MASTER EN SCIENCES INFORMATIQUES

Exploitations graphiques de la base de données des spécifications d'un système d'information

Gena, Alain

Award date:
1983

Awarding institution:
Universite de Namur

FACULTES UNIVERSITAIRES NOTRE-DAME DE LA PAIX (NAMUR)

INSTITUT D'INFORMATIQUE.

EXPLOITATIONS GRAPHIQUES DE LA BASE DE DONNEES DES SPECIFICATIONS D'UN SYSTEME D'INFORMATION.

Memoire présenté par

Alain GENA

en vue de l'obtention du titre de

LICENCIE ET MAITRE EN INFORMATIQUE.

Année académique 1982-1983.

Table des matières

REMERCIEMENTS

Je tiens à exprimer mes remerciements à ceux qui m'ont aidé dans l'élaboration de ce mémoire :

M. F. Bodart ,promoteur de ce mémoire. Par ses avis et ses critiques,il a contribué à la réalisation de ce travail.

M. Y. Pigneur ,pour ses conseils et son dévouement.

Me A-M. Hennebert et M. J-M. Leheureux , pour leur assistance technique.

Mlle B. Scoyer , pour sa constante disponibilité lors de la mise en page de ce mémoire.

Et toutes les personnes sans qui ce mémoire n'aurait pu se réaliser.

INTRODUCTION

L'application de l'informatique au traitement graphique est de plus en plus courante dans des domaines aussi variés tels l'ingénérie, la gestion, l'éducation, l'urbanisme, l'archictecture, la création artistique.

Elle permet d'obtenir, automatiquement ou d'une manière interactive, une représentation claire et concise d'un ensemble complexe d'informations.

L'objet du présent mémoire fût d'étendre et de compléter un interface graphique existant, le Graphical Interface.

Cet interface a été réalisé à l'université du Michigan dans le cadre du projet ISDOS, logiciel d'aide au développement d'un Système d'Information. Il permet une introduction graphique d'informations dans une base de données de spécifications d'un Système d'Information.

L'extension apportée visait à obtenir un outil d'exploitation graphique des informations contenues dans une base de données des spécifications d'un Système d'Information : le Graphical Output Interface.

Ce nouvel outil permettra d'obtenir des représentations complètes et traditionnelles des différents aspects d'un Système d'Information.

Le traitement graphique ne faisant pas encore l'objet d'un enseignement à l'institut d'informatique, nous avons jugé utile de consacrer la première partie de ce mémoire à une synthèse élémentaire des connaissances acquises en ce domaine.

Dans une deuxième partie, nous décrirons les outils graphiques contenu dans le logiciel ISDOS-IDA et nous présenterons l'architecture fonctionnelle de l'interface réalisé.

Enfin, la troisième partie sera consacrée aux problèmes d'implémentation et de réalisation du Graphical Output Interface. Nous y exposerons les différents composants de l'interface.

PARTIE 1 : INTRODUCTION AU TRAITEMENT GRAPHIQUE

Dans cette première partie, nous allons tenter d'établir une synthèse élémentaire et partielle relative à l'utilisation actuelle des outils graphiques.

Nous parlerons de l'outil et de ses caractéristiques tant au niveau matériel (terminaux, imprimantes etc.) qu'au niveau logiciel (standardisation des logiciels graphiques).

Ensuite , nous nous tournerons vers les domaines d'applications en relevant les différents usages actuels du graphique et en tentant de classifier ces usages selon divers critères. Nous nous attarderons essentiellement sur l'utilisation du graphique au sein d'applications de gestion.

En guise d'introduction à ces deux approches, nous allons définir ce que l'on entend par traitement graphique.Nous traiterons plus particulièrement du graphique interactif. Nous essayerons aussi de relever certains avantages liés à ce nouveau mode de dialogue ainsi que les composants essentiels de ce genre d'application.

Chapitre 1 . INTRODUCTION : LE GRAPHIQUE INTERACTIF [FOLE.82]

Nous allons introduire une définition du traitement graphique en informatique par l'intermédiaire de quelques exemples familiers pour la plupart des lecteurs :

- L'utilisation d'imprimantes à impacts permettant d'obtenir des reproductions (par surimpression) de personnages de dessins animés.
- L'utilisation de tables tracantes grâce auxquelles on génère des graphes bi ou tridimensionnels, des histogrammes, des cartes, des dessins architecturaux ou des représentations de circuits.
- L'utilisation d'un terminal graphique avec un clavier et un curseur, afin de manipuler des images d'objets réelsou imaginaires.
- L'utilisation d'un ordinateur individuel destiné non seulement à des applications classiques de traitement de l'information mais aussi à des usages récréatifs par l'intermédiaire d'une vaste gamme de jeux vidéos.
- L'utilisation de microprocesseurs orientés exclusivement vers les jeux vidéos et reliés à une télévision (Atary par exemple).

Bien qu'il existe une grande diversité parmi ces formes d'utilisation de l'informatique graphique, diversité quant au type et à la qualité de l'image, quant au degré de contrôle de l'utilisateur sur l'image, nous pouvons cependant relever une propriété commune :

> une image d'un ou de plusieurs objets est créée et manipulée par un processeur.

Aussi pouvons nous dire que le graphique en informatique concerne la création, la mémorisation et la manipulation de modèles d'objets et de leurs images par un ordinateur.

La notion d'interactivité adjointe au graphique reprend le cas important où l'utilisateur contrôle dynamiquement le contenu des images, le format, la taille, les couleurs sur l'écran par l'intermédiaire d'outils interactifs tel un clavier, un "joystick"...

Les deux premiers exemples d'utilisation cités ci-dessus sont des illustrations de ce que l'on appelle le graphique passif, les trois derniers le graphique interactif.

Avantages d'un outil graphique interactif.

La synthèse d'une image par l'intermédiaire d'un outil graphique interactif constitue l'un des moyens les plus naturels pour communiquer avec l'ordinateur.Nos mécanismes visuels et intellectuels de reconnaissance des formes dans l'espace à deux ou à trois dimensions nous permettent de percevoir et de traiter d'une manière rapide et efficace de nombreuses données si elles nous sont présentées graphiquement.

L'informatique graphique interactive représente le plus important moyen mécanique de production et de reproduction d'image depuis l'invention de la photographie et de la télévision.Le nouvel avantage lié à cette technologie est qu'avec l'aide de l'ordinateur, nous pouvons réaliser des images d'objets abstraits, synthétiques.

Le graphique interactif est une forme d'interaction homme-machine qui combine une communication textuelle ou alphanumérique (via un terminal et un clavier) avec une communication graphique.

L'utilisateur est libéré des frustrations et de la fatigue liées au parcours de nombreuses pages de textes sur listing ou sur terminal. Bien que des images statiques soient souvent de bons moyens de communication d'informations, la possibilité de faire varier dynamiquement une image constitue un avantage appréciable.Cela est spécialement vrai lorsque l'on désire visualiser des phénomènes variant dans le temps, phénomènes tantôt réels (étude du comportement des ailes d'un avion supersonique) tantôt abstraits (désertion en fonction du temps des centres des villes pour les faubourgs et inversement).

Pour exprimer des changements dans le temps, une séquence animée est souvent plus expressive qu'une succession d'images.Une telle séquence peut offrir une vision plus agréable et plus souple d'un changement, surtout lorsque l'utilisateur peut contrôler l'animation en ajustant la vitesse, en choisissant une portion de la scène ainsi que le niveau de détails visualisés...

Une grande partie de la technologie du graphique interactif utilise des techniques matérielles et logicielles pour visualiser soit des mouvements, soit des changements contrôlés par l'utilisateur.

Dans le premier cas, des objets peuvent être bougés par rapport à la position d'un observateur, ou peuvent rester stationnaires tandis que l'observateur se déplace autour d'eux.Un simulateur de vol permet d'entraîner les pilotes en les laissant manoeuvrer un avion simulé au dessus d'un paysage tridimensionnel également simulé. De même, on peut également permettre à un utilisateur de voyager autour ou à l'intérieur

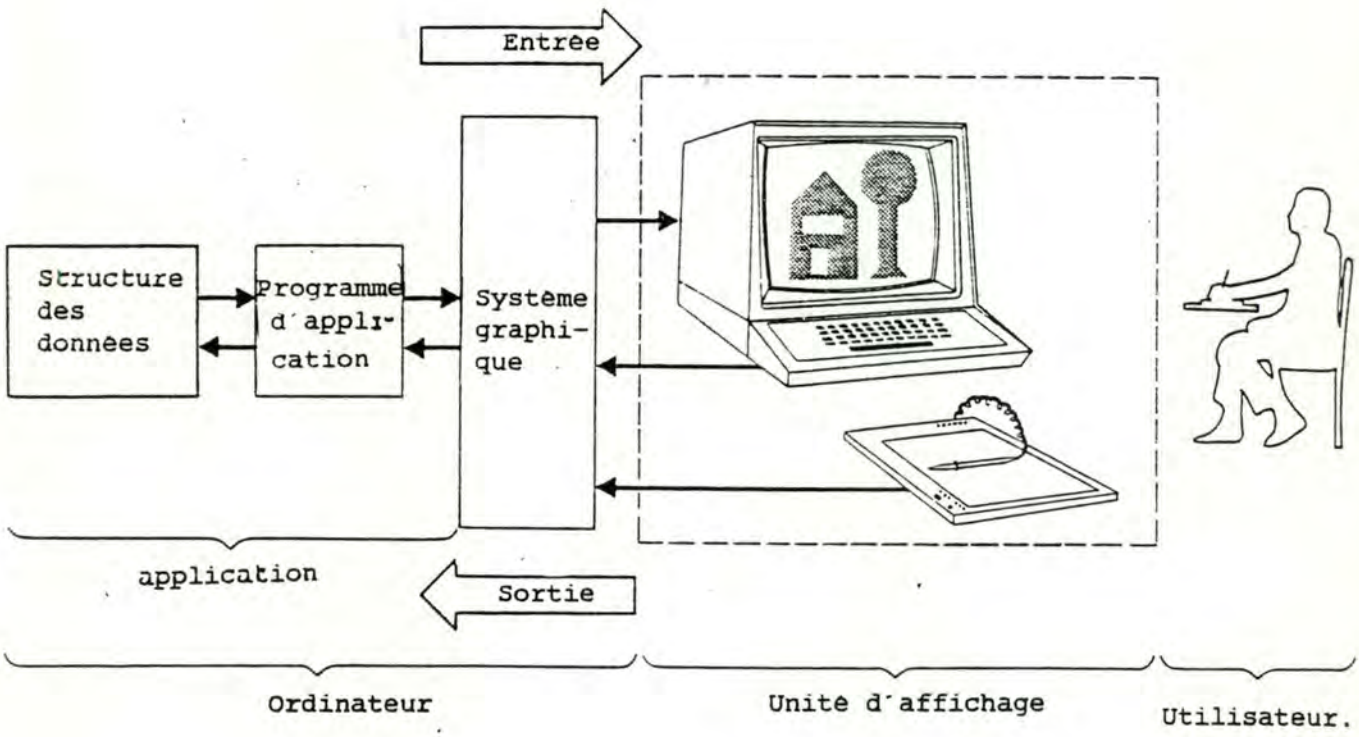
d'un bâtiment, d'étudier une molécule...

Dans le second cas, l'ordinateur peut visualiser les changements d'apparence, de couleurs d'un objet.Il peut faire apparaître, par exemple, la déformation d'une plaque de métal sous l'effet de la chaleur.

En résumé, nous pouvons dire que le graphique interactif nous permet d'obtenir une meilleure communication homme-machine, si l'on a recours à une judicieuse combinaison entre textes et images dynamiques ou statiques.

Composants d'une application graphique interactive

Les différents composants de base d'une application graphique interactive apparaissent sur le schéma ci-dessous [FOLE.82].



Le composant matériel essentiel est un ordinateur central auquel est relié une unité d'affichage constituée d'une unité de sortie (écran sur lequel l'image est affichée) et d'une unité d'entrée.Cette dernière regroupe un ensemble de périphériques logiques comprenant un clavier, des touches permettant d'invoquer des options prédéfinies, un dispositif de

sélection (picking) tel un crayon lumineux permettant d'indiquer un élément au sein d'une image, des estimateurs (valuators) comme des cadrans de contrôle afin d'introduire des valeurs scalaires et enfin un indicateur de position selon les axes de coordonnées X et Y tel un "joystick".Notons qu'une unité physique peut être utilisée pour implémenter une ou plusieurs fonctions logiques.

Les composants logiciels sont au nombre de trois.Nous avons le programme d'application;il enregistre et utilise le second composant, la structure des données de l'application, et envoie des commandes graphiques vers le troisième composant, le système graphique (logiciel graphique).

La structure des données enregistre la description d'objets réels ou abstraits dont les images doivent apparaître à l'écran.Elle contient toutes les informations pertinentes sur des objets tels un circuit, un batiment, un fuselage d'un avion, une fonction mathématique ou statistique...

Ces informations reprennent les attributs des objets tels le type de lignes, leurs couleurs ou même la texture de leur surface.Elles contiennent également un ensemble de coordonnées géométriques décrivant leurs images ainsi que des données précisant les relations entre leurs composants.

De plus il est souvent nécessaire d'enregistrer certaines informations non géométriques telles des propriétés ou des tolérances mécaniques ou électriques.

Le programme d'application décrit la géométrie bi ou tridimentionnelle de l'objet dont l'image doit être visualisée sur la surface de vue par l'intermédiaire du système graphique.

Le système graphique contient un ensemble de sous-routines de sorties graphiques compatible avec un langage de haut niveau tel Fortran ou Pascal. Cet ensemble de sous-routines contrôle une ou plusieurs unités de sorties et engendre l'affichage de l'image sur ces unités. Le système graphique contient également un ensemble de sous-routines d'entrées d'informations graphiques permettant à l'utilisateur de communiquer avec le programme d'application.

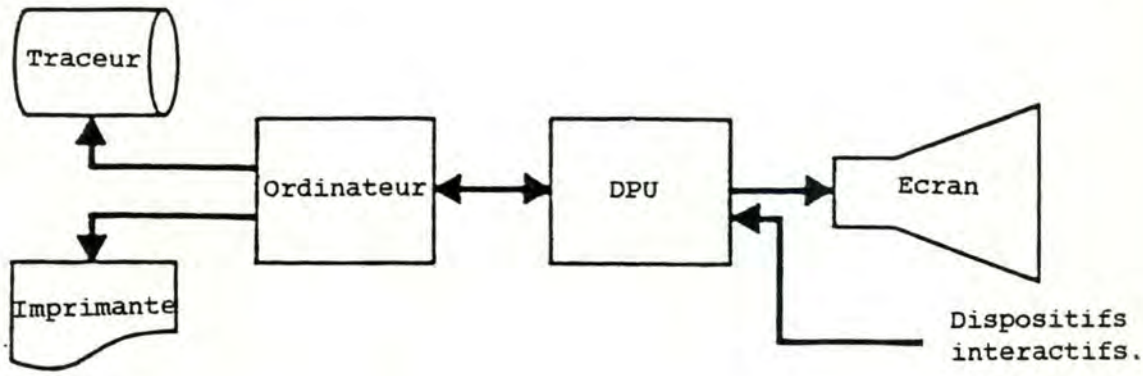
Chapitre 2. APPROCHE TECHNIQUE

Au cours de ce chapitre, nous décrirons dans un premier paragraphe le matériel.Nous parlerons plus précisément des différentes techniques de réalisation de terminaux et d'imprimantes graphiques ainsi que des méthodes d'introduction de données graphiques.Nous aborderons également le problème de la réalisation des couleurs
sur papier et sur écran.

Ensuite, nous traiterons des tentatives de normalisation au sein des logiciels graphiques et des résultats déjà obtenus dans cette voie.

2.1. LE MATERIEL

Le matériel d'un sytème graphique interactif est constitué de quatre composants principaux [FOLE.82] : l'ordinateur,une unité de traitement graphique (Display Processing Unit),une unité d'affichage et une unité d'entrée de données.



Associé à l'ordinateur,le coeur du système,nous trouvons également une imprimante et un traceur (de courbes).

Le DPU peut être comparé à un CPU spécial avec son propre ensemble de commandes,ses propres formats de données et son propre compteur d'instructions. Il exécute une séquence d'instructions d'affichage (display file) afin de créer un dessin sur un écran.Des instructions typiques correspondent au marquage d'un point,au tracé d'une ligne ou d'une chaîne de caractères. Les dispositifs interactifs sont également rattachés au DPU.

Nous allons décrire dans ce paragraphe les différents types de terminaux graphiques en ce qui concerne le mode de réalisation de graphes monochromes.

Pour suivre,nous citerons les différentes techniques d'introduction graphique de données.

Ensuite ,nous parlerons des diverses méthodes d'impression d'images graphiques.

Enfin,nous introduirons le problème de la couleur.

2.1.1. Terminaux graphiques [FOLE.82,MAR1.82]

La technologie des tubes cathodiques est la plus rencontrée aujourd'hui dans la réalisation d'écran graphique.C'est pourquoi nous y consacrerons l'essentiel de notre approche. Nous citerons cependant d'autres technologies qui parviendront peut-être à long terme à réduire ce monopole.

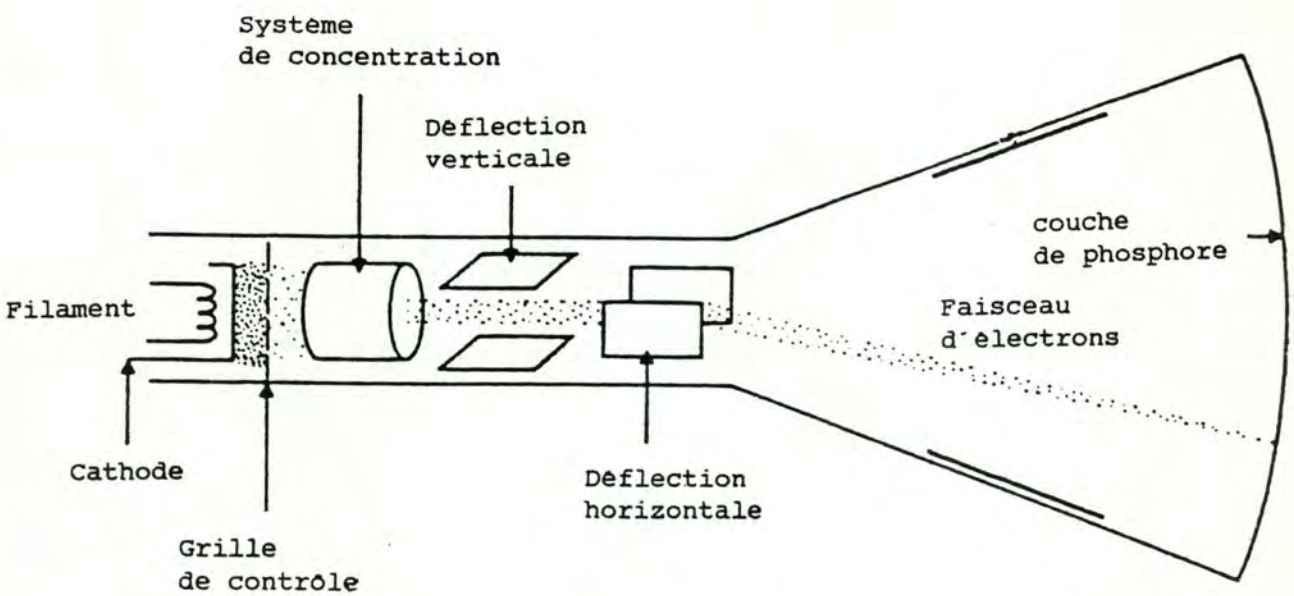
2.1.1.1. Terminaux à tubes cathodiques

Nous allons opérer une première classificaton des terminaux à tube cathodique selon la nécessité ou non de rafraîchir régulièrement l'écran afin d'y maintenir une image.

A. Terminaux à tubes cathodiques avec rafraîchissement d'image

La technologie

Les tubes cathodiques utilisés pour les terminaux graphiques monochromes sont le plus souvent semblables à ceux utilisés pour les télévisions noir et blanc.



Une image résulte de la projection d'un faisceau étroit d'électrons sur une couche de phosphore dont les points sensibilisés émettent une lumière .Le ton dépendra de l'intensité du faisceau.

La lumière phosphorescente s'atténuant d'une manière exponentielle en fonction du temps écoulé, l'entièreté de la figure devra être redessinée de nombreuses fois par seconde afin que la personne observant l'écran ait l'impression de voir une image stable.

La qualité d'un tel système est communément mesurée en terme de résolution, linéarité et vitesse.

La résolution mesure le nombre de points adressables et dépend directement du diamètre du faisceau émis.

La linéarité vérifie si les lignes qui doivent être droites le sont réellement et peut être mesurée par la distance perpendiculaire maximum entre la ligne tracée et une ligne parfaitement droite reliant les mêmes points.

Quant à la vitesse, elle peut être mesurée de différentes facons. Elle peut représenter le temps moyen nécessaire pour déplacer le faisceau vers un point donné à l'écran et faire apparaître une lumière en ce point.

Deux types de réalisation de l'image, deux types de terminaux

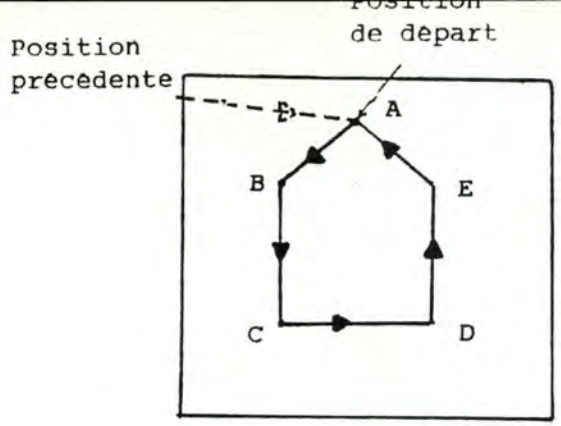
Il existe deux types de terminaux graphiques basé sur la technologie qui vient d'être exposée.Ils diffèrent l'un de l'autre par la manière dont l'image va se dessiner à l'écran.

a. Terminaux à balayage cavalier (Vector refresh display)

Il s'agit de terminaux développés vers le milieu des années soixante et encore d'usage aujourd'hui.

Principe du balayage.

Le principe de création d'un dessin est illustré par le schéma ci-dessous.



La maison apparaissant sur ce dessin est tracée en positionnant le faisceau éteint sur le point de départ A, en l'allumant et en le dirigeant d'une manière continue vers les points indiquant les fins de lignes (B, C, D, E, A).

Le faisceau est asservi en X-Y au sens où il est directement dirigé vers les points appartenant à l'image à visualiser.

Rafraîchissement de l'image.

Un "buffer", souvent appelé "buffer de rafraîchissement", mémorise pour un écran la liste de commandes d'affichage (display list) produite par l'ordinateur.Ces commandes, regroupant des positionnements, des tracés de lignes et des tracés de caractères, sont interprétées par un processeur (DPU), partie intégrante du terminal, qui convertira les valeurs digitales en valeurs analogiques afin de déplacer le faisceau d'électrons et de sensibiliser la couche de phosphore.

La lumière ainsi créée sur le phosphore s'atténuant toutes les 10 ou, tout au plus, les 100 microsecondes, le processeur reparcourra la liste pour rafraîchir le phosphore au moins 30 fois par seconde.

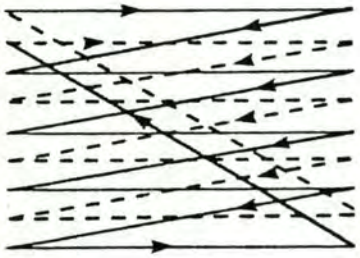
b. Terminaux à balayage de trame (Raster scan display).

Vers le milieu des années septante sont apparus les terminaux à balayage de trame nettement moins coûteux que les précédents (coût lié notamment à la notion d'asservissement en X-Y).

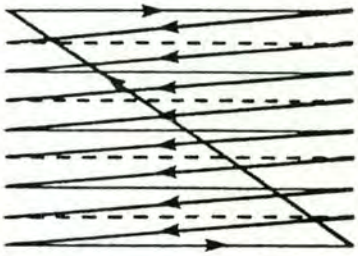
Principe du balayage de trame.

Le dessin à réaliser est divisé en lignes

horizontales. Toutes les parties du dessin apparaissant sur la première ligne sont reproduites, ceci de droite à gauche, ensuite toutes les parties de la seconde ligne etc.



Balayage entrelacé



Balayage non entrelacé

Durant le parcours d'une ligne, l'intensité du faisceau est modulée afin de créer les différents tons allant du noir au blanc. A la fin de la ligne, le faisceau est éteint et repositionné (lignes pointillées) à gauche sur la ligne suivante. Cette dernière peut être celle qui se trouve juste en dessous (balayage non entrelacé) ou bien deux niveaux plus bas (balayage entrelacé).

Dans le cas d'un balayage entrelacé, le faisceau scrute donc d'abord les lignes impaires avant les lignes paires.Il faudra deux passages pour parcourir toute la trame (ensemble des lignes).

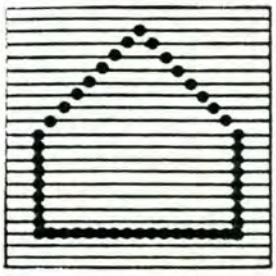
Rafraîchissement de l'image.

Les primitives d'affichage tel le tracé d'une ligne, d'un caractère ou d'un polygône sont enregistrées pour un écran dans un buffer de rafraichissement par l'intermédiaire des points les composant, points appelés "pixels".

L'entièreté de l'image est parcourue 30 fois par seconde ligne par ligne, les lignes étant divisées en un certain nombre de points.

La mémorisation nécessaire dans le cadre d'un tel balayage est accrue au sens où la totalité des dessins s'étendant par exemple sur 512 lignes de 512 points doit être mémorisée explicitement dans une "carte de bits" contenant l'état de chacun des points.

La figure ci-dessous illustre, pour le dessin d'une maison, l'enregistrement de l'état des différents points de l'écran.



D'un autre côté, l'affichage d'une simple image peut être réalisé par une technologie parfaitement semblable à celle d'une télévision.

Quelques commentaires.

Le développement des terminaux à balayage de trames est dû en grande partie à une diminution du coût des buffers de rafraîchissement.

La résolution de ces systèmes n'égale pas encore celle des terminaux à balayage cavalier vu l'absence de matériaux suffisamment rapides. Il faut savoir que tous les points d'une primitive telle le tracé d'une ligne ou d'un rectangle doivent être transformés dans le buffer en leurs nouvelles coordonnées, opération limitée uniquement aux points terminaux des lignes dans le cas du balayage cavalier. Pour un écran de 1000 lignes de 1000 points, un million de points doivent être altérés en moins d'un trentième de seconde. Cela exige des ressources de calcul la plupart du temps indisponibles.

D'un autre côté, avec un système à balayage de trames, le processus de rafaîchissement est indépendant de la complexité (nombre de lignes etc.) de l'image, puisque le materiel est suffisamment rapide pour lire tous les points du buffer durant chaque cycle. Dès lors, il n'y a jamais de scintillement.

Par contre, les terminaux à balayage cavalier scintillent souvent lorsque le nombre de primitives dans le buffer croît d'une manière telle qu'elles ne peuvent plus toutes être lues et traitées en un trentième de seconde. L'image ne sera donc pas rafraîchie suffisamment souvent.

B. Terminaux à tubes cathodiques avec mémoires (Direct view storage tube)

Les systèmes composés d'un tube cathodique à mémoire sont similaires à ceux dotés d'un tube cathodique standard (CRT), si ce n'est qu'ils n'ont pas besoin de rafraîchissement. L'image est enregistrée dans le système comme une distribution de charges sur la surface intérieure de l'écran.

Les désavantages liés à cette technique apparaissent dans la difficulté d'obtenir une variété de tons et dans l'impossibilité d'effacer une zone particulière à l'écran.

L'avantage essentiel réside dans le coût inférieur à celui d'un terminal à balayage cavalier et à ceux de la plupart des terminaux à balayage de trames. Cela est dû à l'absence de buffer de rafraîchissement et au fait que le système de déflexion (orientation du faisceau) n'a pas besoin d'être aussi rapide que dans le cas d'un terminal avec rafraîchissement d'image.

C. Résumé des avantages et des inconvénients des terminaux à tubes cathodiques.

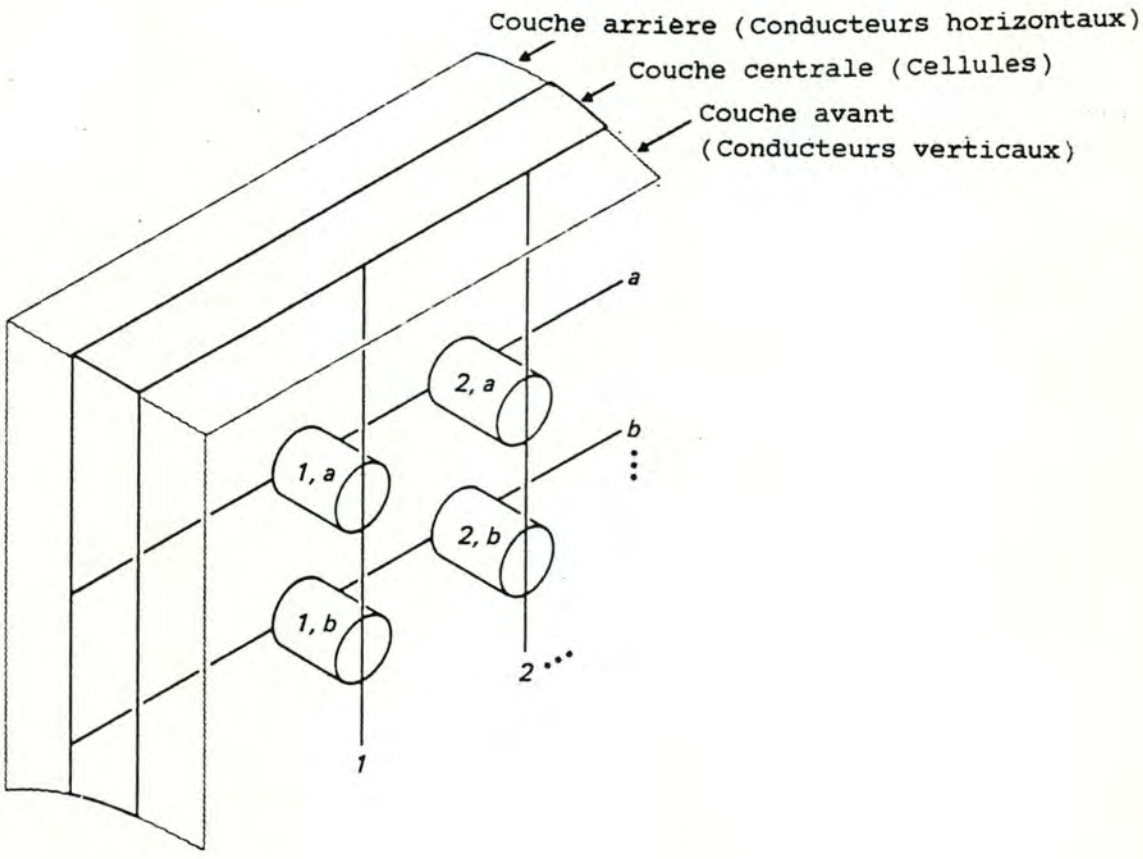
	Tube avec balayage cavalier	Tube avec balayage de trames	Tube à mémoire
Avantages	Très bonne définition	Pas de scintillement Bon marché Très adapté couleurs (1)	Très bonne définition Très bon marché Pas de scintillement
Inconvénients	Coût Tendance au scintillement Réalisation de couleurs difficile (1)	Définition insuffisante	Effacement sélectif impossible Pas de couleurs (1)

(1) Nous parlerons du problème de la couleur ultérieurement. Nous avons déjà précisé cependant qu'il était très difficile d'obtenir une variété de tons en noir et blanc avec les tubes à mémoire.

2.1.1.2. Autres types de terminaux

Nous allons décrire ici brièvement la technologie des terminaux dont la dénomination anglaise est "plasma panel display".
Le panneau, leur principal composant, correspond souvent à une matrice de petits tubes néons. Chaque tube peut être dans un état "on" ou "off" et reste dans cet état jusqu'à un changement explicite.
En général, nous aurons une surface de 20 centimètres carré avec 26 cellules (tubes) par centimètre.

Les tubes ne constituent pas des unités discrètes et indépendantes, mais plutôt des parties d'un panneau intégré constitué de trois couches de verre :



La surface intérieure de la couche avant contient de fines bandes verticales d'un conducteur électrique; la couche centrale contient les cellules et la surface intérieure de la couche arrière contient de fines bandes horizontales d'un conducteur électrique.
Pour allumer l'ampoule (1, a) dans la figure, les tensions sur les lignes 1 et a sont ajustées afin que leur différence soit suffisamment large pour allumer le néon et le faire briller. Ceci fait, une tension moins forte est appliquée afin de maintenir cette brillance.
Pour éteindre le même tube , la tension sur les lignes 1 et a est rendue momentanément inférieure à la tension de maintien.

Nous pouvons citer deux autres technologies qui semblent promises à un fréquent usage dans le futur : les écrans à cristaux liquides (LCD) et les écrans à diodes luminescentes (LED). Les cristaux liquides prévalent par leur faible consommation.

Nous n'aborderons pas enfin la description technique des écrans à lasers.

2.1.2. Techniques d'introduction graphique de données [FOLE.82, MICH.78]

Il existe aujourd'hui une grande variété de techniques et de dispositifs interactifs en graphique.Nous pouvons heureusement établir une classification parmi ceux-ci et structurer ainsi notre étude;la classification adoptée étant basée sur les dispositifs logiques (logical devices) définis par le Siggraph Planning Committee (SGP).

Nous distinguerons (1):

- "locator" : indiquer une position et (ou) une orientation.
- "pick" : sélection d'une entité à l'écran.
- "valuator": permet d'introduire une valeur réelle.
- "Keyboard": introduire une chaîne de caractères.
- "button" : sélection parmi un ensemble d'actions ou de choix alternatifs possibles

2.1.2.1. Locator

Le moyen physique le plus utilisé pour remplir cette fonction est la tablette couplée avec un stylo ou un curseur manuel.

La tablette designe l'espace adressable par l'utilisateur.

Le stylo a une extremité sensible à la pression.L'utilisateur met en contact le stylo avec la tablette et effectue une pression afin d'indiquer au programme d'application qu'il se trouve au point interessé.

De son côté, le curseur manuel possède plusieurs touches pouvant être manipulées pour introduire des commandes.

Lorsque le stylo ou le curseur manuel se trouve près ou sur la tablette, un curseur est généralement affiché à l'écran afin de visualiser la position actuelle.

Un autre moyen est l'usage d'une "souris".La souris a de petites roues à sa base.En déplacant la

---

(1) Vu l'imprécision de la terminologie francaise, nous utiliserons les termes anglais.

souris sur une surface plane, l'utilisateur fait tourner les roues.Un potentiomètre associé à l'une des roues enregistre le déplacement relatif dans deux directions orthogonales.Ce déplacement déterminera la nouvelle position du curseur à l'écran.

On peut encore faire usage d'un "joystick".Il s'agit d'un simple batonnet qui peut bouger de droite à gauche, de haut en bas.A nouveau, un potentiomètre enregistre le mouvement.Il est difficile d'utiliser un tel batonnet pour contrôler directement la position absolue du curseur à l'écran car un léger mouvement manuel est amplifié de 5 à 10 fois au niveau du curseur.

2.1.2.2. Pick

Le seul dispositif qui soit d'une manière inhérente un "pick", c'est à dire un outil de sélection d'une entité à l'écran, est le crayon lumineux.

Cependant son usage est de plus en plus limité, vu la fatigue liée à son utilisation prolongée.La plupart des systèmes utilisent la tablette ou la souris pour simuler cette fonction.

2.1.2.3. Valuator

La majorité des dispositifs fournissant des valeurs réelles sont basés sur des potentiomètres semblable à ceux utilisés pour ajuster le volume, la balance ou la tonalité d'une installation musicale.

Beaucoup de "valuator" sont des potentiomètres rotatifs.On les utilise souvent pour contrôler la rotation d'objets.

Il existe aussi des potentiomètres "glissants", où, pour exprimer des valeurs sans interprétation angulaire, un mouvement linéaire remplace la rotation.

2.1.2.4. Keyboard

Un clavier alphanumèrique est le seul dispositif d'introduction de textes.

2.1.2.5. Button

La plupart du temps, les touches des claviers constituent le dispositif technique employé pour remplir cette fonction.

On pourrait également faire usage des touches disposées sur les curseurs manuels ou les souris.

Grâce à ces dispositifs, l'utilisateur entrera des commandes ou effectuera des choix au sein d'un menu.

2.1.3. Techniques d'impression de graphes.

L'ordinateur a été utilisé pour tracer des images bien avant le développement du graphique interactif.L'outil employé peut être une simple imprimante linéaire ou une imprimante graphique (traceur).

2.1.3.1. Imprimante linéaire (printer)

Les utilisations les plus courantes d'une imprimante linéaire pour réaliser des sorties graphiques comprennent le tracé de cartes de flux, de fonctions à une variable (barres etc) et à deux variables.

Le facteur essentiel permettant de savoir si un dispositif de sorties graphiques est approprié à une application particulière est sa résolution (nombre de points distincts par unité de distance).

La faible résolution des imprimantes linéaires limite leur utilité.

2.1.3.2. Traceurs

La table tracante, l'imprimante à rouleau ou à tambour et l'imprimante électrostatique constituent des illustrations des différentes techniques de réalisation de traceurs.

Le mode de création d'un dessin pour les deux premiers dispositifs cités est similaire à celui employé par les terminaux à balayage cavalier (asservissemment en X-Y)

2.1.4. Les couleurs [FOLE.82, MAR1.81, MAR2.81]

2.1.4.1. Introduction

Depuis quelques années, la couleur semble prendre une place de plus en plus prépondérante dans le graphique.L'apparition d'une technique nouvelle bien adaptée (les terminaux à balayage de trame) n'est certes pas étrangère à ce phénomène.

L'oeil humain, dont la sensibilité est extraordinaire, distingue 160 teintes et 1000 niveaux d'intensités différents.

Selon d'aucuns, la couleur améliore la productivité, réduit la fatigue et diminue le nombre d'erreurs.

L'opérateur peut facilement sélectionner les informations qui le concernent, relever d'un coup d'oeil un changement de tendance, percevoir des exceptions.

La couleur, éventuellement couplée aux techniques statistiques de lissages, de régressions ou d'extrapolations met en évidence des mouvements subtils, des tendances peu évidentes sous formes chiffrées...

Le principal obstacle au développement des terminaux couleurs a toujours été la complexité des logiciels qu'ils impliquent et l'importance des moyens matériels qu'ils entraînent - processeurs, mémoires etc.

De plus il n'existe actuellement aucun standard, ni sur les couleurs, ni sur leur modélisation, ni sur

la définition de l'intensité et de la saturation.

Il est enfin pénible de devoir archiver en noir et blanc ce qui a été réalisé en couleur car il existe peu de modèle de recopie d'écran et ils restent très chers.

2.1.4.2. Couleurs achromatiques [FOLE.82]

Nous parlerons ici brièvement des couleurs achromatiques apparaissant, par exemple, sur une télévision noir et blanc.

La seule caractéristique d'une couleur achromatique est son intensité liée à l'amplitude lumineuse de la source.

Une télevision noir et blanc peut produire de nombreux niveaux d'intensités différents sur un même point (pixel).

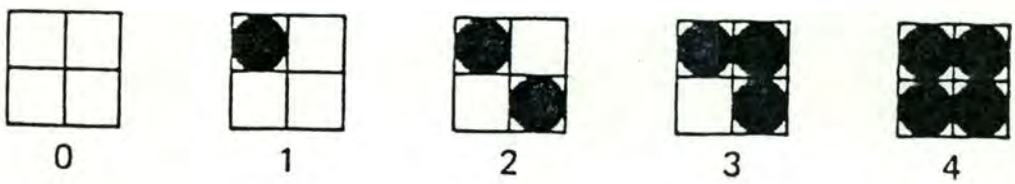
Par contre une grande partie des terminaux et des imprimantes graphiques ne peuvent offrir que deux niveaux : le blanc et le noir.

Comment dès lors étendre l'éventail des intensités disponibles ?

La réponse réside dans l'intégration spatiale réalisée par l'oeil humain.Si nous observons une surface très petite (par exemple 0.05 sur 0.05 cm) à partir d'une distance normale, l'oeil va intégrer les détails au sein de la petite surface et enregistrer uniquement l'intensité globale de la surface.

Ce phénomène est utilisé pour imprimer des photos noir et blanc dans les journaux et dans les magazines.

De même, au niveau des imprimantes et des terminaux, nous pouvons regrouper les points en de petites surfaces de deux points sur deux.Nous obtiendrons ainsi cinq niveaux d'intensités différents.



2.1.4.3. Couleurs chromatiques : principe de colorimètrie

Une couleur est entièrement définie par trois paramètres :

- La teinte :
  Cette notion nous permet de distinguer des couleurs tel le rouge , le vert et le bleu.Elle associe la couleur à une longueur d'onde determinée (ou frèquence).

- La luminosité ou la luminance :
  Elle caractèrise l'intensité d'excitation visuelle (brillance) c'est à dire l'amplitude de la source.

- La saturation :
  C'est le degrè de pureté, l'extension à partir de laquelle la teinte apparaît comme diluée ou mèlangée avec du blanc; en d'autres termes, l'écart par rapport à une couleur achromatique de même brillance (noir, gris, blanc).

La colorimètrie ou science de la couleur définit trois teintes fondamentales : le rouge, le vert et le bleu qui mèlangées donnent du blanc.En mèlangeant 2 à 2 les trois teintes fondamentales, dites aussi primaires additives, on obtient :

bleu + vert = turquoise ou cyan

rouge + vert = jaune

rouge + bleu = magenta (pourpre)

On constate aussi qu'en mèlangeant deux couleurs fondamentales, on obtient une teinte complèmentaire de la troisième : le cyan est le complèmentaire du vert et le jaune est le complèmentaire du bleu (deux couleurs complèmentaires donnent la sensation du blanc).

Or si, par un moyen quelconque, on arrive à soustraire une teinte du blanc, on obtiendra sa complèmentaire :

blanc - bleu = jaune

blanc - vert = magenta

blanc - rouge = cyan

Ceci explique que ces trois teintes, jaune, magenta et cyan, soient appelées les primaires soustractives.

Ces principes de colorimétrie sont essentiels pour la bonne compréhension de la technologie des couleurs en matière d'écran et de photographie, l'additivité des teintes étant exploitées par les écrans et la soustractivité des teintes par la photographie.

2.1.4.4. Réalisation technique

A. Terminaux couleurs

Parmi les trois types de terminaux basés sur la technologie des tubes, deux peuvent permettre la réalisation de couleurs, les terminaux à balayage de trame et les terminaux à balayage cavalier.

Il est, en effet, impossible de fabriquer des écrans couleurs à mémoire, impossibilité liée au principe de mémorisation, obtenue par une émission secondaire d'électrons et entretenue par des canons auxiliaires d'arrosage.

1. Terminaux couleurs à balayage de trame

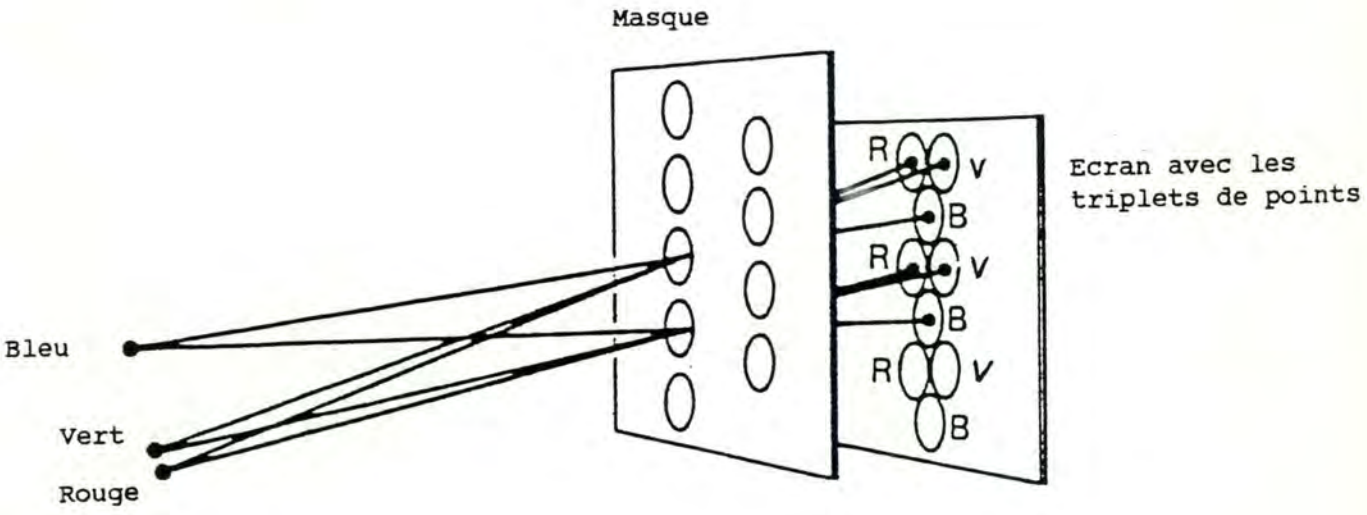
L'écran est recouvert (voir figure) d'une multitude de points luminophores (phosphores) regroupés en triplet, appelé triads, les éléments du triplet ayant la propriété d'émettre, l'un dans le rouge, l'autre dans le vert et le troisième dans le bleu, lorsqu'ils sont soumis à une projection d'électrons.

Comment exciter individuellement ces points de phosphores par des électrons ?

Trois canons différents, arrangés selon la même figure triangulaire que les triads, sont utilisés;chacun d'eux étant spécialisé dans une couleur.L'intensité de la couleur dépend de l'énergie du faisceau, lui-même fonction de la tension d'accélération qui varie selon la modulation portée par les signaux vidéos.De manière que les faisceaux frappent bien les cibles qui leur sont attribuées, un masque est placé devant le phosphore, percé d'une multitude de trous, qui ne laissent passer, pour une couleur donnée, que les électrons qui la concernent.

La couleur sur l'écran n'est donc jamais que le

résultat d'une juxtaposition de teintes, une sensation résultant de l'association de points qui émettent avec des intensités variables selon les trois couleurs fondamentales.



Nous rappelons que, l'émission étant éphémère, le système doit rafraîchir l'information à partir d'une mémoire.

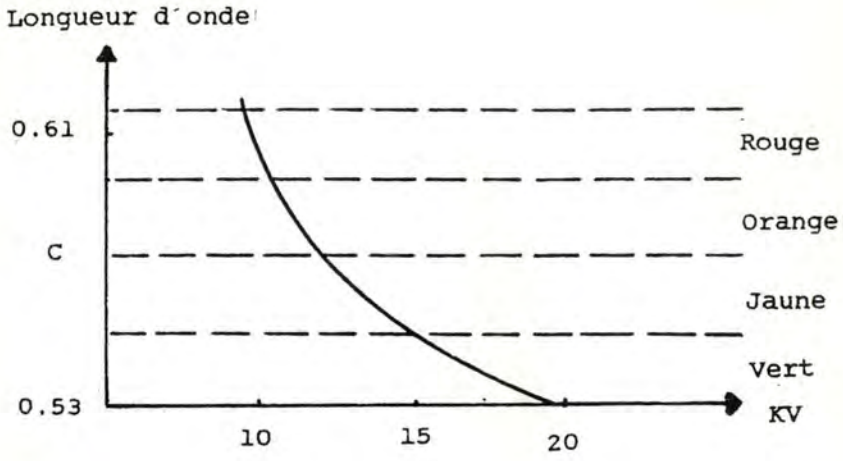
Le tube à masque a un inconvénient majeur : il ne permet pas une résolution suffisante pour les applications graphiques de moyenne complexité (l'unité adressable à l'écran étant le triad c'est-à-dire un ensemble de trois points).

2.Terminaux couleurs à balayage cavalier

Nous rappelons, qu'avec cette technique, le faisceau d'électrons n'est envoyé que sur les seuls points qui appartiennent à l'image à visualiser;le faisceau étant éteint lorsqu'il doit se déplacer vers une destination donnée en passant par des points hors image. Comme pour un oscilloscope, le faisceau est asservi en X et Y, et l'image, comme pour un balayage récurrent, doit être rafraîchie à partir d'une mémoire.

La couleur est obtenue par le contrôle de la pénétration d'électrons dans une succession de deux couches de phosphore qui émettent des longueurs d'onde différentes.Selon la force du faisceau, les électrons traversent la première couche et vont frapper la deuxième, ou s'arrêtent sur la première.Une pénétration partielle donnera des couleurs intermédiaires, composées de rouge et de vert, les quatre couleurs obtenues étant le jaune, le rouge, l'orange et le vert, la variété des couleurs étant fonction de la tension

d'accélération.



Le gros avantage de la présente technique est la possiblité d'obtenir des dessins d'une grande finesse, ses inconvénients étant le coût, la difficulté d'obtenir plusieurs couches et le scintillement qui ne manque jamais d'apparaître lorsque l'image visualisée est complexe (on retrouve les défauts et les avantages des écrans noir et blanc à balayage cavalier).

B. Imprimantes couleurs.

Nous allons parler ici du problème de la recopie d'un écran couleur à balayage de trame.

Soulignons pour commencer les différences entre les deux mécanismes de reproduction de la couleur, photographie et écran, mécanismes revêtant l'un par rapport à l'autre un certain nombre d'incompatibilité :

D'abord, en ce qui concerne le principe de reproduction de la couleur : synthèse additive pour l'écran et synthèse soustractive pour le procédé photographique.

Un objet apparaît jaune sur un écran parce que l'oeil humain n'est pas assez sensible pour s'appercevoir que le jaune est dû à la juxtaposition d'un grand nombre de points rouges et verts.

En photographie, par contre, c'est le colorant jaune qui, en absorbant le bleu et en laissant passer le vert et le rouge, permet à l'objet d'être apercu en jaune.

Mais la différence s'accuse dans la structure des

èlements de la coloration : la gènèration de la couleur sur un écran provient de la juxtaposition "latèrale" des luminophores, alors qu'en photographie, c'est la superposition "verticale" des trois images argentiques qui permet de la recrèer.

Dès lors, lorsque l'on dèsire reproduire un écran couleur, la simple photographie de l'écran par un système optique traditionnel donne des rèsultats dècevants, car elle reproduit fidèlement les triplets de points rouges, verts et bleus et non une image en couleurs fondues.Les primaires, en particulier, sont très mal rendues, car, pour chacune d'elles, le triplet sera constituè de deux points noirs qui apparaîtront sur le papier photographique.

Il fallait donc trouver autre chose.Nous allons citer ici deux exemples parmi les solutions technologiques qui ont été adoptées.

1. Système Calcomp 31

Pour mieux comprendre le principe de fonctionnement d'un système de recopie couleur type Calcomp, il est nècessaire de bien poser les bases de la colorimètrie à synthèse soustractive.

Synthèse soustractive

On se rappelle que les trois couleurs fondamentales sont le rouge, le vert, et le bleu.En les mèlangeant 2 à 2, on obtient la complementaire de la troisième :

bleu + vert = cyan , complèmentaire du rouge

rouge + vert = jaune , complèmentaire du bleu

rouge + bleu = magenta , complèmentaire du vert

Le mèlange de deux couleurs complèmentaires donne du blanc.Ainsi au lieu d'additionner ou d'associer des couleurs (comme en technologie TV) on peut, à partir d'une teinte, soustraire des couleurs pour en obtenir une autre :

blanc - bleu = jaune

blanc - vert = magenta

Blanc - rouge = cyan

A cette fin, on utilise des filtres, qui ont la propriété de ne laisser passer que des radiations de même longueurs d'onde, et d'éliminer, en partie ou en totalité, celles des teintes complémentaires.

De cette facon on peut, par exemple, reconstituer les trois primaires :

- le vert résulte de l'absorption du bleu par un filtre jaune (complementaire du bleu) et de celle du rouge par un filtre cyan (complementaire du rouge).

- le bleu résulte de l'absorption du vert par un filtre magenta et de celle du rouge par un filtre cyan.

- le rouge résulte de l'absorption du bleu par un filtre jaune et de celle du vert par un filtre magenta.

Système Calcomp

Calcomp utilise avec son système 31, un film polaroid qui met en oeuvre ces principes.

Le système contourne le problème de la disposition latérale des luminophores grâce à un dispositif de transmission verticale et désynchronisée des couleurs.L'image à produire est recueillie à partir de trois entrées séparées, rouge, vert, bleu, correspondant aux trois canons du tube.
Ces trois entrées donnent lieu à trois images, qui sont superposées sur un film en trois images fondues et non juxtaposées.

Un tel procédé est évidemment très coûteux.

2. Imprimante à impact Colorplot de Trilog

Trilog propose une imprimante de recopie matricielle et à impact, dont la particularité est liée à la nature de son ruban.

Un support de 19.8 m de long, divisé en trois zones de 6.6 m : une pour le rouge, une pour le vert et une pour le bleu.L'ensemble étant d'après son fournisseur 75 % moins coûteux qu'un système à jets d'encre et 90 % qu'un système de type photographique.

En fait, l'impression se fait en trois phase

: une pour chaque couleur et page par page.Lorsqu'une page est terminée, en rouge par exemple, le mécanisme d'entrainement repositionne le papier au début de la page et libère l'édition de la couleur suivante.

Etant donné que le ruban ne comporte que trois couleurs, Trilog a résolu le problème de la saturation en repassant deux fois sur les mêmes points et celui des demi-teintes en n'imprimant qu'une ligne sur 2.Quant aux autres couleurs, le principe retenu est identique à celui d'un tube de télévision. Pour obtenir du jaune, elle imprime un poin t rouge puis un point vert.

Le coût d'une copie ainsi obtenue est nettement moins élevé que celui d'un Polaroid.Enfin, la taille d'une copie n'est plus limitée à 20 sur 25 cm et peut atteindre 33 cm de large pour une longueur quelconque.

2.1.4.5. Modèle de choix de représentation de couleurs pour terminaux à balayage de trame

Autant il est facile à l'opérateur de clarifier ses souhaits en affichage alphanumérique ou même graphique, autant la couleur est une notion subjective, difficile à préciser surtout lorsque l'écran en propose une vaste palette.C'est d'ailleurs à ce niveau que les habitudes sont et seront les plus boulversées.

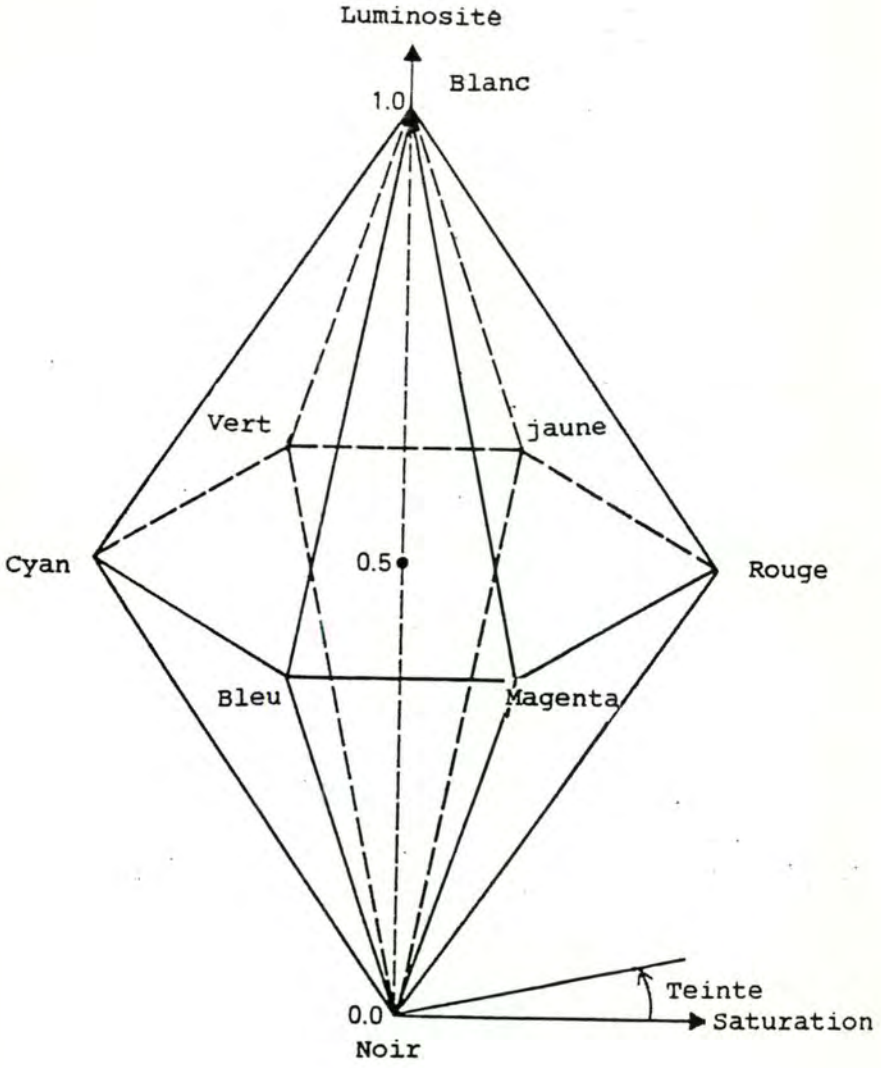
Et du côté des constructeurs, deux écoles existent, liées à la complexité des produits.

En bas de gamme, l'utilisateur a le choix entre 4, 8 et parfois 16 couleurs, parfaitement définies dès l'origine.Chacune porte un nom ou un numéro, que le programmeur appelle par une commande.Il n'existe aucune autre teinte, aucun jeu sur la luminosité, la saturation.

En haut de gamme, la sophistication des matériels a nécessité l'adoption de modèles de couleurs.Ces modèles utiliseront le plus souvent un double cône ou un cube.

Au lieu de définir la teinte comme une zone de caractères numériques ("je veux du violet un peu fané et tendant vers le rouge), l'opérateur positionnera visuellement la couleur sur la figure géométrique et indiquera dans une commande les paramètres correspondants.

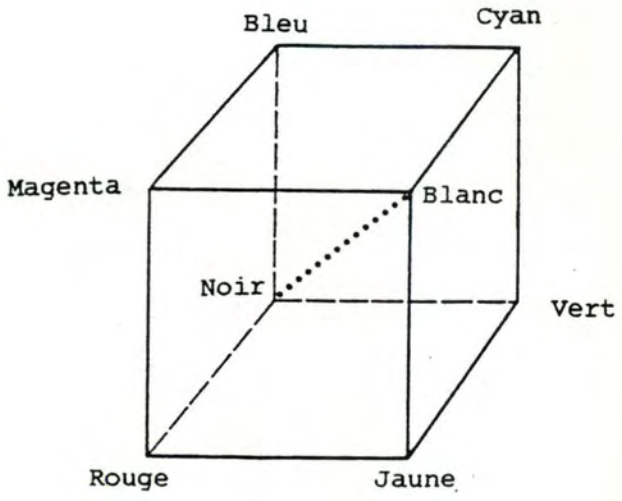
A. Le modèle double cône



La teinte est déterminée par l'angle autour de l'axe vertical du double cône avec du rouge à 0 degré.La saturation est mesurée par la distance orthogonale par rapport à l'axe vertical et varie de 0 à 1.La luminosité vaut 0 (noir) à l'extrémité inférieure du cône et 1 (blanc) à l'extrémité supérieure du cône.

Par exemple si le programmeur désire du rouge plein, il désignera le point de : teinte 0, luminosité 1/2, saturation 1.

B. Le modèle cubique



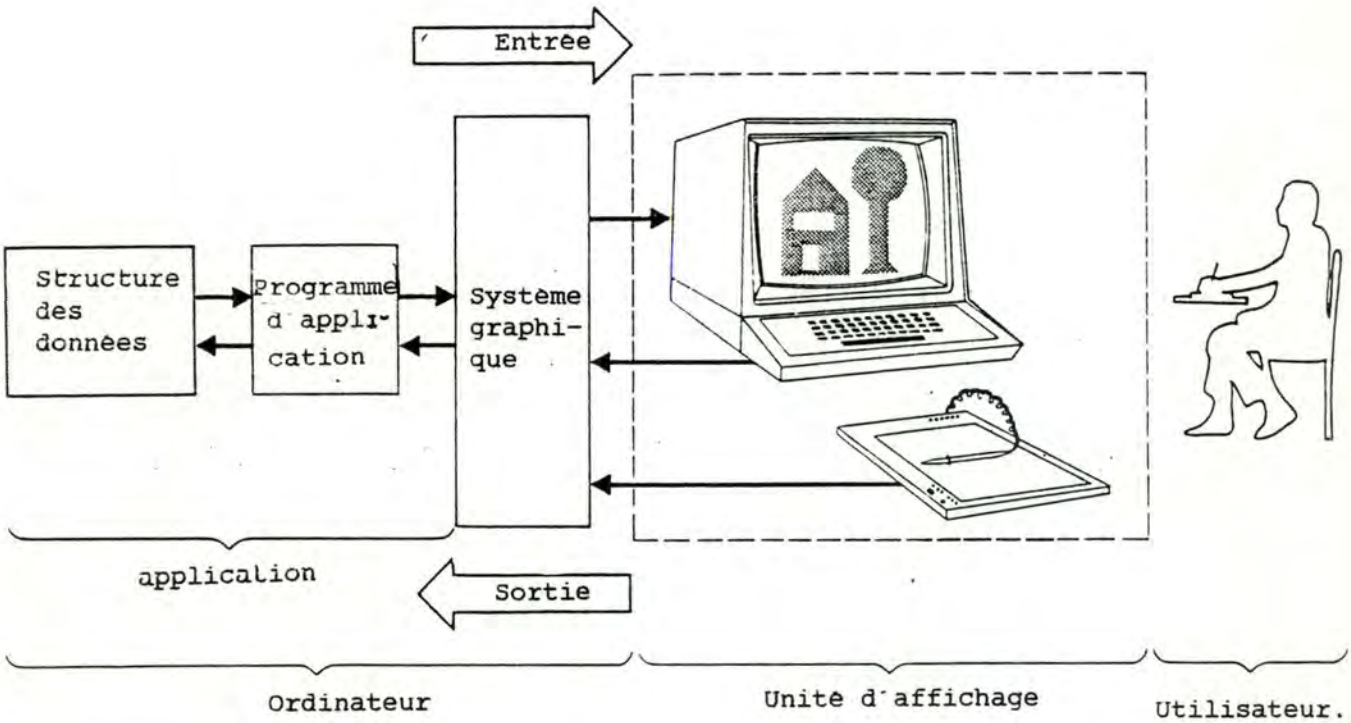
Trois arêtes du cubes constituent les axes d'un système cartésien dont le centre est le point "noir".La longueur des cotés est 1.

L'utilisateur choisit une couleur en désignant un point du cube.

2.2. LES LOGICIELS GRAPHIQUES

Ce paragraphe est consacré à l'étude des problèmes de standardisation des logiciels graphiques.

Aussi il semble opportun de préciser ce que l'on entend exactement par logiciel graphique.Pour ce faire considérons à nouveau la figure des composants d'une application graphique reprise à la fin du premier chapitre.Le dessin représentait ce que le GSPC (Graphics Siggraph Planning Committee) appelle le modèle du programmeur [FOLE.82].



Nous avons déjà précisé que le système graphique (logiciel graphique) contrôlait les différents périphériques et contenait un ensemble de sous-routines de sorties et d'entrées d'informations graphiques compatibles avec un langage de haut niveau.

Le programme d'application quant à lui décrivait la géométrie bi ou tri dimensionnelle de l'objet dont l'image est à visualiser par l'intermédiaire de ces sous-routines.La séparation entre programme d'application et logiciel graphique s'avère de plus en plus fréquente au sein de la plupart des systèmes et constitue déjà un facteur de portabilité.Sur base de celle-ci, des recherches de standardisation des fonctions offertes par les

logiciels graphiques se sont développées...

Nous parlerons dans un premier temps du principal motif , la portabilité, qui a lancé les recherches sur la voie de la standardisation.

Nous énoncerons ensuite quelques exigences et principes concernant la conception d'un standard.Par après, nous effectuerons un bref historique sur ces efforts de standardisation.Cela nous amènera enfin à parler des normes existantes à ce jour , GSPC79 et GKS, et à en décrire les fonctions de bases.

2.2.1. Pourquoi un standard ?

Le principal motif de standardisation des logiciels graphiques est la portabilité du programme d'application, c'est-à-dire la possibilité de transférer une application graphique d'une installation vers une autre avec un minimum de changements dans la programmation.

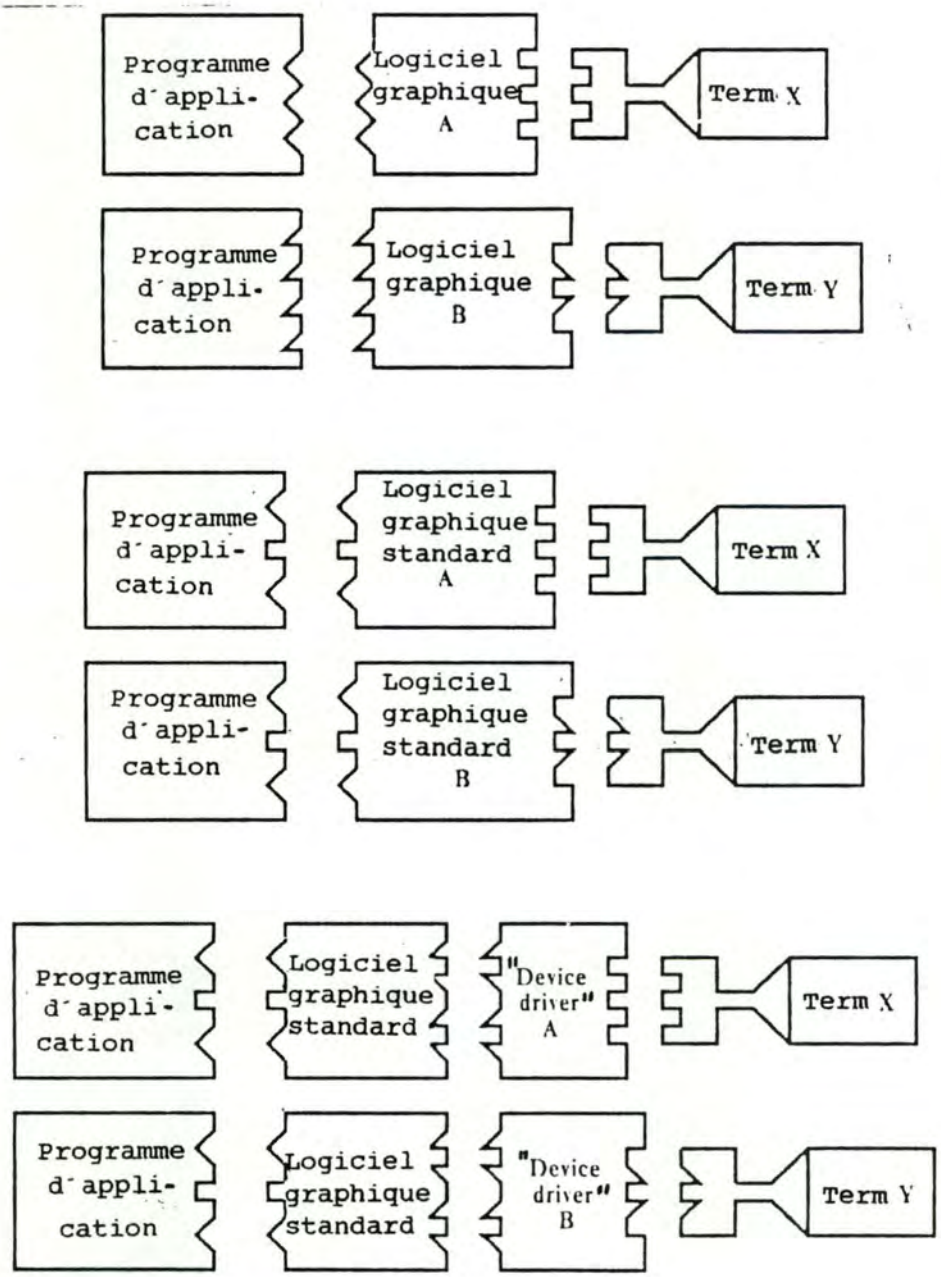
Les difficultés rencontrées pour transférer un programme d'application entre sites ont longtemps écarté de nombreux utilisateurs potentiels.

Un autre problème fût le besoin de former des programmeurs aptes à comprendre la complexité de chaque nouvelle installation graphique rencontrée.La standardisation offre une opportunité de réduire ce problème et d'obtenir la portabilité du programmeur.

Dans une large mesure la portabilité pourrait être atteinte si le matériel graphique était standardisé.Cela ne semble guère envisageable.

Le problème ne peut être résolu que par l'utilisation d'un interface standard de haut niveau entre le programme d'application et les unités graphiques.Cette interface , appelé souvent logiciel graphique, offrira au programmeur d'application un ensemble de fonctions qu'il pourra utilisé dans son programme, indépendament des caratéristiques des unités concernées.

Le problème de la dépendance vis à vis du matériel dans le cadre d'une application graphique est illustré sur la figure ci-dessous [NEWM.14].


Programme d'appli-cation
Logiciel graphique A
Term X
Programme d'appli-cation
Logiciel graphique B
Term Y
Programme d'appli-cation
Logiciel graphique standard A
Term X
Programme d'appli-cation
Logiciel graphique standard B
Term Y
Programme d'appli-cation
Logiciel graphique standard
"Device driver" A
Term X
Programme d'appli-cation
Logiciel graphique standard
"Device driver" B
Term Y

Le premier dessin montre deux logiciels graphiques développés pour deux terminaux différents;chacun d'eux présentant un interface de programmation différent au programme d'application. Nous dirons que ces logiciels dépendent entièrement du terminal utilisé en ce sens qu'ils forcent le programmeur à tailler ses programmes en fonction du terminal.

Dans le deuxième dessin, nous voyons le résultat d'un effort visant à offrir un interface identique au programmeur.Le même programme d'application peut tourner sur deux sites.Nous avons donc atteint une indépendance vis à vis du terminal pour le programmeur; cependant les deux logiciels graphiques diffèrent intérieurement.

Dans le troisième dessin, le logiciel graphique est devenu dans une large mesure lui-même indépendant du terminal.La seule partie du logiciel liée au terminal est le "device driver".Sa fonction est de transformer , en fonction du terminal, le "display file", fichier contenant un ensemble d'appels à des primitives graphiques, le restant du logiciel étant tout à fait portable.

L'interface uniforme présenté par le "device driver" permet aux unités d'affichages et d'entrées graphiques d'être traitées comme des unités logiques dont les différences sont ignorées par le programmeur.

En résumé et à partir de ce qui vient d'être dit, nous pouvons citer trois raisons pour justifier l'apparition d'un standard graphique:

a)permettre au programme d'application utilisant des graphiques d'être facilement portable d'une installation à l'autre.

b)aider la compréhension et l'usage de méthodes graphiques par le programmeur d'application.

c)service de point de référence pour les constructeurs d'équipements graphiques en fournissant une combinaison utile de capacités graphiques pour un terminal.

2.2.2. Exigences et principes pour la conception d'un standard graphique [GKS.82]

Afin d'atteindre les objectifs précités, la conception d'un standard graphique doit être basée sur les exigences suivantes:

a)Le standard doit inclure toutes les fonctions essentielles pour l'ensemble des domaines d'application

du graphique, de la simple sortie passive d'un dessin à la grande application interactive.

b)Le standard doit contrôler d'une manière uniforme l'ensemble des types de terminaux graphiques (cavalier, balayage de trame, à mémoire, avec couleurs...).

c)Le standard doit offrir toutes fonctions exigées par la majorité des applications sans pour cela devenir excessivement important.

Ces exigences sont utilisées afin de formuler un certain nombre de principes utilisés pour juger des différentes alternatives de conception du standard.

Il est possible de contribuer à l'ensemble des objectifs globaux en se concentrant sur certains aspects.
Cinq aspects de conception ont été identifiés, chacun se voyant associer un ensemble de principes:

a)objectifs de conception:

Les principes suivants ne doivent être violés par aucune conception technique.

1)cohérence:
aucune des exigences impératives du standard ne doivent être contradictoires.

2)compatibilité:
aucun autre standard ou règle communément acceptée ne doit être violée.

3)orthogonalité:
les fonctions du standard doivent être indépendantes les unes par rapport aux autres, ou la dépendance doit être structurée et bien définie.

b)capacités fonctionnelles:

les principes suivants sont utilisés pour définir l'étendue du standard.

1)complétude:
on doit inclure toute fonction que la majorité des applications désire utiliser à un niveau donné de fonctionnalité.

2)minimalité:
aucune fonction ne doit être offerte si elle n'est pas nécessaire pour des applications d'un certain niveau de fonctionnalité

3)compacitè:
une application doit être capable d'atteindre un rèsultat dèsirè par un ensemble de fonctions et de paramètres aussi petit que possible.

4)un ensemble riche de fonctions offre un large èventail de facilitès dèpassant les fonctions de base et incluant des capacitès d'ordre plus èlevè.

Il est évident qu'il va falloir faire un choix parmi ces principes. C'est pourquoi il est conseillè de grouper les fonctions du standard par niveau.Tandis que le niveau le plus bas contiendra un ensemble minimum de fonctions, les niveaux supèrieurs connaîtront une extension progressive de cet ensemble qui sera d'une richesse de plus en plus grande.

c)conception de l'interface utilisateur:

Les principes suivants sont utilisès pour dèfinir la conception de l'interface utilisateur.

1)commode à l'usage:
le standard doit permettre la conception d'un interface rèpondant aux souhaits de l'utilisateur.

2)clartè:
les capacitès fonctionnelles et conceptuelles du standard doivent être aisèment comprèhensibles; particulièrement par le programmeur.

3)traitement des erreurs:
le traitement des erreurs doit être suffisament explicite pour le programmeur et l'impact de ces drenières aussi faible que possible pour le système et pour le programme d'application.

d)pèriphèriques graphiques:

Les principes suivants sont associès à l'èventail des pèriphèriques graphiques pouvant être adressè par le standard.

1)indèpendance vis à vis des pèriphèriques:
les fonctions du standard doivent être concues afin de permettre au programme d'application, utilisant ses fonctions, d'adresser les facilitès de pèriphèriques d'entrèe et de sortie graphique diffèrents sans modification de la structure du programme.

2)richesse:
toutes les capacitès d'une large gamme de pèriphèrique d'entrèe et de sortie graphique doivent être accessibles à partir des fonctions du standard.

e)implèmentation:

Ce dernier groupe de principes est liè à l'implèmentation.

1)"implèmentabilitè":
il doit être possible de supporter les fonctions du standard dans la plupart des langages de haut niveau, sur la plupart des "operating systems" et sur la plupart des pèriphèriques graphiques.

2)indèpendance vis à vis du langage:
il doit être possible d'accèder aux facilitès du standard à partir de tout langage de programmation standard.

3)efficacitè:
le standard doit pouvoir être implèmentè sans algorithmes coûteux en consommation.

Ces cinq groupes de principes sont interconnectès.

Par exemple, les objectifs de conception et les capacitès fonctionnelles contribuent à la commoditè à l'usage.

L'efficacitè est ègalement importante quand on considère le temps de rèponse dans un environnement interactif.

Certains principes peuvent être contradictoires, tel la richesse par rapport à la minimalitè, le traitement comprèhensif des erreurs et l'efficacitè...

2.2.3. Historique [HOPG.82]

Les standards dans la conception de logiciels graphiques ont mis longtemps pour apparaître. Tandis que la standardisation dans le domaine des langages de programmation est apparue très tôt [Fortran et Algol 60], il a fallu une longue pèriode avant que, au mieux, certains standards locaux aient ètè dèveloppès.

Diffèrentes raisons peuvent expliquer ce retard.La plus importante est liée au changement continuel dans la perception et la conception du graphique interactif.

La majoritè des gros systèmes ètaient auparavant dèdicacès aux ècrans cavaliers.La percée du temps partagè et l'apparition de tubes à mèmoire peu coûteux ont ouvert le marchè à un plus grand public.

Cette tendance s'est encore accentuée lors de l'apparition de terminaux à balayage de trame.

SEILLAC 1

La majorité des activités actuelles dans le domaine de la standardisation ont trouvé leur origine lors de la réunion organisée par l'IFIP à Seillac en mai 1976 (Seillac 1).

Les participants à cette réunion désiraient définir une méthodologie pour l'informatique graphique;ils ont longtemps traité du problème de la prise en compte ou non par un standard de la partie interactive du graphique.Le point de vue de la majorité fût finalement d'insérer cette partie dans les recherches de standardisation.

La principale résolution de Seillac fût d'établir une distinction claire entre la modélisation d'une image et la vue que le système en donne.Cela signifie que le programmeur décrira des objets dans un système de coordonnées qui lui est propre (word coordinate system) indépendamment de la surface sur laquelle l'objet va être représenté (device coordinate system);le passage de l'un à l'autre étant assuré par le logiciel graphique (viewing transformation)

GSPC et GKS

L'enthousiasme développé à Seillac a conduit le GSPC (graphics standard planning committee), crée en 1974, à s'attacher à la conception du "core graphic system".Deux propositions de standard sont alors apparues en 1977 (GSPC 1977) et en 1979 (GSPC 1979). La version de 1979 reprend la description complète d'un système à trois dimensions (core system) et a servi de base à de nombreuses implémentations aux USA (DI3000 par exemple)

Pendant que les américains définissaient le GSPC core system, un groupe allemand (DIN) travaillait sur le "Graphical Kernel System" (GKS) cherchant aussi à définir un système graphique de base mais limité à l'espace à deux dimensions (GKS 7.0).La dernière version fut acceptée par l'ISO (International Standard Organisation) comme proposition de base (draft proposal)

Dans les deux paragraphes qui vont suivre nous allons décrire brièvement les fonctions offertes par ces deux standards en commencant par GSPC.

2.2.4. GSPC [MICH.78]

Les fonctions offertes par le Core System sont classifiées en quatre grandes catégories:les primitives de sortie (output primitives), les transformations de vue (viewing transformation), la segmentation des images, les dispositifs logiques d'entrée de données graphiques (logical input device).

2.2.4.1. Les primitives de sortie et leurs attributs.

Le programmeur d'application décrit un ou plusieurs objets dans son propre monde (modèle) à deux ou à trois dimensions.

Les lignes, les textes, les symbôles qui constituent l'objet sont explicités dans le système de coordonnées utilisateurs par l'intermédiaire de primitives de sortie (tracer une ligne d'un point à un autre , etc).

Une primitive de sortie est définie par les paramètres passés par l'utilisateur(coordonnées...), une transformation de vue (cfr point 2.2.4.2 et par les valeurs courantes des attributs des primitives.

Les attributs des primitives de sortie sont le type de ligne (continu, pointillé...), la largeur des lignes, l'intensité, la couleur, les types de caractères, l'espacement des caractères, la qualité des caractères, le plan des caractères et l'identifiant d'une sélection (pick- identifier).

Le Core System diffère des logiciels graphiques au sein desquels tous les attributs d'une primitive de sortie sont déterminés par l'intermédiaire de paramètres.

Dans le Core System, chaque attribut de primitive a une "valeur courante" que le programme d'application modifie par l'appel d'une fonction spéciale. L'avantage d'un tel système est de limiter le nombre de paramètres à passer lors de l'appel d'une primitive.

2.2.4.2. Transformation de vue (viewing transformation)

Le programme d'application décrit la vue d'un objet par la spécification d'une transformation de vue.

Une transformation de vue sélectionne une région du monde graphique à afficher et spécifie la facon dont les objets se trouvant dans la zone sélectionnée doivent être projetés sur la surface de vue.

La surface de vue est un rectangle (ou un carré) se trouvant sur la surface des périphériques utilisés.

Le système de coordonnées spécifiant les positions sur la surface de vue est appellé système de coordonnées normalisé des périphériques.

Conceptuellement, une transformation de vue décrit la position d'une caméra qui doit prendre un instantané d'un objet.

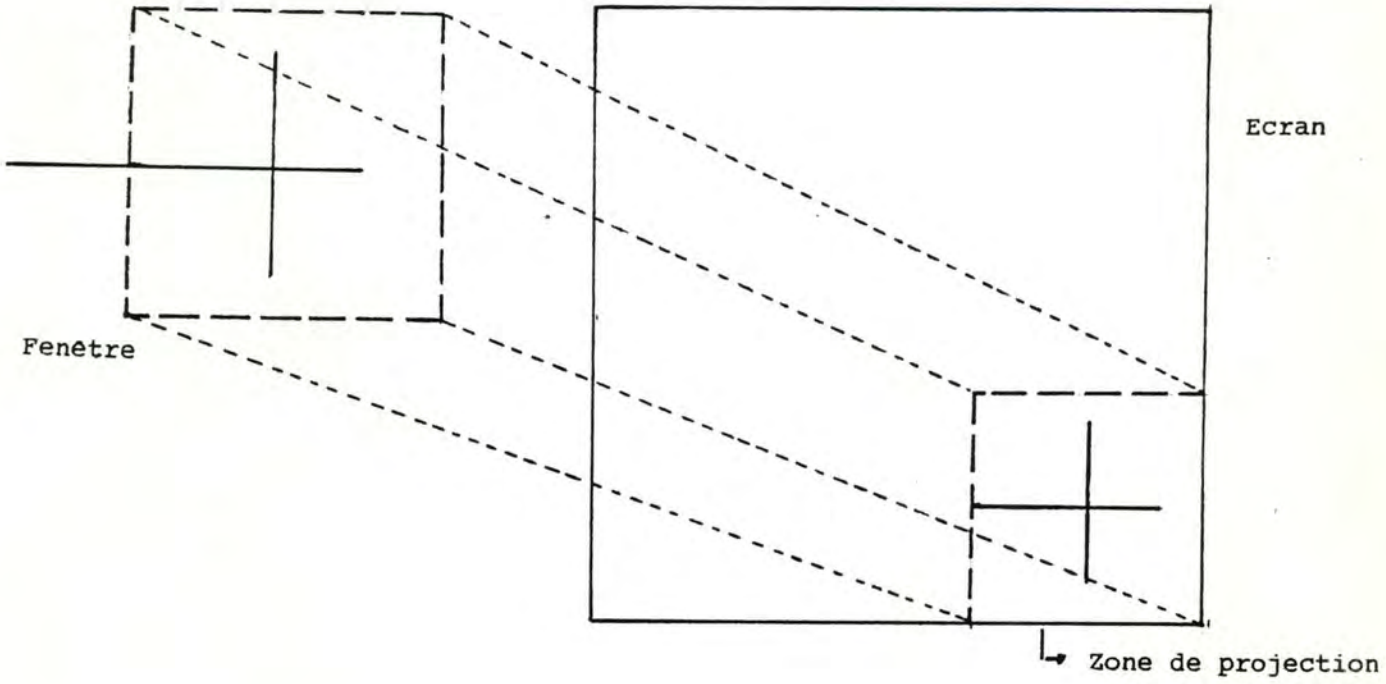
Le Core System traite l'espace à deux dimensions comme un sous-ensemble de l'espace à trois dimensions.Le programmeur travaillant dans l'espace à deux dimensions peut ignorer complètement les éléments intervenant dans la troisième dimension.

Dans l'espace à deux dimensions, une transformation de vue est entièrement spécifiée par une fenêtre dans le système de coordonnées utilisateurs et par une zone de projection (viewport) sur la surface de vue.

La fenêtre peut être tout rectangle dans le plan déterminé par les axes X et Y du système de coordonnées utilisateur.

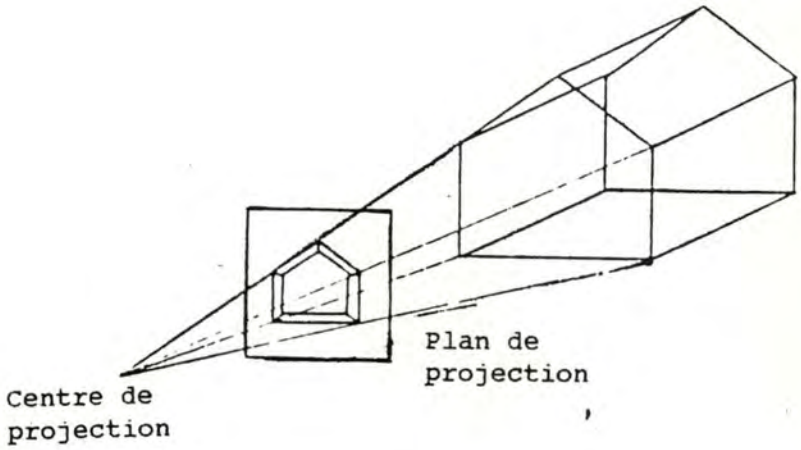
La zone de projection est un rectangle spécifié dans le système de coordonnées normalisé (0-->1 pour X et 0-->1 pour Y) dont les côtés sont parallèles aux côtés horizontaux et verticaux de la surface de vue.

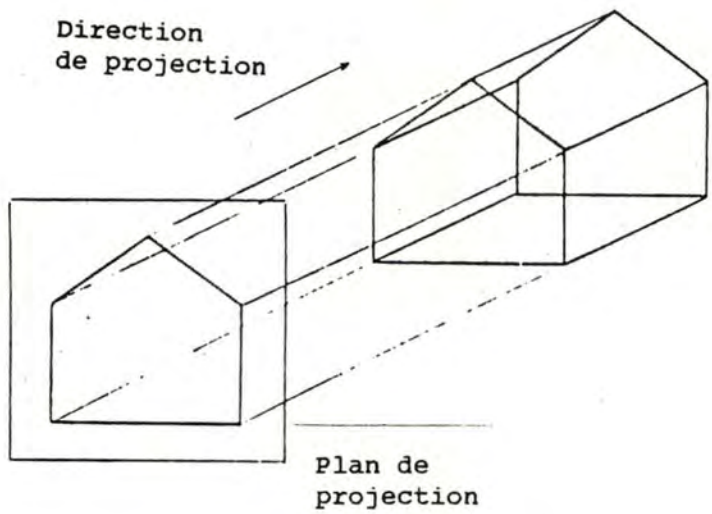
Une fenêtre est utilisée pour "couper" un objet, c'est-à-dire pour déterminer la portion de l'objet devant être visualisée. Après la coupure, cette portion est projetée sur partie de la surface de vue déterminée par la zone de projection.



Dans l'espace à trois dimensions, la situation est plus compliquée puisque la transformation de vue doit inclure une projection de l'espace à trois dimensions vers l'espace à deux dimensions en plus de la coupure et de la projection sur la surface de vue.

Dans le Core System, une projection de l'espace à trois dimensions vers l'espace à deux dimensions est spécifiée, dans le monde graphique du programmeur, par un plan de projection et par un centre de projection ("l'oeil") pour une projection en perspective et une direction de projection pour une projection parallèle.





Deux coupures peuvent être réalisées dans l'espace à trois dimensions.

La première est une coupure "en profondeur".Le programmeur définit deux plans (un vers l'avant et un vers l'arrière) et seuls les objets ou parties d'objets situés entre ces deux plans seront projetés.

La deuxième consiste à définir une fenêtre dans le plan de projection .

La projection de la fenêtre vers la zone de projection est semblable à celle rencontrée dans l'espace à deux dimensions.

2.2.4.3. Segmentation et modification d'image

Un dessin apparaissant sur la surface de vue est défini par un ou plusieurs segments.

Un programme d'application utilise des segments différents afin de pouvoir changer une ou plusieurs parties d'un dessin.

Les changements au sein d'un dessin apparaissent lorsque des segments sont construits à partir de primitives de sortie, lorsque des segments sont supprimé ou lorsque les attributs dynamiques (décrits ci-dessous) des segments ont changé.

Un segment est créé en ouvrant un nouveau segment, en faisant appel à des primitives de sortie à l'intérieur de ce nouveau segment et, enfin, en fermant le segment.

Un segment correspondra à une image, à une unité considérée comme un tout dans un dessin et pouvant éventuellement apparaître et disparaître (un menu par exemple).

Deux types de segments peuvent être créés.

- Les segments retenus sont utilisés lorsque une image peut être affichée plusieurs fois et peut être sujette à des modifications via les attributs dynamiques des segments.

  Les primitives de tels segments sont enregistrées jusqu'au moment où le segment aura été explicitement supprimé.

- Les segments non retenus sont utilisés lorsque une image ne doit être affichée qu'une seule fois.

Les attributs dynamiques des segments retenus sont:

- la visibilité:selon que l'image soit visible ou invisible
- la luminosité:détermine la luminosité de l'image
- détectabilité:dit si oui ou non une image peut être sujette à détection par un dispositif logique de sélection (pick)
- transformation d'image:permet d'effectuer un déplacement, un agrandissement ou une rotation d'images.

2.2.4.4. Dispositifs logiques d'entrée.

L'introduction de la notion de dispositifs logiques est due à la volonté du Core System de faire abstraction des propiétés des périphériques physiques.

Cinq classes de dispositifs logiques d'entrées sont supportées par le Core System:

- dispositif logique de sélection (pick):
  permet à l'opérateur de sélectionner une primitive de sortie, notamment à l'aide d'un crayon lumineux.Le nom du segment associé sera rendu au programme d'application.
- dispositif logique de positionnement (locator):
  utilisation d'une tablette ou d'un "joystick".
- valuator:
  permet d'introduire une valeur réelle par l'intermédiaire notament de boutons rotatifs .
- button:
  indique qu'une certaine touche a été sélectionnée.

Pour terminer cette étude du GSPC Core System, précisons que le système est divisé en quatre niveaux ;selon la complexité des fonctions utilisées, le programmeur travaillera avec le premier, deuxième, troisième ou quatrième niveau.

2.2.5. GKS [GKS.82, HOPG.82]

Nous allons décrire le "graphical kernel system" en soulignant essentiellement ses différences par rapport au GSPC Core System.

Rappelons que le standard proposé ne concernait que les applications graphiques dans l'espace à deux dimensions.

2.2.5.1. Station de travail.

Le concept central du GKS est la station de travail graphique composée d'une seule unité d'affichage et d'un certain nombre de périphériques d'entrée.

La station de travail est définie comme appartenant à l'un des types standards (table tracante, écran à tube mémoire, écran à rafraîchissement...) ce qui permet au programme d'adopter son comportement en fonction du domaine d'application.

Un opérateur peut avoir en même temps plusieurs stations de travail sous son contrôle .Il peut demander notamment l'impression d'un large dessin sur une table tracante tout en observant une partie de ce même dessin sur écran.

Le programmeur d'application a de nombreuses libertés dans l'usage de chacune des stations.Différentes stations peuvent être utilisées pour montrer simultanément plusieurs parties d'un même dessin.

2.2.5.2. "Plume" (pen)

Les primitives graphiques tel le tracé d'une ligne peuvent se voir associer des attributs tel une couleur , une luminosité, un type de ligne...

Deux approches dans la spécification de tels attributs sont possibles.

La première est d'avoir un ensemble d'attributs modaux conservant leur valeur courante jusqu'au changement explicite suivant.Il s'agit là de la méthode conventionnelle de spécification d'attribut utilisée par le GSPC Core System.

Par exemple:

```
couleur (rouge)
épaisseur (large)
type-ligne (pointillée)

    tracer une ligne

couleur (gris)
type-ligne (continue)
```

Ces appels vont tracer deux lignes.La premiere sera rouge, large et pointillée;la deuxieme sera grise, large et continue.

Un attribut particulier reste d'effet jusqu'au moment où il est modifié (ex:épaisseur "large").

Un désavantage lié à cette approche est le besoin d'adapter ces spécifications d'attributs à des périphériques ne pouvant implémenter certains attributs.

Comment tracer une ligne rouge sur un terminal à mémoire ? On laisse en général le soin au constructeur du "device driver" de prendre une décision arbitraire.

Un autre désavantage provient de l'obligation, pour le programmeur, d'accroître la complexité de ses algorithmes par la modification de nombreuses valeurs d'attributs en fonction des exigences de l'utilisateur (exemple:alterner successivement deux types de spécification:ligne rouge, pointillée et large <--> ligne bleue, continue et étroite).

La solution adoptée par GKS n'est plus d'avoir un certain nombre d'attributs modaux mais plutôt d'avoir un attribut majeur par primitive, un numéro de plume.Chaque primitive peut se voir associer une plume parmi l'ensemble des plumes définies.

L'équivalent GKS du programme donné dans le dernier exemple ressemblera à:

```
Pen (1)
Draw line
Pen (2)
draw line
```

On tracera sur un périphérique une première ligne avec la plume 1 et une seconde avec la plume 2.La définition des plumes 1 et 2 dépendra de la station de travail et sera établie par le programmeur d'application.

Ainsi, ce dernier pourra décider d'avoir une plume 1 rouge, large et pointillée et une plume 2 grise, large et continue.

2.2.5.3. Primitive de sortie

GKS définit six primitives de sortie:

- Polyline
- Polymarker
- Text
- Fill area
- Pixed away
- Generalized drawing primitive

Une importante différence entre GKS et GSPC est que dans GKS la notion de position courante n'existe pas.Chaque primitive voit ses coordonnées entièrement définies par les paramètres.

a.Polyline:

permet de tracer un ensemble de lignes connectées.

Constatant que la plupart du temps le programmeur désire tracer un ensemble de lignes pour former une figure, GKS a étendu la primitive "line" de GSPC.

b.Polymarker:

permet de marquer une suite de point par un symbole. Il s'agit là d'une extension (par rapport à GSPC) semblable à la précédente.

c.Text:

permet d'écrire une chaîne de caractères.

Les trois dernières primitives sont moins souvent utilisées et témoignent de l'influence des terminaux à balayage de trame sur le graphique ainsi que du besoin de profiter, même au sein d'un standard, de facilités matérielles coûteuses.

d.Fill area:définit les limites d'une surface dont l'intérieur doit être hachuré ou colorié.

e.Pixel area:moyen utilisé pour spécifier une suite (tableau) de points.

f.Generalized drawing primitive:
fonctions permettant le tracé de cercles, de courbes...

2.2.5.4. Segments

Les segments sont mémorisés dans les stations de travail actives au moment de leur définition.

Cela s'avère suffisant dans beaucoup de cas;cependant il arrive que l'on désire voir apparaître un segment sur une station qui n'était pas active lors de sa création.

L'utilisateur peut très bien définir sur un terminal à balayage de trame un dessin formé de segments et, par la suite, désirer en avoir une copie sur une table tracante.

A cette fin, GKS enregistre une copie des segments.Les copies peuvent être retirées et envoyées vers n'importe quelle station.

Ajoutons encore la possibilité d'insérer un segment dans un autre segment.

2.2.5.5. Vues (viewing).

GKS a trois systèmes de coordonnées différents:

Le programmeur d'application décrit ses dessins dans un système de coordonnées utilisateur. Une première projection est réalisée sur une partie du plan du système de coordonnées normalisé du périphérique.

Une deuxième projection fera la correspondance entre ce système de coordonnées normalisé et le système de coordonnées physiques du périphérique.

La notion de fenêtre interviendra lors de la première projection; la notion de zone de projection (viewport) lors de la deuxième.

2.2.5.6. Entrées de données graphiques

GKS adopte un système de classification en dispositifs logiques à peu près semblable à celui de GSPC.

Chapitre 3 . APPLICATIONS GRAPHIQUES

Dans ce chapitre, nous parlerons des usages representatifs du graphique et de quelques criteres de classification de ses applications.

Ensuite, nous nous attarderons plus particulièrement sur l'utilisation du graphique en gestion vu l'ampleur prise par ce domaine d'application.

3.1. Usages representatifs du graphique [FOLE.82]

L'utilisation de l'informatique pour realiser des graphiques est de plus en plus courante dans de nombreux domaines tel l'industrie, la gestion, l'education...

La liste des applications est enorme et croît rapidement vu la diminution du coût des terminaux.

Nous enoncons ci-dessous quelques exemples representatifs.

3.1.1. Tracé graphique [interactif] en gestion, science et technologie

Le graphique est très souvent utilisé aujourd'hui afin de tracer des courbes de fonctions économiques, mathématiques, physiques; afin de dessiner des histogrammes, des ''tartes''...

Tous ces usages visent à présenter d'une manière claire et concise les tendances, les informations sous-jacentes à un ensemble de données et à accroître ainsi la compréhension de phénomènes complexes.

3.1.2. Cartographie

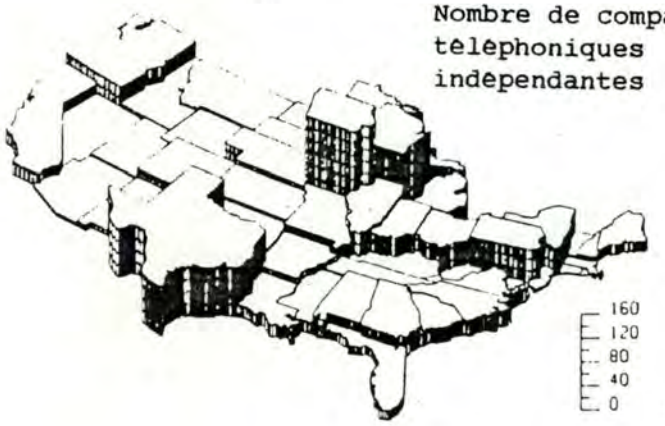
L'outil informatique peut également être utilisé pour obtenir une reproduction précise de répartitions géographiques. Citons les cartes géographiques, les cartes de relief, les cartes météorologiques, les cartes de densités de population...

(a)

Téléphones pour
100 personnes

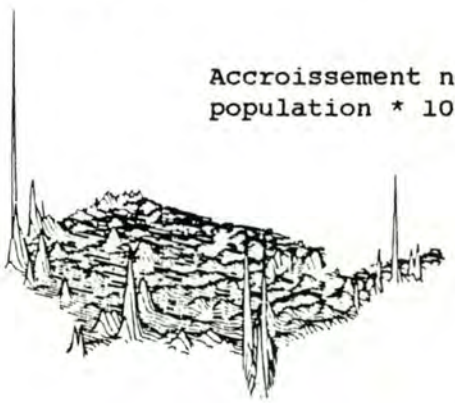
(b)

Nombre de compagnies
téléphoniques
indépendantes



(c)

Accroissement net de la
population * 10 (USA 1970-1979)



(d)

3.1.3. Conception assistée par ordinateur

En CAO, on utilise le graphique interactif afin de concevoir des composants ou des systèmes mécaniques, électriques, électro-mécanique ou électroniques.

Ces systèmes peuvent être des structures (tel un building, une usine chimique, la coque autoporteuse d'une voiture, la coque d'un avion), des systèmes optiques, des réseaux téléphoniques ou informatiques.

Sur base d'un modèle du composant ou du système, l'ordinateur permet au concepteur de réaliser des tests concernant ses propriétés mécaniques, électriques ou thermiques.

Très souvent, le modèle est interprété par un simulateur montrant le comportement du système sur une console et favorisant ainsi le processus interactif de test et de conception.

3.1.4. Simulation et animation

On utilise de plus en plus fréquent des séquences animées, produites par ordinateur, afin de décrire le comportement au cours du temps d'objets réels ou simulés. En visualisant les effets de transformations, nous pouvons étudier des phénomènes scientifiques tels les flux hydrauliques, les réactions chimiques, les déformations d'une structure sous une charge.

Un nouveau domaine d'application à la pointe de la technique est la réalisation de dessins animés exigeant des images de très grande qualité.

Une autre application sophistiquée de l'animation est le simulateur de vol. Le simulateur génère non seulement des vues du monde fixe au-dessus duquel le véhicule se déplace mais également des vues d'effets spéciaux tels les nuages, le brouillard, les lumières de la nuit, et les autres avions.

On peut également simuler les opérations d'un système matériel ou logiciel et visualiser graphiquement les interactions entre les composants ainsi que les changements de valeurs.

Citons enfin, à niveau de performance nettement moindre, la large gamme de jeux vidéos disponibles actuellement sur le marché.

3.1.5. Contrôle de traitement

A l'encontre du simulateur de vols ou des jeux vidéos qui laisse l'utilisateur réagir avec la SIMULATION d'un monde réel ou imaginaire, de nombreuses autres applications permettent à l'utilisateur d'être en contact avec certains aspects du monde réel lui-même.

Les écrans de contrôle dans les raffineries, dans les centrales nucléaires ou dans les réseaux informatiques affichent des données obtenues à partir de senseurs attachés aux composants critiques du système;l'opérateur répond alors à des conditions exeptionnelles.

Dans les aéroports, l'ordinateur visualise, à partir de données fournies par les radars, la situation des différents avions sur la console du controleur aérien qui peut ainsi diriger aisément le traffic.

3.1.6. Art graphique

La création de dessins par l'intermédiaire de l'outil informatique est de plus en plus considéré par certain comme un art.Des mécanismes très sophistiqués sont offerts au créateur du dessin afin de modéliser les objets et afin de représenter la lumière et les ombres.

3.2. Classifications des applications [FOLEY.82]

On peut utiliser divers critères pour classifier les applications graphiques

3.2.1. Le type d'objet ou d'image à produire

Nous pouvons distinguer :
- les dessins d'objets bidimentionnels
- les dessins d'objets tridimentionnels
- les dessins d'objets tridimentionnels avec effacement des arêtes cachées
- les images couleurs bidimentionnelles
- les représentations tridimentionnelles d'objets solides et ombragés avec effacement des surfaces cachées...

On constate que:
- certains objets sont abstraits, d'autres réels.

- une image peut être symbolique (un simple graphique à deux dimensions) ou très réaliste.

3.2.2. Le type d'interaction

Intervient ici le degré de contrôle de l'utilisateur sur l'image.

- Impression d'un graphique à partir d'une base de données par un programme d'application ne demandant aucune intervention extérieure (offline plotting)
- Impression interactive où l'utilisateur fournit des paramètres, fait imprimer, modifie les paramètres, refait imprimer.
- Prédéfinir un objet et voler autour de lui en temps réel sous contrôle de l'utilisateur (cfr simulateur de vol)
- Conception interactive:
  l'utilisateur part d'un écran vierge, crée un objet à partir de composants prédéfinis, le modifie selon sa volonté, effectue des déplacements et des agrandissements .

3.2.3. Le rôle de l'image

La production d'une image peut être une fin en soi ou un moyen pour atteindre cette fin.

En cartographie, le dessin constitue le produit de base tandis que dans la majorité des applications de CAO le dessin constitue une simple visualisation des propriétés géométriques de l'objet analysé.

3.2.4. La relations entres les objets et leurs images

L'utilisateur peut tantôt travailler avec une seule image (impression), tantôt avec une collection structurée d'objets (CAO).

3.3. Le graphique en gestion

Le marché de l'informatique graphique, dominé par les applications industrielles et scientifiques, n'accusait chaque année qu'une lente croissance.Avec l'apparition des terminaux couleurs, un nouveau marché vient de s'ouvrir:celui de la gestion.

Les entreprises s'empressent aujourd'hui d'acquérir inconsidérément un système graphique.

Mais bien souvent la technologie ne leur permet pas d'atteindre les niveaux de satisfactions vantés par ses promoteurs.

Les raisons sont variées.Peut être la fonctionnalité a-t-elle été exagérée ? Plus vraisemblablement la technique, qui paraissait si éclatante lors de la démonstration du vendeur, s'est elle embourbée dans la gadoue des politiques organisationnelles ou des systèmes existants ? Enfin le temps d'apprentissage a-t-il été plus long que prévu ?

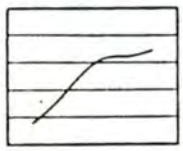
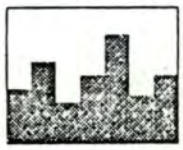
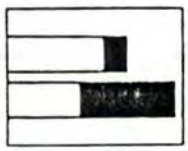
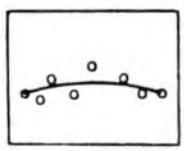
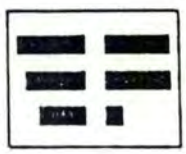
La première partie de cette section sera consacrée aux différentes représentations graphiques traditionnelles ou non en gestion.

Une seconde partie traitera des éventuels bénéfices que peut apporter l'utilisation du graphique dans le cadre de la prise de décision.

3.3.1. Représentations graphiques en gestion

Le recours aux graphiques ne constitue pas en lui-même une innovation en gestion.La nouveauté réside dans l'apparition de techniques informatiques permettant de produire des dessins plus rapidement et à un meilleur prix.Leurs utilisations vont s'étendre et toucheront les niveaux inférieurs dans la hiérarchie des pouvoirs.

Les dessins produits par ordinateur sont, en général, semblables à ceux réalisés manuellement.L'éventail classique reprend les "tartes", les histogrammes, les graphiques linéaires ou curvilignes. Un exemple de ces différentes représentations apparaît ci-dessous [BLAI.82].

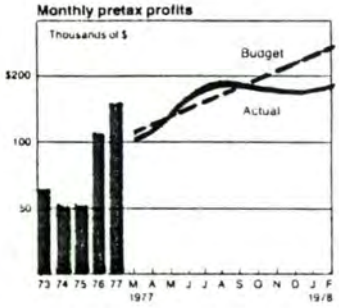
TYPE D'INFORMATION	TYPES DE REPRESENTATION				
	"Tarte"	Courbe	Histogramme	Colonne	Carte
Série chronologique					
Pourcentage					
Comparaison d'informations Comparaison géographique					
Relation entre variables					

Certaines variantes appliquées sur ces schémas de bases peuvent s'avérer utiles.

Blake, par exemple, présente un ensemble de diagrammes mettant en lumière les performances de l'organisation sur base d'un certain nombre d'indicateur [BLAG.78].

Monthly sales

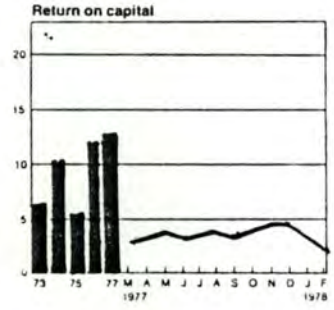
Monthly pretax profits



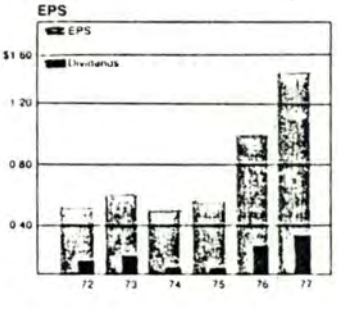
Division contribution to pretax profits (12 months)

$000	Actual	Variance
S.M.C.	930	15%
R.T.	641	14
H.L.	468	+ 4

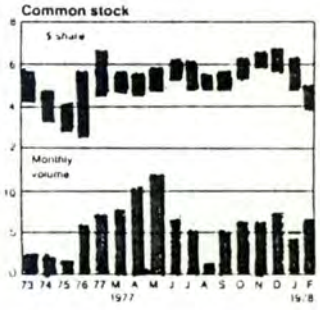
Return on capital



EPS

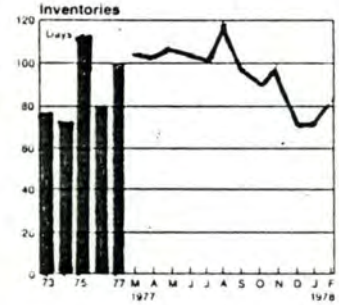


Common stock

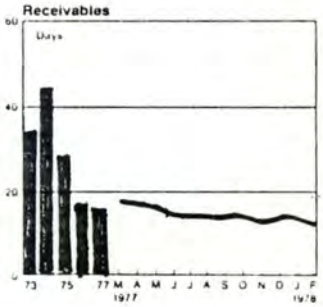


Annual turnover

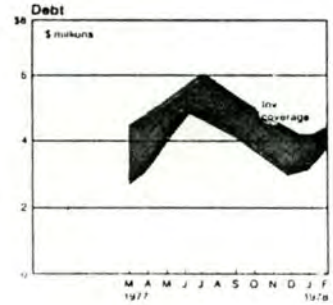
Inventories



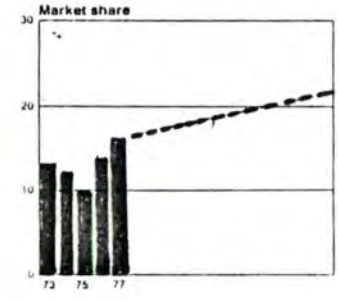
Receivables



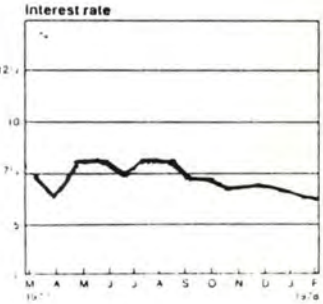
Debt



Market share

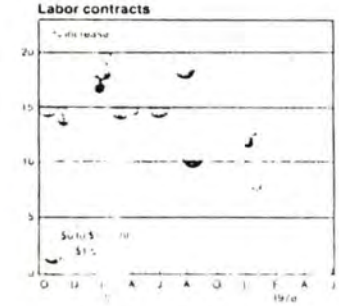


Interest rate

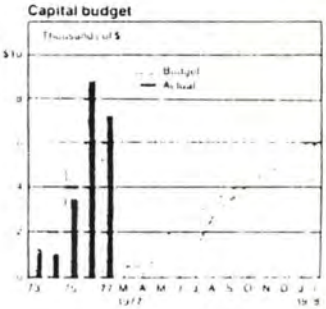


Overhead

Labor contracts



Capital budget



Parmis les dessins de bases, nous avons également repris la carte qui , vu sa complexité, bénéficie probablement plus de l'apport de l'outil informatique.

Les études menées dans le domaine de la cartographie, plus précisément de la cartographie interactive, ont pour but de permettre la représentation, sur des cartes géographiques, de données de nature les plus diverses mais ayant une dimension spaciale.

Il devrait également être possible de superposer différentes cartes ainsi obtenues pour apprécier les répartitions conjuguées de plusieurs d'entre elles ou , le cas échéant , pour mettre en valeur des corrélations entre certains paramètres.

Par ailleurs, il devrait être possible d'analyser avec plus de précision une zone présentant des caractéristiques particulières grâce à un zoom ou encore, par simple désignation de cette zone sur la carte globale au moyen d'un crayon lumineux, d'obtenir l'affichage des informations de synthèse concernant cette zone...

3.3.2. Graphique et productivité du décideur [BLAI.82, JANS.82, MONI.79]

De plus en plus souvent, les "décideurs" doivent faire face à des situations complexes, incertaines et changeantes.

Il leur serait donc utile de disposer de facon rapide et sous forme claire d'un certain nombre d'informations leur permettant d'analyser les situations et de guider leur choix.

Les moyens traditionnels informatiques n'offrent guère de facilités pour rapprocher les données entre elles et encore moins pour les présenter sous une forme synthétique directement compréhensible. Nous ne nous attarderons pas sur les tonnes de listings d'où les responsables d'entreprises sont censés extraire les informations significatives qu'ils cherchent, ni sur les batailles de ces mêmes responsables avec les terminaux qui en disent toujours trop ou pas assez.

Aussi d'aucuns ont affirmé que l'émergence des systèmes informatiques graphiques pourrait être aussi importante en ce qui concerne la productivité du gestionnaire que l'apparition de l'informatique elle-même.

Le graphique offrirait deux grands avantages aux managers :

- il leur permettrait d'abord de faire des économies en ce qui concerne l'une de leurs ressources les plus importantes : le temps.
  Ces économies interviendront lors de l'interprétation et lors de la communication des données.
- il les aiderait ensuite à mieux remplir leur fonction essentielle : la prise de décisions.
  Deux éléments différents sont généralement pris en compte a ce niveau :

  a) Des informations visuelles pouvant être plus facilement assimilées, le manager sera capable d'acquérir en peu de temps un grand nombre de connaissances.

  b) Les tendances ou les exceptions seront facilement perceptibles.

Devant cet enthousiasme, il semble nécessaire d'étudier plus profondément les impacts du graphique sur la productivité du décideur et sur son processus de décisions.

Nous parlerons ainsi des recherches comparant présentation graphique et non graphique et des recherches comparant présentation couleur et achromatique.

3.3.2.1. Graphique et productivité

Réaliser une comparaison objective entre une représentation graphique et une représentation tabulaire ou narrative constitue une tâche très complexe .

La raison essentielle est liée à la difficulté de trouver la version graphique d'un rapport non graphique.

Il est pour ainsi dire impossible que les informations contenue dans un rapport graphique soient équivalentes à celles contenues dans un autre rapport.
De plus les deux représentations devraient être une illustration de ce qu'il y a de mieux dans leur discipline respective.

Notons aussi que la recherche est confrontée au fait que les graphiques sont annotés par une certain nombre d'informations narratives ou mêmes tabulaires.

De nombreuses études contournent ces problèmes pour arriver à des conclusions contradictoires.Une préférence est donnée tantôt à la version graphique ,

tantôt à la version tabulaire.

Il est cependant important de remarquer que les décideurs accompliront d'autant mieux leur tâche que le format des informations manipulées leur est familié.
Le passage d'une forme tabulaire à une forme graphique nécessitera donc une période d'apprentissage...

3.3.2.2. Couleur et productivité

Une grande majorité des travaux sur la couleur ont été réalisés par des spécialistes désirant étudier son impact sur des tâches tel l'identification correcte et rapide d'un objectif.

La rapidité d'identification d'un objectif semble certes un facteur très important pour un pilote d'avion, mais l'est-elle tout autant pour un manager ?

Ce dernier va-t-il retirer un bénéfice significatif grâce à la perception une seconde plutôt d'une information particulière ?

D'autres critères peuvent être pris en compte.Citons :

- la qualité de la décision
- le temps de décision
- la satisfaction du décideur
- le temps d'assimilation
- le temps de rétention
- la capacité à attirer l'attention
- la capacité à maintenir l'attention
- la fatigue visuelle

Les études concernant les impacts de la couleur sont nombreuses. Elles peuvent porter sur un ou plusieurs de ces critères.

Nous reprenons ci-dessous les conclusions d'auteurs qui ont tenté d'en faire une synthèse.

Christ a revu 42 études examinant l'impact de la

couleur comme stimulant dans la localisation et l'identification d'informations. Il concluait en disant :

> "Si le but des utilisateurs est de receuillir des informations en fonction d'un objectif précis, la couleur apparait plus efficace que les changements de tailles, de luminosités, de figures géométriques. Néanmoins elle sera toujours inférieure aux symbôles alphanumériques".

Christ note aussi que les utilisateurs estimaient l'écran couleur supérieur à l'écran noir et blanc bien que rien ne démontre sa meilleure efficacité.
Selon lui, l'opinion générale est que la couleur diminue la monotonie et la fatigue.

Barker et Krebs ont examiné 78 études considérant les effets des couleurs sur les performances humaines.
Ils concluent :

> "L'efficacité relative de la couleur dépend de la tâche accomplie par l'utilisateur.Le recours à la couleur semble plus efficace lorsque la position de l'information recherchée est inconnue."

Chute examine l'impact des couleurs sur l'assimilation et constate que celui-ci n'est guère significatif.

Watson, à l'issue d'une expérience sur des étudiants universitaires, conclut que la rétention d'informations n'est pas améliorée par l'usage de graphiques et de couleurs.

Robertson pense que le terminal couleur est sans aucun doute supérieur au terminal monochrome pour des applications de gestion.
Il décrit différentes études démontrant l'accroissement de la productivité avec l'usage de la couleur;mais conclut par ailleurs qu'il n'y a pas de supports émérites pour la couleur.

> "L'avantage de la couleur paraît si évident que l'on n'éprouve guère le besoin de le prouver".

3.3.2.3. Conclusions

Selon ses promoteurs, l'usage du graphique s'accompagnerait d'un bon prodigieux dans la productivité du décideur.

Généralement, ces prétentions ne sont pas modestes et suggèrent que l'amélioration du décideur est si évidente qu'une rigoureuse démonstration est superflue.

En réalité, les recherches réalisées ont amené à des découvertes équivoques.Bien qu'il faille encore aller plus loin dans ces études, les résultats mitigés déjà obtenus tendent à prouver que l'usage du graphique et de la couleur ne garantit pas un rehausement dans la productivité du décideur.

Au contraire, un mauvais usage du graphique et de la couleur risque d'engendrer des effets opposés...

PARTIE 2 : LE GRAPHIQUE DANS LE CADRE D'UN LOGICIEL D'AIDE AU DEVELOPPEMENT D'UN SYSTEME D'INFORMATION.

Introduction

Cette deuxième partie commencera par une description du graphique dans le cadre d'une méthodologie de développement d'un SI.

Nous parlerons ensuite des outils graphiques disponibles sur ISDOS-IDA, logiciel d'aide au développement d'un SI.

Chapitre 1 . LE GRAPHIQUE DANS UNE METHODOLOGIE DE DEVELOPPEMENT D'UN SI.

Nous rappellerons, dans ce chapitre, la définition d'une méthodologie de développement d'un SI et nous décrirons, selon leur rôle, les différentes représentations graphiques qui lui sont associées

1.1. Méthodologie de développement d'un SI [BODA.83].

F. Bodart et Y. Pigneur ont défini une méthodologie de développement d'un Système d'information (SI) comme un ensemble de règles et de propositions générales relatives à l'emploi de ressources en vue de maîtriser les différentes étapes du cycle de vie d'un système d'information.

Ils précisent la signification donnée aux éléments principaux de cette définition de la manière suivante:

Règles et propositions générales :

"Une méthodologie est vue comme un ensemble de règles et de propositions générales définissant comment mettre en oeuvre des ressources pour la résolution d'une classe de problèmes, sans tenir compte de leurs particularités propres".

Ressources :

Ils distinguent 3 classes de ressources:

1. Les ressources managériales, relatives à la gestion technique et financière du développement d'un projet et à la gestion des impacts organisationnels.

2. Les modèles de représentation du SI, permettant de représenter les différents aspects du SI selon les niveaux d'abstraction appropriés (cfr le modele E/A..).

3. Les techniques et outils informatiques d'aide au développement d'un SI qui doivent servir de support aux fonctions de spécification, de génération, de contrôle, de documentation, d'évaluation, de prévision...

Les outils de représentation graphique joueront un rôle important en ce qui concerne les trois dernières fonctions citées.

Maîtriser les différentes étapes du cycle de vie d'un SI.

La démarche méthodologique retenue s'intéresse essentiellement aux contrôles et aux décisions relatifs au contenu d'un SI à différents moments de son cycle de vie.

Les décisions peuvent être :

- fonctionnelles : définitions des activités du SI nouveau. (règles de traitement, mémorisation des informations...);
- technico-économiques : choix technologiques relatif à la mise en oeuvre des décisions fonctionnelles;
- organisationnelles : changements à introduire dans la structure de l'organisation...

1.2. Rôle du graphique dans une méthologie de developpement d'un SI.

Différentes formes de représentation graphique interviennent dans le cadre d'une méthodologie de développement d'un SI. Leur rôle est d'offrir un outil de communication aisé et compréhensible pour les diverses personnes interessées : utilisateur,concepteur,informaticien...

La conception d'un graphique peut intervenir soit lors de la diffusion de résultats , soit lors de la saisie des informations.

1.2.1. Diffusion de résultats.

Les exemples de représentation graphique réalisés à partir de données recueillies lors de la conception d'un SI peuvent tantôt s'appuyer sur des modèles de la méthodologie, tantôt servir d'outils d'évaluation.

1.2.1.1. Représentations graphiques basées sur les modèles de la méthodologie.

Lors de la définition d'une méthodologie de développement d'un SI, nous avons précisé que la démarche s'appuyait sur un certain nombre de modèles. Ces modèles, dans un souci de communicabilité, se sont vu associer des représentations graphiques.

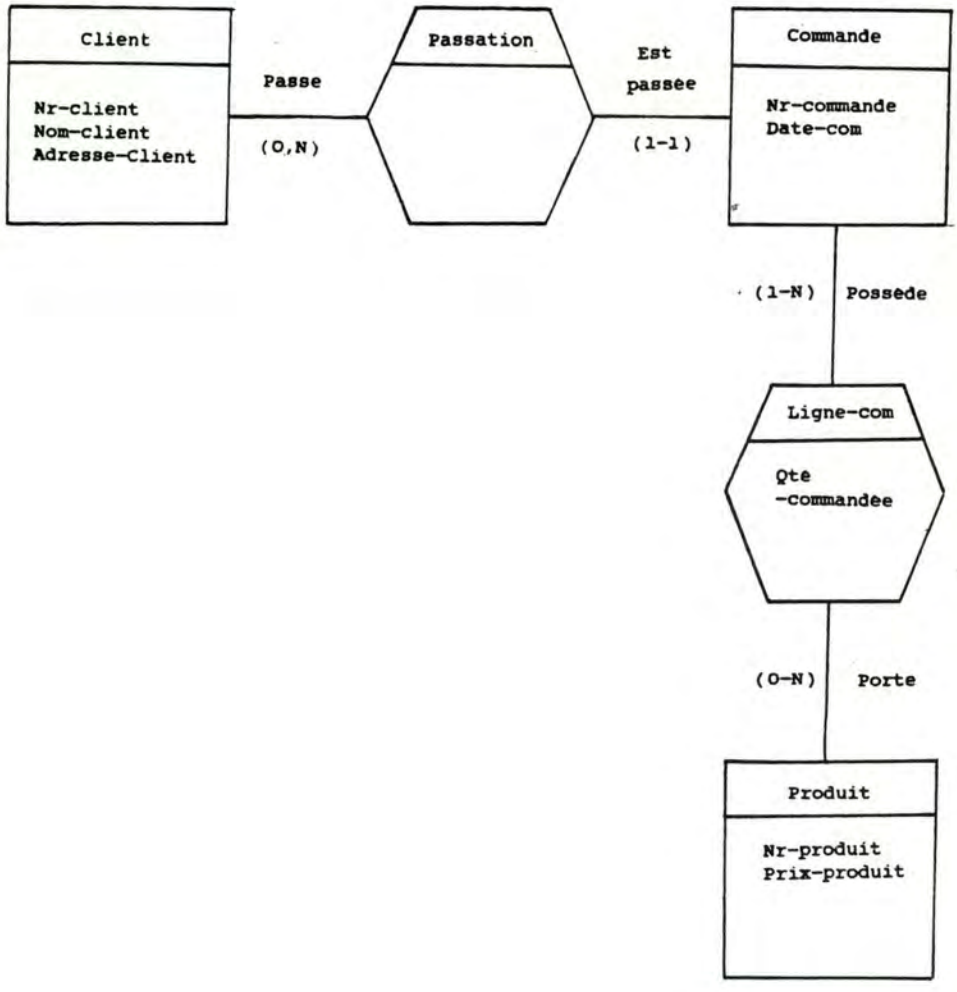
a) le schéma conceptuel.

Parmi les nombreux modèles de structuration des informations, le modèle Entite-Association rencontre une audience croissante parmi les spécialistes des SI et des bases de données. Il propose de modéliser les informations du réel percu à partir de 3 concepts élémentaires :

- entité
- association
- attribut.

La capacité d'expression de la signification fournie par ces 3 concepts de base sera étendue par la notion de contrainte d'intégrité [BODA.83].

La représentation graphique d'un schéma conceptuel des informations basée sur le modèle entité-association se présente sous la forme d'un réseau:



Par convention, on représente:

- Un type d'entité par un rectangle comportant une cartouche où figure le nom du type d'entité.

- Un type d'association par un hexagone relié par des segments de droites aux rectangles qui représentent les types d'entité sur lesquels est defini ce type d'association. Dans la cartouche supérieure de l'hexagone, on indique le nom du type d'association.

  Sur la patte qui relie un type d'entité à l'hexagone, on indique le nom du rôle joué par le type d'entité dans le type d'association.

  En dessous de la patte, on mentionne le couple définissant les valeurs de la connectivité du type d'association pour le type d'entité concerné.

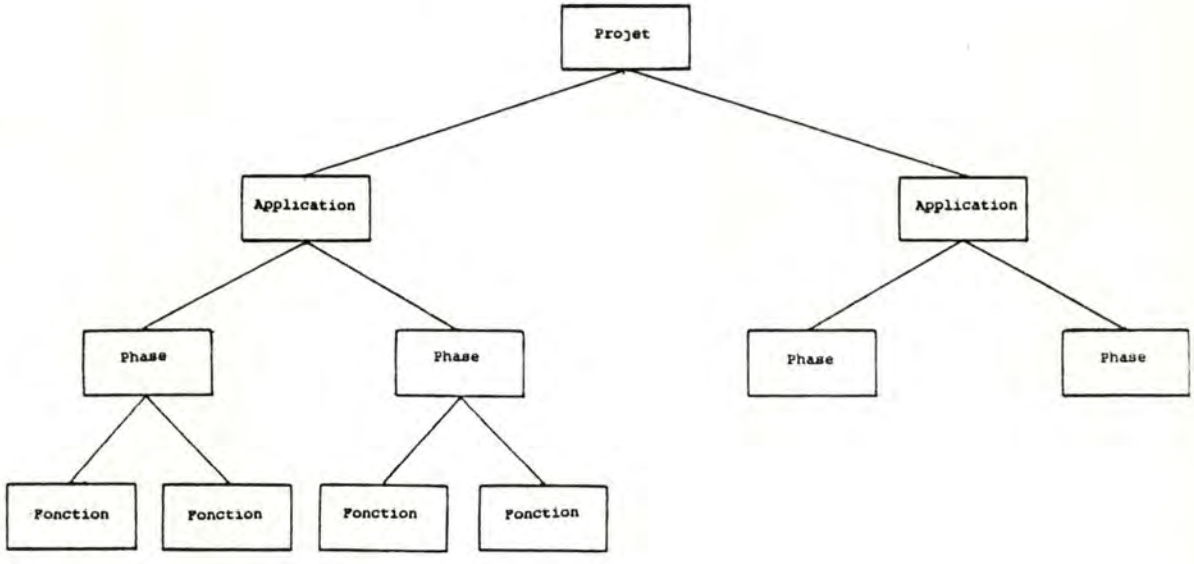
- Le nom d'un attribut d'un type d'entité ou d'association sera inscrit dans la partie inférieure du rectangle ou de l'hexagone qui représente le type d'entité ou d'association.

b) Architecture des traitements.

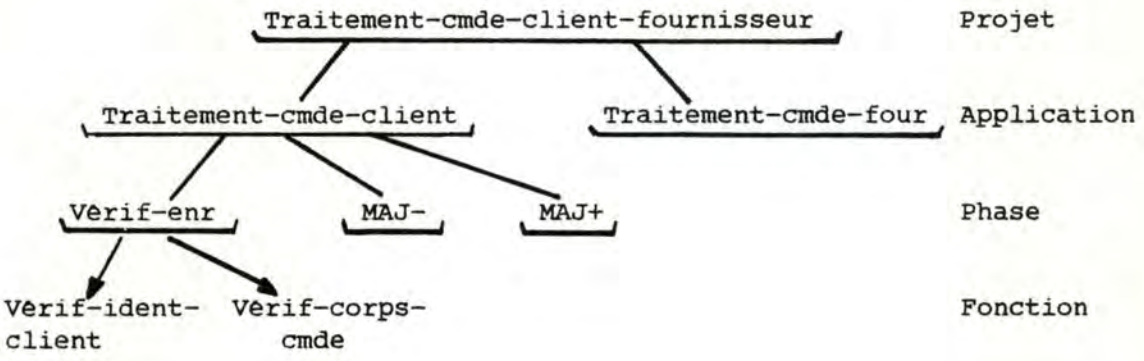
L'objectif du modèle de la structuration des traitements est de fournir aux concepteurs et analystes des critères leur permettant de décomposer un projet en traitements de plus en plus élémentaires [BODA.83].

Tout traitement est décomposé sous forme d'une arborescence dont les différents niveaux sont :

- le projet
- les applications
- les phases
- les fonctions.



L'usage d'une représentation graphique arborescente permet d'obtenir une vue aisée de l'ensemble de la structure des traitements d'un SI donné.



c) Le modèle de la dynamique des traitements

"L'objectif du modèle de la dynamique des traitements est de fournir à l'analyste des concepts et des mécanismes lui permettant de représenter les conditions de déclenchement, d'exécution et d'enchaînement des traitements en vue de caractériser les éléments du SI qui causent la production de messages résultats et les changements d'état de la mémoire du SI."

Le modèle de la dynamique des traitements repose sur deux concepts de base:

- les processus
- l'événement

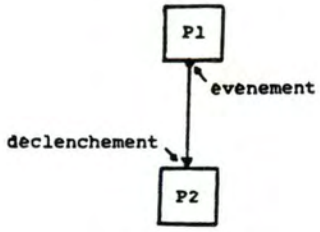
(cfr [BODA.83])

Représentations graphiques des différentes structures dynamiques [BODA.83].

On représentera par convention un événement par un point et un déclenchement par une flèche.

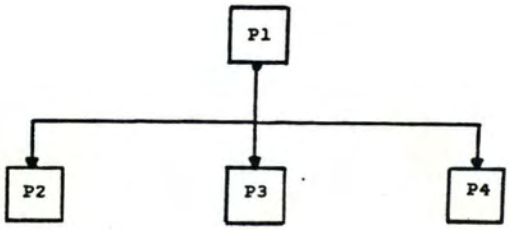
1.Enchaînement séquentiel :

La terminaison d'un processus provoque le déclenchement d'un autre processus.



2.Enchaînement éclaté :

La terminaison d'un processus provoque le déclenchement de plusieurs autres processus.

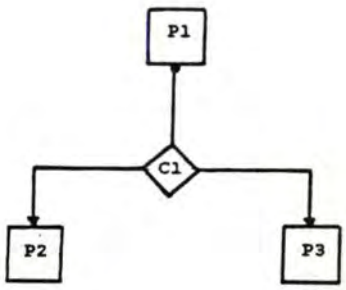


3.Enchaînement multiple :

La terminaison d'un processus provoque le déclenchement multiple de plusieurs processus de même type.

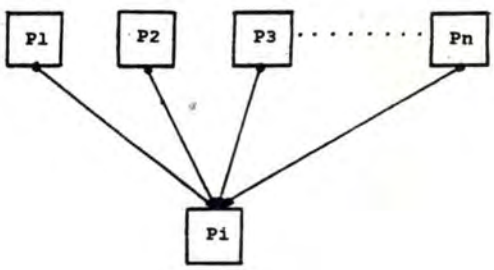
4.Enchaînement conditionnel :

La terminaison d'un processus P1 déclenche un autre processus P2 si la condition C1 est vraie et déclenche un processus P3 si la condition C1 est fausse.



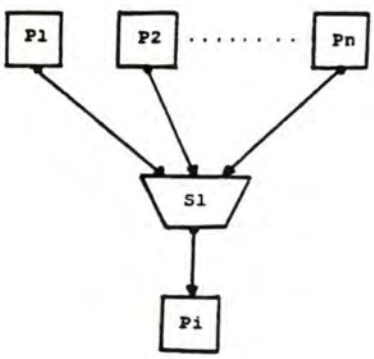
5.Enchaînement convergent :

Le déclenchement d'un processus Pi est provoqué par la terminaison d'un des processus P1 ou P2 ... ou PN.

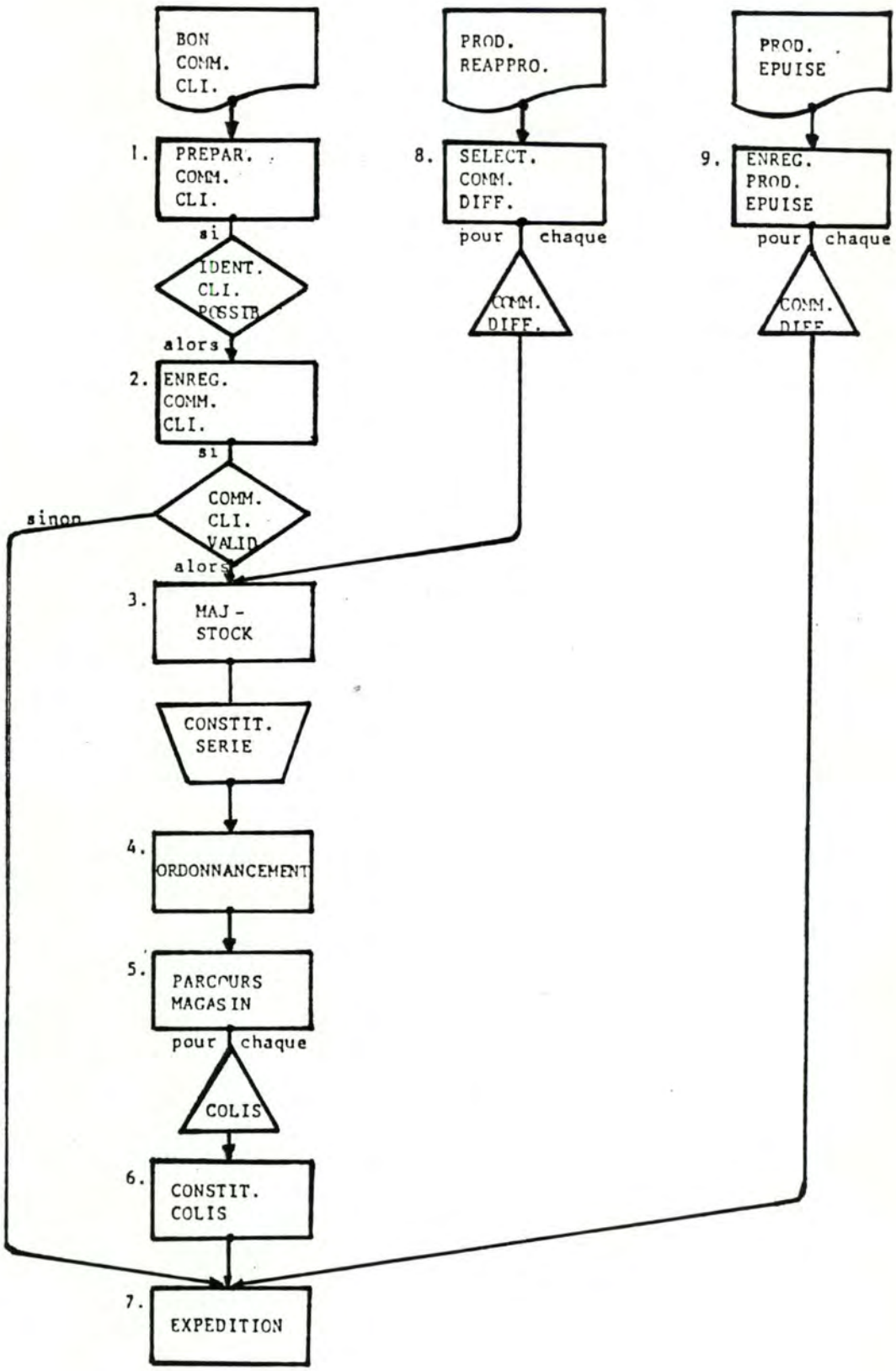


6.Enchaînement synchronisé :

Le déclenchement d'un processus Pi est provoqué par la réalisation d'un point de synchronisation.Cet événement sera lui-même causé par la réalisation d'une condition de jointure exprimant essentiellement une coordination de type ET entre d'autres événements.



Un exemple complet [BODA.83]



Ce schéma est très explicite par lui-même.Afin permettre sa parfaite compréhension,il sera cependant nécessaire de décrire complètement les différents points de synchronisation.

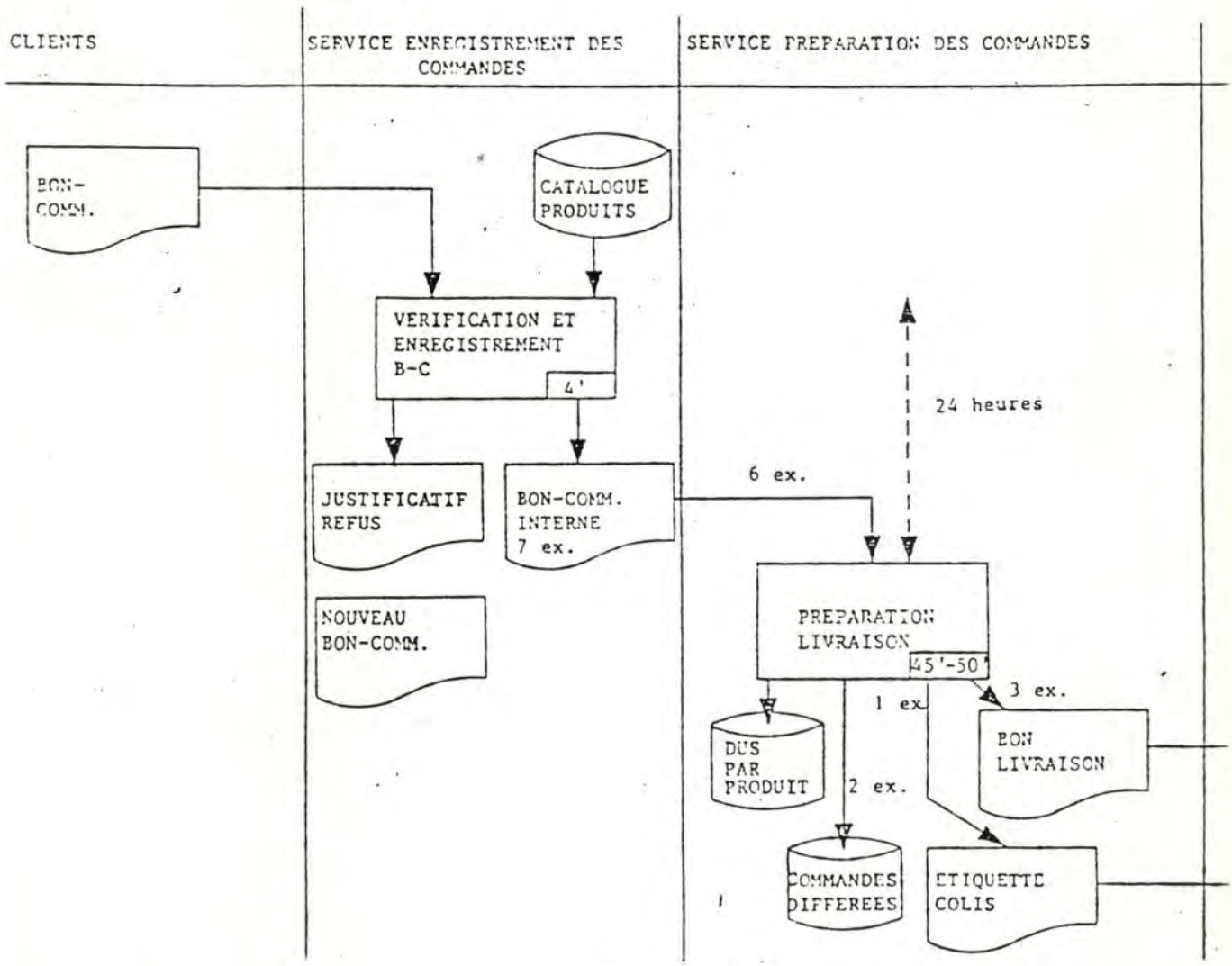
d) Le diagramme des flux

Le diagramme des flux joue un rôle important comme moyen d'expression synthétique du fonctionnement d'un SI,comme révélateur d'anomalies fonctionnelles et structurelles ainsi que comme outil de dialogue avec les utilisateurs.

Etant familière à l'utilisateur ,cette représentation lui offre un intérêt majeur.Elle illustre la circulation de l'information au sein de l'entreprise tel qu'il la percoit quotidiennement [BODA.83].

Exemple

La figure ci-dessous contient un extrait d'un diagramme de flux.



- un client émet un bon de commande qui est reçu et traité dans le service "d'enregistrement des commandes".Ce service procède à la "vérification et enregistrement des commandes".Pour réaliser ce traitement,on dispose "en entrée" du catalogue des produits.Une commande refusée est renvoyée au client avec un justificatif de refus et un nouveau bon de commande.Une commande acceptée donne lieu à la création en 7 exemplaires d'un bon de commande interne.La durée d'une opération de vérification et enregistrement du bon de commande" est estimée à quatre minutes...

- Le délai moyen qui sépare pour une même commande le début du traitement de "vérification et enregistrement du bon de commande" et celui du traitement de "préparation de la livraison" est de l'ordre de 24 heures.

Cet exemple met en évidence les caractéristiques et les conventions de représentations d'un diagramme de flux:

- Il décrit le cheminement des messages,leurs origines et leurs destinations.

- Il fournit une localisation spatiale et temporelle des "points" de naissance,de transformation et de disparition des informations.

- Pour chaque traitement,représenté par un rectangle,on indique les messages et les informations mémorisés en entrée et en sortie.

- On indique la durée unitaire d'un traitement ainsi que le délai qui sépare deux traitements distincts effectués en séquence.La comparaison entre ce délai et cette durée unitaire peut être révélatrice d'un goulet d'étranglement...

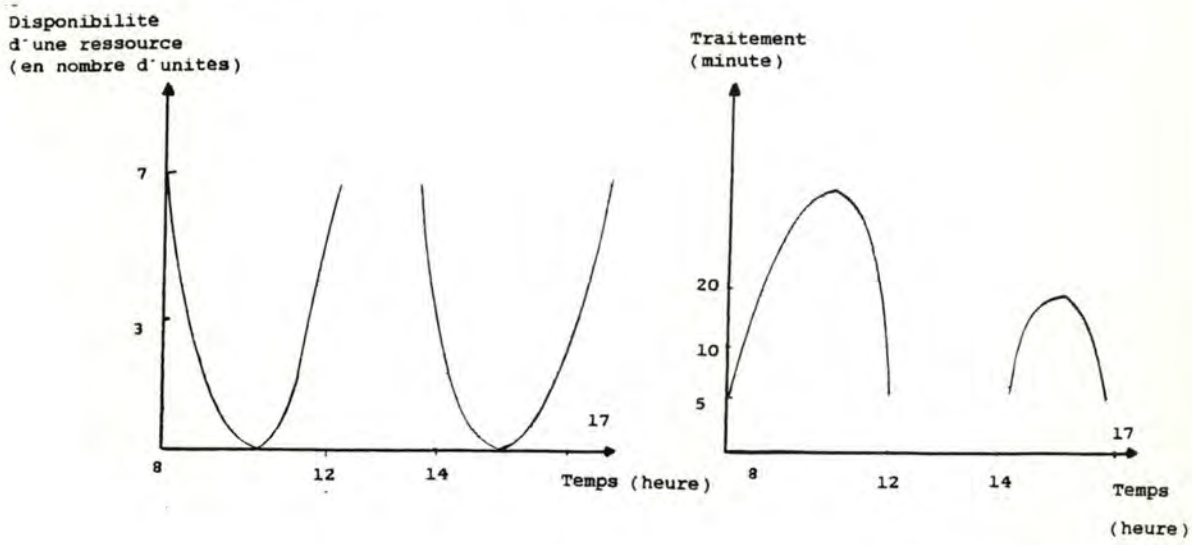
1.2.1.2. Représentations graphiques et outils d'évaluation

Il s'avère parfois intéressant de disposer de graphiques afin d'évaluer le caractère réalisable des spécifications d'un système d'information.

Un préférence sera accordée aux séries chronologiques permettant d'étudier et d'expliciter le

comportement du système en fonction du temps.

Lors d'une simulation du comportement d'un SI,il serait par exemple utile de visualiser graphiquement l'évolution de la disponibilité d'une ressource en parallèle avec la variation du temps d'attente pour un traitement utilisant cette ressource.



Ces deux graphiques mettent en évidence le lien étroit existant entre le temps d'attente avant le traitement et la disponibilité de la ressource : le temps d'attente devient très important lorsque la ressource n'est plus disponible.Il faudrait dès lors accroître le nombre d'unités de cette ressource.

1.2.2. Saisie des informations.

Nous nous sommes intéressés en [2.2.1] aux différents graphiques pouvant être réalisés à partir de la description d'un SI.

Il s'agissait là d'une simple exploitation de données préalablement enregistrées.

Une personne pourrait aussi introduire la description d'un système en utilisant les représentations graphiques associées aux différents modèles de la méthodologie [2.2.1.1].

Par rapport à une méthode traditionnelle d'introduction de données à l'aide d'un langage alphabétique, l'usage d'une méthode graphique présente divers intérêts.

Lien étroit avec la conception du système.

Lorsqu'une personne imagine les différents éléments intervenant dans un système d'information, elle peut préciser ses idées par l'intermédiaire d'une représentation graphique. Tel est l'usage d'un schéma entité-association dans la recherche d'une structure de données cohérentes. Tel est l'usage d'un schéma de la dynamique dans l'étude de la dynamique d'un système.

Une méthode d'introduction graphique permettra l'enregistrement simultané de la description d'un système et de sa représentation graphique.

Introduction facile et assistée.

L'utilisateur n'est point obligé de se familiariser avec un langage alphabétique pour pouvoir décrire un système.

Il se contentera de manipuler les différents éléments intervenant au sein des représentations graphiques.

Chapitre 2. Le graphique dans le cadre du logiciel ISDOS-IDA.

Le logiciel ISDOS-IDA constitue un logiciel d'aide à la conception des SI rassemblant un ensemble intégré d'outils complémentaires exploitant une base de données de spécifications.

Nous rappellerons brièvement dans ce chapitre l'évolution et les caractéristiques de ce logiciel.

Nous décrirons ensuite les outils graphiques disponibles.Ils seront tantôt interactifs tantôt non interactifs.

Lors de la description des outils graphiques interactifs, nous présenterons les spécifications fonctionnelles de l'interface graphique réalisé dans le cadre de ce mémoire, le Graphical Output Interface .

2.1. ISDOS-IDA.

2.1.1. Introduction

Au cours des annees septante,la plupart des departements informatiques connurent une crise de logiciel.

Cette crise provenait du fait que la majorite des grandes applications informatiques se developpaient de facon anarchique,sans methodologie ou outils integres pour les supporter.

Les principales difficultes que l'on a constate durant cette periode sont les suivantes :

- manque d'uniformite dans les regles d'ecriture des specifications et dans la documentation des applications.
- impossibilite de tenir a jour des dossiers.
- difficulte de consultation de ceux-ci.
- communication difficile entre les differentes equipes.
- manque de controle des specifications.

Ces differents points ont amene une incoherence croissante dans le developpement d'applications.

Pour remedier a ces critiques,des methodes et des outils ont ete crees pour aider a la conception,l'analyse,le developpement et la maintenance de sytemes d'information.

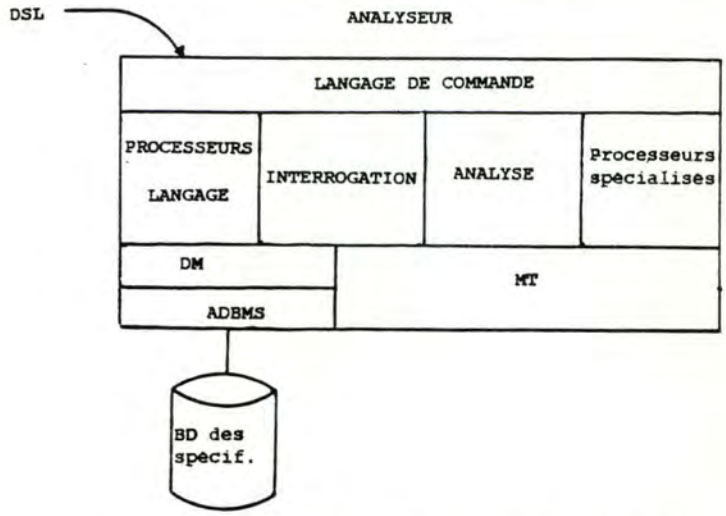
Le logiciel ISDOS est un de ces outils developpe depuis 1969 a l'universite du Michigan sous la direction du professeur D. Teichrow.

Depuis 1977,l'universite de Namur participe conjointement aux recherches. Ses efforts ont abouti a la creation du logiciel IDA.

Dans les prochaines sections,nous detaillerons quelque peu la structure du logiciel IDA.

2.1 2, Structure générale du logiciel IDA

La structure du logiciel IDA peut être représentée par la figure ci-dessous et comprend les éléments suivants :



- un langage de spécification, DSL (Dynamic Statement Language), pour décrire les éléments d'un système d'information. Ce langage possède la même structure que le langage PSL [PSL.82].

- un analyseur, DSA (Dynamic Statement Analyser), comprenant :

  1) un module d'analyse syntaxique des phrases écrites en DSL.

  2) un module d'analyse des commandes de mise à jour.

  3) un langage d'interrogation (Query System)

  4) un module de production de rapports documentaires.

- une base de données contenant les spécifications d'un système d'information.

Pour des raisons de portabilité, d'extensibilité et de maintenance, tout le logiciel IDA est conçu de façon modulaire et est implémenté en Fortran 4.

Le langage DSL

Le langage DSL est un langage de description des spécifications d'un système d'information.

Il est basé sur le modèle Entité-Association et comprend les concepts suivants :

- types d'objets .
- types de relations entre ces objets .
- types de propriétés associées aux types d'objets.

Les caractéristiques principales de ce langage sont les suivantes :

- il s'agit d'un langage de spécification et non exécutable.
- ce langage est non procédural.Ce qui signifie que l'ordre des instructions est quelconque.

Cette propriété permet de répartir dans l'espace et dans le temps des spécifications relatives à un même objet.L'Analyseur permettra de regrouper les différents éléments.

L'Analyseur DSA et l'exploitation de la base de données

La base de données est gérée par le logiciel ADBMS de type Codasyl.

Un analyseur (DSA) permet d'exploiter cette base de données.Il possède son propre langage de commande CLI (Command Language Interface).

Quatre groupes de commandes peuvent être distingués :

- les utilitaires qui remplissent des fonctions de service telle l'initialisation ,la restauration ou le "vidage" de la base de données.
- les commandes de mise à jour (IP,RP,DP) de la base de données. Ces commandes regroupent les fonctions d'ajouts (IP),de suppressions (DP) et de modifications (RP) d'éléments de la base de données.
- un langage d'interrogation (Query Langage) qui permet de sélectionner des objets à partir de critères complexes faisant intervenir le nom des objets,leur propriétés et les relations du langage

DSL.

- les commandes de production de rapports documentaires.

  Ces rapports contiennent de manière complète ou synthétique les descriptions d'éléments dans la base de données.
  Ces descriptions peuvent sortir sous des formes diverse (tables,graphiques,listes...).
  Certains rapports permettent en outre d'effectuer des contrôles de complétude et de cohérence sur les descriptions introduites.

Il existe également des processeurs spécialisés :

- la génération automatique d'une maquette programmée du système futur pour tester le caractère effectif des spécifications.

- La génération automatique d'un programme de simulation pour évaluer la faisabilité du système décrit.

Ces interfaces permettent de tester l'impact de diverses hypothèses de fonctionnement et de favoriser la perception directe par les utilisateurs de ce qu'ils souhaitent obtenir.

2.1.3. Extensions du logiciel IDA : ISLDM et SEM [ISDO.81]

Certains utilisateurs ont émis le souhait de disposer d'un outil qui permette de définir un meta modèle qui exprime plus adéquatement leur propre système d'information.

Pour cela,il est nécessaire de disposer d'un outil permettant à l'utilisateur de définir son propre langage,de créer une base de données contenant des spécifications exprimées dans ce langage particulier et d'en extraire des rapports.

Ces fonctions sont assurées par ISLDM et SEM.

Un langage de description d'un système d'information est défini en méta langage.

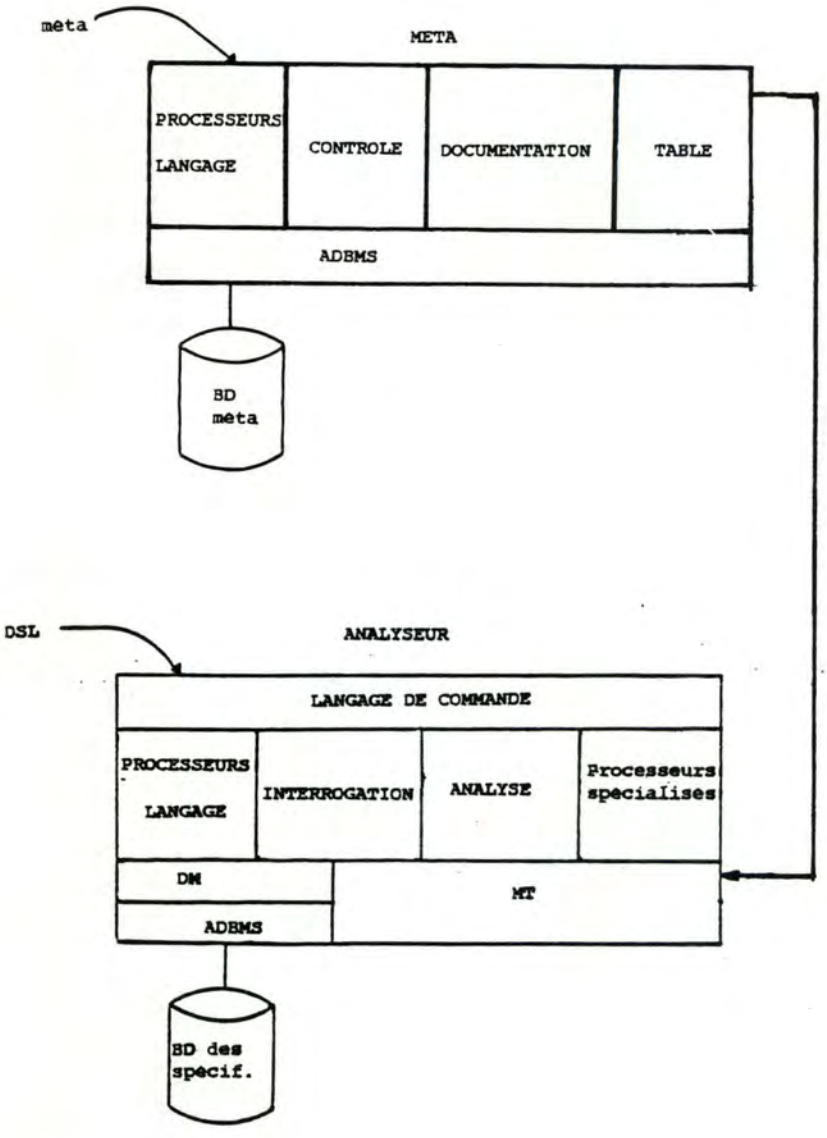
ISLDM analyse la définition du langage via l'Input Parser et stocke celui-ci dans une base de données (la base de données meta).

Des contrôles sont effectués par le Global Analyser et des résultats sont produits par le Documentation Producer.

Lorsque la définition du langage est correcte et satisfaisante pour le concepteur,une documentation est

produite et des tables sont gènèrèes
Ces tables seront exploitèes par SEM.

L'action de SEM est analogue à celle de DSA excepté que SEM est associé à un langage défini par l'utilisateur.
La description de ce langage est contenue dans les tables gènèrée par ISLDM. *



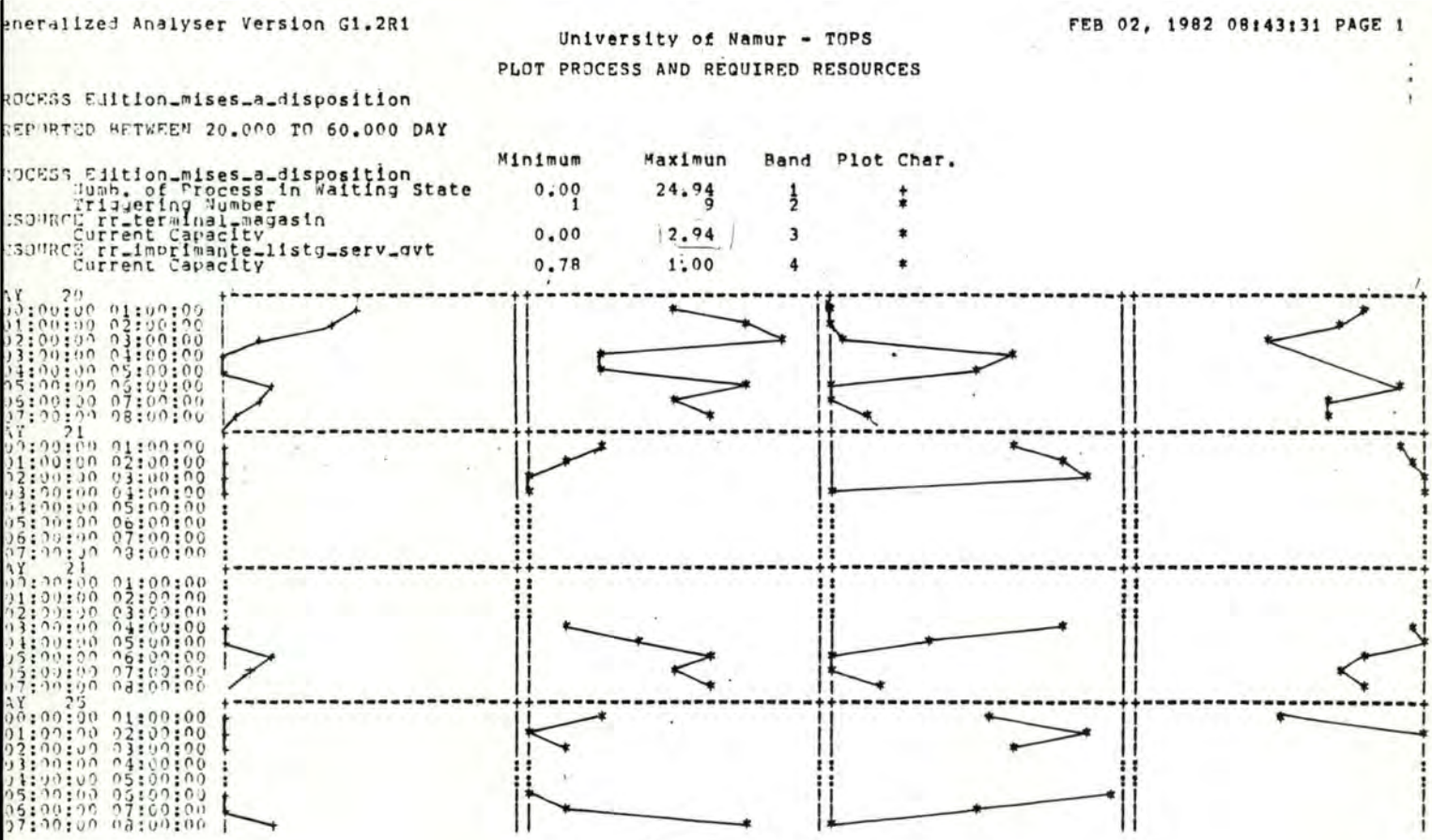
2.2. Graphique non interactif.

Le logiciel ISDOS-IDA permet la réalisation automatique de deux types de représentations graphiques : les représentations de courbes et les représentations arborescentes.

2.2.1. Représentation de courbes.

Des graphes curvilignes peuvent être générés automatiquement
à partir des résultats de la simulation du comportement d'un SI (DSL-SIM).

Ils reprendront par exemple, pour un processsus, l'évolution chronologique du temps moyen entre son déclenchement et son activation, ou, pour une ressource, l'évolution chronologique de sa disponibilité (figure ci-dessous).



2.2.2. Représentations arborescentes [DSA.82]

Le logiciel ISDOS-IDA dispose d'un outil permettant la réalisation de graphes arborescents : l'"EXTENDED PICTURE" (EP).

2.2.2.1. But de l'"Extended Picture".

Après avoir enregistré dans une base de données la description ou une partie de la description d'un système, l'utilisateur a la possibilité de rechercher ces informations dans la base de données et de les imprimer sous plusieurs formes à l'aide des commandes de rapport de DSA (Dynamic Spécification Analyser) .

L'"Extended Picture" présente, pour une liste de noms d'objets en entrée,un graphe arborescent de noms reliés aux précédents par un nom de relation.

2.2.2.2. Format de l'"Extended Picture".

Un graphe est imprimé pour chaque nom d'objet introduit (racine). Ce graphe est une arborescence qui relie chaque objet à un ou plusieurs successeurs, ces successeurs étant des objets reliés aux précédents par une des relations données par l'utilisateur pour autant qu'il soit d'un des types autorisés par l'utilisateur.

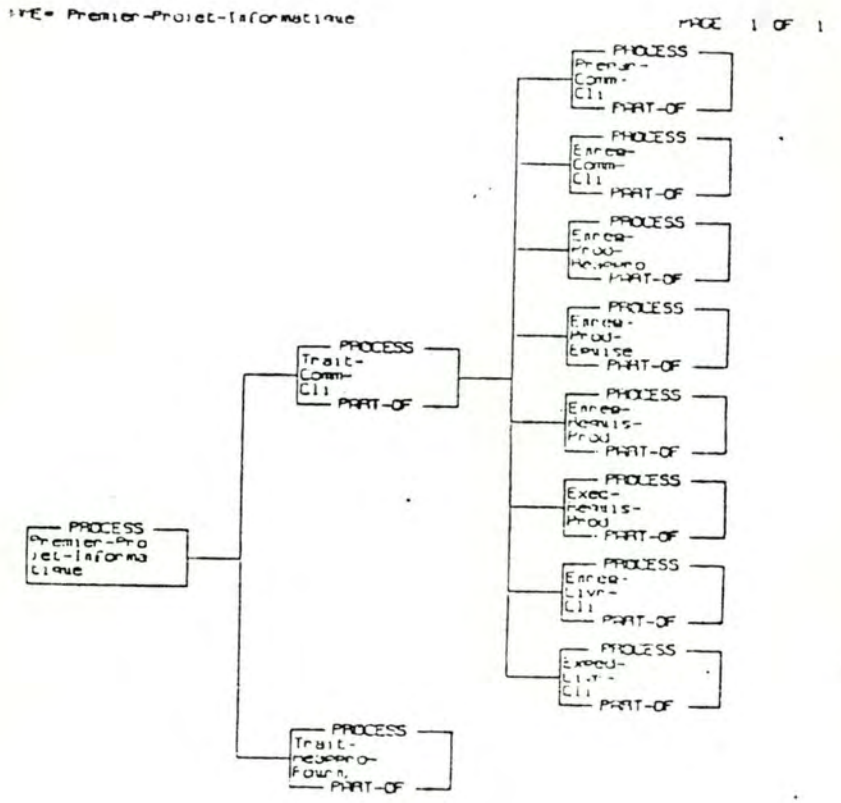
Ce graphe se termine soit parce qu'il n'y a plus de successeur, soit parce que le nombre de niveaux de profondeur a atteint la limite fixée par l'utilisateur, soit parce qu'on a rencontré une boucle.

Chaque noeud du graphe est relié au suivant par une ligne terminée par une flèche (si cette option est choisie).

Si l'utilisateur préfère l'usage d'une imprimante linéaire à celui d'une table tracante, le rapport possède également une numérotation des pages internes qui permettrait un collage tant vers la droite d'une page que vers le bas avec répétition sur la page suivante du dernier noeud de la page précédente.

2.2.2.3. Informations présentées dans l'"Extended Picture".

Considérons l'exemple suivant :



Les parametres fournis par l'utilisateur pour obtenir un tel dessin sont essentiellement :

type d'objets autorisès : "process"

type de relations autorisès : "subpart are"

nombre de niveaux : 3

racine : "Premier-projet-informatique".

Ce schèma nous montre une partie de la structure des traitements d'un système (Premier-projet-informatique).

2.2.2.4. Recherche des informations dans la base.

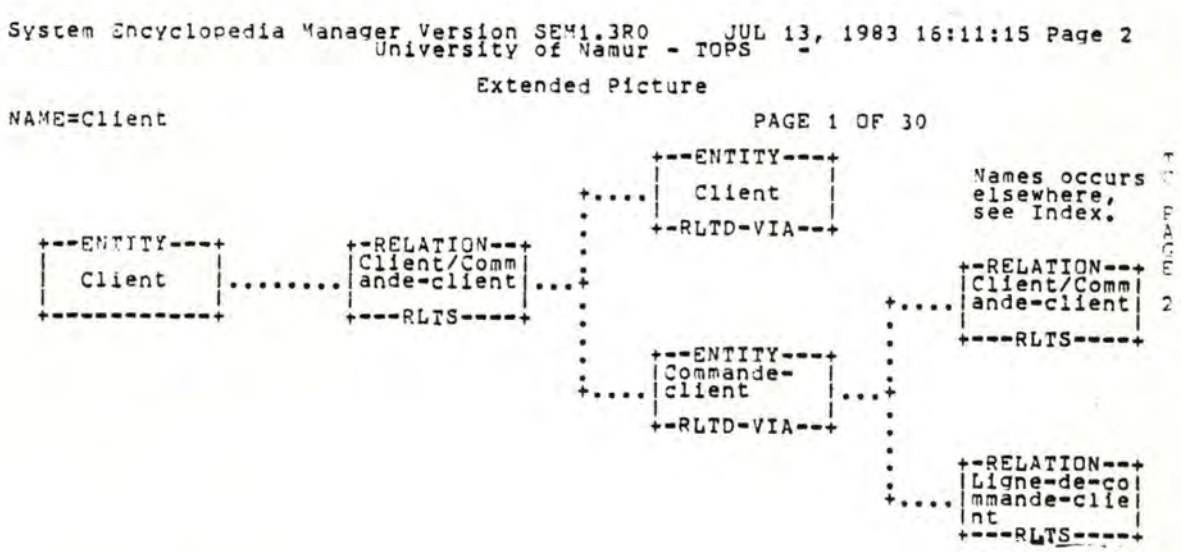
La recherche des informations dans la base de données s'effectue, comme la représentation, d'une manière arborescente à partir de chacune des racines données en entrée et en fonction des types d'objets et des types de relations autorisés.

2.2.2.5. Critiques.

L'"Extended Picture" est un rapport structuré ne pouvant réaliser que des graphes arborescents. Aussi, son usage sera approprié pour représenter graphiquement un ensemble d'informations dont la structure est elle-même arborescente, comme la structuration des traitements.

Par contre, les résultats seront décevants si l'on emploie l'"Extended Picture" pour représenter une structure non arborescente tel un schéma conceptuel des informations ou un schéma de la dynamique des traitements.

La figure ci-dessus illustre les limites de l'EP dans la représentation d'un schéma conceptuel des informations.



2.3. Graphique interactif.

Des outils interactifs ont été insérés dans le logiciel ISDOS-IDA afin de faciliter la création de réseaux.

Les difficultés rencontrées pour réaliser automatiquement de bons réseaux sur un plan pédagogique ont mis en évidence l'intérêt du recours à des moyens interactifs.

Deux outils interactifs existent à ce jour : le "Graphical Interface" (GI) et le "Graphical Output Interface" (GOI).

Le premier,crée à l'Université du Michigan dans le cadre du projet ISDOS, permet une introduction graphique de la description d'un SI.

Le second a été concu dans le cadre du présent mémoire et permet une exploitation graphique de la base de données des spécifications.

Il convient de souligner que le GOI reprend de nombreux éléments introduits dans le GI aussi bien au niveau de son architecture qu'au niveau de ses possibilités graphiques.En ignorant la grande différence entre les objectifs poursuivis par ces deux interfaces,on pourrait considérer le GOI comme une extension du GI.

Cette ressemblance explique la brièveté de la description du GI, les possibilités communes des deux interfaces étant détaillées lors de la description du GOI.

2.3.1. Le "Graphical Interface" (GI) [GIU.82]

Le "Graphical Interface" permet à un utilisateur d'enregistrer d'une manière interactive et graphique la description d'un système.

Afin de situer le rôle de cet interface au sein de la méthodologie, nous énoncerons les concepts de base dans la description d'un système.

Nous décrirons ensuite sommairement l'architecture fonctionnelle du "Graphical Interface".

Nous effectuerons enfin une critique générale de l'interface.

2.3.1.1. Usage méthodologique du GI [GIU.82].

Nous allons décrire ici les concepts de base des méthodes destinées à la représentation d'un système de traitement de l'information à l'aide de l'outil informatique et graphique.

Méthodologie.

Modèle entité-association (E-A	Langage alphabétique	Langage alphanumérique
Information System Description Language (ISDL)	ISDL - CS (Character String)	ISDL - GR (Graphical)
- type d'objet - type de relation - propriétés	- type d'objet - type de relation - propriétés	- figure - type d'arc - annotation
Partition du SDL en des sous-ensembles mutuellement exclusifs	Aspects du système	Type de diagramme
Description du système	Occurence d'objets et relations dans la base de données	Diagramme
Session	Session d'analyse	Session graphique
Description	Phrase	Transaction

1. Méthodologie

Le terme méthodologie désigne ici un moyen de développer un système de traitement de l'information.

2. Modèle entité-association (EA)

Le modèle entité-association est un modèle utilisé pour décrire un système d'information.

Son usage entraine l'identification d'entités (objet ou chose à propos de laquelle une donnée peut être enregistrée),de relations entre entités et d'attributs,donnant des renseignements sur les entités.

3. Langage de description d'un système d'information ISDL [cfr supra point 2.1.1.2]

Un langage de description d'un système d'information est un langage formel.

Les langages basés sur le modèle entité-association consistent en:

v.TP type d'objet
- type de relation
- type d'attribut

Un système particulier est décrit en définissant des objets,leurs propriétés et les relations entre les objets.

Un ISDL peut être construit sur des chaînes de caractères,nous parlerons alors d'un langage alphabétique (ISDL-CS)

Il peut également prendre une forme graphique (ISDL-GR).

4. ISDL-CS

Un langage alphabétique permet de décrire un système par l'intermédiaire de "statements" càd de phrases.

Les phrases sont contrôlées par un analyseur et contribueront à la mise à jour de la base de données. (cfr DSL-DSA)

5 ISDL-GR (graphique)

La forme graphique d'un ISDL nous permet de décrire un système par des diagrammes contenant des figures et des arcs.

Les figures correspondent aux types d'objets et les différents arcs aux types de relations.

Un arc peut relier deux objets.Les relations ternaires et quaternaires se verront associer plusieurs arcs.

6. Partition de la description d'un système

Un système est en général très étendu.Pour faciliter sa description,le système est généralement partitionné.

Dans un langage alphabétique,le système est étudié par types d'aspects et dans un langage graphique par types de diagrammes.

a) Types d'aspects

Un aspect du système va regrouper un sous-ensemble de types d'objets,de types de relations et de types de propriétés.Les partitions des types de relations et des types de propriétés sont généralement exhaustives et mutuellement exclusives tandis qu'un type d'objet peut apparaître dans plusieurs aspects (un process apparaît aussi bien dans la structure des traitements que dans la dynamique).

b) Types de diagrammes

Un type de diagramme est défini par les types d'objets et les types de relations pouvant y apparaître.

7. Description du système

Dans un langage alphabétique,la description du système comporte l'énoncé de phrases relatives aux objets,propriétés et relations qui ont été définies

(usage de rapports)

Dans un langage graphique,la description du système concerne la réalisation de diagrammes qui ont été définis.

8. Session

Quand le système est décrit par l'intermédiaire d'un langage alphabétique,l'entrée actuelle d'information dans la base de données est considérée comme une session;chaque session commencant par l'usage de l'Analyseur.

Avec un langage graphique,une session commence par le branchement d'un terminal graphique et le tracé d'un ou plusieurs diagrammes.

9. Unité dans la description d'un système

Avec un langage alphabétique,l'unité de base de la description d'un système est la phrase (statement).

Dans une session graphique,l'unité est la transaction.

Une transaction graphique diffère d'une phrase par la nature de l'intervention de l'utilisateur.

Pour mettre à jour la description d'un système par un langage alphabétique,l'utilisateur a seulement besoin d'invoquer un processus tel IP.Ce processus va utiliser le fichier créé par l'utilisateur sans requérir une nouvelle intervention de sa part.

Une transaction graphique nécessite l'usage d'un terminal graphique et une intervention continuelle de l'utilisateur durant son exécution.

Tandis que des phrases peuvent être traitées en batch,l'utilisateur doit initialiser spécifiquement chaque transaction.

Une transaction graphique comprend 4 parties :

a) l'invocation

Sélection d'un élément d'un menu afin de déclencher la transaction.

b) un ensemble d'activités interactives de l'utilisateur

Tâches devant être réalisées correctement selon une séquence et une syntaxe déterminée afin d'obtenir le résultat espéré d'une commande du GI.

Le GI guide à cet effet l'utilisateur.

Citons quelques exemples d'activités interactives :

- sélection d'un point à l'écran
- entrer une chaîne de caractères
- contrôle

c) Traitement syntaxique

Un traitement syntaxique est réalisé à partir d'informations déduites de la phase précédente et d'informations existantes au sein d'une des bases de données utilisées par le GI.

d) Mise à jour des bases de données.

La mise à jour des bases de données intervient à la fin de la transaction.

2.3.1.2. Architecture du GI [GIU.82]

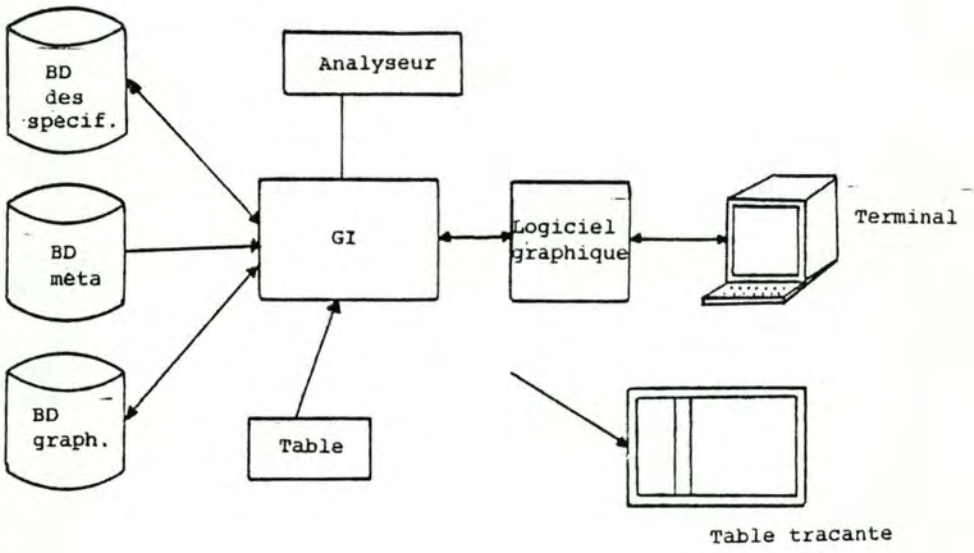
Nous avons déjà précisé que la description fonctionnelle du GI serait sommaire en raison des nombreuses ressemblances existant entre la structure de cette interface et de celle du GOI dont nous exposerons les fonctions en [2.3.2.3.]

Le graphical interface permet à un utilisateur d'introduire d'une manière interactive et graphique la description d'un système.

Le GI peut enregistrer directement une information sur le système sans convertir la représentation sous forme de diagramme en chaîne de caractères.De plus,le diagramme est mémorisé afin de permettre son tracé,avec la position des objets inchangés,lors de sessions

ultérieures.

Les différents composants du système graphique apparaissent sur la figure ci-dessous:



Le système comprend:

1. Un terminal et un logiciel graphique

2. L'interface graphique (GI)

L'interface graphique exécute des commandes,dessine des diagrammes et met à jour les bases de données.

Il offre à l'utilisateur deux niveaux de commandes:

a. Control mode

Ce niveau regroupe les commandes auxquelles l'utilisateur a accès dès l'entrée dans le GI.

Elles regroupent des activités

préliminaires telles la sélection ou l'effacement de diagrammes existant et la création de nouveaux diagrammes (cfr GOI [2.3.2.3.])

b) Diagram mode

Pour travailler dans ce mode, l'utilisateur aura sélectionné un diagramme sur lequel il désire effectuer certaines manipulations.

Il dispose à présent de commandes de deux types :

- celles qui peuvent entraîner une mise à jour d'une des bases de données (ex:ajout d'un objet)
- celles qui ne peuvent entraîner une telle mise à jour (ex: un zoom) (cfr 2.3.2.3)

3. l'analyseur

Faisant référence à la fois à la syntaxe d'un langage alphabétique et à la syntaxe d'un langage graphique, le GI réalise deux étapes d'analyse.

Dans une première étape, l'input de l'utilisateur est contrôlé par rapport à la syntaxe exprimée dans les tables de définition du langage graphique. Si tout est correct, l'input sera alors contrôlé vis à vis de la syntaxe d'un ISDL-CS (exemple: ordre des objets dans la création d'une relation).

Si aucun problème n'a été rencontré, on procédera ensuite à la mise à jour des bases de données.

4. Bases de données

Le GI utilise trois bases de données:

- la base de données méta contenant la descrition de l'ISDL-CS (consultation).
- la base de données utilisateur contenant la description du SI (consultation et maj).
- la base de données graphique contenant les informations nécessaires pour retracer un diagramme créé par l'utilisateur afin de représenter un aspect du système

(consultation et maj).

5.Table

L'administrateur du système initialise le GI pour l'usage d'un langage graphique particulier par la création d'une table de définition du langage.

Cette table comprend:

- la liste des types d'objets pouvant être utilisés.
- la liste des types de relations pouvant être utilisés.
- la liste des différents types de diagramme avec,pour chaque diagramme, la liste des types d'objets,des types de relations disponibles et leur représentation graphique (une figure pour un objet,un vecteur pour une relation).

2.3.1.3. Critiques générales sur le GI

a)Cohérence entre les bases de données.

Un des plus importants problème que peut rencontrer l'utilisateur lors de l'utilisation du GI est celui du maintient de la cohérence entre le contenu de la base de données utilisateur et celui de la base de données graphique.

L'actuelle version du GI n'a rien prévu pour une génération automatique de diagrammes sur base du contenu de la base de données utilisateur.

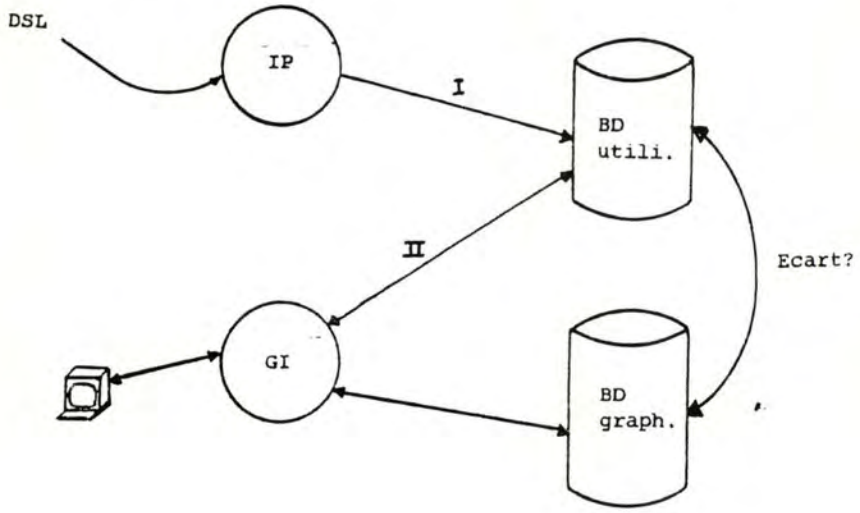
Pour prévenir une incohérence due au fait qu'une mise à jour de la base de données utilisateur peut être réalisée aussi bien d'une manière graphique que par l'intermédiaire de IP, plusieurs précautions devront être prises.

La plus importante est que toute commande entrainant une modification de la base de données utilisateur devant apparaître dans la base de données graphique (notamment la création ou l'effacement d'un objet) doit être réalisée par l'intermédiaire du GI.

Sans cela,il deviendrait possible pour un objet d'apparaître dans un rapport généré à partir de la base de données utilisateur sans pour autant se retrouver

dans un diagramme.

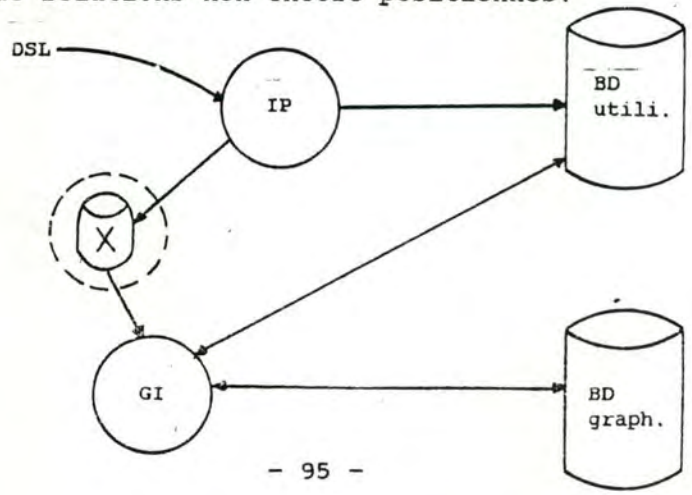
Un objet pourrait aussi avoir été effacé de la base de données utilisateur tout en demeurant dans la base de données graphique.



Solutions possibles

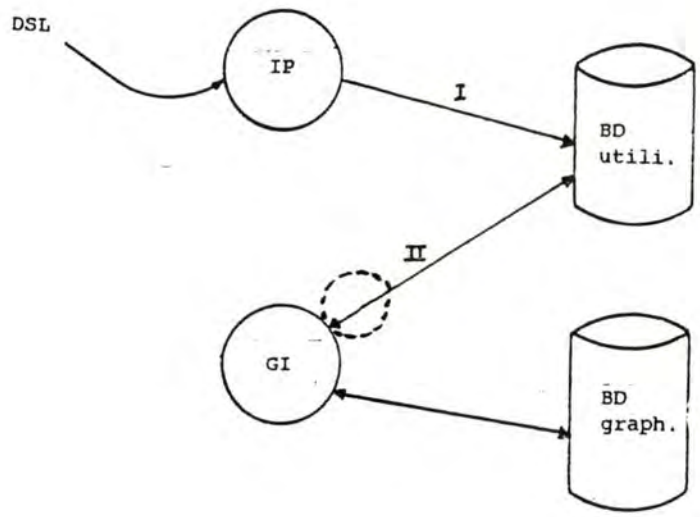
1.Lors de la mise à jour basée sur les phrases du langage,la commande de mise à jour (IP) provoque la création d'un fichier X reprenant la liste des objets et relations à rajouter (effacer,modifier) dans la base de données graphique.

Au cours de la session graphique qui suivra, l'utilisateur verra apparaître cette liste d'objets et de relations non encore positionnés.



2.Au lieu de passer par l'intermédiaire d'un fichier,on pourrait imaginer que le GI effectuerait au début de chaque session un contrôle sur le contenu de la base de données graphique.

Il informerait alors l'utilisateur des modifications intervenues.



Ces deux solutions ne sont certes pas idéales et soulignent encore l'importance du problème rencontré.

b)Extensions des capacités graphiques

Les capacités de représentations graphiques offertes par le GI nécessitent certaines extensions afin de rencontrer les conventions adoptées dans le cadre de la méthodologie.

Formes (figures) disponibles

Le Graphical Interface ne reprend pas,parmi ses figures de bases,des formes spécifiques pour la représentation d'une entité,d'une association,d'un point de synchronisation...

Emboîtement de représentations

Le GI ne permet pas l'emboîtement de représentations.Il interdit par exemple l'introduction de plusieurs noms d'objets au sein d'une même figure.

Il sera dès lors impossible de construire un schéma conceptuel conventionnel, les noms des propriétés devant apparaître dans le dessin d'une entité.

Automatisme dans la représentation.

Le GI a été concu,dans un souci de portabilité,indépendamment de tout langage de spécification.

Ce choix a entraîné certaines limitations au niveau des automatismes dans la représentation d'un aspect du SI.

L'interface ne guidera jamais l'utilisateur lors de la création d'un diagramme en fonction de l'aspect du système représenté.

Il n'associera pas la création d'une entité à l'enregistrement de ses propriétés...

Le Graphical Output Interface (GOI) tentera de résorber ces déficiences,notamment par une extension de la table de définition du langage graphique.

2.3.2. Le "Graphical Output Interface" (GOI)

Nous venons de constater que le Graphical Interface (GI) était confronté au problème presque insoluble de la cohérence de la base de données graphique par rapport à la base de données utilisateur (ou de spécifications).

Aussi,il s'avérait utile de disposer d'un interface qui se contenterait,comme le fait l'"Extended picture",d'exploiter graphiquement la base de données utilisateur.Il permettrait d'obtenir des représentations complètes et traditionnelles des différents aspects du système d'informations:structure des données,dynamique des traitements,diagramme des flux.

La réalisation de ces graphes s'effectuera d'une manière interactive sous le contrôle de l'interface.L'utilisateur sera entièrement guidé et se contentera d'introduire certaines informations influant l'aspect graphique et non le contenu informationnel des différents dessins.

Le recours à un outil non entièrement automatisé s'explique par la difficulté de produire automatiquement un réseau lisible.

Après une présentation synthétique de l'interface et de son architecture,nous consacrerons l'essentiel de cette section à sa description fonctionnelle.L'architecture du GOI étant presque analogue à celle du GI,nous soulignerons les analogies et les différences fonctionnelles entre ces deux interfaces.

2.3.2.1. Introduction

Le GOI,graphical output interface,permet à un utilisateur de construire d'une manière interactive une représentation graphique du contenu d'une base de données des spécifications d'un SI.

Par l'intermédiaire de tables et selon les différents aspects du système,l'administrateur définit divers types de diagrammes. Un diagramme est caractérisé par les types d'objets et de relations qu'il peut contenir ainsi que par leur représentation graphique (figures pour les objets,vecteurs pour les relations)

Afin de ne pas devoir toujours recréer de nouveaux diagrammes,le GOI travaillera également sur une base de données graphique qui conservera un enregistrement des

différents dessins déjà réalisés.

Dans le but d'éviter toute incohérence entre le contenu de la base de données des spécifications et celui de la base de données graphique, l'utilisateur n'aura guère de liberté sur le choix des objets et relations à introduire dans un diagramme.

En aucun cas, il ne pourra décider, de sa propre initiative, d'introduire une information spécifique au sein d'un diagramme.

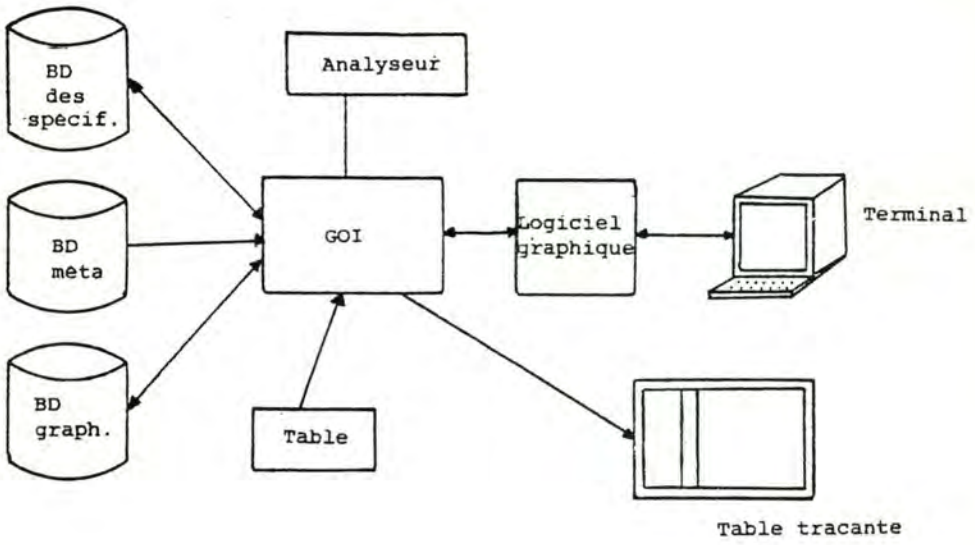
Il incombera au programme de lui indiquer à partir du contenu de la base de données des spécifications les objets et les relations qu'il lui est possible d'ajouter dans la base de données graphique.

Sur base de ces propositions, l'utilisateur pourra effectuer une éventuelle sélection...

Le parcours de la base de données des spécifications s'effectuera d'une manière arborescente à partir de racines fournies par l'utilisateur...

2.3.2.2. Architecture globale.

Les composantes du système graphique réalisé apparaissent sur la figure ci-dessous:



Nous pouvons constater la similitude entre cette architecture et celle du GI représentée en [2.3.1.2] .

Les bases de données

Le graphical output interface travaille,comme le GI,sur trois bases de données:

- la base de données méta contenant la description de l'ISDL-CS (consultation)

- la base de données utilisateur contenant la description du S.I. Contrairement au GI,le GOI ne modifiera pas le contenu de cette base de données .

- la base de données graphique contenant les informations nécessaires pour retracer un diagramme (consultations et modifications).
  La structure de cette base de données [cfr chapitre 2,troisième partie] sera différente par rapport à celle du GI,vu les extensions réalisées notamment au niveau des représentations graphiques.

La table.

La table du GOI permet de définir les divers types de diagrammes souhaités.

Offrant certains automatismes dans la création de dessins et autorisant l'emboîtement de figures,elle constitue une large extension de la table du GI.

Nous en donnerons une description détaillée lors de la troisième partie de ce mémoire.

GOI

La description des différents niveaux de commandes du graphical output interface apparaîtra dans la section suivante.

2.3.2.3. Description fonctionnelle

Les fonctions de base du GOI permettent à l'utilisateur:

- de créer un nouveau diagramme

- de modifier l'aspect graphique d'un diagramme existant

- de compléter un diagramme existant en fonction de modifications effectuées dans la base de données des spécifications.

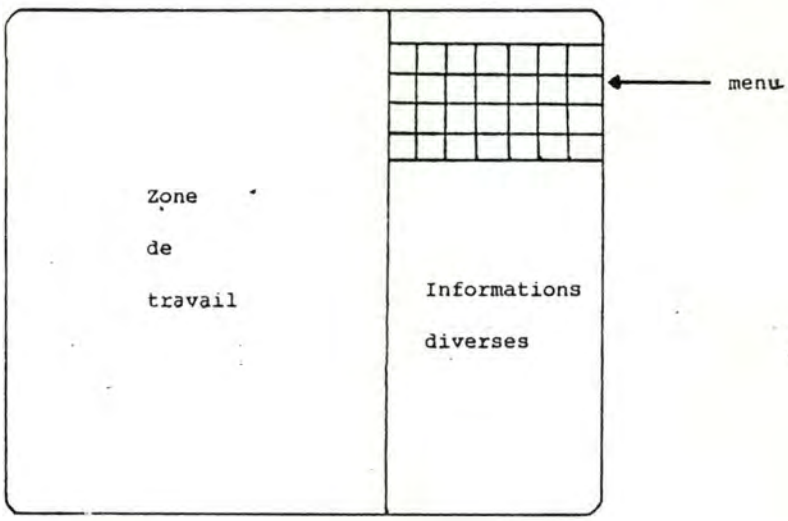
Pour ce faire,il peut travailler dans trois modes de commandes différents:

- "Control mode"

- "Display mode"

- "Création/completion mode"

A chaque mode sera associé un écran dont la division sera toujours semblable à ce qui suit:



a.Menu:
Partie de l'écran sur laquelle les différentes commandes disponibles seront affichées.

b.Zone de travail:
Partie de l'écran sur laquelle un diagramme sera éventuellement dessiné.

c.Messages:
Cette ligne permet à l'utilisateur d'introduire des informations non graphiques.
Elle sert également à l'affichage de messages guidant l'utilisateur lors d'une transaction.

d.Informations diverses:
Partie de l'écran contenant essentiellement des informations utiles à un niveau de commandes donné.

Dans le GI,toutes les commandes de manipulations d'un diagramme ont été regroupées au sein d'un même mode de commandes "diagram mode".

Par contre,le GOI présentera deux modes de commandes différents selon que l'utilisateur désire ajouter des objets (creation/complexion mode) ou effectuer de simples modifications d'ordre graphique (display mode).

Cette distinction est due à la spécificité de la première tâche.

La création ou le complètement d'un diagramme (ajout d'objets) exigera un parcours de la base de données des spécifications du système d'information et une consultation de la base de données méta. Elle nécessitera également l'apparition à l'écran de diverses informations permettant à l'utilisateur de construire son dessin...

2.3.2.3.1. Control mode.

Il s'agit du niveau de commandes auquel l'utilisateur accède au début de chaque session.Il lui permet d'indiquer la tâche qu'il désire effectuer.

Les commandes disponibles sont:

a)Créer un diagramme d'un certain type (create).

Il sera demandé à l'utilisateur de donner,pour un diagramme,son nom, son type ainsi que ses racines c'est-à-dire les noms des objets à partir desquels le programme entamera les parcours de la base de données des spécifications.

Ce programme passera ensuite automatiquement au mode de commande "création/ complexion mode".

Notons qu'à tout diagramme créé sera associé un numéro d'ordre de création.

b)Compléter un diagramme existant en fonction des changements apparus dans la base de données utilisateur. (complete).

Le diagramme sera identifié par son nom ou son numéro d'ordre de création.

Le programme trouvera ses racines dans la base de données graphique et passera ensuite en "creation/completion mode" en affichant le diagramme sélectionné.

c)Afficher un diagramme existant et éventuellement en modifier l'aspect graphique (display)

Le diagramme sera identifié par son nom ou son numéro d'ordre de création.

Le programme passera ensuite en mode de commande "display mode" en affichant le diagramme sélectionné.

d)Imprimer un diagramme existant. (Print)

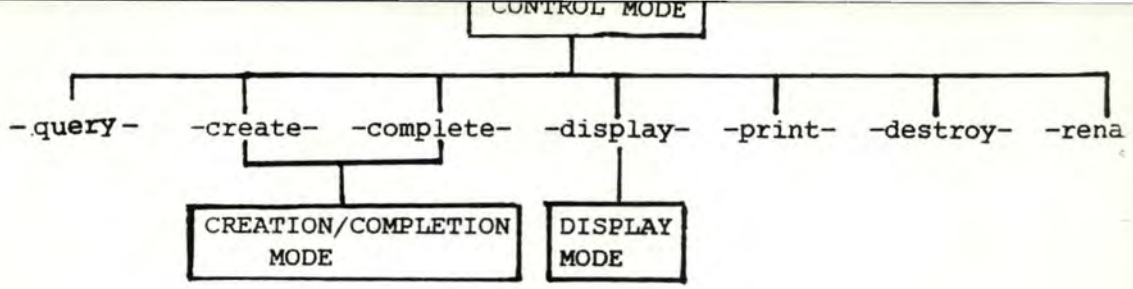
L'utilisateur identifie le diagramme et indique le format de page désiré sur la table tracante.

e)Effacer l'enregistrement d'un diagramme existant identifié par l'utilisateur. (Destroy)

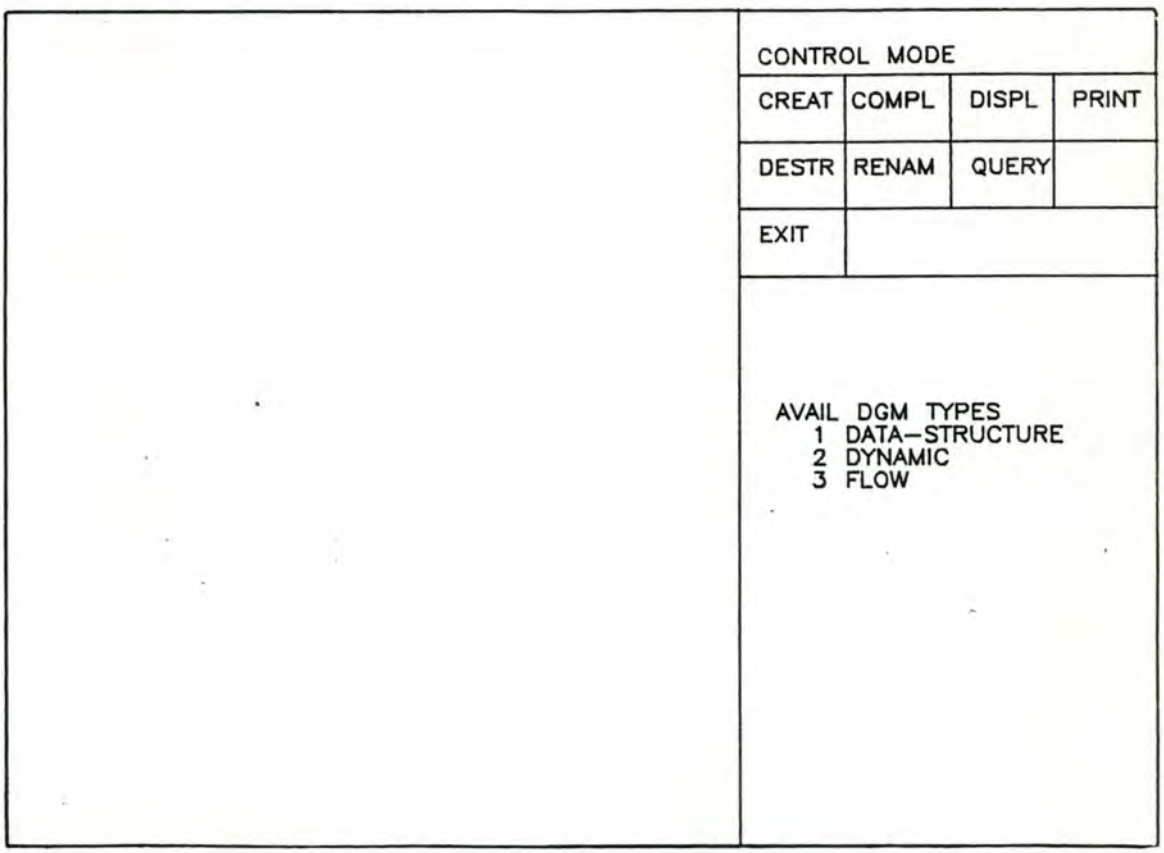
f)Changer le nom d'un diagramme . (Rename)

g)Obtenir la liste de tous les diagrammes existants avec nom,type, numéro de création et nombre d'objets représentés (query)

Nous pouvons illustrer le passage d'un mode de commande à l'autre:



Le lecteur trouvera ci-dessous une reproduction d'un écran associé au présent mode de commande.



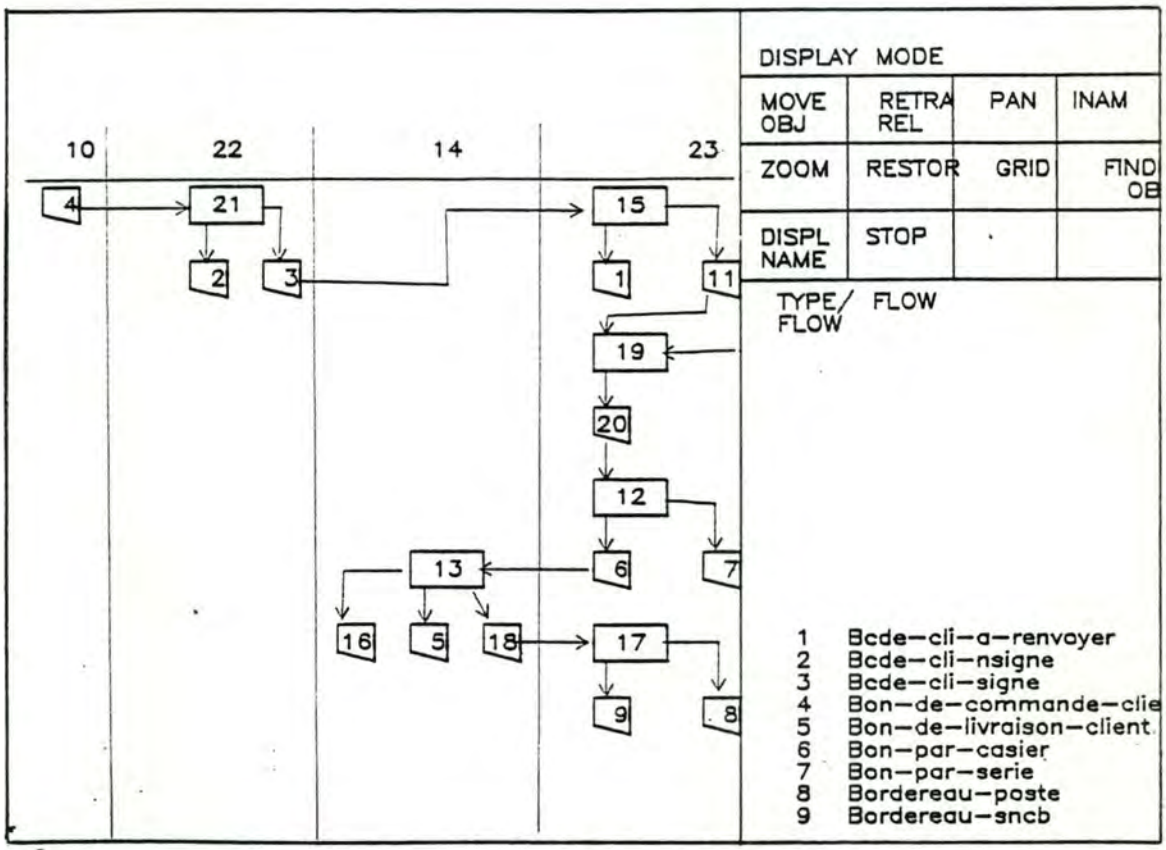
En "control mode", on reprend dans la zone "informations diverses" la liste des types de diagrammes définis.

Notons enfin que les commandes "Destroy","Rename" et "query" étaient déjà disponibles sur le GI et n'ont fait l'objet que de légères modifications.

Par contre,les autres fonctions dépendent davantage des caractéristiques propres de l'interface réalisé.

2.3.2.3.2. Display mode.

L'utilisateur dispose à ce niveau d'un ensemble de commandes lui permettant d'observer et de modifier un diagramme affiché en partie ou entièrement à l'écran.



Nous constatons que seule une partie du diagramme est visible sur la surface de travail.

De plus, les noms des différents objets ne sont pas directement repris et sont remplacés par des numéros; le format des figures étant trop petit que pour pouvoir y inscrire lisiblement un texte.

Ces numéros permettent de retrouver les noms des objets au sein d'une liste dont une partie est affichée dans la zone "informations diverses".

Cette zone reprendra également le nom et le type du diagramme.

Les commandes

La majorité des commandes disponibles "Display mode" étaient déjà offertes par le GI en "diagram mode".

Il convient cependant de souligner que ces commandes ont été étendues en fonction des plus grandes possibilités de représentation graphique du GOI (emboîtements de figures...).

Bien que leurs fonctions soient toujours identiques, les traitements sous-jacents deviendront plus complexes.

a) Déplacer un objet (Move obj)

L'utilisateur sélectionne un objet et indique sa nouvelle position.

Le GOI a étendu cette commande de deux manières:

- Si l'objet a une représentation graphique contenant également d'autres objets (emboîtements de représentations), l'ensemble sera déplacé.

- Les vecteurs des relations auxquelles participe l'objet seront également retracés.

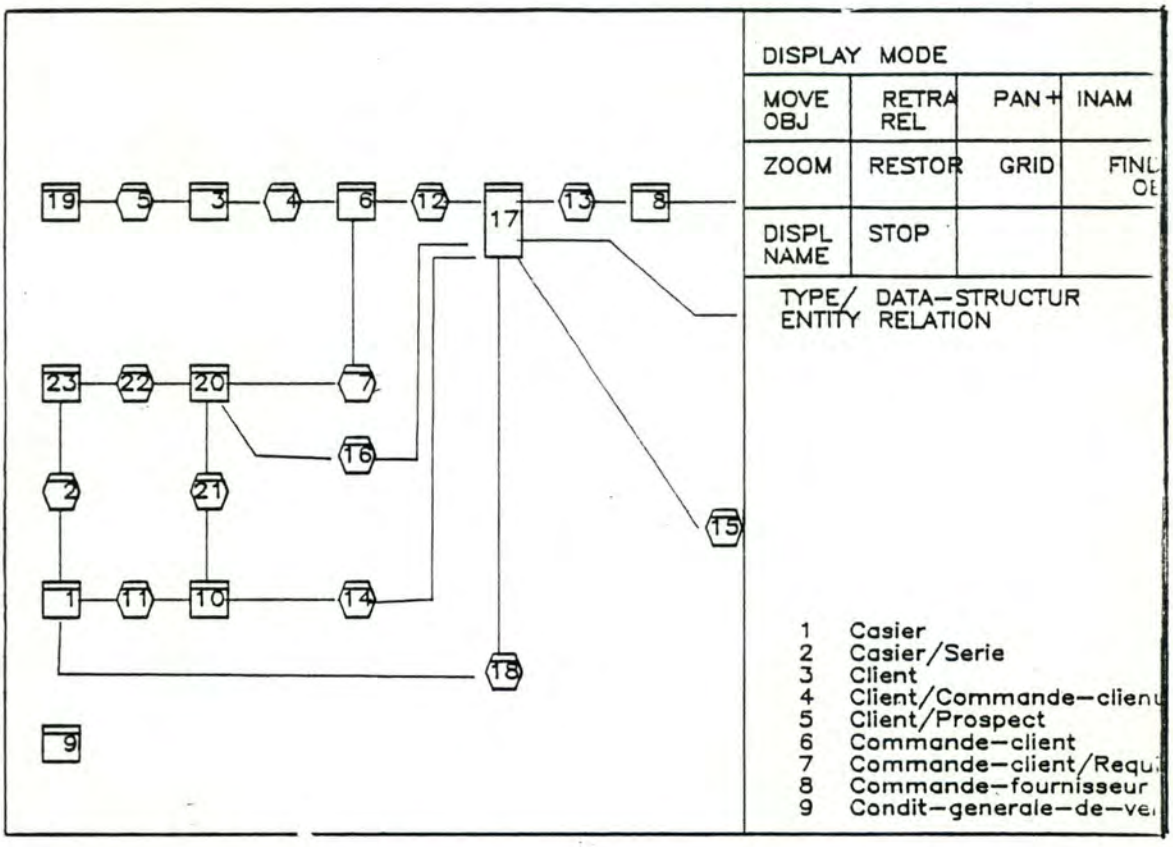
b) Retracer une relation. (Retrace rel)

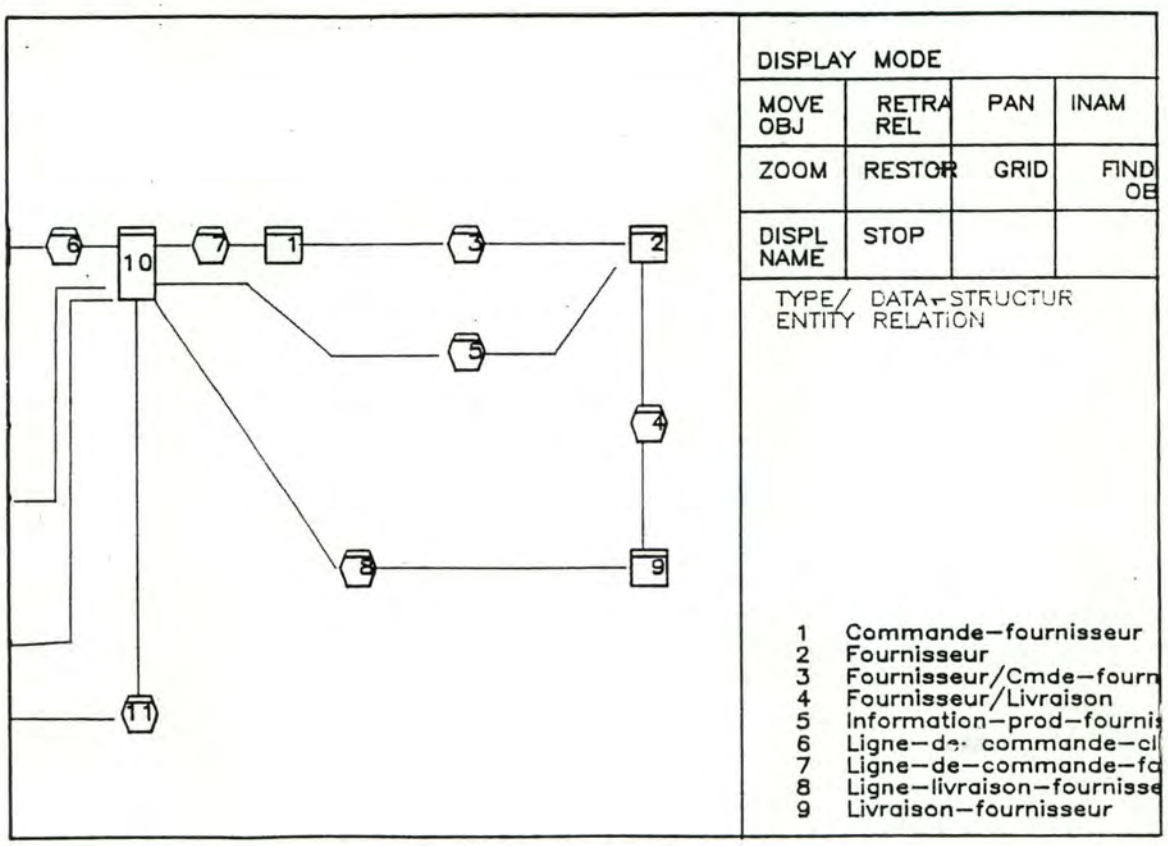
L'utilisateur désire redessiner le(s) vecteur(s) associé(s) à une relation.

Il désignera la relation par les objets qui l'identifient.

c) Déplacer la fenêtre de vue sur le diagramme. (Pan)

Nous reprenons ci-dessous un exemple.


DISPLAY MODE
MOVE OBJ
RETRA REL
PAN +
INAM
ZOOM
RESTOR
GRID
FIND OB
DISPL NAME
STOP
TYPE/ ENTITY
DATA-STRUCTUR RELATION
19
5
3
4
6
12
17
13
8
23
22
20
7
16
2
21
15
1
11
10
14
18
9
1 Casier
2 Casier/Serie
3 Client
4 Client/Commande-clien
5 Client/Prospect
6 Commande-client
7 Commande-client/Requ
8 Commande-fournisseur
9 Condit-generale-de-ve


DISPLAY MODE
MOVE OBJ
RETRA REL
PAN
INAM
ZOOM
RESTOR
GRID
FIND OB
DISPL NAME
STOP
TYPE/ ENTITY
DATA-STRUCTUR RELATION
6
10
7
1
3
2
5
4
8
9
11
1 Commande-fournisseur
2 Fournisseur
3 Fournisseur/Cmde-fourn
4 Fournisseur/Livraison
5 Information-prod-fourni
6 Ligne-de-commande-cl
7 Ligne-de-commande-fo
8 Ligne-livraison-fournisse
9 Livraison-fournisseur

d)déplacer la fenêtre sur la liste des noms d'objets affichés (INAM).

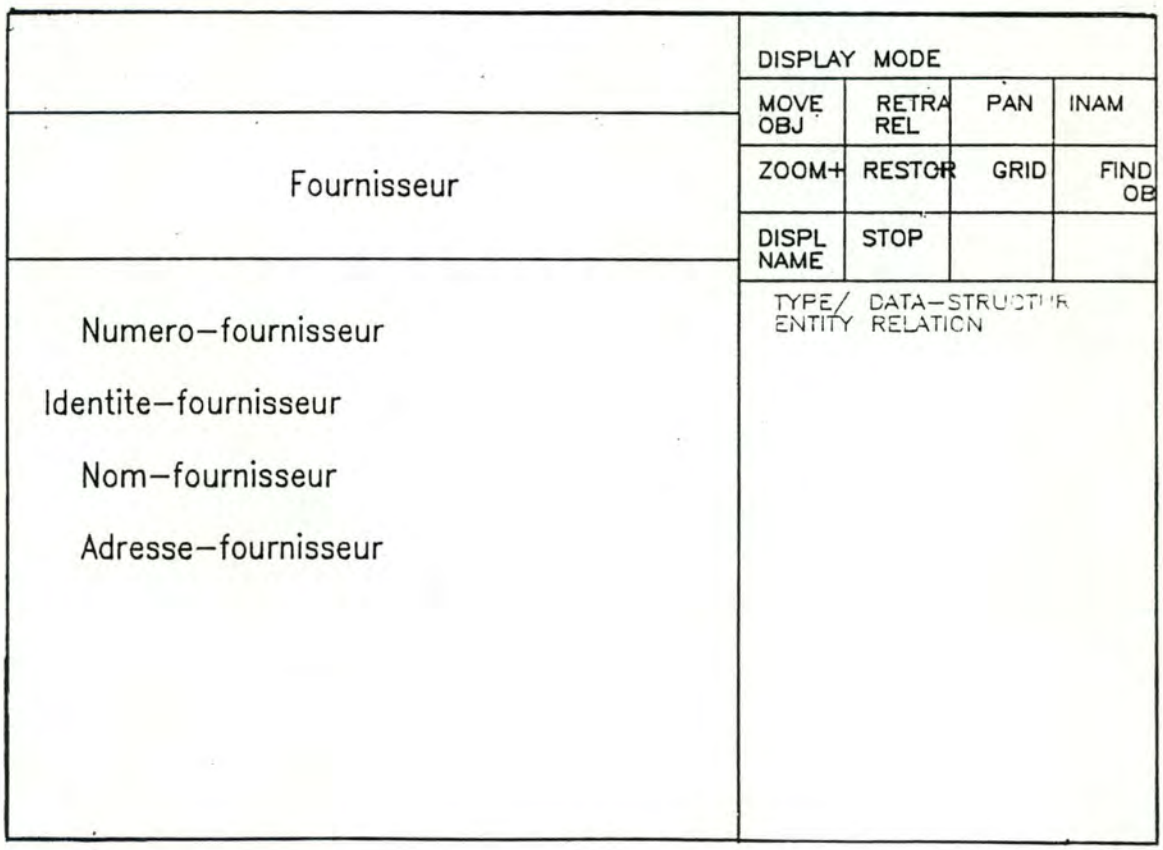
L'utilisateur déplace cette fenêtre en indiquant le numéro de l'objet devant devenir la borne inférieure de la liste des noms affichés.

Les neuf premiers noms de numéro supérieur ou égal au nombre introduit apparaîtront alors à l'écran (cette commande est propre au GOI).

e)Zoom.

L'utilisateur effectue un agrandissement sur une figure choisie.

Cette commande sera particulièrement utile pour pouvoir voir les noms des éléments d'une entité.



f)Remettre l'écran dans son précédent état (Restore)

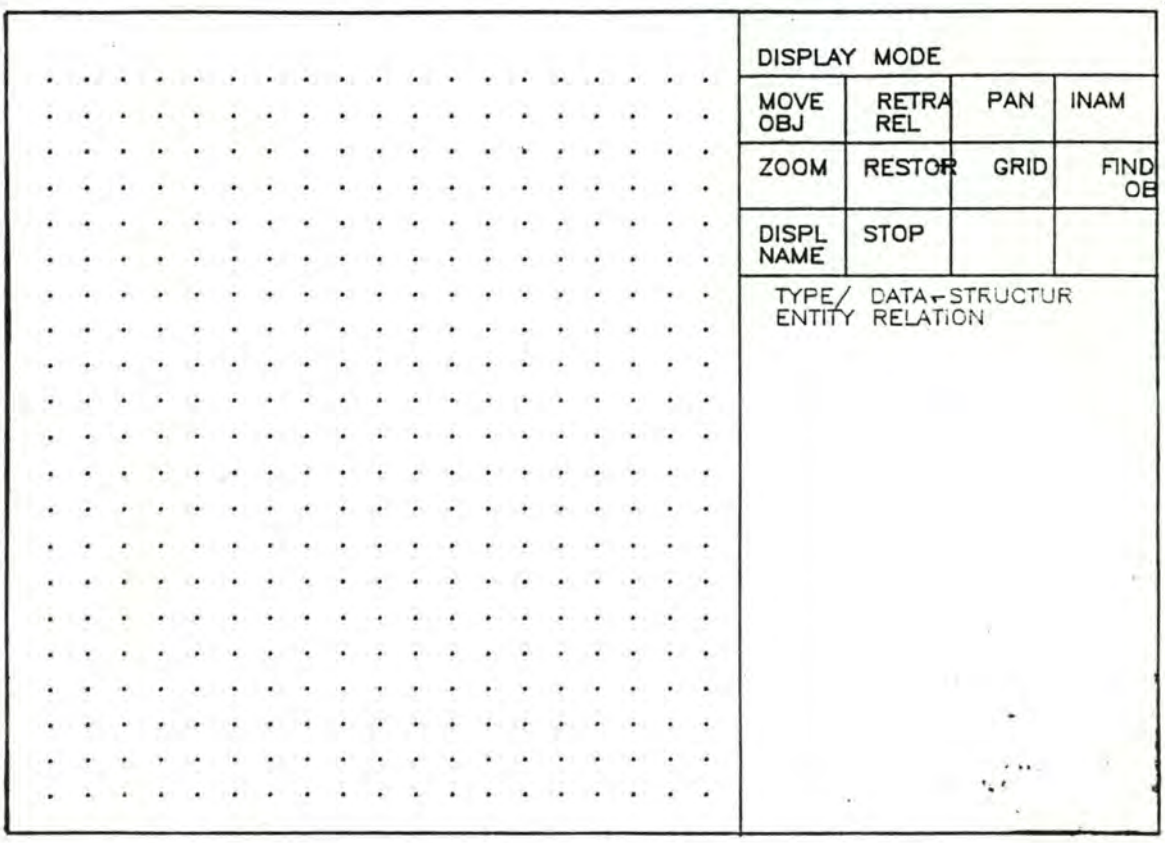
Cette commande est utilisée après un "zoom" ou un "pan" afin de restaurer l'état précédent de l'écran.

Si plusieurs "pan" ont été réalisés successivement, seul le dernier état de l'écran a été mémorisé. Dès lors, si l'utilisateur a recours deux fois à la présente commande, le programme réaffichera la vue du diagramme obtenue lors de l'entrée dans ce mode de commande (vue par défaut).

g)Visualiser la découpe de la zone de travail en case. (grid)

Nous parlerons de la fonction de cette découpe lorsque nous traiterons des problèmes d'implémentation (Système de coordonnées exposé lors de la troisième partie de ce mémoire).

Retenons simplement que les unités adressables dans la zone de travail sont des "cases" ne pouvant contenir au maximum qu'une figure .



h)Trouver un objet dans un diagramme. (Find obj)

L'utilisateur donne le nom de l'objet.La figure dans laquelle se trouve cet objet sera indiquée.

Si cette figure ne se situe pas actuellement dans la zone de travail, un déplacement de la fenêtre sera effectué,le nouveau centre de l'écran étant la figure. (cette commande est propre au GOI)

i)Afficher le nom d'un objet. (Displ. name)

Cette commande permet d'obtenir le nom de l'objet dont la figure apparaît dans une case indiquée par l'utilisateur .

Ce nom apparaîtra sur la ligne des messages. (cette commande est propre au GOI)

j)Retourner au mode de commande "Control mode" (Stop)

L'utilisateur a terminé ses manipulations sur le présent diagramme.

2.3.2.3.3. Création/completion mode.

Nous allons retrouver ici l'essentiel des commandes disponibles en "Display mode ".

L'utilisateur disposera en plus des commandes "continue","add" et "stop" dont les fonctions seront expliquées ci après.

Principe de création d'un diagramme.

Le programme parcourt la base de données des spécifications à partir de la (ou des) racine(s) du diagramme et en suivant les relations

autorisées.

Supposons que la relation présente soit R et qu'elle touche les objets A,B,C.Après avoir vérifié si ces objets sont d'un type autorisé, le programme va veiller à ce que l'utilisateur place ces objets (s'ils ne le sont pas déjà).

Soit,par exemple,l'objet A à placer.Cet objet devient l'"information courante" (à ajouter).L'utilisateur en est informé.Il peut soit s'occuper directement du placement de l'objet,soit effectuer quelques opérations préparatoires.Pour placer l'objet,il lui suffit de sélectionner la commande "ADD".

Par opérations préparatoires,nous entendons des opérations tel le déplacement d'un objet,la recherche d'un objet,un appel à la commande "Pan",etc... .

Tous les objets d'une relation une fois placés,la relation elle-même devient "information courante à ajouter" et sera automatiquement tracée.

Notons que l'utilisateur peut ne pas désirer représenter un objet dans un diagramme.Dans ce cas,il lui suffit de sélectionner la commande "CONTINUE" (au lieu de "ADD") lorsque cet objet est courant.

A la fin du parcours de la base de données des spécifications,le programme retourne en "Control mode"

Principe de complètement d'un diagramme.

Le programme vérifie si tous les objets et toutes les relations enregistrées dans la base de données graphique existent toujours dans la base de données des spécifications.Les informations qui n'ont plus de correspondants seront supprimées.

Ensuite le programme parcourt la base de données des spécifications à partir de la (des) racine(s) du diagramme et en suivant les relations autorisées.

Pour chaque objet,pour chaque relation

sélectionnée, le programme va voir si il ou elle se trouve dans la base de données graphique. Dans la négative, il (elle) devient "information courante"...

Notons que les objets volontairement écartés par l'utilisateur sont tout de même enregistrés dans la base de données graphique.

Liste des commandes.

- MOVE OBJ:Déplacer un objet
- RETRA REL:Retracer une relation
- PAN:Déplacer la fenêtre de vue sur le diagramme
- INAM:Déplacer la fenêtre de vue sur la liste des noms d'objets affichés.
- ZOOM:Agrandissement
- RESTOR:Remettre l'écran dans son état précédent
- GRID:Visualiser la découpe de la zone de travail en grille
- FIND OBJ:Trouver un objet
- DISPL NAME:Afficher le nom d'un objet
- ADD:Ajouter l'"information courante"
- CONTINUE:Omettre l'"information courante"
- STOP:
  - Interruption de la création ou du complètement. Tous les objets et relations restants sont considérés comme écartés
  - Retour au niveau de commande "Control mode"

L'écran:

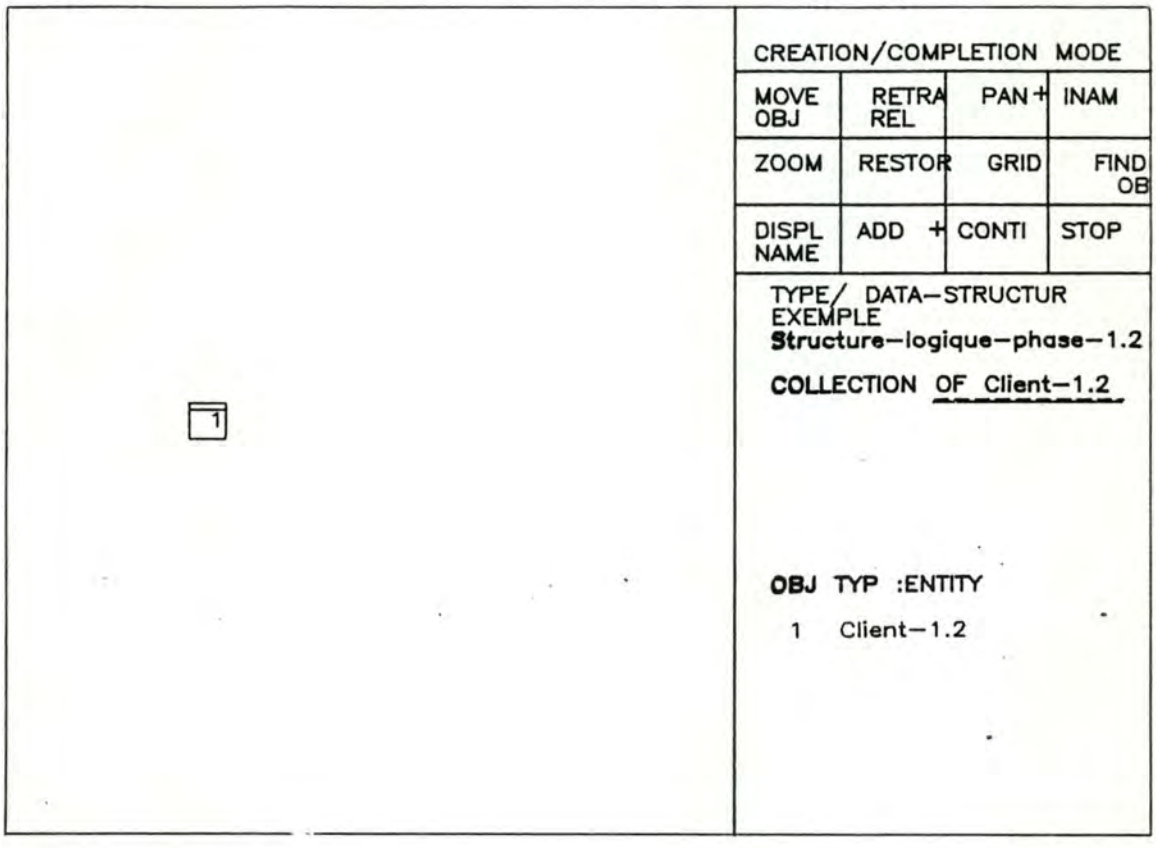
La "zone informations diverses" reprend:

- Le type du diagramme courant
- Le nom du diagramme courant

- L'instruction DSL correspondant à la relation qui a permis d'obtenir l'information courante.Cette dernière y sera soulignée. (CFR client 1.2)

- Si l'information courante est un objet,son type sera indiqué.

- La liste (ou une partie) des numéros et des noms des objets apparaissant déjà dans la zone de travail

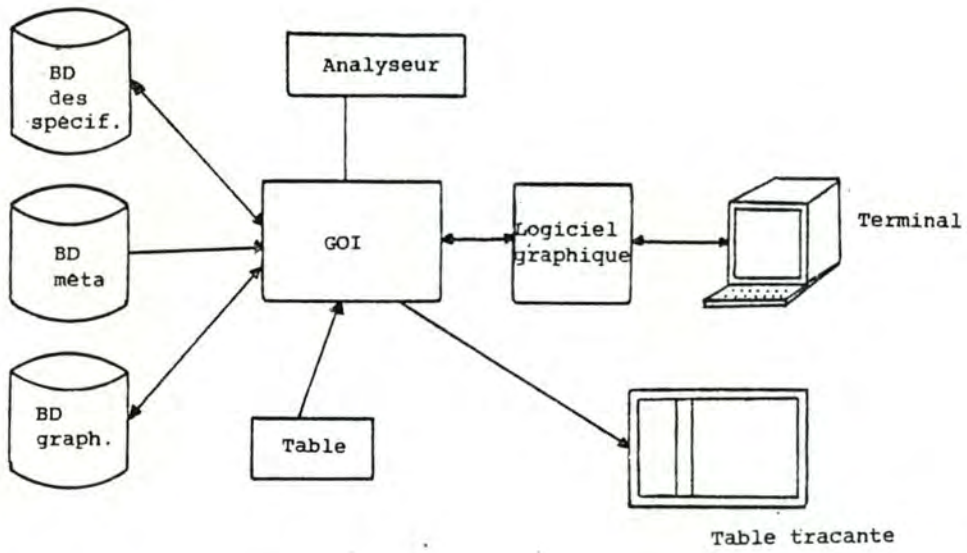


Remarquons,pour terminer ce chapitre et cette partie,que la création et le complètement d'un diagramme ont été regroupés en un seul mode de commandes car,pour l'utilisateur,ces deux activités revêtiront la même apparence.

Les contrôles supplémentaires réalisés lors du complètement d'un diagramme seront invisibles de l'extérieur.

PARTIE 3 : REALISATION DU GRAPHICAL OUTPUT INTERFACE

Afin d'illustrer et de décrire les différentes tâches accomplies lors de la réalisation du GOI, reprenons le schéma de son architecture générale.



a. Table de définition d'un diagramme [chapitre 1]

Il a fallu étendre la table de spécification d'un langage graphique du GI afin d'offrir à l'utilisateur une plus grande richesse au niveau des représentations graphiques.

b. Base de données graphiques [chapitre 2]

Le graphical Output Interface permettant la réalisation de dessins plus complexes, il a également fallu modifier la structure des informations nécessaires pour retracer un diagramme.

c. Graphical Output Interface [chapitre 3]

Après avoir modifié la structure de la table et la structure de la base de données, le travail de programmation pouvait commencer.

En fonction des différences d'objectifs, en fonction des extensions des capacités graphiques, des extensions de la table et en fonction du changement de base de données graphique, nous avons réalisé tantôt des ajoûts de nouveaux modules, tantôt des extensions ou des adaptations de modules du GI.

Chapitre 1 . LA TABLE DE DEFINITION DES DIAGRAMMES

La table du Graphical Output Interface permet de définir les différentes représentations graphiques que l'on désire obtenir. Elle assure la portabilité de l'interface,sa programmation étant indépendante de tout principe et de toute conception locale.

Cette table constitue une extension de la table de définitions d'un langage graphique du GI.

Les principales améliorations apportées sont reprises ci-dessous:

Représentations des objets.

Le nom d'un objet n'est plus systématiquement centré au sein d'une figure.L'utilisateur définira lui-même dans la table la position du nom.

La table du GOI permet de définir des emboitements de représentations. Le nom d'un objet peut apparaître dans la figure d'un autre objet.

Le nom d'un objet peut être tracé automatiquement et parrallèlement à un vecteur d'une relation.

Le nom d'un objet peut être positionné manuellement.

L'utilisateur peut définir dans la table la taille de la figure d'un type d'objet.

Représentation des relations.

Les vecteurs d'une relation peuvent être tracés avec une flèche ou sans une flèche à leur extrémité.

Les vecteurs d'une relation peuvent être tracés avec un point ou sans un point à leur origine.

Le nom d'une relation peut apparaître sur un dessin. Il sera positionné soit manuellement soit parallèlement à un des vecteurs de la relation.

Nous allons ci-après donner une description de la table du GOI. Le lecteur pourra consulter le manuel de l'administrateur du GI [GIAD.82] s'il désire réaliser une comparaison plus approfondie entre les tables des deux interfaces.

1.1. Description générale de la table.

La personne chargée de construire la table du GOI, communément appelée l'administrateur du système, va devoir accomplir différentes tâches:

1.1.1. Définir les types de diagrammes.

Les types de diagrammes sont identifiés dans le GOI par un entier qui doit être unique. Chaque diagramme doit aussi avoir un nom unique décrivant l'aspect du système qu'il représente.

1.1.2. Définir les objets légaux pour un type de diagramme

Chaque type de diagramme peut voir son contenu limité à un sous-ensemble des objets légaux du langage de spécifications.

1.1.3. Définir les relations légales pour un type de diagramme.

Les types de relations retenues dans un type de diagramme devront être limitée à l'ensemble nécessaire pour représenter l'aspect du système repris par le type de diagramme.

1.1.4. Définir le mode de représentation des types d'objets.

Le choix des figures assignées aux objets joue un rôle très important dans la présentation des informations. L'administrateur du système devra préciser les figures employées par type de diagramme et par type d'objet.
L'éventail disponible est repris ci-dessous:

1. Cercle
2. Deux lignes horizontales parallèles
3. Moitié gauche d'un rectangle
4. Carré
5. Boite de décision
6. Carte d'E/S
7. Octogone
8. Rectangle
9. Document
10. Tambour (symbole du disque)
11. Boite tronquée
12. Parallélogramme

13. Rien (aucune ligne)
14. Symbole d'entrée externe
15. Symbole de sortie externe
16. Symbole de sortie interne
17. Symbole d'entrée interne
18. Capsule
19. Carré contenant un triangle
21. Symbole d'une bande
25. Symbole d'une entité
26. Dessin d'une association
27. Dessin d'un point de synchronisation
28. Triangle

Les numéros identifient les figures.

L'administrateur devra également définir la taille de la figure d'un type d'objet (en nombre de cases).Il précisera encore si un objet d'un certain type peut apparaître plusieurs fois dans un diagramme et s'il peut inclure dans sa représentation graphique d'autres objets.

L'administrateur déterminera ainsi le mode de représentation d'un type d'objet par un nombre de six chiffres.

9 9 99 99

1. Le premier chiffre indique si un objet du type concerné peut apparaître plusieurs fois dans un diagramme (ex:system-paramater)

Si 0 = non
1 = oui

2. Le deuxième chiffre indique si la figure associée à un objet du type concerné peut contenir d'autres objets. (ex:entité<-->élément)

Si 0 = non
1 = oui

3. Les deux chiffres suivants définissent la taille de la figure:

le premier donne la largeur en nombre de cases
le deuxième donne la hauteur en nombre de cases

4. Les deux derniers chiffres donnent le numéro de la figure.

1.1.5. Définir le mode de représentation des types de relations

L'administrateur définit le mode de représentation d'un type de relations dans un type de diagramme par un nombre de trois chiffres: 9 9 9

1.Le premier chiffre définit le type de lignes des vecteurs de connexions de la relation:

0 : aucune ligne
1 : ligne continue
2 : ligne pointillée
3 : ligne hachurée

Rappelons qu'un vecteur est composé d'un ou plusieurs segments de droite contigus.

2. Le deuxième chiffre indique si le dernier segment de droite d'un vecteur de la relation doit se terminer par une flèche.

0 : non
1 : oui

3.Le dernier chiffre indique si le premier segment de droite d'un vecteur de la relation doit commencer par un point.

0 : non
1 : oui

1.1.6. Définir le placements des noms d'objets par type

L'administrateur caractérise la manière de placer les noms des objets d'un type par type de diagramme grâce à un nombre de cinq chiffres:

9 9 999

1. Le premier chiffre indique le type de positionnement du nom:

0: positionnement manuel

1: positionnement parallèle à un vecteur.

Ce vecteur sera le premier vecteur reliant le Fromcorner et le Tocorner [YAMA.77] de la relation courante si les trois derniers chiffres du nombre sont égaux à 0 sinon ce sera le dernier.

2: positionnement dans la case courante sur une ligne déterminée par la valeur d'un compteur incrémenté à chaque nouvel ajoût d'un nom dans la case, nom ayant un type de positionnement semblable.

La case courante est la case du fromcorner de la relation ayant permis de trouver l'objet actuel.

Ce type de positionnement est particulièrement adapté pour insérer des noms d'éléments dans le dessin d'une entité.

Remarquons que, si le nombre de lignes déjà enregistrées dans la case devient supérieur à une certaine limite, la hauteur de la figure de l'objet englobant sera agrandie automatiquement.

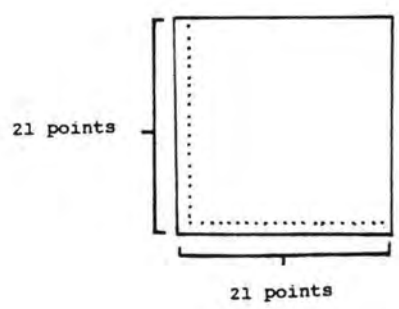
3: positionnement dans une case sélectionnée par l'utilisateur
(le nom et la figure de l'objet appaîtront dans cette case).

2. Le deuxième chiffre indique le type de cadrage du nom (par rapport à la coordonnée expliquée en 3)

1. à gauche
2. centré
3. à droite

3. Le dernier chiffre donne la coordonnée du nom au sein de la case:

La case est divisée en 21 points sur 21



La coordonnée sera exprimée par le numéro d'un point;le coin inférieur gauche étant le point 1 et le coin supérieur droit étant le point 441 (21*21)

1.1.7. Définir le placement des noms des relations par type

L'administrateur définit le mode de placement du nom d'un type de relation par un chiffre:

0:le nom n'est pas affiché
1:le nom est positionné par l'utilisateur
2:le nom est parallèle au dernier segment de droite du vecteur de la relation.

Le nom qui apparaîtera sera celui donné par l'administrateur dans la deuxième section de la table,section définie ci-après.

1.2. Structure de la table

La table est divisée en 5 sections différentes:

- Types d'objets
- Types de relations
- Types de diagrammes
- Modes de représentations des objets et des relations
- Modes de placements des noms des objets et des relations

1.2.1. Section des types d'objets

Les types d'objets définis dans le langage de spécification(ISDL-CS) sont repris par valeur croissante de l'OBCODE (nombre identifiant d'un type d'objet dans la base de données meta)

L'OBCODE est justifié à droite dans les colonnes 1-5 d'une ligne. Le nom de l'objet est justifié à gauche à partir de la colonne 7 d'une ligne.

```
Exemple : colonne 1 2 3 4 5 6 7

                      5   ATTRIBUTE
                    8 5   SET
```

1.2.2. Section des types de relation

Les types de relations définis dans le langage de spécification (ISDL-CS) sont repris par valeur croissante du RTCODE (nombre identifiant d'un type de relation dans la base des données méta). Le RTCODE est justifié à droite dans les colonnes 1-5 d'une ligne. Le nom de la relation est justifié à gauche à partir de la colonne 7 d'une ligne.

1.2.3. Section des types de diagramme

Deux lignes minimums sont nécessaires pour définir un type de diagramme.

-La première définit le type.

-La deuxième (et éventuellement les suivantes) donne les types d'objets et les types de relations autorisés pour ce type de diagramme.

Format:

Ligne de définition du type:

col 1-5:
Numéro identifiant du type de diagramme (justifié à droite)

col 7-36:
Nom du type de diagramme (justifié à gauche)

col 38-40:
Taille du diagramme en nombre de cases (justifié à droite)

- Si le nombre introduit est positif, il désignera la largeur du diagramme, la hauteur étant calculée à partir de la largeur en fonction du rapport des côtés d'une page de la table traçante (rapport défini dans un "*DEFN" de la librairie GRDI). La hauteur sera toujours inférieure ou égale à la largeur. La vue par défaut du diagramme reprendra le coin inférieur gauche de sa surface.

- Inversément, si le nombre introduit est négatif, la valeur absolue désignera la hauteur du diagramme. La largeur sera automatiquement calculée et sera toujours inférieure ou égale à la hauteur. La vue par défaut du diagramme reprendra le coin supérieur gauche de sa surface.

Ligne du contenu légal du type:

col 7-80:
Numéros des types d'objet (OBCODE) et des types de relation (RSCODE) avec au moins un blanc entre chaque numéro.

Un nombre séparateur est placé entre le dernier numéro d'objet et le premier numéro de relation.La valeur de ce séparateur est donnée par la constante (*defn) CONT-SGMT et est égale à "999".

Soulignons le fait que les types de relation seront identifiés par des RSCODE;cela signifie que l'administrateur devra également préciser le fromcorner et le tocorner du type de relation [YAMA.77].Cette précision n'engendrera pas l'exclusion des informations fournies par les autres coins de la relation;mais elle permettra d'obtenir les renseignements nécessaires aux placements des noms de certains types d'objet.

Si la description du contenu légal d'un type de diagramme doit prendre plusieurs lignes,le string "CC" servira de symbole de continuation.

Exemple de définition d'un type de diagramme:

```
1 data-stucture
25 30 40 75 85 95 999 6521 CC
7512 8013 8512
```

Dans cet exemple,on définit un diagramme de type "data-structure" (schéma-conceptuel) et de numéro 1.

Les numéros (OBCODE) des types d'objets autorisés sont:
- 25 (Element dans DSL)
- 30 (Entite)
- 40 (Groupe)
- 75 (Relation)
- 85 (Set)
- 95 (Système paramètre)

Les numéros déterminant les types de relations autorisées sont:

- 6521
  Relates (65),le fromcorner étant le coin 2 et tocorner le coin 1
- 7512
  Collection of (75),le fromcorner étant le coin 1

et le tocorner le coin 2

- 8013
  Connectivity is (80),le fromcorner étant le coin 1 et le tocorner le coin 3
- 8512
  Consist of (85),le fromcorner étant le coin 1 et le tocorner le coin 2

1.2.4. Section déterminant les modes de représentations des objets et des relations

Cette section décrit par type de diagramme le mode de représentation des types d'objets et des types de relations.

Deux principes régissent la présentation de cette section:

1. L'ordre des descriptions des modes de représentations par type de diagrammes doit être semblable à celui adopté pour la définition des types de diagrammes.

exemple:

Si la section définissant les types de diagrammes est:

```
1 DATA-Structure                30

    1 2 999 3

2 DYNAMIC                      -40

    7 8 999 5 7
```

La section décrivant les modes de représentation des types d'objets et des types de relations pourrait être:

```
11124  11125  999  100

901128  1105  999  110  100
```

La première ligne définira les représentations graphiques des types d'objets et des types relations du premier type de diagramme (DATA-STRUCTURE).

La deuxième ligne définira les représentations graphiques des types d'objets et des types de relations du deuxième type de diagramme (DYNAMIC).

2. Les modes (types) de représentations des types d'objets

et des types de relations d'un type de diagramme sont presentes dans une sequence identique à celle du contenu de ce type de diagramme.

En reprenant l'exemple du principe précédent,on constate que pour le diagramme de type DATA-STRUCTURE:

-L'objet de type 1 voit sa représentation définie par le nombre 11124

-L'objet de type 2 voit sa représentation définie par le nombre 11125

-La relation de type 3 voit sa représentation définie par le nombre 100

Pour comprendre la signification des nombres déterminant la représentation graphique d'un type d'objet ou d'un type de relation, nous devrons nous référer à la définition des modes de représentations des types d'objets et des types de relations enoncée précédemment (points 1.1.4 et 1.1.5)

Remarquons que le séparateur entre types d'objet et types de relation sera à nouveau "999"

Le symbole de continuation d'une ligne correspondra toujours au string "CC".

Exemple:

Nous allons décrire ici le mode de representation du diagramme de type "DATA STRUCTURE" (défini précédemment) :

0 11125 0 11126 0 90000 999 100 0 00

Explication:

Rappelons que,selon le deuxième principe caractérisant la présentation de cette section,le premier nombre de la ligne décrit la représentation graphique du premier type d'objet autorisé dans le présent type de diagramme,etc.

- L'objet d'OBCODE 25 ne se voit associer aucune figure (0).Il s'agit pour DSL d'un élément.

- L'objet d'OBCODE 30 est une entité.Sa représentation graphique sera: (0) 1 11 25

  a.le dessin d'une entité (cfr point 1.1.4)
  b.avec une largeur et une hauteur d'une case
  c.pouvant contenir d'autres objets

- L'objet d'OBCODE 40 ne se voit associer aucune figure.Il s'agit d'un groupe.

- L'objet d'OBCODE 75 est une relation.Sa représentation graphique sera:

  (0) 1 11 25

  a.le dessin d'une relation
  b.avec une largeur et une hauteur d'une case
  c.pouvant contenir d'autres objets

- L'objet d'OBCODE 85 ne se voit associer aucune figure. Il s'agit d'un SET.

- L'objet d'OBCODE 95 est un système paramètre.Aucune figure ne lui est associée mais son nom pourra apparaitre plusieurs fois (90000).

Seule la relation de RSCODE 6521 (relates) aura une représentation graphique.Ses vecteurs seront tracés en traits continus sans flèches ni points.

1.2.5. Section de détermination du placement des noms

Cette section définit par type de diagramme le mode de placement des noms d'objets de chaque type et des noms de relations de chaque type (voir codification point 1.6).Ces définitions seront présentées selon les principes déja exposés pour la section précédente.

Exemple:

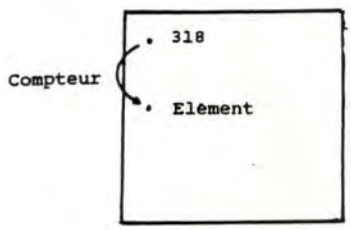
Nous allons décrire ici le mode de placement des noms d'un diagramme de type "data-structure".

21318 32389 21317 32389 0 11000 999 CC 0 0 0 0

Explication:

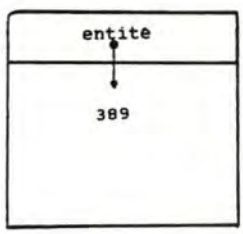
- "21318":

  Un élément est placé dans une case sur une ligne déterminée par la valeur d'un compteur incrémenté à chaque nouvelle insertion (le premier chiffre "2").La première ligne commencera au point 318 (les trois derniers chiffres).Le nom de l'objet sera cadré à gauche (le deuxième chiffre).



- "32389":

  le nom d'une entité sera placé dans une case sélectionnée par l'utilisateur (3).Il sera centré (2) au point 389 .



- "21317":

  le placement du nom d'un groupe est semblable à celui du nom d'un élément si ce n'est que le nom

sera un peu plus à gauche (318 -> 317).

- "32389":

  le nom d'une relation est traité d'une manière analogue à celui d'une entité.

- "11000":

  le nom d'un système paramètre sera placé parallèlement à un segment de droite d'un vecteur (le premier de la relation courante)

Les noms des types de relation ne seront pas affiché (0000)

Les cinq sections de la table que nous venons de décrire seront séparées par une ou plusieurs lignes commencant par un "C".

Un exemple complet est repris ci-après.Le lecteur pourra y trouver une définition des trois principaux types de diagramme: structure des données,dynamique des traitements,diagramme des flux.

LEGAL OBJECT DEFINITIONS

```
  5 ATTRIBUTE
 15 CONDITION
 25 ELEMENT
 30 ENTITY
 40 GROUP
 45 INTERFACE
 52 DAT-INTERVAL
 55 MENU
 60 MESSAGE
 65 PROCESS
 75 RELATION
 85 SET
 90 SYNCHRO-POINT
 95 SYSTEM PARAMETER
```

LEGAL RELATION-DEFINITIONS

```
  5 ACTIVE
 10 ADDS
 15 APPLIES
 25 ATTRIB
 50 CAPACITY
 65 RELATIONS
 75 CONDITION-OF
 80 CONNECTIVITY
 85 CONSISTS-OF
 95 ON-CONDITION-OF
 96 ON-CONNECTIVITY
100 ON-INPUT-OF
101 ON-INPUT
105 ON-PROCESS
106 ON-PROCESS
110 ON-RELIES
111 ON-RELIES
113 ON-RESOURCE
114 ON-RESOURCING
115 ON-SHORTMEM
120 ON-TIMEIN
121 ON-TIMEIN
123 ON-TIMEOUT
124 ON-TIMEOURING
125 ON-TIMEOUTMEM
126 ON-TIMEOUT
127 ON-TIMEOUTER
129 ON-TIMEOUTS
130 ON-TIMEOURING
131 ON-TIMEOUTMEM
155 OPERATIVE
180 OPERATE-OPT
191 OPERATE-IF
192 OPERATE-IFNOT
195 IDENTIFIES
220 MODIFIES
225 PART-OF
230 PERFS-OUP
255 PRIORITY
256 REC-COPIES
265 REFERENCES
270 REMOVES
275 REQUIRES
276 ROLE-NAME
277 SAME-NODE
280 SEE-HEAD
285 SHARABABLE
290 SHARES
295 SIM-VAL-INC
296 SIM-VAL-ADD
297 SIM-VAL-READ
298 SIM-VAL-WRITE
299 HIGH
301 DIST
302 DYNAMIC
303 SUBSET-OF
304 SUBTYPE-OF
305 ON-INC-PRIO
306 ON-INC-PRIO
310 ON-PROCESS-PRIO
311 ON-PROCESS-PRIO
315 ON-GROUP-TRIG
316 ON-GROUP-TRIGNOT
320 ON-GROUP-TRIG
325 ON-GROUP-TRIG
335 UNIT-PRICE
340 UTILISES
345 USES
350 VALUES-ARE
355 VALUES-IS
```

DIAGRAM DEFINITIONS

```
 1 DATA-STRUCTURE                                    30
   25 30 40 75 85 95 999 9521 7512 8013 8512
 2 DYNAMIC                                          60
   5 15 60 65 90 95 999 9512 9612 9712 10012 10112 10212 CC
   10512 10612 10712 31512 31612 31712 CC
   31512 31612 31712 30512 30612 30712 CC
   31012 31112 31212 32012 32512 32612
 3 FLOW                                             30
   45 50 65 999 15512 19112 21221 25520
```

SHAPE

```
   0 11125 0 11126 0 900000 999 100 0 0 0
   901128 1105 1109 1104 1127 900000 999 111 111 111 111 111 111 CC
   111 111 111 111 111 111 111 111 111 111 111 111 CC
   111 111 111 111 111 111 111
   2229 901109 2104 999 110 110 100 110
```

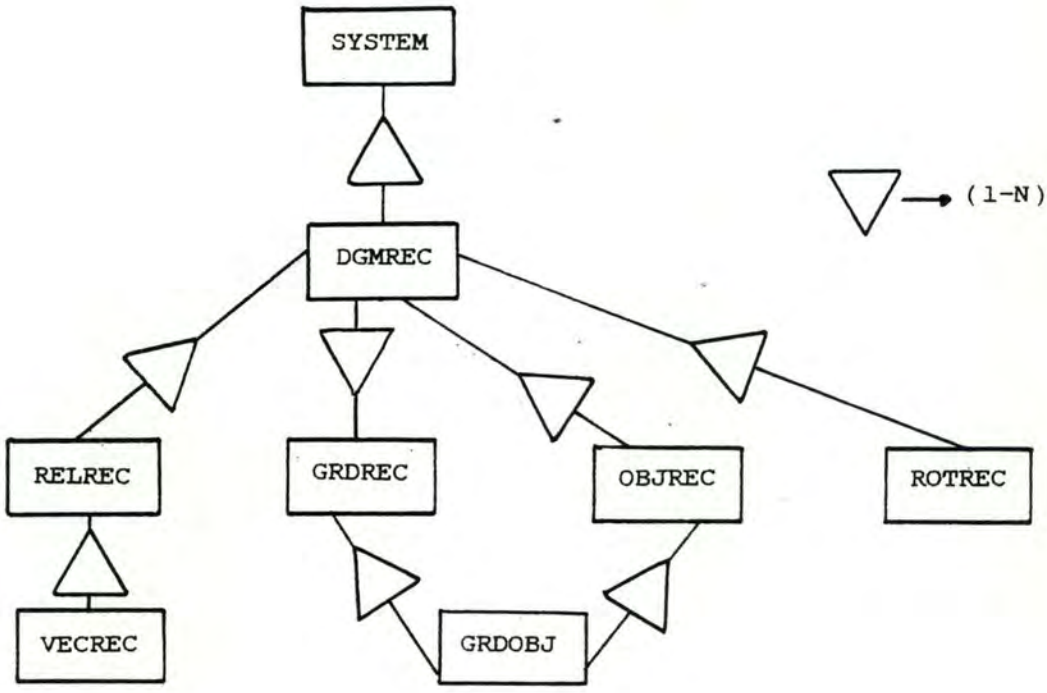
NAME

```
   21318  32389 21317 32369 0 11000 999 0 0 0 0
   32153 32220  32326 32220 32304 11000 999 0 0 0 0 0 0 0 0 0 0 0 0
   0 0 0 0 0 0 0 0 0 0 0 0 0 0
   1000 32326 32230 999 0 0 0 0 0
```

Chapitre 2. LA BASE DE DONNEES GRAPHIQUE

La base de données graphique du Graphical Output Interface contient la description des différents diagrammes existants. L'interface devra y avoir accès pour tout affichage ou toute impression.

Le schéma ci-dessous décrit la structure de cette base de données, structure tout à fait différente par rapport à celle adoptée dans le GI.



Le système de gestion de base de données utilisé est ADBMS.

ADBMS répond aux normes CODASYL et permet l'usage d'items simples ou répétitifs (fixe,variable) et autorise des chemins d'accès 1-N doté de leurs inverses [HERS.77].

Description des enregistrements

Tous les items employés sont simples.

DGMREC :
contient des informations sur un diagramme

NAMDGM : nom du diagramme
NUMDGM : numéro du diagramme
DGMTYP : type du diagramme
GRDSIZ : taille du diagramme

GRDREC :
contient des informations sur l'état d'une case dans un diagramme

NUMGRD : numéro de la case
GRDINC : valeur du compteur de lignes
(nombre de lignes déjà insérées dans la case)

OBJREC
contient des informations sur un objet dans un diagramme

OBJTYP : type de l'objet
NAMOBJ : nom de l'objet
SHPSIZ : taille de l'objet

ROTREC :
contient des informations sur une racine d'un diagramme.

NAMROT : nom de la racine

GRDOBJ :
enregistrement crée pour obtenir un chemin M-N entre un enregistrement GRDREC et un enregistrement OBJREC, un objet pouvant avoir plusieurs représentations graphiques et une case pouvant contenir plusieurs objets.

NPOSIT : position du nom de l'objet
dans la case.

NANGLE : angle définissant le plan
de base sur lequel sera dessiné
le nom

RELREC
contient des informations sur une relation d'un diagramme.

RELTYP : type de la relation
KYOBJ1 :
KYOBJ2 : Clé dans la base de données graphiques
KYOBJ3 : des objets intervenant dans la relation
KYOBJ4 :
NMPOS1 : position du nom de la relation
(dans la case qui lui est associée)
NANGLE : angle définissant le plan de base
sur lequel sera dessiné le nom.

VECREC :
contient des informations sur un vecteur d'une relation (un vecteur relie deux objets d'une relation et est constitué d'un ou plusieurs segments de droite)

KYOBJ1 : clé dans la base de données
graphique de l'objet origine du vecteur
KYOBJ2 : clé dans la base de données
graphique de l'objet extrémité du
vecteur
GRID01 :
! cases contenant les points par lesquels
! doit passer le vecteur
!
!
GRID09 :
LOCAT1 :
!
! coordonnées au sein des cases
! par lesquelles doit passer le vecteur
!
LOCAT9 :

Description des chemins d'accès

ALLDGM :
accès aux différents diagrammes

origine : SYSTEM
extrémité : DGMREC
triè sur NAMDGM

DGMGRD :
accès aux cases d'un diagamme

origine : DGMREC
extrémité : GRDREC
triè sur NUMGRD

DGMOBJ :
accès aux objets d'un diagramme

origine : DGMREC
extrémité : OBJREC
triè sur NAMOBJ

DGMREL :
accès aux relations d'un diagramme

origine : DGMREC
extrémite : RELREC
FIFO

DGMROT :
accès aux racines d'un diagramme

origine : DGMREC
extrémité : ROTREC
FIFO

GRDOB1 :

origine : GRDREC
extrémité : GRDOBJ
FIFO

GRDOB2 :

origine : OBJREC
extrémité : GRDOBJ
FIFO

GRDREL :
accès à une case associé à une relation

origine : GRDREC
extrémité : RELREC

RELVEC ;
accès aux vecteurs d'une relation

origine : RELVEC
extrémité : VECREL
FIFO

Le lecteur trouvera en annexe 1 la description en DDL de la base de données

Chapitre 3. GRAPHICAL OUPUT INTERFACE

Nous allons décrire,dans ce chapitre,l'architecture du Graphical Output Interface.Nous présenterons ensuite le système de coordonnées et de référence employé lors de la construction et lors de l'affichage d'un diagramme.

3.1. Architecture du GOI

Le Graphical Output Interface est composé de 5 modules :

GOI (Graphical Output Interface)

GOD (Graphical Output Database)

GRDI (Graphics Control)

BEN (Plotter Control)

DI (Terminal Control)

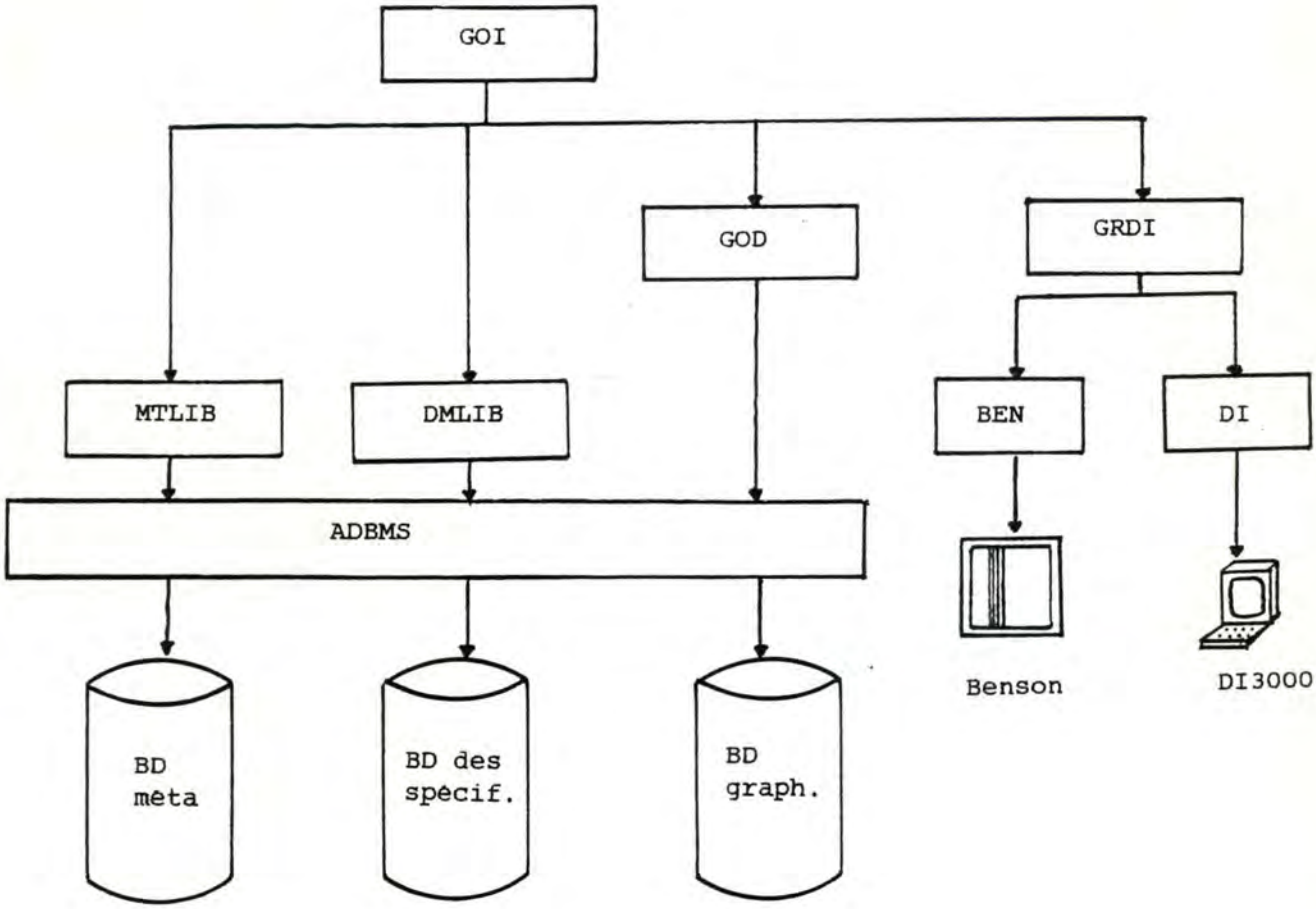
Il utilise également deux autres librairies appartenant au logiciel SEM [ISDO.81]:

MTLIB : ensemble de routines d'accès à la base de données méta [PERE.78]

DMLIB : ensemble de routines d'accès à la base de données des spécifications [BOYD.80]

Les routines de MTLIB,DMLIB et GOD feront appel aux primitives du système de gestion de base de données ADBMS [HERS.77].

L'architecture générale du Graphical Output Interface apparait sur le schéma ci-dessous .



Cette architecture est presque identique à celle du Graphical Interface d'où la librairie RMLIB (Relationship Manipulation) a complètement disparu.

La librairie DI correspond à la librairie TK, la librairie GRDI à la librairie GR et la librairie GOD à la librairie GD.

Le module BEN (Plotter control) est propre au Graphical Output Interface.

Seules les modules BEN et DI dépendent de l'installation graphique locale.

Le lecteur trouvera en annexe 2 une description des 5 modules de base du GOI, soulignant, selon le cas, les ajoûts (A), les modifications (M) ou les extensions (E) de routines.

Nous reprenons brièvement ci-après les fonctions assumées par ces modules

GOI (Graphical Output Interface)

Le module GOI est le plus important du système;ses 64 routines contrôlent l'ensemble de la session graphique.

GOD

Cette librairie contient 47 routines d'accès à la base de données graphique (création,modification,effacement d'enregistrement) .

Ces routines sont propres au GOI(sa base de données étant différente de celle du GI);elles font appel aux primitives du système de gestion de base de données ADBMS [HERS.77].

GRDI

Cette librairie contient 49 routines contrôlant l'affichage et la saisie à l'écran ainsi que l'impression sur la table tracante.

Elle est indépendante des périphériques physiques reliés au système et du logiciel graphique utilisé.Toute liason avec ces unités se fera par l'intermédiaire des routines de "BEN" ou de "DI" .

DI

Cette librairie contient un ensemble de routines graphiques faisant appel aux primitives du logiciel DI3000.Elle devra être recodée en fonction de l'installation locale.

BEN

Cette librairie contient un ensemble de routines graphiques faisant appel aux primitives de la table tracante [BENSON].

Elle devra être recodée en fonction de l'installation locale.

3.2. Système de coordonnées et de références.

Le dessin d'un diagramme peut s'étendre sur une surface d'une longueur et d'une largeur prédéfinies dans la table du GOI.

Lorsque l'on désire afficher le dessin,on retire de cette surface une portion limitée.

Le système employé est semblable à celui de la vision d'une caméra qui peut se déplacer selon notre gré au dessus d'un champ.

La surface d'un diagramme est divisée en cases référencables individuellemen.Ces cases peuvent éventuellement contenir une (et une seule) figure.

Les axes des X et des Y correspondront respectivement au bord inférieur au bord latéral gauche de l'écran.

Nous reprenons ci-dessous les principales données utilisées lors de l'emploi de ce système de coordonnées et de références :

- (SXLL,SYLL)

  désigne les coordonnées du coin inférieur gauche de l'écran. Dans le cas présent,la valeur des deux variables sera : (0,0).

- (IXLL,IYLL)

  désigne les coordonnées du coin inférieur gauche du diagramme. Ces coordonnées varieront en fonction du déplacement de la caméra (de la "fenêtre").

- NGRIDX

  désigne le nombre de cases apparaissant à l'écran sur l'axe des X.

- NGRIDY

  désigne le nombre de cases apparaissant à l'écran sur l'axe des Y.

- GRDDXY

  désigne la taille d'une case (hauteur ou largeur du carré).

- GRSTBF(2500,2)

tableau reprenant le statut des cases.Si une case est occupée,ce tableau permettra d'identifier l'objet qui y a pris place ainsi que la taille de la figure qui lui est associée.

- SCRWDTH

  désigne la largeur totale de l'écran.

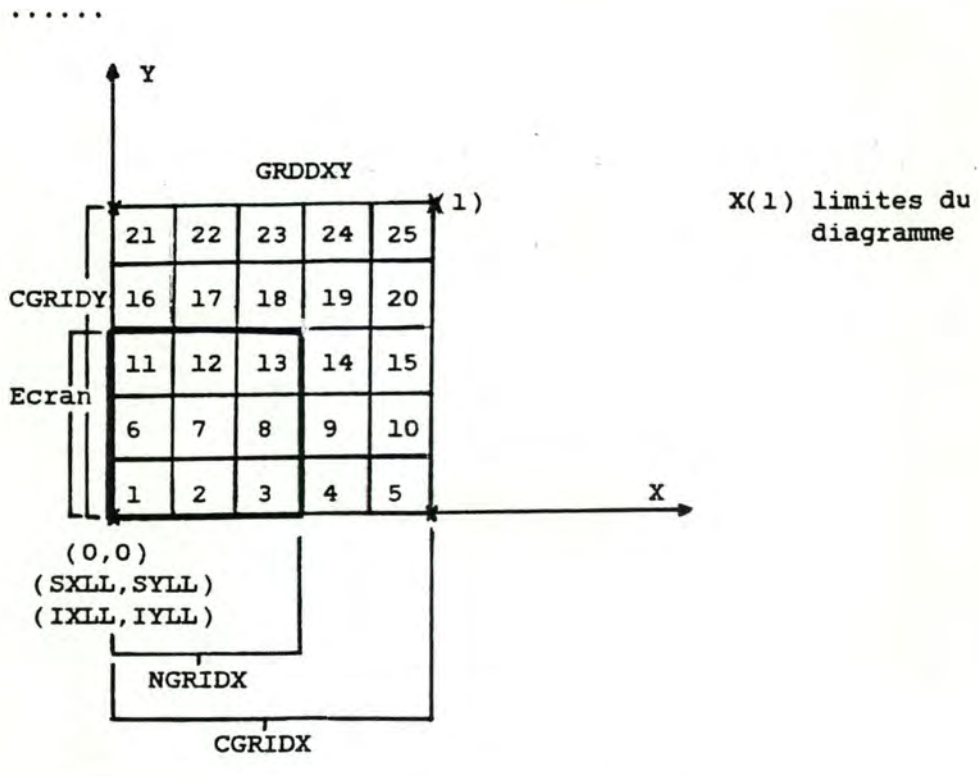
- SCRHGT

  désigne la hauteur totale de l'écran.

- MNWDTH

  désigne la largeur de la surface de l'écran réservée au menu

Nous pouvons visualiser le rôle de ces différentes données sur le schéma suivant :



Notons que,au départ,sans zoom ni pan (déplacement de la caméra), (IXLL,IYLL) est égale à (0,0) .Cela signifie que la caméra est fixée sur le coin inférieur gauche du diagramme (sauf

demande contraire dans la table de définition des diagrammes).

Comment trouver la valeur (a,b),c'est à dire les coordonnées du coin inférieur gauche d'une case ?

Soit NBOX,le numéro de cette case :

```
a = (NBOX-(((NBOX-1)/cgridx)*cgridx)-1)*GRDDXY+IXLL    [1]

b = ((NBOX-1)/cgridx)*GRDDXY+IYLL                      [2]
```

Les divisions intervenant dans ces deux formules sont des divisions entières.

Expliquons,à titre indicatif,la première formule qui permet de trouver l'abscisse du coin inférieur gauche d'une case [1].

- La division ((NBOX-1)/cgridx))) nous rend le numéro de la ligne sur laquelle apparaît la case (0 pour la première ligne,1 pour la seconde).

- On multiplie ce numéro de ligne par le nombre de cases apparaissant sur une ligne (cgridx).

- En soustrayant le résultat de cette multiplication incrémenté de 1 au numéro de la case,on obtiendra le numéro de la colonne à laquelle appartient cette case.

- On multiplie ce nouveau résultat décrémenté de 1 par la largeur d'une case.

- On ajoute enfin l'abscisse du coin inférieur gauche du diagramme. Cette abcisse variera selon la position de la caméra (voir explication d'un pan ou d'un zoom).

Un ZOOM

Lorsque l'utilisateur désire effectuer un zoom,il précise la valeur de deux variables :

- NBOX1Z désigne le numéro de la case représentant le coin inférieur gauche de la surface sur laquelle le zoom va être effectué.
- NBOX2Z désigne le numéro de la case représentant le coin supérieur droit de la surface sur laquelle le zoom va être effectué.

A partir des valeurs de ces deux variables,on trouve la largeur et la longeur de la nouvelle surface visible en nombre de cases et NGRIDX est mis à jour.

GRDDXY est également mis à jour en fonction de NGRIDX.On

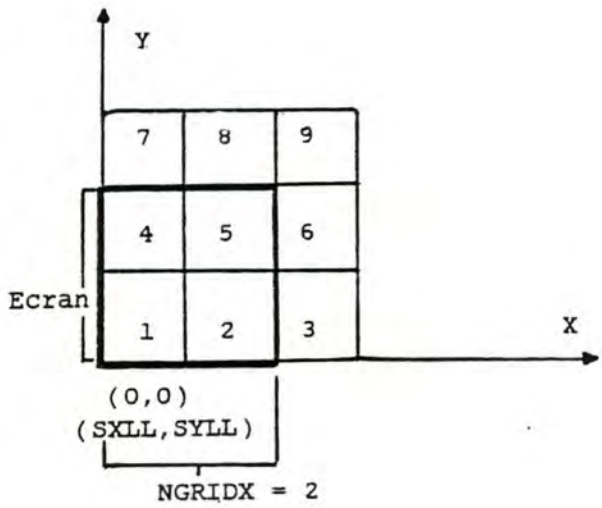
peut également trouver la nouvelle coordonnée du coin inférieur gauche du diagramme,ce dernier s'étant déplacé si le zoom ne s'est pas fait sur le coin inférieur gauche.

```
IXLL = SXLL-(NBOX1Z-((NBOX1Z/CGRIDX)*CGRIDX)-1)*GRDDXY

IYLL = SYLL-((NBOX1Z-1)/CGRIDX)*GRDDXY
```

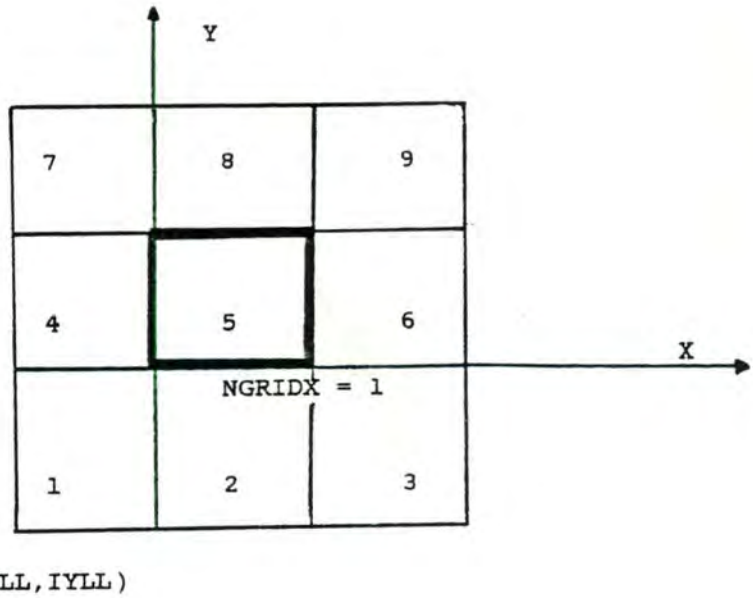
Nous pouvons représenter graphiquement les principes de réalisation d'un zoom :

Avant le zoom



Les deux croix désignent les limites de la surface sur laquelle va être effectué le zoom.Dans le cas présent,NBOX1Z et NBOX2Z seront égaux.

Après le zoom



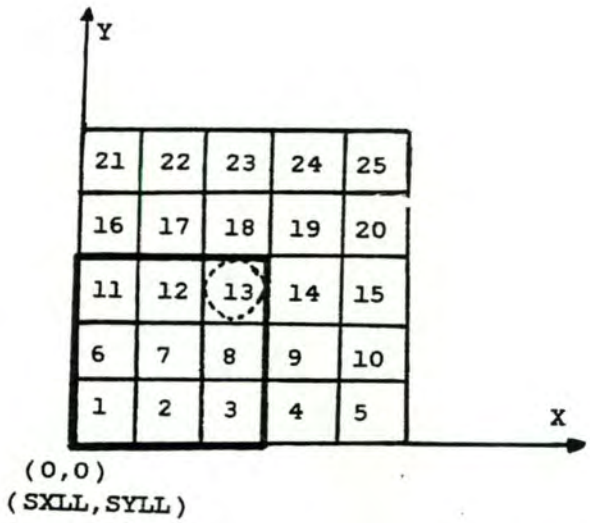
Un PAN

Le PAN correspond à un déplacement de la caméra.La partie visible du diagramme va changer.

L'utilisateur précise le numéro de la case devant devenir le nouveau centre de la surface d'affichage.

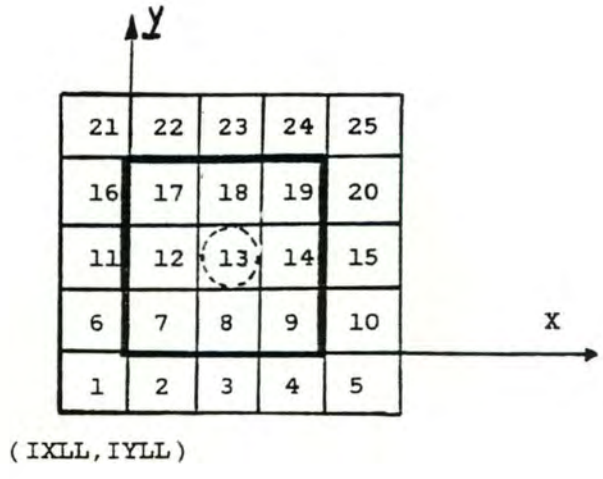
Nous illustrons ci-dessous la réalisation d'un PAN.

Avant le PAN



La grille numéro 13 doit devenir le nouveau centre de l'écran.

Après le PAN



Les nouvelles valeurs de (IXLL,IYLL) sont :

IXLL = IXLL-IX+SXLL+(SCWDTH-MNWDTH)/2

valeur de l'abcisse du centre de la surface de travail

IYLL = IYLL-IY+SYLL+SCRHGT/2

où (IX,IY) représente les coordonnées du coin inférieur gauche de la grille devant devenir le nouveau centre de l'écran.

Avantage de la découpe en cases.

La division de la surface en cases permet d'éviter facilement les problèmes de superposition de figures.Une figure ne pourra être introduite dans une case que si celle-ci est libre...

CONCLUSION

Le but de ce travail était de compléter et d'étendre un interface graphique existant, le Graphical Interface, en vue de disposer d'un interface qui se contente d'exploiter graphiquement la base de données des spécifications d'un Système d'Information.

Le Graphical Output Interface rencontre ce besoin et permet d'obtenir des représentations complètes et traditionnelles des différents aspects d'un Système d'Information.

Afin d'alleger quelque peu l'importance du travail nécessaire à la réalisation de cet interface, nous avons décidé de recourir autant que possible aux principes et aux routines du Graphical Interface.

En fonction des différences d'objectifs , en fonction des extensions des capacités graphiques , des extensions de la table et en fonction du changement de base de données graphique , nous avons réalisé tantôt des ajouts, tantôt des extensions ou des adaptations de modules.

Lors la réalisation du Graphical Output Interface, nous avons également essayé de répondre aux différents critères et aux différentes exigences devant intervenir lors de la conception d'un logiciel graphique.

Afin de respecter les principes de complétude et de minimalité des fonctions disponibles au sein d'un logiciel graphique, le GOI offre un ensemble étendu de primitives graphiques tout en restant indépendant des besoins et des désirs propres à certaines applications.

Son usage est aisé, l'interface guidant continuellement l'utilisateur lors de la construction d'un diagramme. Le traitement des erreurs est simple et l'impact de ces dernières aussi faible que possible.

Les fonctions du GOI sont indépendantes de tout périphérique graphique. Les routines faisant appel à ces unités sont localisées et aisément adaptables.

La portabilité du système est encore assurée par l'usage d'une table pour définir les différentes représentations graphiques intervenant dans le cadre d'une méthodologie de développement d'un Système d'Information. La programmation ne fût influencée par aucun langage de spécification , ni par aucune convention de représentation

particulière.

Ces différents choix permettront l'installation de l'interface sur tout système ISDLM/SEM, quelque soit le langage de spécification utilisé.

Si le système réalisé a l'avantage d'être portable et facile à utiliser, il présente cependant certaines limites au niveau des automatismes offerts lors de la construction d'un diagramme.

Le choix d'une indépendance totale vis à vis d'une représentation graphique particulière peut, dans certains cas, alourdir la tâche de l'utilisateur.

BIBLIOGRAPHIE

PARTIE 1

- [BERG.78] D. BERGERON ,R. BONO ,D. FOLEY
"Graphics programming using the core system", Computing Surveys ,vol 10,no 4,December 1978

- [BLAG.78] G. BLAKE
"Graphic shorthand as an aid to managers", Harvard Business Review ,March-April,vol 56,no 2,1978

- [BLAI.82] I. BLAKE
"Graphical user interface for business information system", MIS Quarterly , December 1982

- [DRIV.83] R. DRIVER ,C. WATSON
"The influence of computer graphics on the recall of information" , MIS Quarterly ,March 1983

- [DURE.82] J. DURETT ,J. TREZONA
"How to use color displays effectively" Byte ,vol 7,number 4,April 1982

- [FOLE.82] J. FOLEY ,A. VANDAM
"Fundamentals of interactive computer graphics" ,Addison-Wesley,Systems programming series,1982

- [GKS.82] "Graphical Kernel System : functional description", Draft International Standard ISO/DIS,January 1982

- [HOPG.82] F. HOPGOOD
"The road to graphics standards", Computer-aided Design ,vol 14,number 4,July 1982.

- [JANS.80] R. JANSON
"Graphic Indicators or operations" Harvard Business Review ,vol 58,number 6,November-December 1980.

- [MAR1.81] C. MARSON
"Terminaux:la couleur en plus [1]" , 01 Informatique ,no 147,fevrier 1981.

- [MAR2.81] C. MARSON
"Terminaux:la couleur en plus [2]", 01 Informatique ,no 148,mars 1981.

- [MONI.79] P. MONIN
"Techniques graphiques inreractives pour l'aide a la decision",
01 Informatique ,no 127,Janvier-Fevrier 1979

- [MICH.78] J. MICHENER ,A. VANDAM
"A functional overview of the core system with glossary", Computing Surveys ,vol 10,no 4,December 1978.

- [NEWN.78] W. NEWMAN ,A. VANDAM
"Recent efforts towards graphics standardization", Computing Surveys ,vol 10,no 4,December 1978.

PARTIE 2 ET PARTIE 3

- [BODA.83] F. BODART ,Y. PIGNEUR
"Conception assistée des applications informatiques.
Pemière partie :l'étude d'opportunité et l'analyse conceptuelle",
Ed. Masson,Paris,1983.

- [BOYD.80] L. BOYDSTEM ,K. KANG ,Y. YAMAMOTO
"DM subsystem G2.0", MEMO 232,ISDOS PROJECT,University of Michigan,Augustus 1980,ref 763-2238.

- [DSA.82] "DSA : Manuel de réfèrence",Institut d'Informatique,
Facultès Universitaires de Namur, 1982.

- [HERS.77] E. HERSEY
"A data base management system [ADBMS] based on

DBTG 71" , ISDOS PROJECT ,University of Michigan,November 1977,Ref 7730-0191-0.

- [GIA.82] "GI Administrator's Manual" ISDOS PROJECT,University of Michigan,Augustus 1982,Ref 82GI3-0402-0.

- [GIU.82] "GI User's Manual", ISDOS PROJECT ,University of Michigan,Augustus 1982,Ref 82GI2-0402-0.

- [ISDO.81] "An introduction to the use of information system language definition manager" (ISLDM) and the encyclopedia manager (SEM)" , ISDOS PROJECT ,University of Michigan,September 1981,ref 81SEM-0320-2.

- [PERE.78] M. PEREGOY "MT subsystem G2.0" ,June 1978 ,ISDOS PROJECT ,University of Michigan,MEMO 235.

- [PSL.82] "PSL/PSA user's reference Manual version 5.2", nr ref 171,ISDOS PROJECT,University of Michigan,July 1982.

- [YAMA.77] Y. YAMAMOTO g"The structure and contents of the meta data base" ISDOS PROJECT ,University of Michigan,Meta System Memorandum,November 1977,ref 763-2238.

ANNEXE 1

DDL de la base de donnees graphique

```
RECORD  DGMREC
ITEM    NAMDGM  CHAR    030
ITEM    NUMDGM  INTEG   031
ITEM    DGMTYP  INTEG   031
ITEM    GRDSIZ  INTEG   031

RECORD  GRDREC
ITEM    NUMGRD  INTEG   031
ITEM    GRDINC  INTEG   031

RECORD  GRDOBJ
ITEM    NPOSI1  INTEG   031
ITEM    NANGLE  INTEG   031

RECORD  OBJREC
ITEM    OBJTYP  INTEG   031
ITEM    NAMOBJ  CHAR    030
ITEM    SHPSIZ  INTEG   031

RECORD  RELREC
ITEM    RELTYP  INTEG   031
ITEM    KYOBJ1  INTEG   031
ITEM    KYOBJ2  INTEG   031
ITEM    KYOBJ3  INTEG   031
ITEM    KYOBJ4  INTEG   031
ITEM    NMPOS1  INTEG   031
ITEM    NANGLE  INTEG   031

RECORD  ROTREC
ITEM    NAMROT  CHAR    030

RECORD  SDGREC
ITEM    KYDGRM  INTEG   031

RECORD  VECREC
ITEM    KYOBJ1  INTEG   031
ITEM    GRID01  INTEG   031
ITEM    GRID02  INTEG   031
ITEM    GRID03  INTEG   031
ITEM    GRID04  INTEG   031
ITEM    GRID05  INTEG   031
ITEM    GRID06  INTEG   031
ITEM    GRID07  INTEG   031
ITEM    GRID08  INTEG   031
ITEM    GRID09  INTEG   031
ITEM    KYOBJ2  INTEG   031
ITEM    LOCAT1  INTEG   031
ITEM    LOCAT2  INTEG   031
ITEM    LOCAT3  INTEG   031
ITEM    LOCAT4  INTEG   031
ITEM    LOCAT5  INTEG   031
ITEM    LOCAT6  INTEG   031
ITEM    LOCAT7  INTEG   031
ITEM    LOCAT8  INTEG   031
ITEM    LOCAT9  INTEG   031

SET     ALLDGM  SORTED      NAMDGM
OWNER   SYSTEM
MEMBER  DGMREC
```

```
SET     DGMGRD  SORTED      NUMGRD
OWNER   DGMREC
MEMBER  GRDREC

SET     DGMOBJ  SORTED      NAMOBJ
OWNER   DGMREC
MEMBER  OBJREC

SET     DGMREL  FIFO
OWNER   DGMREC
MEMBER  RELREC

SET     DGMSDG  FIFO
OWNER   DGMREC
MEMBER  SDGREC

SET     DGMROT  FIFO
OWNER   DGMREC
MEMBER  ROTREC

SET     GRDOB1  FIFO
OWNER   GRDREC
MEMBER  GRDOBJ

SET     GRDOB2  FIFO
OWNER   OBJREC
MEMBER  GRDOBJ

SET     GRDREL  FIFO
OWNER   GRDREC
MEMBER  RELREC

SET     RELVEC  FIFO
OWNER   RELREC
MEMBER  VECREC
```

ANNEXE 2

Description des modules

Nous allons décrire ci-après les 5 modules de base du GOI, en soulignant,selon le cas,les ajoûts (A),les modifications (M) ou les extensions (E) de routines.

1. GOI (Graphical Output Interface)

Le module GOI est le plus important du système;ses 66 routines contrôlent l'ensemble de la session graphique.

1. *MAIN,GIMAIN (M)

Description

Programme principal de l'interface

2. GIBDGM(nmode,ierr) (A)

Description

Cette routine contrôle la construction d'un diagramme en "creation/completion mode" .

Elle exploite une matrice contenant différentes informations sélectionnées sur le système et indique à l'utilisateur les objets et les relations à insérer dans un diagramme.

Paramètres

NOM	USAGE	TYPE	Description
NMODE	ENTREE	ENTIER	indique si l'on crée (0) un diagramme ou si l'on complète (1) un diagramme
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

3. GIBMAT(ierr) (A)

Description

Cette routine construit une matrice reprenant l'ensemble des informations obtenue sur le système à partir d'une racine donnée.

Paramètre

NOM	USAGE	TYPE	Description
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

4. GICTRL(ierr) (M)

Description

Il s'agit de la routine de contrôle de niveau le plus élevé au sein de l'interface.

Elle permet à l'utilisateur de choisir l'une des commandes disponibles en "Control Mode".

Paramètre

NOM	USAGE	TYPE	Description
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

5. GIACTM(ierr) (M)

Description

Cette routine affiche le type et le nom du diagramme courant.

Paramètre

NOM	USAGE	TYPE	Description
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

6. GIARCH(ierr) (GI)

Description

Cette routine ferme et rèouvre la base de donnèes graphique afin d'archiver les diffèrentes informations introduites.

Paramètre

NOM	USAGE	TYPE	Description
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

7. GIBGNS(ierr) (M)

Description

Cette routine initialise une session.Elle enregistre le contenu de la table,ouvre les bases de donnèes et appelle les routines d'initialisations graphiques.

Paramètre

NOM	USAGE	TYPE	Description
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

8. GICCMD(NMODE,ierr) (A)

Description

Cette routine contrôle la session graphique en "creation/completion mode".

Elle construit les tables internes nècèssaires à l'affichage des phrases du langage de spècification,affiche le menu adèquat et appelle les routines permettant la construction d'un diagramme (GIBDGM et GIBMAT).

Paramètre

NOM	USAGE	TYPE	Description
NMODE	ENTREE	ENTIER	indique si l'on crée (0) ou si l'on complète (1) un diagramme
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

9. GICHRT(ierr) (A)

Description

Cette routine contrôle la validité des racines d'un diagramme.

Paramètre

NOM	USAGE	TYPE	Description
IERR	SORTIE	ENTIER	si la valeur de cette variable n'est pas nulle,une racine est incorrecte

10. GICKDG(ierr) (A)

Description

Cette routine vérifie si les objets et les relations décrivant un diagramme enregistré dans la base de données graphique ont toujours leur correspondant dans la base de données des spécifications.

Les enregistrement des objets ou des relations pour lesquels ont n'a pas pu établir une telle correspondance seront effacés.

Paramètre

NOM	USAGE	TYPE	Description
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

11. GICPLT(ierr) (A)

Description

Cette routine est appelée lorsque l'utilisateur désire compléter un diagramme existant.

Après avoir obtenu l'identification du diagramme,elle contrôlera la cohérence de son contenu actuel (GICKDG) et passera en "creation/completion mode" (GICCMD).

Paramètre

NOM	USAGE	TYPE	Description
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

12. GICRDG(irdrw) (E)

Description

Cette routine permet de créer un diagramme.Elle demande à l'utilisateur le nom,le type et les racines du diagramme.Elle lui assigne un numéro d'ordre de création et entre en "creation/completion mode" (GICCMD).

Paramètre

NOM	USAGE	TYPE	Description
IRDRW	ENTREE	ENTIER	si la valeur de cette variable n'est pas nulle ,le diagramme n'a pu être crée

13. GICROB(objkey,frcorn,tocorn,ierr) (E)

Description

Cette routine permet de créer et d'enregister la description d'un objet dans un diagramme.

Paramètre

NOM	USAGE	TYPE	Description
OBJKEY	ENTREE	ENTIER	clé de l'objet dans la base utilisateur
FRCORN	ENTREE	ENTIER	clé dans la base graphique du fromcorner de la relation courante
TOCORN	ENTREE	ENTIER	clé dans la base graphique du tocorner de la relation courante
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

14. GICRRL(ierr) (E)

Description

Cette routine permet de créer et d'enregister la représentation d'une relation dans un diagramme.

Paramètre

NOM	USAGE	TYPE	Description
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

15. GICRRT(ierr) (A)

Description

Cette routine enregistre dans la base de données graphique les noms des différentes racines d'un diagramme.

Paramètre

NOM	USAGE	TYPE	Description
IERR	SORTIE	ENTIER	si la valeur de cette variable n'est pas nulle,une racine est incorrecte

16. GIDNNA (A)

Description

Cette routine affiche les noms d'objets à partir d'un certain numéro (cfr commande INAME).

17. GIDGCK(dgtyp,ierr) (GI)

Description

Cette routine vérifie si un numéro (DGTYP) correspond à un type de diagramme défini dans la table.

Paramètre

NOM	USAGE	TYPE	Description
DGTYP	ENTREE	ENTIER	numéro du type de diagramme
IERR	SORTIE	ENTIER	n'est pas égale à zéro si le numéro est incorrecte

18. GIDGRL(ierr) (E)

Description

Cette routine va rechercher dans la base de données graphique toutes les relations d'un diagramme.

Elle enregistre les différents noms dans un tableau.

Paramètre

NOM	USAGE	TYPE	Description
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

19. GIDLDG(ierr) (M)

Description

Cette routine efface de la base données graphique toutes les informations concernant un diagramme (y compris son contenu).

Paramètre

NOM	USAGE	TYPE	Description
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

20. GIDONA (A)

Description

Cette routine affiche le nom d'un objet sélectionné par l'utilisateur.

Parametre

NOM	USAGE	TYPE	Description
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

21. GIDOBT

Description (A)

Cette routine affiche la forme alphabètique du mètatype d'un objet.

Paramètre

NOM	USAGE	TYPE	Description
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

22. GIDRDG (M)

Description

Cette routine retire de la base de donnees graphique toutes les informations concernant un diagramme et appelle les routines graphiques appropriees pour proceder a son trace.

Parametre

NOM	USAGE	TYPE	Description
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

23. GIDROB(nbox,ierr) (GI)

Description

Cette routine appelle les routines graphiques necessaires au trace de la figure d'un objet.

Parametre

NOM	USAGE	TYPE	Description
NBOX	ENTREE	ENTIER	numero de la case dans laquelle doit etre dessine l'objet
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

24. GIDRRL(relnum,erdrsw,ierr) (E)

Description

Cette routine decode le type de representation d'une relation et trace,via les routines graphiques appropriees,ses differents vecteurs.

Paramètre

NOM	USAGE	TYPE	Description
RELNUM	ENTREE	ENTIER	numéro de la relation
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

25. GIDRVC(rntyp,idraw,ierr) (M)

Description

Cette routine trace chaque vecteur d'une relation donnée.

Paramètre

NOM	USAGE	TYPE	Description
RNTYP	ENTREE	ENTIER	numéro de la relation
IDRAW	ENTREE	ENTIER	0 -> effacer , 1 -> tracer
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

26. GIDSPL(ierr) (A)

Description

Cette routine est appelée lorsque l'utilisateur désire entrer en "Display Mode".

Elle demande à l'utilsateur de procéder à l'identification d'un diagramme,elle affiche le menu adéquat,dessine le diagramme et attend une nouvelle commande.

Paramètre

NOM	USAGE	TYPE	Description
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

27. GIDSST(numr,isw,irc) (A)

Description

Cette routine affiche la phrase du langage de spécification associée à une relation.

Elle provient du rapport STFS [].

Paramètre

NOM	USAGE	TYPE	Description
NUMR	ENTREE	ENTIER	numéro de la relation
ISW	ENTREE	ENTIER	indique si la phrase doit être effacée (0) ou affichée (1)
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

28. GIENDS(ierr) (M)

Description

Cette routine effectue les différentes tâches devant être réalisées à la fin d'une session et ferme les bases de données.

Paramètre

NOM	USAGE	TYPE	Description
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

29. GIFDRT(ierr) (A)

Description

Cette routine vérifie dans la base de données des spécifications l'existence des différentes racines d'un diagramme.

Paramètre

NOM	USAGE	TYPE	Description
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

30. GIFOBJ(ierr) (A)

Description

Cette routine indique la position d'un objet identifié par l'utilisateur.

Paramètre

NOM	USAGE	TYPE	Description
IERR	SORTIE	ENTIER	est égale à 1 si un PAN a été effectué

31. GIGTFM(Formid,Formad) (A)

Description

Cette routine rend l'adresse d'une phrase (form adress).

Elle provient du rapport STFS.

Paramètre

NOM	USAGE	TYPE	Description
FORMAD	ENTREE	ENTIER	clé dans la base méta de la phrase
FORMID	ENTREE	ENTIER	DSID de la phrase

32. GIINPD(linsin,inpbuf,ierr) (GI)

Description

Cette routine enregistre les différents types de diagrammes définis dans la table ainsi que leur contenu légal.

Paramètre

NOM	USAGE	TYPE	Description
LINSIN	E/S	ENTIER	nombre de lignes lues
INPBUF	E/S	STRING	buffer contenant la dernière ligne lue
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

33. GIINTG(intgr,ierr) (GI)

Description

Cette routine lit une chaîne de caractères représentant une valeur entière introduite sur un clavier et la converti en entier.

Paramètre

NOM	USAGE	TYPE	Description
INTGR	SORTIE	ENTIER	la valeur entière introduite
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

34. GIMESG(ival,inpstr,iptr,len) (GI)

Description

Cette routine affiche les messages destinés à l'utilisateur.

Paramètre

NOM	USAGE	TYPE	Description
IVAL	ENTREE	ENTIER	une valeur entière à afficher
INPSTR	ENTREE	STRING	buffer contenant un string à afficher
IPTR	ENTREE	ENTIER	pointeur vers le début du string dans le buffer
LEN	ENTREE	ENTIER	longueur du string

35. GIMKVC(keyrln,irndta,imark,ierr) (GI)

Description

Cette routine place une marque à côté du premier segment de droite d'un vecteur d'une relation.

Paramètre

NOM	USAGE	TYPE	Description
KEYRLN	ENTREE	ENTIER	clé de la relation dans la base graphique
IRNDTA	ENTREE	ENTIER	tableau contenant des informations sur la relation
IMARK	ENTREE	ENTIER	indique si la marque doit être effacée (0) ou affichée (1)
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

36. GIMVEC(nparts) (E)

Description

Cette routine permet à l'utilisateur de construire manuellement un vecteur entre deux objets d'une relation.

Paramètre

NOM	USAGE	TYPE	Description
NPARTS	ENTREE	ENTIER	indique le nombre d'objets intervenant dans la relation

37. GIPAN(nbox) (A)

Description

Cette routine permet à l'utilisateur de réaliser un PAN en "creation/completion mode".Elle sauvera toutes les informations nécessaires à la poursuite de la construction d'un diagramme.

Paramètre

NOM	USAGE	TYPE	Description
NBOX	ENTREE	ENTIER	indique le numéro de la case devant devenir le nouveau centre de l'écran

38. GINASG(linsin.,inpbuf,asytyp,ierr) (E)

Description

Cette routine lit dans la table les descriptions des représentations graphiques des objets et des relations pour chaque type de diagramme.

Paramètre

NOM	USAGE	TYPE	Description
LINSIN	E/S	ENTIER	nombre de lignes déjà lues dans la présente section de la table
INPBUF	E/S	STRING	buffer contenant la dernière ligne lue
ASGTYP	ENTREE	ENTIER	désigne la section courante de la table (0 -> type de représentation , 1 -> type de positionnement du nom)
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

39. GIOBNM (A)

Description

Cette routine trouve le mode de positionnement du nom pour un type d'objet donnè.

40. GIOBRP (A)

Description

Cette routine trouve le mode de representation d'un type d'objet donnè.

41. GINMDG(ierr) (M)

Description

Cette routine permet à un utilisateur de modifier le nom d'un diagramme.

Paramètre

NOM	USAGE	TYPE	Description
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

42. GIPLOB(objkey,nmode,frcorn,tocorn,relnum,ierr) (A)

Description

A l'appel de cette routine,un nouvel objet courant à ajouter a èté indiquè à l'utilisateur.Celui-ci peut à prèsent choisir l'une des commandes disponibles en "creation/completion mode".

Parametre

NOM	USAGE	TYPE	Description
OBJKEY	ENTREE	ENTIER	clè de l'objet dans la base utilisateur
NMODE	ENTREE	ENTIER	indique si l'on crèe (0) ou si l'on complète (1) un diagramme
FRCORN	ENTREE	ENTIER	clè dans la base graphique du fromcorner de la relation courante
TOCORN	ENTREE	ENTIER	clè dans la base graphique du tocorner de la relation courante
RELNUM	ENTREE	ENTIER	numèro de la relation
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

43. GIPLOT(ierr) (A)

Description

Cette routine effectue l'impression d'un diagramme sur une table tracante.

Paramètre

NOM	USAGE	TYPE	Description
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

44. GIPRNT(ierr) (A)

Description

Cette routine contrôle l'impression d'un diagramme sur une table tracante.Elle demande à l'utilisateur de choisir un diagramme et d'indiquer le format de page dèsirè.Elle effectue ensuite l'impression en appelant GIPLOT.

Paramètre

NOM	USAGE	TYPE	Description
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

45. GIQDNM(ierr) (GI)

Description

Cette routine retire et affiche les noms,les types et le nombre d'objets de tous les diagrammes définis dans la bases de données graphiques.

Paramètre

NOM	USAGE	TYPE	Description
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

46. GIQSYS(ierr) (GI)

Description

Cette routine initialise et contrôle l'affichage des renseignements donnés par la routine GIQDNM.

Paramètre

NOM	USAGE	TYPE	Description
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

47. GIRCON(iobfnd,ierr) (GI)

Description

Cette routine recherche dans la base de données graphique une relation attachée à un groupe d'objet.

Paramètre

NOM	USAGE	TYPE	Description
IOBFND	ENTREE	ENTIER	tableau contenant les clès des objets dans la base graphique
IERR	SORTIE	ENTIER	est différent de zèro si aucune relation n'est trouvée

48. GIRDNM(namlen,ierr) (GI)

Description

Cette routine lit le nom d'un objet donnè par l'utilisateur.

Paramètre

NOM	USAGE	TYPE	Description
NAMLEN	SORTIE	ENTIER	longueur du nom de l'objet
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

49. GIRDEF(objtyp,typnam,ierr) (GI)

Description

Cette route retire de la table le nom d'un type d'objet.

Paramètre

NOM	USAGE	TYPE	Description
OBJTYP	ENTREE	ENTIER	numèro du type de l'objet
TYPNAM	SORTIE	STRING	nom du type de l'objet
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

50. GIRLNH(relnum) (A)

Description

Cette routine retire de la table le nom d'un type de relation

Paramètre

NOM	USAGE	TYPE	Description
RELNUM	ENTREE	ENTIER	numèro du type de la relation
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

51. GIRLRL(nbox1,nbox2,ipan) (E)

Description

Cette routine retrace automatiquement les vecteurs de relations après qu'un objet ait été bougé.

Paramètre

NOM	USAGE	TYPE	Description
NBOX1	ENTREE	ENTIER	numèro de l'ancienne case de l'objet
NBOX2	ENTREE	ENTIER	numero de la nouvelle case de l'objet
IPAN	ENTREE	ENTIER	cette variable vaut 1 si un PAN a été effectuè

52. GIRMUC(relnum,ierr) (A)

Description

Cette routine retrouve chaque vecteur de relations connecté à un objet donné,le marque et permet à l'utilisateur de le retracer manuellement.

Paramètre

NOM	USAGE	TYPE	Description
RELNUM	ENTREE	ENTIER	numéro du type de la relation
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

53. GIRLUC(nbox,relnum,ipan) (E)

Description

Cette routine retrouve chaque vecteur d'une relation donnée et les retrace.

Paramètre

NOM	USAGE	TYPE	Description
NBOX	ENTREE	ENTIER	le numéro de la case de l'origine ou de l'extrémité du vecteur
RELNUM	ENTREE	ENTIER	numéro du type de la relation
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

54. GIRLRP(relnum,lintyp,narrow,npoint) (A)

Description

Cette routine rend des informations sur la représentation graphique d'un type de relation.

Paramètre

NOM	USAGE	TYPE	Description
RELNUM	ENTREE	ENTIER	numèro du type de la relation
LINTYP	SORTIE	ENTIER	type de ligne pour les vecteurs
NARROW	SORTIE	ENTIER	indique si une flèche doit être dessinée (1) ou non (0) à l'extrèmitè des vecteurs
NPOINT	SORTIE	ENTIER	indique si un point doit être dessinè (1) ou non (0) à l'origine des vecteurs
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

55. GIRTDG(ierr) (M)

Description

Cette routine permet à un utilisateur de retrouver un diagramme dans la base de donnèes graphique.Si aucune erreur n'est intervenue dans l'identification du diagramme,differents indicateurs seront mis à jour en fonction de son type.

Paramètre

NOM	USAGE	TYPE	Description
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

56. GIRTOB(nbox,keyobj,iobtyp,ishape,ierr) (GI)

Description

A partir du numèro de la case dans laquelle se trouve un objet,cette routine rend sa clè (database),son type et son numèro de figure.

Paramètre

NOM	USAGE	TYPE	Description
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

57. GIRTRL(ierr) (A)

Description

Cette routine permet de retracer manuellement les vecteurs d'une relation qui sera identifiée par l'utilisateur.

Paramètre

NOM	USAGE	TYPE	Description
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

58. GIRSTR(menu) (A)

Description

Cette routine permet d'effectuer une restauration de l'écran en "creation/completion mode".Elle sauvera toutes les informations nécessaires à la poursuite de la construction du diagramme.

Paramètre

NOM	USAGE	TYPE	Description
MENU	SORTIE	ENTIER	indique le menu qui doit être redessiné à l'écran (1 -> control, 2 -> creation/completion , 3 -> display)

59. GISDUC(ierr) (A)

Cette routine enregistre dans des tableaux les types des relations et les types des objets autorisés au sein du diagramme courant.

Paramètre

NOM	USAGE	TYPE	Description
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

60. GISTTA(ierr) (A)

Description

Cette routine crée les tableaux internes nécessaires pour l'affichage de phrases.

Paramètre

NOM	USAGE	TYPE	Description
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

61. GITBLD(ierr) (M)

Description

Cette routine lit la table de définition des diagrammes créée par l'administrateur du système.

Paramètre

NOM	USAGE	TYPE	Description
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

62. GIUOBN(objkey,erdsw) (A)

Description

Cette routine souligne le nom d'un objet dans une phrase.

Paramètre

NOM	USAGE	TYPE	Description
OBJKEY	ENTREE	ENTIER	clé de l'objet dans la base utilisateur
ERDSW	ENTREE	ENTIER	= 1 -> effacer , = 2 -> tracer

63. GIOBZM(ierr) (A)

Description

Cette routine permet d'effectuer un "zoom" sur un objet.

Paramètre

NOM	USAGE	TYPE	Description
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

64. IGIRLN(irntyp) (GI)

Description

Cette fonction trouve le numèro (d'ordre de dèfiniton) d'un type de relation donnè.

Paramètre

NOM	USAGE	TYPE	Description
IRNTYP	ENTREE	ENTIER	in +2CODE du type de la relation
IGIRLN	SORTIE	ENTIER	le numèro du type de la relation

65. IGIYNO(dumval) (GI)

Description

Cette fonction permet à l'utilisateur d'entrer un "Y" (yes) ou un "N" (no).

Paramètre

NOM	USAGE	TYPE	Description
IGIYNO	SORTIE	ENTIER	= 1 -> Yes , = 0 -> No

66. GIMUOB(ipan) (E)

Description

Cette routine permet à l'utilisateur de changer la position d'un objet.

Paramètre

NOM	USAGE	TYPE	Description
IPAN	SORTIE	ENTIER	<> 0 si un PAN (1) a été effectué

2. GOD

Cette librairie contient 48 routines d'accès à la base de données graphique.Ces routines sont propres au GOI(sa base de données étant différente de celle du GI);elles font appel aux primitives du système de gestion de base de données ADBMS [ ].

1. LGDADK(keydgm)

Description

Cette fonction vérifie si "keydgm" est une clé de diagramme valide.

Paramètre

NOM	USAGE	TYPE	Description
KEYDGM	ENTREE	ENTIER	clé du diagramme à contrôler

2. GDARCU(ierr)

Description

Cette routine archive sur disque la copie actuelle de la base de données graphique.

Paramètre

NOM	USAGE	TYPE	Description
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

3. GDCHUC(keyvec,ivcary,ierr)

Description

Cette routine change les champs de l'enregistrement d'un vecteur (vecrec).

Paramètre

NOM	USAGE	TYPE	Description
KEYVEC	ENTREE	ENTIER	clé du vecteur dans la base de données
IVCARY	ENTREE	ENTIER	tableau contenant les nouvelles informations sur le vecteur
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

4. GDCRGO(nposil,nangle,keygrd,keyobj,keygro,ierr)

Description

Cette routine crée un enregistrement GRDOBJ en le reliant à un objet (keyobj) et à une case (keygrd)

Paramètre

NOM	USAGE	TYPE	Description
IDGDAT	ENTREE	ENTIER	tableau contenant des informations sur le diagramme
DNMSTR	ENTREE	STRING	nom du diagramme
KEYDGM	SORTIE	ENTIER	clé du diagramme enregistré dans la base de données
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

5. GDCRGR(numgrd,keydgm,keygrd,ierr)

Description

Cette routine crée l'enregistrement d'une case (GRDREC) en le reliant à un diagramme (keydgm).

Paramètre

NOM	USAGE	TYPE	Description
NUMGRD	ENTREE	ENTIER	numéro de la case
KEYDGM	ENTREE	ENTIER	clé du diagramme courant dans la base de données
KEYGRD	SORTIE	ENTIER	clé de la case enregistrée dans la base de données
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

6. GDCROB(keydgm,objtyp,namobj,shpsiz,keyobj,ierr).

Cette routine crée l'enregistrement d'un objet (objrec) en le reliant à un diagramme (keydgm)

Paramètre

NOM	USAGE	TYPE	Description
KEYDGM	ENTREE	ENTIER	clé du diagramme courant
OBJTYP	ENTREE	ENTIER	type de l'objet
NAMOBJ	ENTREE	STRING	nom de l'objet
SHPSIZ	ENTREE	ENTIER	taille de la figure de l'objet
KEYOBJ	SORTIE	ENTIER	clé de l'objet enregistré dans la base de données
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

7. GDCRRL (keydgm,keyrln,ierr)

Description

Cette routine crée l'enregistrement d'une relation en le reliant à un diagramme (keydgm)

Paramètre

NOM	USAGE	TYPE	Description
KEYDGM	ENTREE	ENTIER	clé du diagramme courant
KEYRLN	SORTIE	ENTIER	clé de la relation enregistrée dans la base de données
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

8. GDCRRT(namrot,keydgm,keyrot,ierr)

Description

Cette routine crée l'enregistrement d'une racine en le reliant à un diagramme (keydgm).

Paramètre

NOM	USAGE	TYPE	Description
NAMROT	ENTREE	STRING	nom de la racine
KEYDGM	ENTREE	ENTIER	clé du diagramme courant
KEYROT	SORTIE	ENTIER	clé de la racine enregistrée dans la base de donnés
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

9. GDCRVC(keyrln,ivcary,keyvec,ierr)

Description

Cette routine crée l'enregistrement d'un vecteur en le reliant à une relation (keyrln).Les valeurs à assigner aux différents champs sont reprises dans le tableau IVCARY

Paramètre

NOM	USAGE	TYPE	Description
KEYRLN	ENTREE	ENTIER	Clé de la relation dans la base de données
IVCARY	ENTREE	ENTIER	tableau contenant les nouvelles valeurs des champs de l'enregistrement
KEYVEC	SORTIE	ENTIER	clé du vecteur enregistré dans la base graphique
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

10. GDCADG(nodgms)

Description

Cette routine donne le nombre de diagrammes existant dans la base de donnees graphique

Paramètre

NOM	USAGE	TYPE	Description
NODGMS	SORTIE	ENTIER	nombre de diagramme enregistrés

11. GDCAGR(keygrd,numrel,ierr)

Description

Cette routine donne le nombre d'enregistrements in RELREC reliès à un enregistrement GRDREC donnée (keygrd).

Paramètre

NOM	USAGE	TYPE	Description
KEYGRD	ENTREE	ENTIER	clè de l'enregistrement GRDREC
NUMREL	SORTIE	ENTIER	nombre de RELREC relié au GRDREC donné
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

12. GDCAGO(keygrd,numgro,ierr)

Description

Cette routine donne le nombre d'enregistrements GRDOBJ reliès à un enregistrement GRDREC donné (keygrd).

Paramètre

NOM	USAGE	TYPE	Description
KEYGRD	ENTREE	ENTIER	clé de l'enregistrement GRDREC
NUMGRO	SORTIE	ENTIER	nombre de GRDOBJ relié au GRDREC donné
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

13. GDDLDG(keydgm,ierr)

Description

Cette routine efface l'enregistrement d'un diagramme donné (keydgm).

Paramètre

NOM	USAGE	TYPE	Description
KEYDGM	ENTREE	ENTIER	clé du diagramme dans la base de données
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

14. GDDLGO(keygro,ierr)

Description

Cette routine efface l'enregistrement GRDOBJ identifié par keygro.

Paramètre

NOM	USAGE	TYPE	Description
KEYGRO	ENTREE	ENTIER	l'enregistrement GRDOBJ dans la base de données
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

15. GDDLGR(keygrd,ierr)

Description

Cette routine efface l'enregistrement d'une case donnée (keygrd).

Paramètre

NOM	USAGE	TYPE	Description
KEYGRD	ENTREE	ENTIER	clé de la case dans la base de données
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

16. GDDLRT(keyrot,ierr)

Description

Cette routine efface l'enregistrement ROTREC (racine) donné (keyrot).

Paramètre

NOM	USAGE	TYPE	Description
KEYROT	ENTREE	ENTIER	clé de la racine dans la base données
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

17. GDDLRN(keyrln,ierr)

Description

Cette routine efface les enregistrements d'une relation (keyrln) et de tous les vecteurs qui lui sont associés.

Paramètre

NOM	USAGE	TYPE	Description
KEYRLN	ENTREE	ENTIER	clè de la relation dans la base de donnèes
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

18. GDDOKY(keyobj,ierr)

Description

Cette routine efface l'enregistrement d'un objet (keyobj).

Paramètre

NOM	USAGE	TYPE	Description
KEYOBJ	ENTREE	ENTIER	clè de l'objet dans la base de donnèes
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

19. GDFDNM(dnmstr,keydgm,ierr)

Description

Cette routine rend,à partir du nom d'un diagramme,la clè du diagramme.

Paramètre

NOM	USAGE	TYPE	Description
DNMSTR	ENTREE	STRING	nom du diagramme
KEYDGM	SORTIE	ENTIER	clè du diagramme dans la base de donnèes
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

20. GDFDNU(numdgm,keydgm,ierr)

Description

Cette routine rend la clè d'un diagramme à partir du numèro du diagramme.

Paramètre

NOM	USAGE	TYPE	Description
NUMDGM	ENTREE	ENTIER	numèro du diagramme
KEYDGM	SORTIE	ENTIER	clè du diagramme dans la base de donnèes
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

21. GDFGRN(numgrd,keydgm,keygrd,ierr)

Description

Cette routine rend la clè d'une case associèe à un diagramme (keydgm) à partir du numèro de la case.

Paramètre

NOM	USAGE	TYPE	Description
NUMGRD	ENTREE	ENTIER	numèro de la case
KEYDGM	ENTREE	ENTIER	clè du diagramme courant
KEYGRD	SORTIE	ENTIER	clè de la case dans la base de donnèes
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

22. GDFOBN(namobj,keydgm,keyobj,ierr)

Description

Cette routine rend la clé d'un objet associé à un diagramme à partir du nom de l'objet.

Paramètre

NOM	USAGE	TYPE	Description
NAMOBJ	ENTREE	STRING	nom de l'objet
KEYDGM	ENTREE	ENTIER	clé du diagramme courant
KEYOBJ	SORTIE	ENTIER	clé de l'objet dans la base de données
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

23. GDFNDG(keydgm,retrve,dnmstr,idgdat,nobjs,ierr)

Description

Cette routine rend les informations décrivant le diagramme qui est le membre suivant du chemin ALLDGM (next member of ALLDGM set).

Paramètre

NOM	USAGE	TYPE	Description
KEYDGM	E/S	ENTIER	clé du diagramme.
RETRVE	ENTREE	ENTIER	indique quelles informations retirer (1 -> uniquement la clé,2 -> tout)
DNMSTR	SORTIE	STRING	nom du diagramme
IDGDAT	SORTIE	ENTIER	tableau contenant des informations sur le diagramme
NOBJS	SORTIE	ENTIER	nombre d'objets dans le diagramme
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

24. GDFNG1(keygrd,keygro,ierr)

Description

Cette routine rend la clé de l'enregistrement GRDOBJ qui le membre suivant du chemin GRDOB1 dont l'origine est identifiée par "keygrd".

Paramètre

NOM	USAGE	TYPE	Description
KEYGRD	ENTREE	ENTIER	clé d'une case dans la base de données
KEYGRO	E/S	ENTIER	clé de l'enregistrement GRDOBJ dans la base de données
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

25. GDFNG2(keyobj,keygro,ierr)

Description

Cette routine rend la clé de l'enregistrement GRDOBJ qui est le membre suivant du chemin GRDOB2 dont l'origine est identifiée par "keyobj".

Paramètre

NOM	USAGE	TYPE	Description
KEYOBJ	ENTREE	ENTIER	clé d'un objet dans la base de données
KEYGRO	E/S	ENTIER	clé de l'enregistrement GRDOBJ dans la base de données
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

26. GDFNOB(keydgm,keyobj,ierr)

Description

Cette routine rend la clé de l'enregistrement de l'objet qui est le membre suivant du chemin DGMOBJ dont l'origine est identifiée par "keydgm".

Paramètre

NOM	USAGE	TYPE	Description
KEYDGM	ENTREE	ENTIER	clé d'un diagramme dans la base de données
KEYOBJ	E/S	ENTIER	clé de l'objet dans la base de données
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

27. GDFNRL(keydgm,keyrln,ierr)

Description

Cette routine rend la clé de l'enregistrement de la relation qui est le membre suivant du chemin DGMREL dont l'origine est identifiée par "keydgm".

Paramètre

NOM	USAGE	TYPE	Description
KEYDGM	ENTREE	ENTIER	clé d'un diagamme dans la base de données
KEYRLN	E/S	ENTIER	clé de la relation dans la base de données
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

28. GDFNRT(keydgm,keyrot,ierr)

Description

Cette routine rend la clé de l'enregistrement de la racine qui est le membre suivant du chemin DGMROT dont l'origine est identifiée par "keydgm".

Paramètre

NOM	USAGE	TYPE	Description
KEYDGM	ENTREE	ENTIER	clé d'un diagramme dans la la base de données
KEYROT	E/S	ENTIER	clé de la racine dans la base données
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

29. GDFOW1(keygro,keygrd,ierr)

Description

Cette routine rend la clé de la case qui est l'origine du chemin GRDOB1 identifié par le membre "keygro".

Paramètre

NOM	USAGE	TYPE	Description
KEYGRO	ENTREE	ENTIER	clé d'un enregistrement GRDOBJ
KEYGRD	SORTIE	ENTIER	clé de la case dans la base de données
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

30. GDFOW2(keygro,keyobj,ierr)

Description

Cette routine rend la clé de l'objet qui est l'origine du chemin GRDOB2 identifié par "keygro".

Paramètre

NOM	USAGE	TYPE	Description
KEYGRO	ENTREE	ENTIER	clé d'un enregistrement GRDOBJ
KEYOBJ	SORTIE	ENTIER	clé de l'objet dans la base de données
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

31. GDFOW3(keyrln,keygrd,ierr)

Description

Cette routine rend la clé de la case qui est l'origine du chemin GRDREL identifié par "keyrln".

Paramètre

NOM	USAGE	TYPE	Description
KEYRLN	ENTREE	ENTIER	clé d'une relation dans la base de données
KEYGRD	SORITE	ENTIER	clé de la case dans la base de données
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

32. GDFORL(keydgm,keyobj,npart,keyrln,irndta,ierr)

Description

Cette routine permet de retrouver toutes les relations dont fait partie l'objet identifié par keyobj.

Paramètre

NOM	USAGE	TYPE	Description
KEYDGM	ENTREE	ENTIER	clè du diagramme courant
KEYOBJ	ENTREE	ENTIER	clè de l'objet dans la base de donnèes
NPART	SORTIE	ENTIER	corner de la relation occupè par l'objet
KEYRLN	SORTIE	ENTIER	clè de la relation
IRNDTA	SORTIE	ENTIER	tableau contenant des informations sur la relation
IERR	SORTIE	ENTIER	si <> 0 ,aucune relation n'a ètè trouvèe

33. GDGTGO(keygro,npos1,nangle,ierr)

Description

Cette routine rend les valeurs des champs de l'enregistrement GRDOBJ identifiè par keygro.

Paramètre

NOM	USAGE	TYPE	Description
KEYGRO	ENTREE	ENTIER	clè d'un enregistrement GRDOBJ
NPOSI1	SORTIE	ENTIER	valeur du premier champ
NANGLE	SORTIE	ENTIER	valeur du second champ
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

34. GDGTGR(keygrd,numgrd,grdinc,ierr)

Description

Cette routine rend les valeurs des champs de l'enregistremnt GRDREC (case) identifiè par "keygrd".

Paramètre

NOM	USAGE	TYPE	Description
KEYGRD	ENTREE	ENTIER	clè d'une case dans la base de donnnès
NUMGRD	SORTIE	ENTIER	numèro de la case
GRDINC	SORTIE	ENTIER	nombre de lignes dèjà insèrèes dans la case
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

35. GDGTOB(keyobj,objtyp,namobj,shpsiz,ierr)

Description

Cette routine rend les valeurs des champs de l'enregistrement OBJREC (objet) identifiè par "keyobj".

Paramètre

NOM	USAGE	TYPE	Description
KEYOBJ	ENTREE	ENTIER	clè d'un objet dans la base de donnèes
OBJTYP	SORTIE	ENTIER	type de l'objet
NAMOBJ	SORTIE	STRING	nom de l'objet
SHPSIZ	SORTIE	ENTIER	taille de la figure de l'objet
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

36. GDGTRT(keyrot,namrot,ierr)

Description

Cette routine rend la valeur des champs de l'enregistrement ROTREC (racine) identifiè par "keyrot".

Paramètre

NOM	USAGE	TYPE	Description
KEYROT	ENTREE	ENTIER	clé d'une racine dans la dans la base de données
NAMROT	SORTIE	STRING	nom de la racine
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

37. GDGTVC(keyrln,keyvec,ivcary,ierr)

Description

Cette routine permet d'obtenir les valeurs des champs de chaque vecteur d'une relation identifiée par "keyrln".

Paramètre

NOM	USAGE	TYPE	Description
KEYRLN	ENTREE	ENTIER	clé d'une relation dans la base de données
KEYVEC	E/S	ENTIER	clé d'un vecteur dans la base de données
IVCARY	SORTIE	ENTIER	tableau reprenant les informations sur le vecteur
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

38. GDINIT(ierr)

Description

Cette routine initialise ADBMS et ouvre la base de donnés graphique.

Paramètre

NOM	USAGE	TYPE	Description
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

39. GDMAXN(maxdgm,ierr)

Description

Cette routine rend le numéro le plus élevé assigné à un diagamme.

Paramètre

NOM	USAGE	TYPE	Description
MAXDGM	SORTIE	ENTIER	numéro le plus élevé assigné à un diagramme

40. GDURRG(keyrln,keygrd,irndta,ierr)

Description

Cette routine met à jour les champs d'une relation (keyrln) et relie cette dernière à une case (keygrd).

Paramètre

NOM	USAGE	TYPE	Description
KEYRLN	ENTREE	ENTIER	clé de la relation dans la base de données
KEYGRD	ENTREE	ENTIER	clé d'une case dans la base de données
IRNDTA	ENTREE	ENTIER	tableau contenant les nouvelles valeurs des champs de l'enregistrement
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

41. GDRTDG(keydgm,dnmstr,idgdat,ierr)

Description

Cette routine rend les valeurs des champs de l'enregistrement d'un diagramme identifié par "keydgm".

Paramètre

NOM	USAGE	TYPE	Description
KEYDGM	ENTREE	ENTIER	clé d'un diagramme dans la base de données
DNMSTR	SORTIE	STRING	nom du diagramme
IDGDAT	SORTIE	ENTIER	tableau contenant les données sur le diagramme
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

42. GDSETD(keydgm,ierr)

Description

Cette routine rend courant le diagramme de clé keydgm.

Paramètre

NOM	USAGE	TYPE	Description
KEYDGM	ENTREE	ENTIER	clé du diagramme à rendre courant
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

43. GDTERM(ierr)

Description

Cette routine ferme la base de données graphique.

Paramètre

NOM	USAGE	TYPE	Description
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

44. GDXCDB(icurdb,ierr)

Description

Cette routine change la base de données courante qui peut être soit la base de donnees graphique soit la base de donnees des spécifications.

Paramètre

NOM	USAGE	TYPE	Description
ICURDB	ENTREE	ENTIER	désigne la base de donnees à rendre courante (1 -> BD graphique, 2 -> BD utilisateur)
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

45. GDXGRI(keygrd,grdinc,ierr)

Description

Cette routine change la valeur du champ GRDINC de l'enregistrement d'une case identifié par "keygrd".

Paramètre

NOM	USAGE	TYPE	Description
KEYGRD	ENTREE	ENTIER	clé d'une case dans la base de donnees
GRDINC	ENTREE	ENTIER	nouvelle valeur pour le champ GRDINC
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

46. GDXOBS(keyobj,shpsiz,ierr)

Description

Cette routine change la valeur du champ SHPSIZ de l'enregistrement de l'objet identifié par "keyobj".

Paramètre

NOM	USAGE	TYPE	Description
KEYOBJ	ENTREE	ENTIER	clé d'un objet dans la base de données
SHPSIZ	ENTREE	ENTIER	nouvelle valeur du champ SHPSIZ
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

47. GDXDGF(keydgm,nfield,dnmstr,ivalue,ierr)

Description

Cette routine change la valeur d'un champ spécifique de l'enregistrement d'un diagramme identifié par "keydgm".

Paramètre

NOM	USAGE	TYPE	Description
KEYDGM	ENTREE	ENTIER	clé d'un diagramme dans la base de données
NFIELD	ENTREE	ENTIER	désigne le champ à modifier
DNMSTR	ENTREE	STRING	nom du diagramme
IVALUE	ENTREE	ENTIER	nouvelle valeur pour le champ (si celui-ci est de type entier)
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

48. IDGGOB(KEYDGM)

Description

Cette fonction rend le nombre d'objets associés à un diagramme donné (keydgm).

Paramètre

NOM	USAGE	TYPE	Description
KEYDGM	ENTREE	ENTIER	clé d'un diagramme dans la de données
IGDGOB	SORTIE	ENTIER	nombre d'objet dans le diagramme

3. GRDI

La librairie GRDI contient de nombreuses routines de la librairie GR du Graphical Interface.Ces routines seront très souvent modifiées (M) ou étendue (E).La plupart des modifications corresponderont à l'introduction d'appels aux routines de la librairies BEN (primitives de la table tracante).

Les ajoûts (A) de routines seront essentiellement liés aux extensions apportées aux possibilités de représentations graphiques.

1. GRAGRS(nbox1,nbox2,nadsw) (A)

Description

Cette routine vérifie si deux cases sont adjacentes.

Paramètre

NOM	USAGE	TYPE	Description
NBOX1	ENTREE	ENTIER	numéro de la première case
NBOX2	ENTREE	ENTIER	numéro de la deuxième case
NADSW	SORTIE	ENTIER	= 1 si les deux cases sont adjacentes ,= 0 sinon
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

2. GRARSW(nsw) (A)

Description

Cette routine met à jour le booléen indiquant si un vecteur doit être tracé avec une flèche.

Paramètre

NOM	USAGE	TYPE	Description
NSW	ENTREE	ENTIER	= 1 -> oui , = 0 -> non

3. GRMSG(msgnum) (GI)

Description

Cette routine affiche différents messages (d'erreurs et autres).

Paramètre

NOM	USAGE	TYPE	Description
MSGNUM	ENTREE	ENTIER	numéro du message

4. GRBEN(pagesw,outsw) (A)

Description

Cette routine initialise une image sur la table tracante.

Paramètre

NOM	USAGE	TYPE	Description
PAGESZ	ENTREE	ENTIER	taille de la page à imprimer
OUTSW	ENTREE	BOOLEEN	indique s'il faut initialer la table tracante ou mettre fin à son usage

5. GRBPIK(IX,IY,nbox,pctnum,key) (M)

Description

Cette routine permet à l'utilisateur de sélectionner soit une case dans la zone de travail soit une commande dans un menu.

Paramètre

NOM	USAGE	TYPE	Description
IX,IY	SORTIE	ENTIER	coordonnées lues
IX,IY	ENTREE	ENTIER	si PCTNUM = 4,il faut trouver la case contenant le point (IX,IY)
NBOX	SORTIE	ENTIER	numéro de la case contenant le point (IX,IY)
PCTNUM	ENTREE	ENTIER	Où sélectionne-t-on? 1 = zone de travail 2 = menu 3 = trouver la case contenant le point (IX,IY) 4 = position à l'intérieur d'une case donnée

6. GRCARD(idx,idy) (E)

Description

Cette routine trace le dessin d'une carte d'entrée-sortie.

Paramètre

NOM	USAGE	TYPE	Description
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

7. GRCBXS(nbox,keyobj,ishape,isize) (GI)

Description

Cette routine rend la clè de l'objet enregistrè dans la base de donnèes graphique occupant la case de numèro "nbox".

Paramètre

NOM	USAGE	TYPE	Description
NBOX	ENTREE	ENTIER	numèro de la case
KEYOBJ	SORTIE	ENTIER	clè de l'objet dans la base graphique
ISHAPE	SORTIE	ENTIER	numèro de la figure de l'objet
ISIZE	SORTIE	ENTIER	taille de la figure de l'objet

8. GRCOBJ(nbox,ierr) (A)

Description

Cette routine place au milieu de l'écran la figure de l'objet se trouvant dans la case "nbox" si celle-çi n'apparaît pas dans la partie actuellement visible du diagramme.

Paramètre

NOM	USAGE	TYPE	Description
NBOX	ENTREE	ENTIER	numèro de la case
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

9. GRCTLM (M)

Description

Cette routine affiche le menu du mode de commandes "Control Mode".

10. GRDCBX(idx, idy) (E)

Description

Cette routine trace le dessin d'un message.

Paramètre

NOM	USAGE	TYPE	Description
IDX	ENTREE	ENTIER	longueur de la figure
IDY	ENTREE	ENTIER	largeur de la figure

11. GRDCCM (A)

Description

Cette routine affiche le menu du mode de commandes "Creation/Completion mode".

12. GRDDSM (A)

Description

Cette routine affiche le menu du mode de commandes "Display Mode".

Paramètre

NOM	USAGE	TYPE	Description
IDX	ENTREE	ENTIER	longueur de la figure
IDY	ENTREE	ENTIER	largeur de la figure

13. GRDMND(IDX,IDY) (E)

Description

Cette routine trace un losange.

14. GRDNNA(nstart)

Description

Cette routine affiche dans la zone "informations diverses" les noms des objets apparaissant à l'écran et de numéro supérieur à "nstart".

Signalons que les objets apparaissant à l'écran sont numérotés par ordre alphabétique.

Paramètre

NOM	USAGE	TYPE	Description
NSTART	ENTREE	ENTIER	borne inférieur pour les numéros d'objets affichés

15. GRDRUM(idx,idy) (E)

Description

Cette routine trace le symbole d'un disque (tambour).

Paramètre

NOM	USAGE	TYPE	Description
IDX	ENTREE	ENTIER	longueur de la figure
IDY	ENTREE	ENTIER	largeur de la figure

16. GRFDBK(nbox,mode) (GI)

Description

Cette routine place une marque dans la case d'un menu selectionnée par l'utilisateur.Si "mode" est égal à zéro,la marque est effacée.

Paramètre

NOM	USAGE	TYPE	Description
NBOX	ENTREE	ENTIER	numéro de la case
NMODE	ENTREE	ENTIER	= 0 -> effacer , = 1 -> tracer

17. GRGRDF (GI)

Description

Cette routine enregistre dans un tableau les coordonnées X-Y des différents points définissant la grille associée à un écran.

18. GRGRLN(nbox,grdinc,reldep) (A)

Description

Cette routine est utilisée pour placer le nom d'un objet sur une ligne déterminée d'une case.

Sur base d'un numéro de ligne (grdinc),elle calcule la position relative du point à partir duquel une chaine de caractère doit être dessinée au sein d'une case.

Paramètre

NOM	USAGE	TYPE	Description
NBOX	ENTREE	ENTIER	numéro de la case
GRDINC	E/S	ENTIER	nombre de lignes deja insérées dans la case
RELDEP	SORTIE	ENTIER	déplacement relatif à effectuer dans la case

19. GRGRLL(nbox, ixmin, iymin, ierr) (GI)

Description

Cette routine rend la coordonnée du coin inférieur gauche de la case de numéro "nbox".

Paramètre

NOM	USAGE	TYPE	Description
NBOX	ENTREE	ENTIER	numéro de la case
IXMIN IYMIN	ENTREE	ENTIER	coordonnées du coin inférieur gauche de la case.
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

20. GRGRSW (GI)

Description

Si la découpe en cases d'un diagramme est affichée, elle est effacée ; et inversement.

21. GRGRSZ (GI)

Description

Cette routine définit la taille d'une case. Elle est appelée après chaque redéfinition de l'écran (comme un zoom ou un pan).

22. GRINIT (M)

Description

Cette routine initialise l'écran ainsi que les principales variables graphiques.

23. GRKIBX(idx,idy) (E)

Description

Cette routine trace un carrè tronquè.

Paramètre

NOM	USAGE	TYPE	Description
IDX	ENTREE	ENTIER	longueur de la figure
IDY	ENTREE	ENTIER	largeur de la figure

24. GRMCLR(itype) (GI)

Description

Cette routine redessine un écran.

Paramètre

NOM	USAGE	TYPE	Description
ITYPE	ENTREE	ENTIER	indique le menu à rèafficher (1 -> control , 2 -> creation/completion)

25. GRMKVC(nbox1,locn1,nbox2,locn2,ibeam) (GI)

Description

Cette routine place une marque sur un segment de droite dèfini par les paramètres recus.

Paramètre

NOM	USAGE	TYPE	Description
NBOX1	ENTREE	ENTIER	case de l'origine du segment de droite
LOCN1	ENTREE	ENTIER	position au sein de cette case
NBOX2	ENTREE	ENTIER	case de l'extrémité du segment de droite
LOCN2	ENTREE	ENTIER	position au sein de cette case
IBEAM	ENTREE	ENTIER	0 -> effacer , 1 -> tracer

26. GRMTXT(len,txtstr,lineno,ishape,istore,nmode) (M)

Description

Cette routine est appelée pour afficher un texte sur une ligne de la zone "informations diverses".

Paramètre

NOM	USAGE	TYPE	Description
LEN	ENTREE	ENTIER	longueur du texte
TXTSTR	ENTREE	STRING	le texte
LINENO	ENTREE	ENTIER	ligne dans la zone "informations diverses" où doit apparaître le texte
ISHAPE	ENTREE	ENTIER	spécifie la figure qui doit être tracée à côté du texte (> 0 -> objet , = 0 -> rien , < 0 -> relation)
ISTORE	ENTREE	ENTIER	valeur à insérer dans le buffer
NMODE	ENTREE	ENTIER	= 0 -> effacer , = 1 -> afficher

27. GROBZM(nbox,isize) (A)

Description

Cette routine effectue un zoom sur l'objet se trouvant dans la case de numéro "nbox".

Paramètre

NOM	USAGE	TYPE	Description
NBOX	ENTREE	ENTIER	numéro de la case contenant l'objet
ISIZE	ENTREE	ENTIER	taille de la figure de l'objet

28. GROPUT(nbox,ishape,ierr,isize) (E)

Description

Cette routine contrôle et effectue le tracé de la figure d'un objet.

Paramètre

NOM	USAGE	TYPE	Description
NBOX	ENTREE	ENTIER	numéro de la case où va apparaître l'objet
ISHAPE	ENTREE	ENTIER	numéro de la figure de l'objet
IERR	SORTIE	ENTIER	indique si une erreur est intervenue
ISIZE	ENTREE	ENTIER	taille de la figure de l'objet

29. GROREM(keyobj,nbox) (GI)

Description

Cette routine efface un objet affiché à l'écran et met à jour le statut de la (ou des) case (s) concernée (s).

Paramètre

NOM	USAGE	TYPE	Description
KEYOBJ	ENTREE	ENTIER	clè de l'objet dans la base graphique
NBOX	ENTREE	ENTIER	numèro de la case où se trouve l'objet

30. GRPACK(val1,val2,valpak,pfctr) (M)

Description

Cette routine concatène les nombres "val1" et "val2" dans la variable "valpak".

Paramètre

NOM	USAGE	TYPE	Description
VAL1	ENTREE	ENTIER	nombre à concatèner à gauche
VAL2	ENTREE	ENTIER	nombre à concatèner à droite
VALPAK	ENTREE	ENTIER	rèsultat de la concatènation
PFCTR	ENTREE	ENTIER	nombre de chiffres de VAL2

31. GRPAN(nbox) (E)

Description

Cette routine permet à un utilisateur de changer la position du centre "logique" d'un diagramme affichè à l'écran.

Paramètre

NOM	USAGE	TYPE	Description
NBOX	ENTREE	ENTIER	numèro de la case qui doit devenir le nouveau centre de la zone de travail

32. GRPONT(ix,iy,idrsw) (A)

Description

Cette routine marque à l'écran le point de coordonnnées (ix,iy).

Paramètre

NOM	USAGE	TYPE	Description
IX,IY	ENTREE	ENTIER	coordonnées du point à marquer
IDRSW	ENTREE	ENTIER	= 0 -> effacer , 1 -> marquer

33. GRPTSW(NSW) (A)

Description

Cette routine modifie la valeur de la variable booléenne indiquant si un point doit être marqué au début d'un vecteur.

Paramètre

NOM	USAGE	TYPE	Description
NSW	ENTREE	ENTIER	= 1 -> un point est marqué = 0 -> pas de point marqué

34. GRPTXT(txtstr,lenstr,nbox,nangle,typind,linesw,numsw,erdrsw) (A)

Description

Cette routine trace un texte à partir d'une position donnée et selon des attributs donnés.

Paramètre

NOM	USAGE	TYPE	Description
TXTSTR	ENTREE	ENTIER	texte à afficher
LENSTR	ENTREE	ENTIER	longueur du texte
NBOX	ENTREE	ENTIER	case où sera placé le texte
LOCN	ENTREE	ENTIER	position dans la case
NANGLE	ENTREE	ENTIER	angle définissant le plan de base pour le tracé du texte
TYPIND	ENTREE	ENTIER	type de cadrage du texte
LINESW	ENTREE	ENTIER	= 0 -> le texte est imprimé sur une seule ligne = 1 -> on utilise les "-" pour couper le string et on l'imprime avec des petits caractères = 2 -> on utilise les "-" pour couper le string et on l'imprime avec des caractères adaptés à sa longueur
NUMSW	ENTREE	ENTIER	<> 0 -> le string sera précédé d'un numéro
ERDSW	ENTREE	ENTIER	= 0 -> effacé , <> 0 afficher

35. GRPARL(idx,idy) (E)

Description

Cette routine dessine un parallèlogramme.

Paramètre

NOM	USAGE	TYPE	Description
IDX	ENTREE	ENTIER	longueur de la figure
IDY	ENTREE	ENTIER	largeur de la figure

36. GRPTDP(ix,iy,rdx,rdy,nbox,locn) (A)

Description

Cette routine calcule la coordonnée d'un point connu relativement à un autre point.

Paramètre

NOM	USAGE	TYPE	Description
IX,IY	E/S	ENTIER	coordonnées du point
RDX	ENTREE	ENTIER	déplacement relatif sur l'axe X
RDY	ENTREE	ENTIER	déplacement relatif sur l'axe Y
NBOX	E/S	ENTIER	coordonnées de la case contenant le point
LOCN	E/S	ENTIER	position dans la case

37. GRRBTX(nbox,txtstr,lenstr,ibeam) (M)

Description

Cette routine centre un texte donné à l'intérieur d'une case.

Paramètre

NOM	USAGE	TYPE	Description
NBOX	ENTREE	ENTIER	numéro de la case
TXTSTR	ENTREE	STRING	le texte à centrer
LENSTR	ENTREE	ENTIER	longueur du texte
IBEAM	ENTREE	ENTIER	= 0 -> effacer , = 1 -> tracer

38. GRRCFV(nbox,locn,ix,iy) (M)

Description

Cette routine calcule la coordonnée d'un point à partir de sa position dans une case.

Paramètre

NOM	USAGE	TYPE	Description
NBOX	ENTREE	ENTIER	numéro de la case
LOCN	ENTREE	ENTIER	position dans la case
IX,IY	SORTIE	ENTIER	coordonnées du point

39. GRRDRW(mode) (M)

Description

Cette routine redessine un écran après l'avoir effacé.

Paramètre

NOM	USAGE	TYPE	Description
MODE	ENTREE	ENTIER	indique le menu à réafficher (1 -> control , 2 -> creation/completion , 3 -> display)

40. GRRSTR(mtype) (M)

Description

Cette routine restaure l'état précédent d'un écran (après un zoom ou un pan).

Paramètre

NOM	USAGE	TYPE	Description
MTYPE	ENTREE	ENTIER	indique le menu à réafficher (CFR GRRDRW)

GRSBXS(nbox,keyobj,ishape,ierr,isize) (M)

Description

Cette routine modifie le statut d'une case après l'affichage ou l'effacement d'un objet.

Paramètre

NOM	USAGE	TYPE	Description
NBOX	ENTREE	ENTIER	numéro de la case
KEYOBJ	ENTREE	ENTIER	clé de l'objet dans la base graphique
ISHAPE	ENTREE	ENTIER	numéro de la figure de l'objet
IERR	SORTIE	ENTIER	indique si une erreur est intervenue
ISIZE	ENTREE	ENTIER	taille de la figure de l'objet

GRSCDF (GI)

Description

Cette routine divise l'écran en tracant les lignes séparant les différentes zones définies (zone de travail,zone des menus,zone "informations diverses").

Paramètre

NOM	USAGE	TYPE	Description
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

GRSGRS(isize) (M)

Description

Cette routine calcule la taille d'un diagramme en nombre de cases.

Paramètre

NOM	USAGE	TYPE	Description
ISIZE	ENTREE	ENTIER	= 0 -> on désire connaître la taille du diagramme courant <> 0 -> on désire assigner ISIZE à la taille du diagramme courant
ISIZE	SORTIE	ENTIER	taille du diagramme courant sur l'axe X

41. GRTERM (M)

Description

Cette routine efface l'écran et remet le terminal en mode alphanumérique.

42. GRUOBN(lineno,nposit,lenprt,erdrsw) (A)

Description

Cette routine trace une ligne en dessous du nom d'un objet dans une phrase du langage de spécification.

Paramètre

NOM	USAGE	TYPE	Description
LINENO	ENTREE	ENTIER	numéro de la ligne dans la zone "informations diverses" où apparaît le nom de l'objet
NPOSIT	ENTREE	ENTIER	position du nom au sein de la ligne
LENPRT	ENTREE	ENTIER	longueur du nom
ERDRSW	ENTREE	ENTIER	= 0 -> effacer , = 1 -> tracer

43. GRUNPK(valpak,val1,val2,pfctr) (M)

Description

Cette routine "déconcatène" le nombre contenu dans la variable "valpak".

Paramètre

NOM	USAGE	TYPE	Description
VALPAK	ENTREE	ENTIER	nombre à déconcatèner
VAL1	SORTIE	ENTIER	nombre correspondant à la partie gauche de VALPAK
VAL2	SORTIE	ENTIER	nombre correspondant à la partie droite de VALPAK
PFCTR	ENTREE	ENTIER	nombre de chiffre de VAL2

44. GRVCTR(nbox1,locn1,shaps1,nbox2,locn2,shaps2,cmode,lntype) (M)

Description

Cette routine trace un vecteur entre l'objet contenu dans la case de numéro "nbox1" et l'objet contenu dans la case de numéro "nbox2".

Paramètre

NOM	USAGE	TYPE	Description
NBOX1	ENTREE	ENTIER	case du premier objet
LOCN1	SORTIE	ENTIER	position dans la case de l'origine du vecteur
SHAPS1	ENTREE	ENTIER	taille de la figure du premier objet
NBOX2	ENTREE	ENTIER	case du second objet
LOCN2	SORTIE	ENTIER	position dans la case de l'extrémité du vecteur
SHAPS2	ENTREE	ENTIER	taille de la figure du second objet
CMODE	ENTREE	ENTIER	0 -> effacer , 1 -> afficher
LNTYPE	ENTREE	ENTIER	type de ligne pour le vecteur
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

45. GRARRO(ix1,iy1,ix2,iy2,cmode) (GI)

Description

Cette routine trace une flèche à l'extrémité d'un vecteur.La coordonnée de cette extrémité est (ix1,iy1).

Paramètre

NOM	USAGE	TYPE	Description
IX1,IY1	ENTREE	ENTIER	coordonnées de l'extrémité du vecteur
IX2,IY2	ENTREE	ENTIER	coordonnées d'un deuxième point du vecteur
CMODE	ENTREE	ENTIER	0 -> effacer , 1 -> afficher

46. GRVMCR(mode,nbox,locn,lntype) (M)

Description

Cette routine permet la crèation manuelle d'un vecteur à l'écran.

Paramètre

NOM	USAGE	TYPE	Description
MODE	ENTREE	ENTIER	0 -> dèplacement , 1-> tracer , 2 -> tracer avec une flèche
MODE	SORTIE	ENTIER	1 -> continuer la crèation 0 -> fin de la crèation
NBOX	SORTIE	ENTIER	case sèlectionnèe
LOCN	SORTIE	ENTIER	position sèlectionnèe dans la case
LTYPE	ENTREE	ENTIER	type de ligne pour le vecteur

47. GRVMRB(mode,nbox,locn,iline) (M)

Description

Cette routine permet de reconstruire manuellement un vecteur enregistrè dans la base de donnèes graphique.

Paramètre

NOM	USAGE	TYPE	Description
MODE	ENTREE	ENTIER	0 -> dèplacement ,1 -> tracer , 2 -> tracer avec une flèche
NBOX	ENTREE	ENTIER	numèro de case pour le vecteur
LOCN	ENTREE	ENTIER	position au sein de la case
ILINE	ENTREE	ENTIER	0 -> effacer , 1 -> afficher

48. GRZOOM(nbox1,nbox2,ierr) (M)

Description

Cette routine permet à l'utilisateur d'effectuer un zoom sur un zone de l'écran.

Paramètre

NOM	USAGE	TYPE	Description
NBOX1	ENTREE	ENTIER	case correspondant au coin inférieur gauche de surface agrandie
NBOX2	ENTREE	ENTIER	case correspondant au coin supérieur droit de la zone agrandie
IERR	SORTIE	ENTIER	indique si une erreur est intervenue

49. GRGSIN (GI)

Description

Cette routine initialise le tableau reprenant le statut des cases.

4. DI

Cette librairie contient un ensemble de routines graphiques faisant appel aux primitives du logiciel DI3000.Elle devra être recodée en fonction de l'installation locale.

Les fonctions graphiques utilisées sont relativement simples afin d'assurer la portabilité du système;elles ne font appel qu'au premier niveau du logiciel DI3000.

1. DIBEAM(ix,iy)

Description

Cette routine rend la position courante à l'écran.

Paramètre

NOM	USAGE	TYPE	Description
IX,IY	SORTIE	ENTIER	coordonnées du point courant

2. DIDA(ix,iy)

Description

Cette routine trace un segment de droite à partir du point courant jusqu'au point (ix,iy).

Paramètre

NOM	USAGE	TYPE	Description
IX,IY	ENTREE	ENTIER	coordonnées de l'extrémité du segment de droite

3. DIDI(ix,iy)

Description

Cette routine trace d'une manière relative un segment de droite à partir du point courant.Le déplacement relatif est défini par les valeurs ix,iy.

Paramètre

NOM	USAGE	TYPE	Description
IX,IY	ENTREE	ENTIER	déplacement relatif à effectuer

4. DIER

Description

Cette routine efface l'écran.

DIMA(ix,iy)

Description

Cette routine effectue un déplacement absolu vers le point (ix,iy).

Paramètre

NOM	USAGE	TYPE	Description
IX,IY	ENTREE	ENTIER	coordonnées du point

5. DIMI(ix,iy)

Description

Cette routine effectue un déplacement relatif de "ix" sur l'axe des X et de "iy" sur l'axe des Y.

Paramètre

NOM	USAGE	TYPE	Description
IX,IY	ENTREE	ENTIER	déplacement relatif

6. DIPEN(ix,iy)

Description

Cette routine marque le point (ix,iy).

Paramètre

NOM	USAGE	TYPE	Description
IX,IY	ENTREE	ENTIER	coordonnées du point

7. DITERM

Description

Cette routine remet le terminal en mode alphanumérique.

8. DIARC(rad,di,sangl,eangl)

Description

Cette routine trace un arc dont le centre est le point courant de l'écran.Le rayon est égal à "rad".Les angles de départ et de fin sont déterminés respectivement par "sangl" et "eangl".

Paramètre

NOM	USAGE	TYPE	Description
RAD	ENTREE	ENTIER	rayon de l'arc
DI	ENTREE	ENTIER	nombre de segments de droites composant l'arc
SANGL	ENTREE	ENTIER	angle de depart (degrè)
EANGL	ENTREE	ENTIER	angle d'arrivée (degrè)

9. DIGCUR(ix,iy,ib)

Description

Cette routine demande à l'utilisateur d'introduire un coordonnées (ix,iy).Elle rend également la touche sélectionnée lors de cette introduction.

Paramètre

NOM	USAGE	TYPE	Description
IX,IY	ENTREE	ENTIER	coordonnées du point introduit
IB	ENTREE	ENTIER	touche sélectionnée

10. DIINIT

Description

Cette routine initialise le terminal en mode graphique.

11. DILINE(lntype)

Description

Cette routine permet de définir le type de ligne.

Paramètre

NOM	USAGE	TYPE	Description
LNTYPE	ENTREE	ENTIER	le type de ligne

12. DIRECT(delh,delv)

Description

Cette routine dessine,à partir de la position courante,un rectangle de longueur "delv" et de largeur "delh".

Paramètre

NOM	USAGE	TYPE	Description
DELH	ENTREE	ENTIER	longueur du rectangle
DELV	ENTREE	ENTIER	largeur du rectangle

13. DITRAN

Description

Cette routine transmet les ordres contenus dans le buffer du terminal.

14. DITXT(nchars,txtstr,isxc,isyc,ch,cvxsize,ysize,nangl)

Description

Cette routine assure l'impression d'un texte.

Paramètre

NOM	USAGE	TYPE	Description
NCHARS	ENTREE	ENTIER	longueur du texte
TXTSTR	ENTREE	STRING	texte à afficher
ISXC ISYC	ENTREE	ENTIER	coordonnées pour l'impression du texte
CH	ENTREE	ENTIER	type de cadrage horizontale
CV	ENTREE	ENTIER	type de cadrage verticale
XSIZE	ENTREE	ENTIER	longueur d'un caractère
YSIZE	ENTREE	ENTIER	largeur d'un caractère
NANGLE	ENTREE	ENTIER	angle définissant le plan de base du texte

15. DIWIND(nwd)

Description

Cette routine permet de definir un fenêtre sur la zone de travail.

Paramètre

NOM	USAGE	TYPE	Description
NWD	ENTREE	ENTIER	= 1 -> reduire la fenêtre à la surface de travail = 0 -> rétablir la fenêtre générale

5. BEN

Cette librairie contient un ensemble de routines graphiques faisant appel aux primitives de la table tracante [BENSON].

Elle devra être recodée en fonction de l'installation locale.

Elle reprend les fonctions graphiques non interactives des routines de DI (tracé relatif ou absolu,déplacement relatif ou absolu,etc)