



## THESIS / THÈSE

### MASTER EN SCIENCES INFORMATIQUES

MEANDRES ou Proposition d'une interface pour un Module d'Evaluation et d'ANalyse De REsultats Statistiques

Gillo, Xavier

*Award date:*  
1991

*Awarding institution:*  
Universite de Namur

[Link to publication](#)

#### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

#### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

FACULTES UNIVERSITAIRES  
NOTRE-DAME DE LA PAIX



NAMUR

INSTITUT D'INFORMATIQUE

**MEANDRES**  
ou  
**Proposition d'une interface pour un Module  
d'Evaluation et d'ANalyse De RESULTats  
Statistiques**

X. Gillo

Promoteur : F. Bodart

Mémoire présenté en vue de  
l'obtention du titre de licencié  
et maître en informatique

Année académique 1990-1991

RUE GRANDGAGNAGE, 21, B - 5000 NAMUR (BELGIUM)

## **Résumé**

Ce mémoire présente MEANDRES, une interface conviviale pour un Module d'Evaluation et d'ANalyse De RESULTats Statistiques. Ces mesures sont produites par DSL/SIM, l'outil de simulation du comportement des systèmes d'information de l'atelier logiciel IDA.

L'interface proposée a pour but de faciliter la consultation de ces résultats statistiques et l'élaboration de rapports reprenant les mesures que l'utilisateur juge significatives. La conception de cette interface repose largement sur la mise en œuvre de critères et de règles ergonomiques communément admis.

MEANDRES se veut conforme aux normes de Ms-Windows 2, un environnement multi-fenêtré tournant sur les P.C.

## **Abstract**

This study describes MEANDRES, a user-friendly man-machine interface for a module for evaluation and analysis of statistical results. The row data are produced by DSL/SIM, a tool that simulates the behaviour of information systems.

This man-machine interface is introduced in order to ease the examination of the statistical results and the creation of reports based on those that the user judge representative. The design of the user-interface rests on applying largely accepted ergonomical criteria and rules.

MEANDRES is fully compatible with the Ms-Windows 2 multi-windowing PC-based environment.

## Remerciements

Tous nos remerciements vont au Professeur François Bodart qui nous a permis de réaliser ce mémoire. Nous sommes très sensibles à la confiance qu'il nous a accordée et à la très grande compréhension dont il a fait preuve à notre égard.

Nous tenons également à remercier toute l'équipe IDA et celle du projet TRIDENT : nous pouvions toujours compter sur leur aide et sur leurs conseils. Une telle disponibilité n'a pas de prix pour un "mémoire".

Un remerciement tout spécial à Madame Christiane Decoux-Leroy pour sa gentillesse et la disponibilité dont elle a fait preuve dans le cadre d'un travail qui ressemblait curieusement à celui du présent mémoire.

Nous sommes également très reconnaissants à l'équipe du Centre de Calcul non seulement d'avoir mis à notre disposition ses nombreuses compétences, mais surtout de s'être constamment souciée de notre labeur.

Merci à tous ceux et celles que nous n'avons pas cités et qui, même modestement, nous ont aidés.

Enfin, merci à toi, Laurence, sans qui nous n'aurions pas eu la force d'aller jusqu'au bout.



## Table des matières

Introduction.....	1
<b>1. Contexte et tâche.....</b>	<b>3</b>
1.1 Contexte et objectif du mémoire.....	3
1.2 Statistiques.....	9
1.2.1 Mesures concernant les processus.....	10
1.2.2 Mesures concernant les ressources.....	11
1.2.3 Mesures concernant les points de synchronisation.....	12
1.3 Analyse de la tâche.....	13
1.3.1 Mise au point.....	13
1.3.2 Utilisateurs.....	13
1.3.3 Tâche.....	14
<b>2. Présentation de l'outil de consultation de statistiques et d'élaboration de rapports.....</b>	<b>20</b>
2.1 Pour faciliter la lecture de ce chapitre.....	20
2.1.1 Présentation succincte de l'interface de l'outil.....	20
2.1.2 Conventions typographiques.....	21
2.1.3 Concepts.....	21
2.1.4 Structure du chapitre.....	22
2.1.5 Hypothèse.....	23
2.2 Fenêtres.....	24
2.2.1 Fenêtre principale.....	24
2.2.2 Fiche.....	26
2.3 Menu Fichier.....	27
2.3.1 Nouveau.....	27
2.3.2 Ouvrir.....	28
2.3.3 Sélectionner un objet.....	30
2.3.4 Enregistrer.....	32
2.3.5 Enregistrer sous.....	33
2.3.6 Fermer.....	34
2.3.7 Aperçu avant impression.....	35
2.3.8 Effectuer la mise en page.....	36
2.3.9 Imprimer.....	37
2.3.10 Quitter.....	38
2.4 Menu Edition.....	40
2.4.1 Annuler.....	40
2.4.2 Couper.....	41
2.4.3 Copier.....	41
2.4.4 Coller.....	42
2.4.5 Effacer.....	42
2.4.6 Tout sélectionner.....	43
2.5 Menu Processus.....	44
2.5.1 St. globales.....	44
2.5.2 St. périodiques.....	48
2.5.3 St. chronologiques.....	51
2.5.4 Commentaires.....	55
2.5.5 Calendrier.....	57

2.5.6 Ressources requises .....	59
2.5.7 Définir fiche standard .....	59
2.6 Menu Ressources.....	61
2.6.1 St. globales.....	61
2.6.2 St. périodiques .....	64
2.6.3 St. chronologiques .....	65
2.6.4 Commentaires.....	67
2.6.5 Processus requérants.....	68
2.6.6 Définir fiche standard .....	69
2.7 Menu Pt. de synchr.....	70
2.7.1 St. globales.....	70
2.7.2 Commentaires.....	75
2.7.3 Définir fiche standard .....	76
2.8 Menu Ecran .....	77
2.8.1 Aide .....	77
2.8.2 Informations.....	77
2.8.3 Réorganiser .....	78
2.8.4 Masquer la fiche .....	79
2.8.5 Afficher les règles.....	79
2.8.6 Liste des documents.....	80
2.9 Sélection et déformation.....	83
2.9.1 Mécanismes de sélection .....	83
2.9.2 Modifier la taille des informations .....	85
<b>3. Commentaires et justifications .....</b>	<b>87</b>
3.1 Règles ergonomiques .....	87
3.2 Fenêtres .....	88
3.2.1 Fenêtre principale .....	88
3.2.2 Fiche .....	89
3.3 Menu Fichier .....	90
3.3.1 Nouveau.....	90
3.3.2 Ouvrir.....	90
3.3.3 Sélectionner un objet .....	92
3.3.4 Enregistrer .....	93
3.3.5 Enregistrer sous .....	94
3.3.6 Fermer.....	94
3.3.7 Aperçu avant impression .....	95
3.3.8 Effectuer la mise en page.....	95
3.3.9 Imprimer .....	95
3.4 Menu Edition.....	96
3.4.1 Annuler .....	96
3.4.2 Couper .....	96
3.4.3 Copier .....	97
3.4.4 Coller .....	97
3.4.5 Effacer .....	97
3.4.6 Tout sélectionner .....	98
3.5 Menu Processus.....	98
3.5.1 St. globales.....	99
3.5.2 St. périodiques .....	100
3.5.3 St. chronologiques .....	101
3.5.4 Commentaires .....	102
3.5.5 Calendrier .....	103
3.5.6 Ressources requises .....	104
3.5.7 Définir fiche standard .....	104

3.6 Menu Ressources.....	105
3.7 Menu Pt. de synchr.....	106
3.7.1 St. globales.....	106
3.8 Menu Ecran .....	106
3.8.1 Aide .....	107
3.8.2 Informations.....	107
3.8.3 Réorganiser .....	108
3.8.4 Masquer la fiche .....	108
3.8.5 Afficher les règles.....	108
3.8.6 Liste des documents.....	109
3.9 Sélection et déformation.....	110
3.9.1 Mécanismes de sélection .....	110
3.9.2 Modifier la taille des informations .....	110
<b>4. Critiques et améliorations .....</b>	<b>112</b>
4.1 Changements importants.....	112
4.1.1 Scinder la gestion des fiches et celle des rapports.....	112
4.1.2 Simulations et rapports .....	114
4.1.3 Texte libre.....	114
4.2 Liste des documents .....	115
4.3 Modifier le standard des fiches .....	116
4.4 Divers .....	118
<b>5. A propos de la conception d'interfaces .....</b>	<b>123</b>
5.1 Itinéraire .....	123
5.2 Suggestions pour faciliter le travail du concepteur d'interfaces .....	125
<b>Conclusion .....</b>	<b>128</b>
<b>Bibliographie .....</b>	<b>130</b>

---

## **Introduction**

Quel est l'objet de notre mémoire ? Dans quel contexte s'inscrit-il ? Voilà ce que le premier chapitre essaie de préciser. Après avoir rappelé certaines considérations théoriques, nous y développons brièvement certains aspects importants pour la conception de notre interface.

Le deuxième chapitre constitue le noyau du mémoire. Il présente tout d'abord succinctement l'interface afin que le lecteur puisse en avoir une vision globale. Après avoir exposé les principaux concepts mis en œuvre, ce chapitre décrit, menu par menu, commande par commande, l'outil de consultation de résultats statistiques et d'élaboration de rapports.

Le troisième chapitre justifie les choix que nous avons opérés lors de la conception de l'interface. Nous y ajoutons également certains commentaires que nous inspirant certaines situations rencontrées lors de l'élaboration de notre outil. Ce chapitre essaie de faire sentir au lecteur qu'une interface est le fruit de nombreuses décisions, que ces décisions ne sont pas toujours prises en fonction de règles ergonomiques strictes et parfois contradictoires, que la créativité et l'imagination trouvent leur place dans cet univers de rigueur.

La lecture du deuxième chapitre demande un réel effort tant il est difficile d'imaginer concrètement la réalité de l'interface à partir d'un texte seul. C'est pourquoi nous avons été amenés à réaliser une maquette permettant de se faire plus facilement une idée de nos propositions. Cette maquette est détaillée dans les annexes.

Grâce à cet outil de communication, nous avons pu recueillir les avis de différentes personnes. Nous avons eu la chance de réunir des informaticiens dont les préoccupations sont complémentaires de sorte que les critiques émises furent à chaque fois fort différentes. Certains ont examiné les aspects fonctionnels de l'interface : l'outil améliore-t-il réellement la tâche du concepteur de systèmes d'information ? D'autres se sont plus souciés des aspects conceptuels : les notions autour desquelles s'articule notre outil sont-elles de bon aloi ? Enfin, d'autres encore se sont préoccupés de l'ergonomie

de notre outil : respecte-t-il les nombreuses règles qui régissent cette discipline si empirique qu'est la conception d'interfaces hommes/machines. Les critiques et suggestions émises par ces différentes personnes ont été rassemblées dans un quatrième chapitre.

Enfin, un cinquième chapitre retrace l'itinéraire de notre mémoire. Il met en évidence certaines difficultés que nous avons rencontrées, et propose quelques solutions pour faciliter le travail des concepteurs d'interfaces.

---

## Contexte et tâche

### 1.1 CONTEXTE ET OBJECTIF DU MEMOIRE

Le programme de consultation de statistiques que nous proposons s'inscrit dans le contexte général de la conception des systèmes d'information (S.I.), et plus précisément dans le cadre de la méthode IDA (Interactive Design Approach).

Il ne nous appartient pas de résumer ici cette méthode. Nous renvoyons le lecteur profane à l'ouvrage "*Conception assistée des systèmes d'information*" des Professeurs F. Bodart et Y. Pigneur [Bod89]. Toutefois, pour la facilité du lecteur, il nous semble opportun de rappeler certains éléments qui lui permettront de cerner le contexte spécifique dans lequel s'inscrit notre mémoire. Pour atteindre cet objectif, nous nous permettons de puiser largement dans le cinquième chapitre de l'ouvrage cité ci-dessus.

Pour réaliser leur tâche de conception de S.I., les analystes suivent des méthodes, disposent de modèles et d'outils. Ils peuvent vérifier la validité de leur travail grâce à un ensemble de règles. Celles-ci leur permettent de déterminer si les spécifications sont bonnes (c'est-à-dire complètes, cohérentes et non-redondantes par rapport au modèle utilisé). Mais ces spécifications fonctionnelles décrivent-elles quelque chose de bon, c'est-à-dire fournissent-elles "une bonne solution sous l'angle de l'aptitude à réaliser les objectifs que l'organisation assigne au S.I." ?

Pour répondre à cette question, deux alternatives s'offrent aux concepteurs de S.I. :

- attendre la mise en œuvre du système pour réaliser une vérification a posteriori;
- expérimenter dès que possible des éléments de la solution proposée.

Pour des raisons évidentes, c'est l'approche expérimentale qu'a adoptée la méthode IDA. Cette dernière propose trois types de prototypes (pas tous implémentés) :

- le prototypage des *performances* du S.I. afin d'évaluer si la solution fonctionnelle proposée est *réalisable* (faisable) compte tenu des ressources disponibles,
- le prototypage des *règles de mémorisation et de traitement* afin d'évaluer la *conformité* de la solution proposée aux besoins informationnels,
- le prototypage des *dialogues* pour apprécier si les interfaces hommes-machines sont adaptés aux exigences des différents postes de travail."

Dans notre mémoire, nous ne nous intéressons qu'au prototypage des performances des S.I.

L'objectif majeur de l'évaluation de la faisabilité par simulation est de "contrôler que le comportement modélisé garantit le respect des performances attendues de la solution envisagée, compte tenu du niveau d'utilisation des ressources et de la charge estimée du futur système".

Pour atteindre cet objectif, il est nécessaire que deux importants prérequis soient satisfaits :

- d'une part, les performances attendues de la solution à élaborer devraient avoir été précisées et donc faire partie intégrante de sa spécification. Celles-ci peuvent notamment porter sur des délais de traitement à respecter, des volumes à absorber par période, des taux de service ou d'utilisation de ressources à garantir.
- d'autre part, une estimation quantitative du comportement devrait être disponible. Celle-ci devrait être élaborée à partir de mesures de comportement, réel ou simulé, du système d'information sous étude. Ces mesures de comportement devraient non seulement permettre d'évaluer les performances du système futur, mais aussi de localiser et expliquer les problèmes potentiels, c'est-à-dire les divergences entre les performances attendues et estimées.

De plus, ces mesures du comportement doivent pouvoir être établies directement à partir des seules spécifications du système futur et principalement celles concernant les trois aspects suivants :

- le comportement dynamique du S.I. (cfr modèle de la dynamique des traitements [Bod89,70]),
- les moyens requis (cfr modèle des ressources [Bod89,91]),
- la charge estimée (cfr échéanciers des événements externes [Bod89,89]).

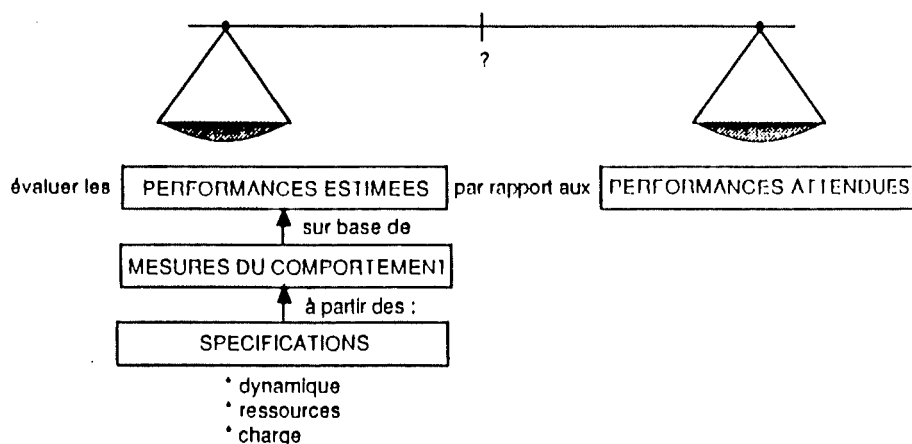


Illustration 0 : Objectif de l'évaluation de la faisabilité

Dans l'environnement IDA, les mesures de comportement correspondent aux résultats statistiques fournis par un outil logiciel d'évaluation : DSL/SIM.

La caractéristique la plus remarquable de ce logiciel est qu'il assure "la génération automatique d'un programme de simulation au départ des seules spécifications enregistrées dans une base de données." *Les résultats statistiques sont donc produits en "exécutant" les spécifications fonctionnelles du S.I.*

L'architecture fonctionnelle de DSL/SIM est composée des trois modules suivants :

- la *génération* automatique d'un programme de simulation qui correspond à une version exécutable des spécifications du comportement enregistrées dans le dictionnaire de l'environnement de spécification IDA;
- l'*exécution* proprement dite du programme de simulation qui collecte des mesures dans une base de données statistiques;
- l'*exploitation* de cette base de données statistiques, description posthume du comportement simulé, pour produire des rapports d'analyse de ce comportement."

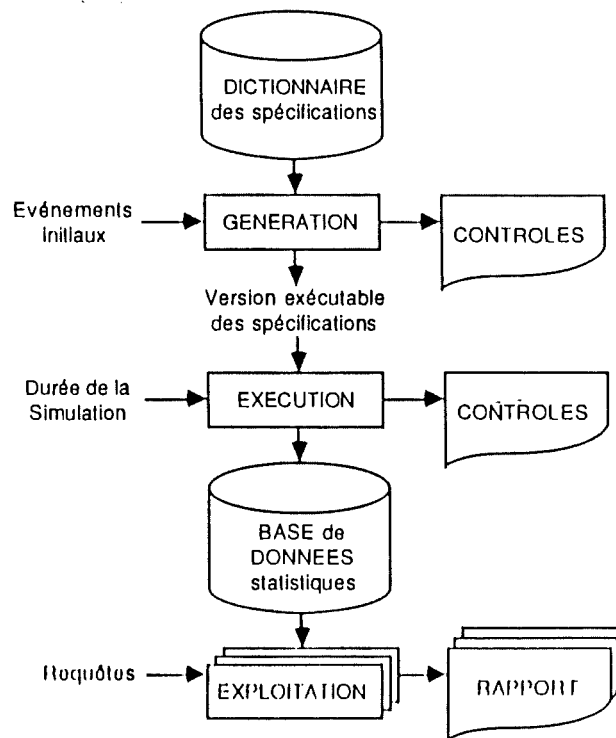


Illustration 1 : Architecture fonctionnelle de DSL/SIM

### Génération de la simulation

"L'objectif de ce composant est de générer automatiquement un programme de simulation à partir des seules spécifications enregistrées dans le dictionnaire. Les spécifications extraites concernent la dynamique des traitements, la charge estimée et les moyens ou ressources requises.

Les seuls paramètres que l'utilisateur doit préciser sont les types d'événements initiaux du ou des schémas dynamiques qu'il souhaite sélectionner, simuler et donc évaluer.



Préalablement à la production du programme de simulation, l'outil effectue un certain nombre de contrôles pour assurer que la simulation souhaitée est possible. Un rapport signale à l'utilisateur les éventuelles anomalies détectées qui subsistent dans les spécifications et empêchent l'exécution de la simulation correspondante."

#### Exécution de la simulation

"L'objectif ici est de simuler le comportement spécifié sur une période de temps, par l'exécution du programme généré à l'étape précédente, en vue de collecter des mesures sur le fonctionnement simulé.

L'utilisateur se contente uniquement de préciser la durée de simulation voulue. Il peut éventuellement interrompre la simulation, visualiser certaines variables d'état caractéristiques de l'exécution, ajuster la durée de simulation et relancer son exécution.

Il peut aussi obtenir une trace d'exécution, plus ou moins détaillée, qui reprend chronologiquement tous les événements détectés en cours de simulation. Pendant toute la simulation, des mesures de comportement sont collectées dans une base de données statistiques."

#### Exploitation de la simulation

"L'objectif de ce composant est d'extraire certaines mesures élémentaires, collectées dans la base de données statistiques, et de les traiter pour produire des rapports d'analyse du comportement simulé. Ces rapports doivent permettre l'évaluation des performances obtenues par la simulation et leur comparaison avec celles attendues par le concepteur. Ils doivent également faciliter la détection des causes d'un comportement anormal ou jugé non satisfaisant.

Ces rapports sont obtenus à la demande de l'utilisateur qui adresse à l'outil des requêtes concernant le contenu et la forme des rapports souhaités. Ces requêtes doivent notamment mentionner :

- les types d'objets pour lesquels on souhaite obtenir des résultats statistiques;
- la période de référence, à l'intérieur de la période simulée, pour laquelle on veut produire des résultats;
- la nature des statistiques à calculer : valeurs agrégées, séries chronologiques, etc.
- la forme de présentation des résultats : tableaux, courbes, histogrammes, schémas quantifiés, etc."

A l'heure actuelle, les rapports se présentent uniquement sous forme de listings, généralement assez volumineux<sup>1</sup>. Ces listings, quoique bien conçus, présentent un certain nombre d'inconvénients :

- le volume de ces listings rend fastidieuse la recherche de résultats statistiques particuliers d'un objet ;
- contrairement à ce qui est écrit ci-dessus, les informations sont uniquement présentées sous forme de textes ou de tableaux de nombres, accroissant ainsi l'effort du concepteur qui souhaite repérer rapidement un comportement anormal du S.I. Pour s'en convaincre, il suffit de comparer les deux illustrations ci-dessous :

---

<sup>1</sup> Nous invitons le lecteur à prendre connaissance de la structure de ces listings en annexe. Il pourra ainsi se faire une idée des difficultés que rencontre le concepteur qui doit utiliser ces listings pour analyser le comportement d'un S.I.

PROCESS Exec-Requis_Prod		Number of				
[Low Bnd	Upp Bnd[	Trig	Incp	Intr	Resm	Term
08h	09h	1	2	0	1	1
09h	10h	2	1	0	0	0
10h	11h	2	2	0	0	3
11h	12h	2	2	0	0	1
12h	13h	2	1	3	0	1
13h	14h	0	0	0	3	1
14h	15h	1	3	0	0	2
15h	16h	2	2	0	0	2
16h	17h	3	2	0	0	2
17h	18h	0	0	2	0	1
18h	19h	2	0	0	0	0

Illustration 2 : Informations telles que présentées sur un listing

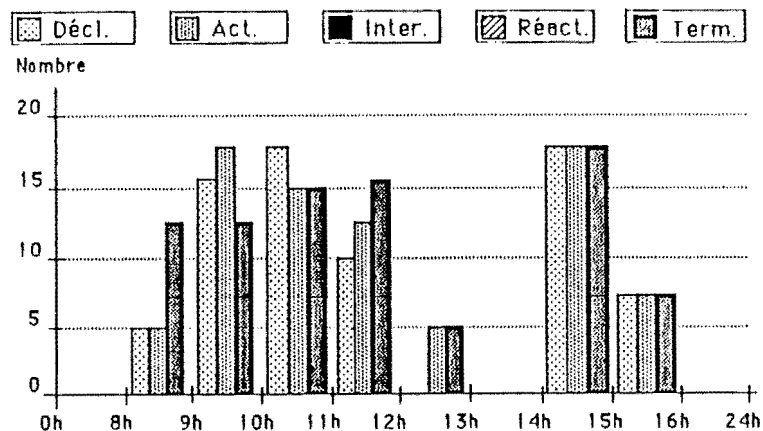


Illustration 3 : Informations analogues présentées sous forme d'histogramme

- les listings reproduisent toutes les mesures statistiques, même celles qui ne sont pas pertinentes à un moment donné du processus d'analyse du concepteur. Celui-ci doit donc produire un effort supplémentaire pour séparer le bon grain de l'ivraie.

Bref, l'ergonomie assez pauvre de ce type de rapport nuit à la bonne exploitation des résultats statistiques, ce qui est en contradiction avec la raison d'être de ces rapports. Cette constatation a également été faite par d'autres étudiants amenés à utiliser DSL/SIM :

“Le principal grief procède de l'impossibilité, à notre connaissance, d'obtenir des rapports à "envergure variable"” [Reu87,24].

“Le volume et la précision des résultats statistiques, chronologiques sont stupéfiants. Pris isolément, les résultats d'une simulation sont difficilement interprétables; lorsqu'on les compare, ils deviennent plus parlants. De plus, on ne dispose pas de sortie graphique des résultats, c'est pourquoi le temps [que nous avons] consacré à réaliser les diagrammes n'est pas négligeable. Une amélioration en ce sens serait la bienvenue, afin d'assurer la meilleure continuité possible entre l'input et l'output” [Jou89].

“La forme de présentation tabulaire des résultats de la simulation pose des difficultés d'interprétation étant donné leur aspect peu convivial. Il est donc nécessaire de disposer d'un environnement plus ergonomique qui permet non seulement une présentation

graphique de ces résultats, mais qui présente également un environnement d'exploitation multifenêtres offrant ainsi au concepteur une capacité maximale de navigation dont l'intérêt se révélera prédominant pour les outils que nous proposons" [Lef87,5].

**Il est donc opportun de proposer au concepteur un outil de consultation de statistiques et d'élaboration de rapports plus adéquat à sa tâche d'évaluation des performances d'un S.I. Ceci constitue l'objet du présent mémoire.**

## 1.2 STATISTIQUES

Avant de préciser la nature de la tâche que notre outil est censé faciliter, il nous semble utile d'exposer l'ensemble des mesures statistiques dont dispose le concepteur.

Nous pensons que cela permettra au lecteur qui ne maîtrise pas DSL/SIM de comprendre plus facilement, d'une part, la nature de la tâche du concepteur, et, d'autre part, l'outil proposé. Nous profitons également de cette section pour préciser l'un ou l'autre concept important, ainsi que certains termes de vocabulaire.

De nouveau, nous empruntons ce qui suit à l'ouvrage "*Conception assistée des systèmes d'information*" [Bod89]. Le lecteur qui souhaite disposer de plus de détails sur les statistiques, en particulier sur la manière dont elles sont produites, trouvera réponse à ses questions dans l'ouvrage [IDA88].

Les statistiques produites par DSL/SIM portent :

- soit sur des **nombres** de changements d'état caractéristiques du comportement des objets considérés (exemple : le nombre de déclenchements de processus d'un certain type);
- soit sur des **durées** de certains états caractéristiques par lesquels évoluent les objets considérés (exemple : la durée d'attente d'un processus);
- soit encore sur des **volumes** ou nombres d'objets dans certains états caractéristiques à un moment donné (exemple : nombre de processus en attente ou capacité disponible d'une ressource à l'instant t)".

Les mesures effectuées concernent essentiellement trois types d'objets :

- les processus,
- les ressources,
- les points de synchronisation.

En fonction de la nature de la période de temps sur laquelle on les calcule, ces mesures peuvent être de trois types :

- *globales* : les mesures concernent l'entièreté de la période de référence (par exemple, du 5<sup>ème</sup> au 20<sup>ème</sup> jour de la simulation);
- *chronologiques* : les mesures concernent des cellules d'observation se répétant dans une période de référence plus large (par exemple, toutes les heures du 5<sup>ème</sup> au 20<sup>ème</sup> jour de la simulation);
- *périodiques*<sup>1</sup> : ces mesures sont des mesures chronologiques cumulées pour chacune des cellules d'observation (par exemple, heure par heure d'une journée moyenne reconstruite à partir de la 5<sup>ème</sup> à la 20<sup>ème</sup> journée de la simulation).

Signalons toutefois que dans le cas des points de synchronisation, seules des mesures globales sont produites.

---

<sup>1</sup> Le lecteur trouvera dans la littérature consacrée à DSL/SIM le terme "chronologique cumulée" plutôt que "périodique". Nous avons préféré la deuxième expression pour des raisons de concision.

## 1.2.1 Mesures concernant les processus

La notion de processus est issue du modèle de la dynamique des traitements.

“Un *processus* représente l’exécution d’une procédure qui correspond à un traitement effectué dans le système d’information. C’est donc une entité dynamique qui apparaît à chaque exécution de la procédure et dont la progression peut être observée par certains changements d’états remarquables”.

Ces états peuvent être au nombre de cinq : un processus peut être

- *déclenché*, c’est-à-dire créé en réaction à la survenance d’un événement;
- *activé*, c’est-à-dire réellement exécuté; pour cela, il est nécessaire que toutes les ressources qui lui sont nécessaires lui soient allouées;
- *interrompu*, cela survient si l’une au moins des ressources nécessaires à l’exécution du processus n’est plus disponible;
- *réactivé*, le processus interrompu peut être réactivé si les ressources requises sont à nouveau disponibles;
- *terminé*, dans ce cas les ressources allouées au processus sont libérées.

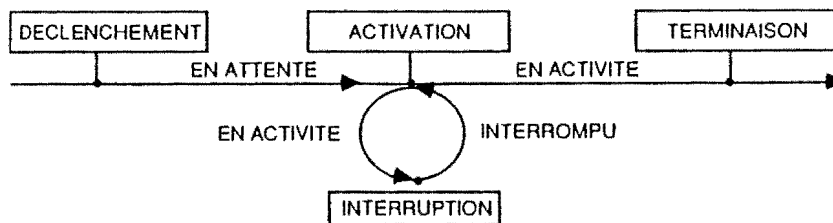


Illustration 4 : Evolution d'un processus dans le temps

Le comportement des processus, pour une période de référence donnée, peut notamment se mesurer par :

“Les *nombre de déclenchements*, d’*activations*, d’*interruptions*, de *réactivations* et de *terminaisons* correspondent à de simples comptages de ces différents changements d’état pour les processus, d’un type donné, dans la période considérée.”

La *durée d’attente d’activation*<sup>1</sup> “correspond à la durée comprise entre le déclenchement et l’activation du processus; il s’agit donc d’une durée pendant laquelle le processus est en attente de ressources, momentanément utilisées par d’autres processus”.

“La *durée d’interruption* est la durée pendant laquelle un processus, qui a été activé, est interrompu par un processus prioritaire qui lui retire temporairement, par préemption, les ressources qu’il utilisait.”

La *durée d’attente* est la somme de la durée d’attente d’activation et de la durée d’interruption. Elle mesure donc le temps durant lequel le processus attend une ou plusieurs ressources allouées à des processus concurrents.

“La *durée de repos* est également une durée pendant laquelle un processus est bloqué et en attente de ressources indisponibles, mais qui le sont également pour tout

<sup>1</sup> ATTENTION ! Contrairement à l’ouvrage d’où sont issues ces définitions, nous faisons une différence entre “durée d’attente” et “durée d’attente d’activation”. Le concept de “durée d’attente” de l’ouvrage est devenu le concept de “durée d’attente d’activation” dans notre mémoire.

autre processus, en fonction de leur calendrier de disponibilité (pause, nuits et week-ends, par exemple).”

“La *durée d’activité* est la durée pendant laquelle le processus est actif et exécute les règles de sa procédure en disposant des ressources nécessaires. Cette mesure est en fait la simple restitution statistique de la durée d’exécution spécifiée par le concepteur.”

“La *durée totale d’exécution* s’étend de la date de déclenchement à celle de terminaison du processus; elle correspond à la somme des durées précédentes pour un même processus<sup>1</sup>.”

“A chacun des états possibles d’un processus est également associée une mesure précisant le volume de processus d’un type donné dans cet état à un moment donné.”

## 1.2.2 Mesures concernant les ressources

“Une *ressource* représente un certain volume de moyens physiques identiques dont l’organisation doit disposer et dont une partie peut être requise par un traitement pour pouvoir s’exécuter; son indisponibilité totale ou partielle peut entraîner le blocage temporaire d’un processus qui la requiert.”

Une ressource “est généralement caractérisée par une propriété - ou une contrainte - de *disponibilité* qui fixe, pour les périodes pendant lesquelles la ressource est disponible, la *capacité* ou volume maximal que peuvent se partager, à un moment donné, les processus qui la requièrent.”

“Les périodes de disponibilité sont regroupées dans un *calendrier* exprimé sous la forme d’intervalles de temps qui se reproduisent périodiquement dans une période plus large.”

La capacité d’une ressource, “qui indique la quantité maximale de ressource disponible pour les traitements, peut être complétée par une propriété qui fixe le nombre maximal de processus qui peuvent se partager, dans les limites de cette capacité, l’utilisation de la ressource.”

Le comportement des ressources, sur une période de référence donnée, peut être analysé à partir des mesures suivantes :

“Les *nombre de réquisitions, d’allocations et de désallocations* sont obtenus par simple comptage : combien de fois un processus a-t-il requis, acquis et libéré une certaine quantité de la ressource dans la période considérée.”

“Le *volume en attente* ou *taille de la file d’attente* correspond au nombre de processus, en attente ou interrompus, faute de pouvoir disposer d’une quantité suffisante de la ressource à un moment donné.”

“La *capacité* ou *volume utilisé* correspond à la quantité totale de la ressource que se partage un ensemble de processus, en activité, à un moment donné.”

“Le *taux d’utilisation* correspond au rapport entre les capacités utilisées et maximales.”

“La *capacité résiduelle* correspond à la différence entre les capacités maximale et utilisée. Il s’agit donc de la quantité de ressource encore disponible à un moment donné pour d’éventuels processus qui se déclencheraient et en seraient demandeurs.”

---

<sup>1</sup> Attention à ne pas comptabiliser deux fois la durée d’attente (sic).

### 1.2.3 Mesures concernant les points de synchronisation

La notion de point de synchronisation est également issue du modèle de la dynamique des traitements.

“Le mécanisme de *synchronisation* est utilisé pour représenter une situation dans laquelle il faut attendre la survenance d'une combinaison de deux ou plusieurs événements avant de provoquer une réaction particulière dans le système d'information. Il s'agit donc d'un mécanisme de coordination d'événements qui permet d'en construire un nouveau à partir d'autres plus élémentaires : ce nouvel élément, appelé réalisation de la synchronisation, survient donc lorsque tous les événements plus élémentaires, dits contributeurs à la réalisation, ont été constatés. Dès cet instant, il s'agit d'un événement comme un autre : il en a les mêmes propriétés et peut dès lors avoir les mêmes effets.”

Le comportement des synchronisations, sur une période de référence donnée, peut être caractérisé par des mesures sur le nombre et la durée des réalisations. Le comportement de leurs événements contributeurs peut également se mesurer en nombre et en durée de contributions.

“Le *nombre de réalisations* correspond à un simple comptage des réalisations d'un point de synchronisation dans la période considérée.”

“La *durée d'une réalisation* correspond à la durée comprise entre la date de la réalisation et celle de survenance du plus ancien événement ayant contribué à cette réalisation.”

“La *durée d'une contribution* correspond à la durée comprise entre la date de la survenance d'un événement et celle de réalisation d'une synchronisation à laquelle il a effectivement contribué.”

“Il faut remarquer que, pour une réalisation donnée, la durée de contribution du plus ancien événement correspond à la durée de réalisation et celle du plus récent, donc du plus "en retard", est toujours nulle.”

L'événement le plus contraignant c'est-à-dire celui qui participe en dernier lieu à la réalisation (c'est donc celui qui a la durée de réalisation la plus faible), fait l'objet de deux estimations :

- la mesure de l'événement le plus contraignant *en fréquence* cumule pour toutes les réalisations les fréquences où un type d'événements est considéré comme le plus contraignant;
- la deuxième estimation est basée sur la comparaison de la différence entre le total des durées de réalisation et le total des durées de contribution pour chaque type d'événements qui ont participé aux réalisations. Si l'on cumule ces différences pour tous les types d'événements, et qu'ensuite on rapproche chaque différence de la somme des différences, on déterminera un classement des types d'événements contraignants *en durée*.

## 1.3 ANALYSE DE LA TACHE

### 1.3.1 Mise au point

Le but d'une interface est de "permettre à une personne de réaliser le mieux possible la tâche qu'elle a à accomplir, c'est-à-dire avec précision, rapidité et sans efforts inutiles (c'est-à-dire étrangers à la nature de la tâche)" [Bod90].

Cette définition donnée au début du cours "Interface homme/machine" du Professeur F. Bodart illustre bien l'importance que revêt l'analyse de la tâche qu'est censée faciliter une interface.

Idéalement, toute conception d'interface homme/machine devrait être précédée d'une telle analyse. Malheureusement, dans le cadre d'un mémoire, une telle entreprise est souvent incompatible d'une part avec le temps imparti, d'autre part avec les compétences des étudiants. En effet, il faut bien être conscient du fait que, de même qu'on ne s'intronise pas du jour au lendemain épistémologue, on ne devient pas non plus ergonome simplement le temps d'un stage dont nous n'avons pas bénéficié. Notre avis rejoint celui que B. Sacré et de J. Lefèvre émettent dans leur mémoire :

"... l'étude de profils cognitifs demande des connaissances psychologiques et le travail de plusieurs personnes pendant plusieurs mois." [Lef87,5]

Alors, que faire ?

Nous avons la chance de disposer du mémoire cité ci-dessus. L'un des objectifs de cet ouvrage est de présenter un certain nombre de diagrammes et de fonctionnalités ayant pour objet d'aider le concepteur de S.I. à interpréter les résultats des simulations exécutées à l'aide de DSL/SIM. Autrement dit, une proposition d'amélioration des moyens de consultation des mesures statistiques a déjà été réalisée.

Cette étude se base sur de nombreuses discussions qu'ont pu avoir ses auteurs avec des utilisateurs de DSL/SIM (RATP, équipe IDA, ...). Bien qu'exclusivement empirique (de l'aveu de ses auteurs), cette étude n'en demeure pas moins très fiable sur le plan des fonctionnalités proposées et de l'ergonomie sous-jacente. Il est probable qu'une analyse de la tâche menée par des professionnels relèverait certaines lacunes et envisagerait d'autres améliorations, mais dans l'état actuel de nos connaissances tant dans le domaine des interfaces homme/machine que dans celui de l'outil DSL/SIM, il nous semble que les propositions de B. Sacré et de J. Lefèvre constituent une base tout à fait honorable.

C'est pourquoi le travail qui suit découle moins d'une réelle analyse de la tâche que nous aurions pu effectuer que d'une concrétisation des propositions évoquées.

Cette importante mise au point faite, nous allons quand même essayer de cerner la tâche que notre outil doit faciliter.

### 1.3.2 Utilisateurs

Nous avons dit précédemment que notre mémoire s'inscrit dans le contexte général de l'exploitation des résultats des simulations du comportement d'un S.I. Plus précisément, il vise à améliorer les rapports d'analyse du comportement simulé. Rappelons



également que ces rapports ont pour but de permettre l'évaluation des performances et leur comparaison avec celles attendues par le concepteur, ainsi que de faciliter la détection d'un comportement anormal ou jugé non satisfaisant.

Ceci nous renseigne sur les utilisateurs potentiels de notre outil : il s'agit des concepteurs de S.I. soucieux de déterminer si leurs spécifications rencontrent bien les exigences de performances attendues.

Selon la taxonomie généralement utilisée [Van91], ces utilisateurs sont experts, voire maîtres dans leur domaine. Notre outil doit donc refléter ce degré de maîtrise de la tâche.

Ce haut degré implique plusieurs points :

- Les utilisateurs connaissent bien les termes et concepts qu'ils auront à manipuler avec notre outil (c'est important, notamment pour le choix du vocabulaire à utiliser dans les menus).
- Les utilisateurs sont plus soucieux d'une interface puissante et efficace qui leur permettent d'accéder rapidement aux informations recherchées en un nombre relativement restreint de commandes.
- L'interface doit être flexible, c'est-à-dire offrir le plus de liberté possible afin de permettre aux utilisateurs non seulement de conserver leurs habitudes de travail, mais aussi de concevoir des rapports qu'ils jugent pertinents. En particulier, on veillera à fournir plusieurs manières d'atteindre un même but, des possibilités de "court-circuits", de personnaliser l'interface, ...
- Dans le même ordre d'idée, il faut particulièrement être attentif à donner l'impression aux utilisateurs qu'ils pilotent réellement le système. Ceux-ci doivent prendre l'initiative des actions. On essaiera donc de ne pas être trop directif, de ne pas offrir des feed-back intempestifs, ...

Il faut bien être attentif à ne pas confondre "expert" et "fréquent". Le premier adjectif concerne le degré de connaissance que possède l'utilisateur de sa tâche, indépendamment de tout outil. Le deuxième adjectif rend compte du taux d'utilisation de l'outil qui doit aider l'utilisateur dans sa tâche. Cette distinction prend un caractère particulièrement important dans le cas qui nous occupe dans la mesure où la tâche d'analyse du comportement d'un S.I. ne constitue pas le travail essentiel d'un concepteur. L'analyse du comportement d'un S.I. n'est qu'une étape d'un processus de conception de longue haleine. Ainsi, le concepteur risque-t-il de ne pas utiliser très fréquemment l'outil proposé si l'on se place sur l'échelle de temps de la conception du S.I., mais, paradoxalement, quand il l'utilisera au moment du processus d'analyse, il le fera fréquemment (l'analyse du comportement d'un S.I. est un travail itératif).

Concrètement, ces considérations nous incitent à minimiser la complexité de l'interface et à faire en sorte qu'elle reflète le mieux possible les processus de pensée de l'utilisateur (réduction du fossé qui sépare l'univers psychologique de la tâche et l'univers physique de l'interface). En effet, un trop grand nombre d'options et un mode d'utilisation de l'interface relativement éloigné des méthodes d'analyse idiosyncrasiques entraîneraient un effort non négligeable de la part de l'utilisateur pour se remémorer le fonctionnement de l'outil chaque fois qu'il entre dans un cycle d'analyse du comportement du S.I.

### 1.3.3 Tâche

Bien que, pour les raisons évoquées précédemment, nous n'avons pas pu ni voulu consacrer le temps nécessaire à analyser la tâche du concepteur, nous souhaitons attirer l'attention du lecteur sur quelques points.

Il faut prendre pleinement conscience de l'importance que revêt l'étape d'analyse de la tâche dans le processus de conception d'une interface homme-machine. En effet, le but de cette discipline informatique étant de permettre de réaliser le mieux possible la tâche que l'utilisateur doit accomplir, il est impératif de cerner le plus précisément possible les caractéristiques de cette dernière. En réalité, le travail de l'ergonome consiste à réduire le fossé qui sépare deux univers : celui, psychologique, dans lequel l'utilisateur pense la tâche, et celui, physique, de l'interface dans lequel il accomplit les actions nécessaires à la réalisation de la tâche. Autrement dit, en vertu du paradigme de l'ergonomie moderne qui, contrairement à la perspective taylorienne, stipule qu'il faut adapter le travail à l'homme et non l'inverse, l'ergonome doit façonner, modeler le système informatisé, en tout cas sa partie visible, pour diminuer les efforts cognitifs que doit inévitablement produire l'utilisateur.

Afin de caractériser la tâche, l'ergonome dispose de quelques modèles, dont celui de D. A. Norman. Ce modèle décompose une tâche à accomplir en sept étapes :

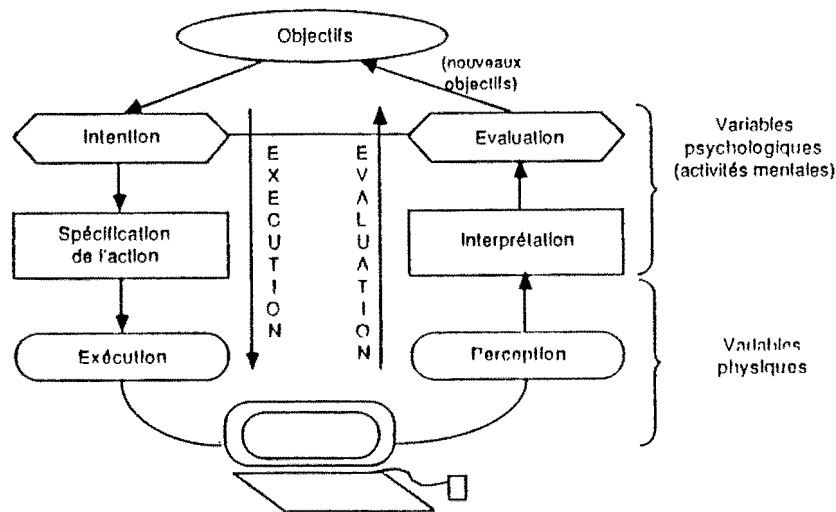


Illustration 5 : Schéma du modèle de la tâche de D. A. Norman.

Ce modèle est intéressant à plus d'un titre. Tout d'abord, il met en évidence les deux temps d'une tâche : l'exécution et l'évaluation. Nous trouvons particulièrement judicieux d'avoir placé le processus d'évaluation sur un pied d'égalité avec celui d'exécution. Cette dichotomie correspond à la conception cybernétique de la communication (école de Palo-Alto) qui met en exergue la notion de feed-back. Ensuite, ce modèle fait ressortir la nécessité de décomposer un objectif en intentions qui sont réalisables en termes d'actions du système informatisé (on retrouve le vieux réflexe des informaticiens). Par conséquent, il propose un moyen pour caractériser et structurer la tâche globale à accomplir. De plus, et c'est là à notre avis le principal intérêt de ce modèle, la décomposition doit se réaliser dans un souci permanent de réduction des deux univers précédemment évoqués.

Malheureusement, comme tout modèle, il est difficile à appliquer par manque de critères opérationnels (ce processus itératif laisse une grande part à l'arbitraire). C'est pourquoi, nous ne l'avons par réellement utilisé, même si nous avons constamment sa philosophie en tête. On peut déplorer cet état de fait, car il ne fait aucun doute qu'une décomposition formelle de la tâche générale en sous-tâches aurait fourni un cadre de référence solide qui aurait garanti non seulement la légitimité de l'interface mais surtout sa complétude. En effet, l'un des reproches fondamentaux que l'on peut adresser à notre interface est qu'elle ne met en œuvre qu'une manière de réaliser la tâche du concepteur, non seulement sans justifier la pertinence de cette vision de la tâche, mais surtout sans avoir vérifié qu'il n'existe pas une meilleure manière de procéder. Nous sommes pleinement conscients de cette faiblesse, mais nous soutenons que dans le cadre d'un

mémoire il est utopique de vouloir mener de front une analyse de la tâche sérieuse et l'élaboration d'une interface. Le lecteur se rendra compte que la concrétisation d'une vision même imparfaite de la tâche est loin d'être un travail négligeable. Il vaut peut-être mieux une bonne concrétisation d'une vision imparfaite de la tâche qu'une concrétisation bâclée d'une très bonne analyse de la tâche.

Il est temps maintenant de lever une ambiguïté qu'entretient le titre de notre mémoire. En effet, l'outil que nous proposons n'est pas à proprement parler un outil d'analyse et d'évaluation des résultats statistiques si on entend par là un logiciel qui, de lui-même, aiderait explicitement l'utilisateur à repérer les comportements pathologiques. Il ne s'agit donc pas d'un système expert qui disposerait d'un ensemble de règles formalisant une méthodologie d'analyse et d'évaluation. Notre outil, plus modeste, n'a pour but que de *faciliter la consultation des statistiques et l'élaboration de rapports* qui sont en quelque sorte des sous-tâches de la tâche globale d'analyse du comportement d'un S.I. Il incombe dès lors à l'utilisateur de mener lui-même la stratégie d'analyse des résultats statistiques la plus appropriée sachant qu'il dispose des matériaux de base pour l'aider dans sa démarche. En toute honnêteté, nous devons reconnaître que notre outil n'est pas indépendant d'une certaine vision du processus d'analyse d'un S.I. En effet, nous basons notre concept de navigation sur le modèle de la dynamique. Il aurait été intéressant de réfléchir sur la pertinence de ce mode de navigation, et éventuellement d'en proposer d'autres (hypertextes).

Idéalement, pour bien comprendre les tâches du concepteur auxquelles nous nous intéressons - consulter des résultats statistiques et élaborer des rapports - il aurait donc fallu définir une méthodologie d'analyse des résultats statistiques, dégager toutes les sous-tâches et les quantifier afin de déterminer leur importance relative, ... Cela représente un travail considérable, suffisant pour en faire un sujet de mémoire à part entière. Il nous a été impossible de consacrer un temps suffisant à acquérir une bonne connaissance des processus d'analyse. C'est d'ailleurs une des raisons pour lesquelles nous avons basé notre travail sur le mémoire déjà cité dont les auteurs ont eu l'occasion de se servir suffisamment de DSL/SIM pour en retirer une expérience intéressante.

Nous nous permettons de reprendre certaines considérations méthodologiques émises par les Professeurs F. Bodart et Y. Pigneur [Bod89]. Ces indications donnent certaines pistes en vue de repérer des comportements anormaux ou non satisfaisants, et par là même éclairent quelque peu la tâche du concepteur.

Les résultats statistiques concernant les processus "doivent pouvoir être exploités pour détecter les *attentes anormalement longues* et les *interruptions* trop fréquentes ou trop longues de certains processus.

Dans un premier temps, le seul examen des nombres caractéristiques de l'évolution des processus permet déjà de s'assurer que les processus déclenchés ont pu être activés et se sont terminés normalement.

Dans un second temps, l'analyse des durées et volumes caractéristiques devrait facilement mettre en évidence des attentes et/ou interruptions trop longues ou trop nombreuses dont il conviendrait de découvrir les causes par une analyse du comportement des ressources utilisées."

En ce qui concerne les ressources, les mesures statistiques produites peuvent servir à "identifier les *ressources critiques* dont la trop forte utilisation ou la trop faible disponibilité expliquent les attentes et/ou interruptions anormalement longues ou fréquentes.

Toutefois, sachant que les performances attendues d'un système pourraient être obtenues au prix d'une sous-utilisation, anti-économique, des ressources, il pourrait également s'avérer judicieux de détecter les ressources sous-utilisées, à l'aide de ces résultats.

En particulier, l'examen des volumes en attente et du taux d'utilisation des ressources requises par des processus aux attentes anormales doit normalement permettre une détection assez aisée des ressources critiques. Pour celles-ci, il ne resterait alors plus qu'à déterminer si ce comportement suspect est dû à la capacité maximale ou au calendrier de disponibilité spécifié."

Enfin, les résultats statistiques concernant les points de synchronisation "sont principalement destinés à localiser les *retards éventuels* dans la réalisation des synchronisations, donc *dans le déclenchement des processus* en aval de celles-ci. La localisation de ces retards correspond en fait à la détection des événements - dits contraignants - qui surviennent tardivement et retardent ainsi le déclenchement de processus.

En effet, l'examen des résultats statistiques concernant les synchronisations devrait permettre très facilement d'identifier les points de synchronisation critiques, ceux dont les durées de réalisation sont anormalement longues. Et pour ceux-ci, l'analyse peut se poursuivre par la détection des contributions contraignantes; sachant qu'un événement contribuant est d'autant plus contraignant qu'il contribue tardivement à la réalisation, c'est-à-dire que sa durée de contribution est courte.

Si cette interprétation des résultats a permis de détecter un type d'événement contraignant, il s'agira de poursuivre l'évaluation du comportement en la centrant principalement sur la partie du schéma de la dynamique en amont de ce type d'événement."

Ces considérations mettent bien en évidence l'importance de la notion de *navigation*. Dans la mesure où le processus d'interprétation des résultats statistiques est un processus non structuré, il est impératif que notre outil offre des facilités de navigation. **La possibilité pour le concepteur de pouvoir accéder à tout moment à n'importe quelle information est donc capitale.**

Ces considérations mettent également l'accent sur le fait que le concepteur doit comparer des mesures statistiques entre elles. Dès lors, **il faut lui offrir des formes de présentation des informations adaptées à sa tâche de comparaison** : le *camembert* et l'*histogramme*. Le *camembert* traduit la relation des parties à un tout et convient particulièrement pour mettre en évidence des données dont la valeur se distingue des autres valeurs présentées. L'*histogramme* traduit une variation de mesures en fonction de certains paramètres et convient bien pour l'étude des variations dans le temps de certaines mesures. Précisons que ces techniques de présentation ne sont pas les seules. Nous aurions pu imaginer d'autres formes de présentation, peut-être moins liées à une vision de la tâche essentiellement basée sur la dynamique des processus.

Enfin, parce que le concepteur doit comparer les mesures simulées à celles estimées, parce qu'il doit communiquer ses observations, **il faut lui offrir la possibilité de créer un rapport qui reflète avec pertinence les mesures jugées significatives.**

Le lecteur attentif a sans doute remarqué que la réalisation de ces trois mots d'ordre apporteraient une solution aux trois critiques que nous avons précédemment émises vis-à-vis de la forme actuelle des rapports (listings) : absence de facilité de navigation, pauvreté des formes de présentation et profusion d'informations non nécessairement pertinentes.

Pour atteindre ce triple objectif, nous allons mettre en œuvre les propositions émises par B. Sacré et J. Lefèvre dans le premier chapitre intitulé "Outils d'interprétation" de leur mémoire. Globalement, ces outils sont [Lef87,35]:

Mesures présentées	Formes de présentation	Type de mesures
Durée moyenne d'attente de repos d'activité	Camembert	Globales
Nombre moyen de processus en attente en activité	Camembert	
Nombre de déclenchements d'activations d'interruptions de terminaisons	Texte	
Capacité moyenne Nombre de points d'entrée utilisés file d'attente moyenne	Camembert Camembert Texte	Globales
Nombre de réalisations de contributions	Texte	Globales
Temps moyen de réalisation de participation	Texte	
Evénements contraignants en fréquence en durée	Camembert Camembert	
Nombre de déclenchements d'activations d'interruptions de terminaisons	Histogramme	Chronologiques
Nombre de déclenchements d'activations d'interruptions de terminaisons	Histogramme	Périodiques
Durée moyenne d'attente	Histogramme	Chronologiques
Durée moyenne d'attente de repos	Histogramme	Périodiques
Nombre moyen de processus en attente en activité	Histogramme	Chronologiques
Capacité moyenne Nombre de points d'entrée utilisés	Histogramme Histogramme	Chronologiques
File d'attente moyenne	Histogramme	Chronologiques
Calendrier	Segments de droites	Calendrier

Nous ne remettons pas en question la pertinence des outils exposés ci-dessus. Toutefois, en vertu de l'importance de la liberté d'action à laisser à l'utilisateur, il nous semble utile de compléter l'éventail des mesures proposées ci-dessus. En effet, bien que nous souscrivons entièrement au choix des mesures à représenter sous forme de camembert ou d'histogramme qui représentent un plus par rapport à la situation existante.

tante, nous estimons que le concepteur doit avoir également à sa disposition toutes les informations qu'il trouvait dans les listings. C'est à lui qu'incombe le soin de savoir si dans le cadre de son analyse de telles informations sont utiles.

Pour terminer, insistons sur le fait que notre outil doit s'articuler autour de deux grandes fonctionnalités : la consultation de résultats statistiques d'une part, et l'élaboration de rapports d'autre part. Si nous attirons votre attention sur cette dichotomie, c'est parce que selon le poids que l'on accorde à chacune de ces parties, on obtient des interfaces sensiblement différentes. Ainsi, dans notre outil nous avons inconsciemment privilégié l'élaboration d'un rapport. Nous verrons comment équilibrer cette dichotomie.

Nous pouvons maintenant passer à la présentation de notre outil de consultation des résultats statistiques et d'élaboration de rapports.

## 2

---

# Présentation de l'outil de consultation de statistiques et d'élaboration de rapports

## 2.1 POUR FACILITER LA LECTURE DE CE CHAPITRE

### 2.1.1 Présentation succincte de l'interface de l'outil

L'interface que nous décrivons dans ce chapitre obéit aux normes et conventions de l'environnement multi-fenêtré Ms-Windows 2. Toutefois, pour pouvoir capturer plus facilement certaines illustrations, nous avons été amenés à exécuter la maquette de notre outil<sup>1</sup> sous l'environnement plus évolué Ms-Windows 3. Par conséquent, que le lecteur ne soit pas surpris si certaines illustrations (par exemple les messages) adoptent un format propre à cet environnement plus récent.

Si le lecteur n'est pas un habitué de l'environnement multi-fenêtré en général, ou de Ms-Windows en particulier, nous lui conseillons de parcourir les guides de l'utilisateur, ou tout au moins la très utile classification des objets interactifs réalisée par I. Provot et J. Vanderdonck [Pro90c].

L'interface se compose d'une fenêtre principale possédant une barre de menus déroulants et une zone de travail. La zone de travail fournit une vue partielle sur le rapport que l'utilisateur élabore. Les menus permettent à l'utilisateur de choisir les objets et les informations à consulter. La visualisation de ces informations s'effectue grâce à l'affichage de fenêtres filles.

---

<sup>1</sup> Pour plus de renseignements sur cette maquette, se référer à l'annexe 5.

A chaque fenêtre fille correspond un seul objet (c'est-à-dire un processus, une ressource, ou un point de synchronisation) et un seul type d'informations (c'est-à-dire se rapportant à des statistiques globales, périodiques ou chronologiques). Nous appellerons désormais *fiches* les fenêtres filles dans la mesure où ces dernières constituent une unité logique de présentation.

L'utilisateur peut choisir entre afficher une fiche standard, c'est-à-dire une fiche présentant un certain nombre d'informations textuelles et graphiques prédéfinies, et afficher une fiche personnalisée, c'est-à-dire une fiche présentant des informations textuelles ou graphiques sélectionnées explicitement. A tout moment, l'utilisateur peut ajouter, supprimer une ou plusieurs informations de la fiche courante. Il peut redéfinir le standard des fiches. Il a bien sûr toute liberté d'afficher plusieurs fiches, de les déplacer (à l'intérieur de la zone définie par la fenêtre principale), de les masquer, de les redimensionner (une fonction est prévue pour réorganiser l'affichage des fiches de manière optimale).

Si l'utilisateur souhaite garder une trace matérielle des informations les plus significatives, il peut les transférer sur la zone de travail de la fenêtre principale grâce au mécanisme habituel du copier/coller. Il peut les redimensionner, les déplacer (une règle est prévue pour aider l'utilisateur à positionner les dessins de manière précise). Ainsi, il compose en toute liberté un rapport qu'il peut enregistrer et imprimer. Bien entendu, il a la possibilité de créer simultanément plusieurs rapports.

## 2.1.2 Conventions typographiques

Les noms de menus et d'items de menus sont mis en caractères gras.

Les noms de touches du clavier, de contrôles de boîtes de dialogue (cases à cocher, boutons, ...), ainsi que les noms de documents sont mis en caractères italiques. Les définitions de certains termes sont également présentées en italique.

## 2.1.3 Concepts

Précisons quelques termes de vocabulaire que nous utilisons abondamment dans la suite du chapitre. Ces termes correspondent aux principaux concepts autour desquels s'articule notre interface.

*Courant* : Un document est dit courant quand son support (fenêtre fille, zone de travail de la fenêtre principale) est le support actif. Un rapport est dit courant quand il est affiché dans la zone de travail de la fenêtre principale. Il est donc possible que le rapport courant ne soit pas le document courant. Par contre, une fiche courante est toujours le document courant.

Concrètement, le document courant se distingue des autres par la coche qui précède son nom dans la liste des documents du menu **Ecran**. Le rapport courant se distingue des autres rapports par le fait que son nom n'est pas mis entre parenthèses dans la liste des rapports du menu **Ecran**.

*Document* : Terme utilisé pour désigner indistinctement une fiche ou un rapport.



- Fiche :** Ensemble d'informations concernant un type de statistiques (globales, périodiques, chronologiques) associées à un objet (de type processus, ressource, ou point de synchronisation) présentes à un moment donné dans une fenêtre fille de l'application. La fiche est l'outil de consultation.
- Fiches sœurs :** Fiches se rapportant à un même objet (par exemple, à un même processus). A tout processus, et à toute ressource peuvent correspondre trois fiches sœurs : une fiche pour chaque type de statistiques. Un point de synchronisation ne possède qu'une seule fiche (statistiques globales).
- MEANDRES :** Module d'Evaluation et d'ANalyse Des REsultats Statistiques - Nom de l'outil décrit dans le présent chapitre.
- Rapport :** Ensemble hétérogène d'informations présentes à un moment donné dans la zone de travail de la fenêtre principale de l'application. Ces informations sont celles que l'utilisateur décide de sauvegarder sous forme de fichier, ou de document imprimé du fait de leur importance dans sa tâche d'analyse.

#### 2.1.4 Structure du chapitre

La structure de ce chapitre est assez simple : nous allons commencer par décrire les différentes fenêtres de l'application; ensuite, nous allons décrire les items des différents menus, menu par menu.

Pour décrire les items des menus, nous avons adopté la structure suivante :

- Type :**
- *Commande* : l'item exécute immédiatement une action.
  - *Commande étendue* : l'item présente une boîte de dialogue avant d'exécuter l'action ainsi précisée.
  - *Descripteur d'état* : l'item détermine un état du système. Il fonctionne comme un interrupteur on/off et est précédé par une coche (✓) si il est activé.
- Mnémonique :** Caractère de l'item qui permet à l'utilisateur de sélectionner l'item en pressant la touche correspondante.
- Accélérateur :** Combinaison de touches qui permet à l'utilisateur de sélectionner directement (c'est-à-dire sans dérouler le menu) l'item associé.
- Action :** Brève description de la fonctionnalité de l'item.
- Boîte de dialogue :** Description de la boîte de dialogue associée à l'item si celui-ci est de type "commande étendue". Rien sinon.
- Effets :** Description précise des effets de la sélection de l'item. Explication de l'utilisation de la boîte de dialogue.
- Messages :** Description de la transformation éventuelle du curseur et description des messages engendrés par la sélection de l'item et/ou par le choix des options de l'éventuelle boîte de dialogue.

### **2.1.5 Hypothèse**

Dans notre mémoire, nous faisons l'hypothèse simplificatrice que tous les résultats statistiques issus d'une exécution de DSL/SIM sont contenus dans un et un seul fichier qui sert d'input à notre outil. La réalité est en fait toute autre : les informations nécessaires à notre outil sont éparpillées dans plusieurs fichiers, en vertu du principe de non-redondance.

Nous pensons que notre hypothèse est plus qu'un simple artifice destiné à faciliter notre travail. En effet, dans la mesure où une bonne interface doit proscrire tout ce qui est étranger à la tâche à réaliser, il est impensable d'imposer à l'utilisateur d'avoir à gérer explicitement plusieurs fichiers d'entrée dont la diversité n'est aucunement liée à la tâche qu'il doit effectuer. Imaginez l'effort que représente l'obligation d'ouvrir plusieurs fichiers : quels fichiers doit-on ouvrir ? Sont-ils tous dans le bon répertoire ? Que se passe-t-il s'il en manque un ? Sont-ils cohérents entre eux? ...

Il faudrait donc qu'en amont, on modifie DSL/SIM pour regrouper toutes les informations dans un seul fichier, ou qu'on crée un utilitaire, sorte d'interface entre DSL/SIM et MEANDRES, qui réalise ce travail. Cette dernière solution ne fait qu'ajouter une étape à un processus de travail déjà bien lourd. Nous pensons qu'une refonte de DSL/SIM, surtout au niveau des dialogues, est nécessaire. Ce serait l'occasion d'intégrer notre outil qui présente une interface complètement différente du mode de dialogue de DSL/SIM.

## 2.2 FENETRES

### 2.2.1 Fenêtre principale

#### Description

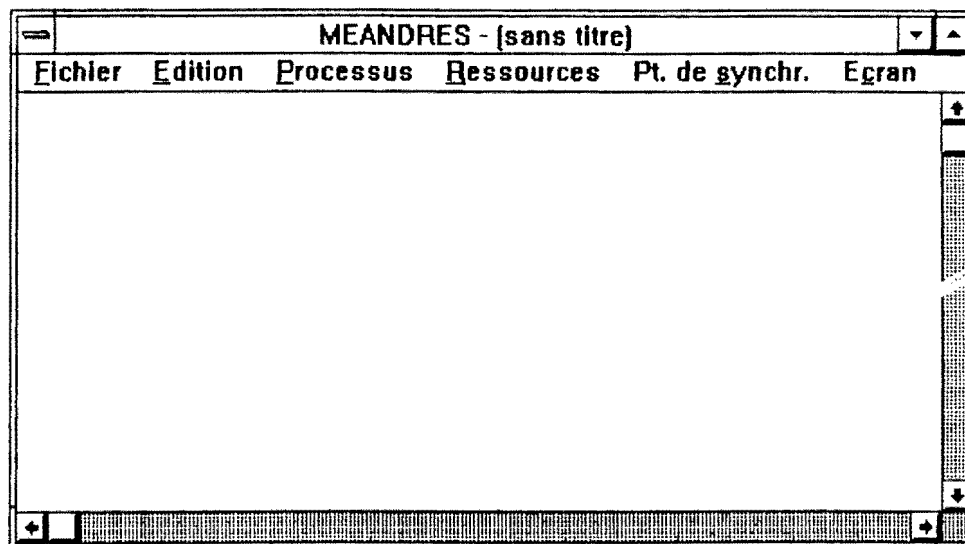


Illustration 6 : Fenêtre principale<sup>1</sup>

Il s'agit d'une fenêtre traditionnelle. En effet, elle est composée de :

- Une barre de titre horizontale au sommet. Cette barre de titre a pour fonction d'identifier la fenêtre. Le titre est formé par la concaténation du nom de l'application et du nom de la simulation dont on exploite les résultats statistiques : *MEANDRES - nom-de-la-simulation*<sup>2</sup>.
- Une case de menu système, située en haut à gauche de la fenêtre. Ce menu abrite les commandes à la gestion de la fenêtre en tant qu'objet : **Taille précédente**, **Déplacement**, **Dimension**,  **Icône**, **Plein écran**, **Fermer**. Ces menus permettent de suppléer à l'absence éventuelle de souris.
- Une case **Plein Ecran**, située en haut à droite de la fenêtre.
- Une case **Icône**, située à gauche de la case **Plein Ecran**.



Illustration 7 : Icône de l'application MEANDRES

<sup>1</sup> Attention ! L'illustration n'est pas totalement fidèle : la zone d'information n'est pas représentée.

<sup>2</sup> Le nom d'une simulation n'est pas forcément un identifiant. Le bon sens recommande toutefois que l'utilisateur donne des noms différents aux simulations réalisées. On peut malheureusement imaginer le cas de deux fenêtres correspondant à deux simulations différentes mais portant le même nom. C'est à l'utilisateur qu'incombe le choix de se faciliter le travail, ou non !

- Une barre de menus horizontale positionnée juste au dessous de la barre de titre. Les menus sont : **Fichier, Edition, Processus, Ressources, Pt. de synchr., Ecran.**
- Une zone de travail qui permet de visualiser le rapport courant. La zone de travail correspond à la partie imprimable des rapports, c'est-à-dire que son extrémité gauche correspond à la marge gauche des rapport, et celle de droite à la marge droite.
- Une barre de défilement vertical à droite de la zone de travail.
- Une barre de défilement horizontal en bas de la zone de travail.
- Une zone d'information, située en bas à gauche de la zone de travail. Cette zone d'information indique le numéro de la page courante du rapport ou les dimensions du dessin sélectionné sur le rapport.

### Causes d'affichage

On distingue quatre événements qui occasionnent l'affichage de la fenêtre principale :

#### Demande d'exécution du programme

Quand l'utilisateur demande à Windows d'exécuter le programme MEANDRES (par exemple, en cliquant deux fois sur le nom de l'application au niveau de l'Exécutif MS-DOS), celui-ci affiche la fenêtre principale intitulée *MEANDRES - (sans titre)*. La zone de travail contient le rapport vide *Rapport 1*. La zone d'information contient le numéro de la page courante (c'est-à-dire *Page 1*).

#### Demande d'exécution du programme à partir d'un rapport

Quand l'utilisateur demande à Windows d'exécuter le programme MEANDRES à partir d'un rapport (par exemple, en cliquant deux fois sur le nom du rapport au niveau de l'Exécutif MS-DOS), celui-ci affiche la fenêtre principale intitulée *MEANDRES - nom\_de\_la\_simulation\_correspondant\_au\_rapport*. La zone de travail permet de visualiser le début du rapport, et la zone d'information indique le numéro de la page courante (*Page 1*).

#### Demande d'exploitation d'une autre simulation

Si l'utilisateur souhaite exploiter les résultats d'une autre simulation que celle en cours, il a la possibilité d'ouvrir une nouvelle fenêtre principale (grâce à la commande **Ouvrir**). Cette nouvelle fenêtre porte le titre *MEANDRES - nom\_de\_la\_nouvelle\_simulation*. Sa zone de travail contient le rapport vide *Rapport 1*. La zone d'information contient le numéro de la page courante (c'est-à-dire *Page 1*).

#### Demande de consultation d'un rapport correspondant à une autre simulation

L'utilisateur peut consulter un rapport correspondant à une autre simulation (grâce à la commande **Ouvrir**). Dans ce cas, le rapport sélectionné sera affiché dans la zone de travail d'une nouvelle fenêtre principale intitulée *MEANDRES - nom\_de\_la\_nouvelle\_simulation\_correspondant\_au\_rapport*. La zone d'information indique le numéro de la page courante (*Page 1*).

## 2.2.2 Fiche

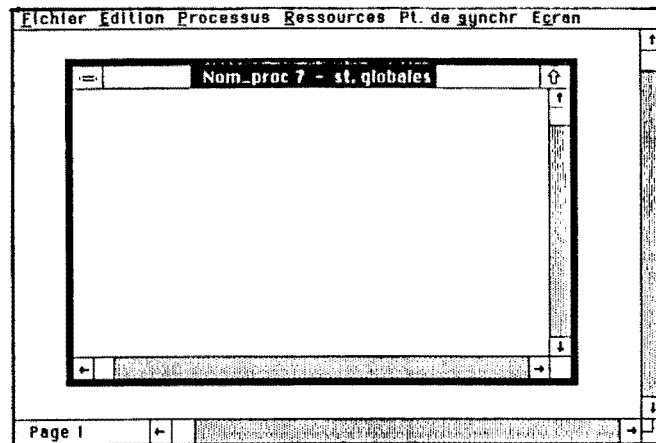


Illustration 8 : Fiche à l'intérieur de la fenêtre principale

Nous avons vu qu'une fiche est une fenêtre fille. Elle comprend :

- Une barre de titre horizontale au sommet. Cette barre de titre a pour fonction d'identifier la fenêtre. Le titre est formé par la concaténation du nom de l'objet et du type de mesures statistiques qui lui sont associés : *nom\_de\_l'objet - type\_de\_statistiques*.
- une case de menu système, située en haut à gauche de la fenêtre. Ce menu abrite les commandes propres à la fenêtre.
- une case **Plein Ecran**, située en haut à droite de la fenêtre.
- une zone de travail qui permet de visualiser les différentes informations statistiques. Ces informations apparaissent selon un ordre prédéterminé. Il n'est pas possible de modifier cet ordre. Nous ne détaillerons pas dans ce mémoire l'ordre d'affichage des informations.
- une barre de défilement vertical à droite de la zone de travail.
- une barre de défilement horizontal en bas de la zone de travail.

On peut déplacer une fiche à l'intérieur de la surface définie par la zone de travail de la fenêtre principale de l'application, mais en aucun cas elle ne peut sortir de cette zone.

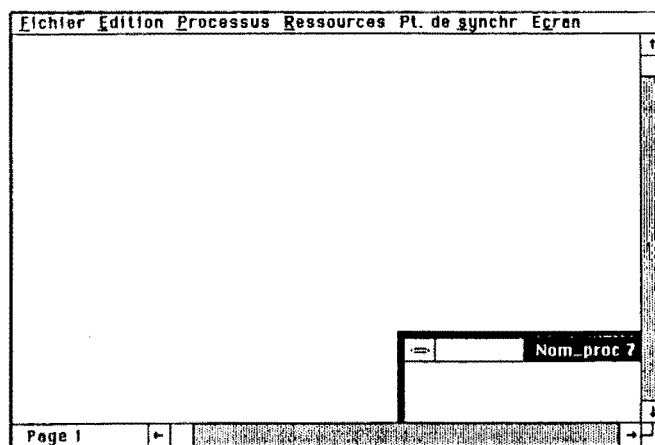


Illustration 9 : Déplacement d'une fiche

## 2.3 MENU FICHIER

Comme toutes les applications qui exploitent des fichiers de données, MEANDRES possède un menu **Fichier**. Ce menu fournit toutes les commandes nécessaires pour ouvrir, créer, sauver, imprimer des fichiers, ... bref, toutes les commandes pour gérer les fichiers en tant que tels.

Le menu **Fichier** comporte dix items.

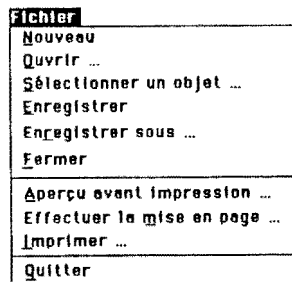


Illustration 10 : Menu **Fichier**

### 2.3.1 Nouveau

Type : Commande.

Mnémo. : N

Accél. : Aucun.

Actif : Toujours.

Action : Permettre à l'utilisateur d'élaborer un nouveau rapport. Contrairement aux normes de Ms-Windows, cette commande ne provoque pas de réinitialisation de l'application (entre autres, le rapport précédent n'est pas perdu et peut toujours être affiché dans la zone de travail).

Effets :

- Rendre la zone de travail vierge.
- Afficher *Page 1* dans la zone d'information.
- Mettre le nom du rapport précédent entre parenthèses dans la liste des rapports du menu **Ecran**.
- Insérer dans la liste des rapports du menu **Ecran** le nom par défaut du nouveau rapport précédé d'une coche : *Rapport n* où n représente le numéro d'ordre de création de ce rapport. Ainsi, si l'utilisateur a demandé l'exécution de MEANDRES à partir d'un rapport, le nom par défaut du nouveau rapport est *Rapport 1*. Par contre, s'il a demandé l'exécution pure et simple de MEANDRES, nous savons qu'un rapport vierge est présent dans la zone de travail sous le nom de *Rapport 1*; dans ce cas, l'utilisation de la commande **Nouveau** créera le rapport *Rapport 2* ...

A priori, l'utilisateur peut créer autant de rapports qu'il le souhaite.

Messages : Aucun.

### 2.3.2 Ouvrir

Type : Commande étendue.

Mnémo. : O

Accél. : Aucun.

Actif : Toujours.

Action : Permettre à l'utilisateur d'ouvrir les fichiers contenant les informations nécessaires à l'exploitation des résultats statistiques (suffixe SIM) ainsi que les fichiers contenant un rapport (suffixe RAP).

Remarque importante : A l'heure actuelle, nous faisons l'hypothèse que les données concernant une simulation se trouvent dans un seul fichier. Un deuxième fichier contient la description du schéma de la dynamique associé à la simulation. Le premier fichier doit impérativement exister, le deuxième peut être absent.

Boîte de dialogue : Il s'agit de la boîte de dialogue modale<sup>1</sup> d'ouverture classique. Elle comprend :

- Un champ d'édition uni-linéaire permettant de spécifier un nom de fichier éventuellement précédé d'un chemin d'accès.
- Une liste de sélection de fichiers et de répertoires.
- Un bouton de commande *Ok*.
- Un bouton de commande *Annuler*.

Par défaut, le champ d'édition contient l'indication suivante *\*.RAP*; la liste de sélection contient la liste des fichiers de suffixe RAP présents dans le répertoire courant, ainsi que la liste des répertoires et volumes accessibles.

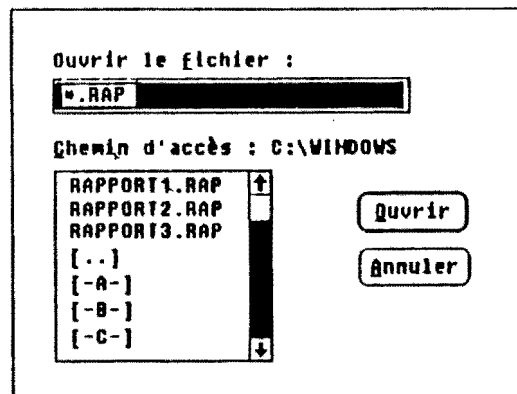


Illustration 11 : Boîte de dialogue de la commande **Ouvrir**

<sup>1</sup> Une *boîte de dialogue modale* est une "boîte de dialogue d'un type particulier exigeant une réponse immédiate de la part de l'utilisateur. Cela signifie qu'il est incapable de faire quoi que ce soit d'autre tant qu'il n'a pas introduit les informations demandées par la boîte de dialogue. Presser le bouton de la souris hors de la boîte de dialogue aura pour seul effet de produire un son indiquant le désaccord de la machine"[Pro90c,53].

Effets :

1) Ouverture d'une simulation

L'utilisateur doit sélectionner le fichier source. MEANDRES procède à la lecture des données et à la modification de la fenêtre principale.

Si la fenêtre principale n'est encore associée à aucune simulation (cela se produit quand l'utilisateur demande l'exécution de MEANDRES), alors son titre change : *MEANDRES - nom de la simulation*; la zone de travail et la zone d'information restent inchangées (respectivement vierge, et contenant *Page 1*). Sinon, il y a affichage d'une nouvelle fenêtre principale intitulée *MEANDRES - nom de la nouvelle simulation* (cfr. section concernant la fenêtre principale).

2) Ouverture d'un rapport

L'utilisateur doit sélectionner un fichier contenant un rapport.

Si aucune simulation n'est encore associée à la fenêtre principale (*MEANDRES - (sans titre)*) :

- Changement du titre de la fenêtre principale : *MEANDRES - nom de la simulation correspondant au rapport*.
- Affichage du début du rapport dans la zone de travail.
- Affichage de *Page 1* dans la zone d'information.
- Apparition du nom du rapport, précédé d'une coche, dans la liste des rapports du menu **Ecran**.
- Suppression de *Rapport 1* de la liste des rapports du menu **Ecran**.

Si une simulation est déjà associée à la fenêtre et si le rapport correspond à cette simulation :

- Affichage du début du rapport dans la zone de travail.
- Affichage de *Page 1* dans la zone d'information.
- Apparition du nom du rapport, précédé d'une coche, dans la liste des rapports du menu **Ecran**.
- Mise entre parenthèses du nom du rapport précédent dans la liste des rapports du menu **Ecran** ou suppression de *Rapport 1* de cette liste si aucun rapport n'était affiché dans la zone de travail.

Si une simulation est déjà associée à la fenêtre et si le rapport ne correspond pas à cette simulation :

- Affichage d'une nouvelle fenêtre principale intitulée *MEANDRES - nom de la nouvelle simulation correspondant au rapport*.
- Affichage du début du rapport dans la zone de travail de cette nouvelle fenêtre.
- Affichage de *Page 1* dans la zone d'information de cette nouvelle fenêtre.
- Apparition du nom du rapport, précédé d'une coche, dans la liste des rapports du menu **Ecran** de cette nouvelle fenêtre.

Messages : Transformation du curseur en sablier durant toute la durée de la lecture des différents fichiers.

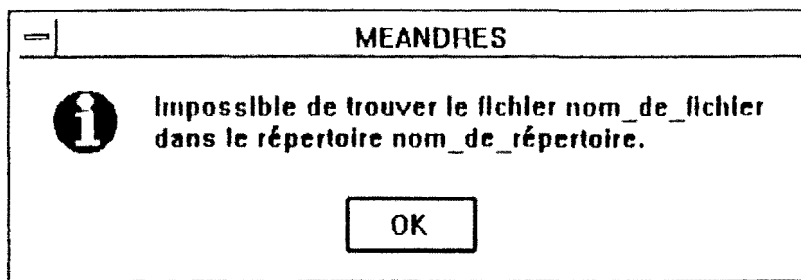
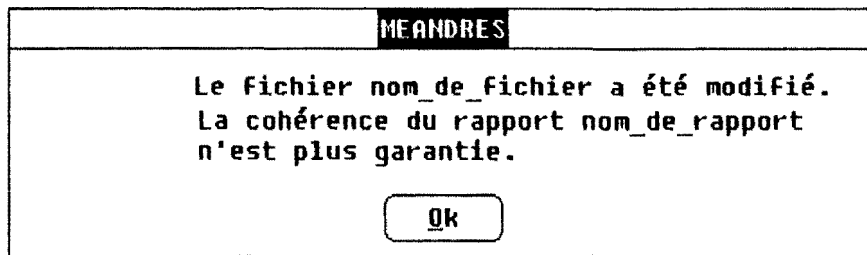
Si l'utilisateur ouvre un rapport, deux conditions doivent être satisfaites pour garantir l'intégrité du lien unissant les rapports aux données d'une simulation :

- Le fichier des données d'une simulation ne peut pas avoir été modifié (en tout cas, pas sous le même nom).

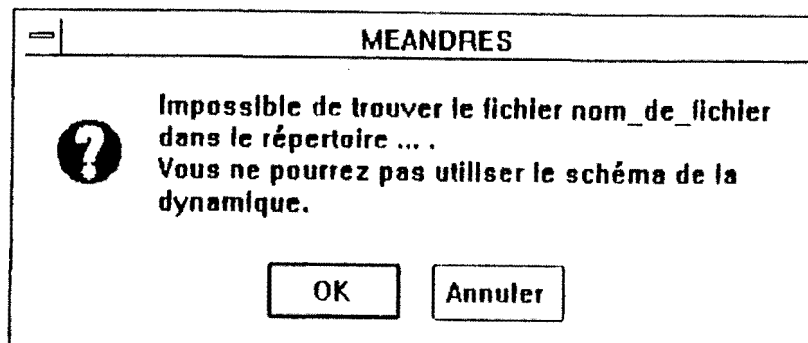


- Le fichier des données d'une simulation doit exister, et il doit se trouver dans le même répertoire que celui des fichiers des rapports.

Le non-respect de ces conditions entraîne l'affichage des messages suivants :



Si le fichier contenant la description du schéma de la dynamique est absent :



### 2.3.3 Sélectionner un objet

Type : Commande étendue.

Mnémo. : S

Accél. : Aucun.

Actif : Dès qu'une simulation a été associée à la fenêtre principale.

Action : Permettre à l'utilisateur de sélectionner un objet (processus, ressource, point de synchronisation) et des mesures statistiques associées. Autrement dit, permettre à l'utilisateur d'afficher une nouvelle fiche.

- Boîte de dialogue :
- Elle est composée de :
- Trois boutons de commande (*Processus*, *Ressources*, *Pt. de synchr.*) qui affichent la liste des objets correspondants.
  - Trois listes de sélection simple qui permettent de sélectionner un objet. Les items des listes sont triés par ordre alphabétique.
  - Trois boutons de commande (*Globales*, *Périodiques*, *Chronologiques*) qui affichent la boîte de dialogue permettant de choisir les informations à consulter. Ces boîtes de dialogues seront décrites plus tard.
  - Un bouton de commande *Schéma* qui affiche le schéma de la dynamique. Ce bouton est grisé si le fichier contenant la description du schéma est indisponible.
  - Un bouton de commande *Annuler*

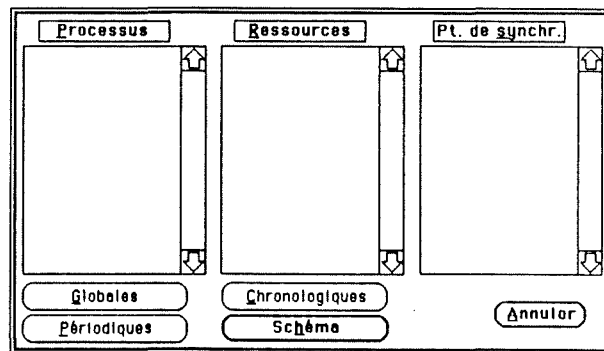


Illustration 12 : Boîte de dialogue de la commande **Sélectionner un objet**

Effets : Les effets de cette commande qui permet de sélectionner les informations à consulter, seront décrits en détails lorsque nous examinerons les menus **Processus**, **Ressources** et **Pt. de synchr.**

Contentons-nous, pour l'instant, de décrire, la manière d'utiliser la boîte de dialogue spécifique à cette commande, c'est-à-dire la boîte de dialogue qui permet de choisir l'objet dont on souhaite consulter les résultats statistiques.

Il existe deux façons de sélectionner un objet :

1) Utiliser les listes de sélection

Quand l'utilisateur choisit la commande **Sélectionner un objet**, les différentes listes de sélection sont vides. En cliquant sur un des boutons *Processus*, *Ressources* ou *Pt. de synchr.*, l'utilisateur fait afficher dans la liste de sélection correspondante l'ensemble des objets de ce type et vide les deux autres listes. Il ne lui reste plus qu'à naviguer à travers la liste et à sélectionner un objet particulier. Signalons que la sélection d'un processus fait apparaître la liste des ressources (directes ou non) que requiert ce processus. Symétriquement, la sélection d'une ressource fait apparaître la liste des processus qui requièrent cette ressource. Naturellement, l'utilisateur peut profiter de cette facilité pour sélectionner un des objets qui viennent d'être affichés dans l'autre liste, plutôt que de cliquer sur le bouton de commande correspondant au type de cet objet. Un seul objet, toutes listes confondues, peut être sélectionné à la fois.

## 2) Utiliser le schéma de la dynamique

Quand l'utilisateur clique sur le bouton de commande *Schéma*, une fenêtre fille (modale) apparaît. Cette fenêtre affiche le schéma de la dynamique associé à la simulation. Si l'utilisateur avait sélectionné un processus ou un point de synchronisation dans la boîte de dialogue, alors le schéma est centré sur cet objet. L'utilisateur n'a plus qu'à naviguer (scrolling) et à sélectionner un processus ou un point de synchronisation. Idéalement, les objets du schéma pour lesquels on ne dispose pas de mesures statistiques devraient être grisés, à l'exception toutefois des conditions d'enchaînement dont on pourrait visualiser les probabilités. Après avoir sélectionné un objet, la fenêtre s'efface et les listes de sélection de la boîte de dialogue reflètent le choix opéré.

Une fois un objet sélectionné, l'utilisateur clique sur le bouton de commande qui lui permet de choisir, par l'intermédiaire d'une nouvelle boîte de dialogue, les informations à consulter.

### Remarques :

- L'utilisateur peut choisir la commande **Sélectionner un objet** pour activer une fiche qui est déjà présente (son nom est repris dans la liste des documents du menu **Ecran**). Dans ce cas, la boîte de dialogue de sélection des informations de la fiche reflète, bien entendu, l'état de cette fiche. Nous verrons qu'il y a d'autres méthodes plus rapides pour activer une fiche qui est déjà présente.
- L'utilisateur peut également sélectionner cette commande pour afficher une nouvelle fiche qui possède une fiche sœur déjà présente. De nouveau, il existe d'autres méthodes plus efficaces pour afficher une telle fiche.
- Pour une illustration du fonctionnement de la boîte de dialogue, nous renvoyons le lecteur en annexe.

Messages : Aucun.

## 2.3.4 Enregistrer

Type : Commande.

Mnémo. : E

Accél. : Aucun.

Actif : Dès qu'une simulation a été associée à la fenêtre principale.

Action : Sauver le rapport courant en l'enregistrant sur un support magnétique. On utilisera cette commande de préférence pour sauver les modifications apportées au rapport depuis son dernier sauvetage.

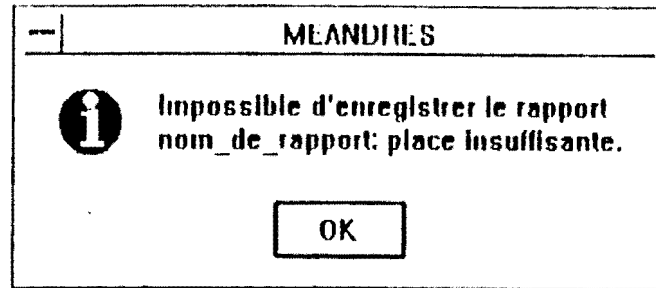
Boîte de dialogue : Si le rapport courant n'a pas encore été sauvé, alors la commande affiche la boîte de dialogue que nous décrivons dans le paragraphe consacré à la commande **Enregistrer sous**.

**Effets :** Si le rapport courant n'a pas encore été sauvé (il porte toujours le nom par défaut *Rapport n*), alors cette commande se comporte exactement comme la commande **Enregistrer sous**. Sinon, cette commande a pour effet de réécrire directement le rapport dans le fichier dont le nom et l'emplacement ont été spécifiés lors du tout premier sauvetage.

**Remarque :** Il n'est pas obligatoire que le rapport courant soit le document courant.

**Messages :** Transformation du curseur en sablier durant toute la durée de l'enregistrement.

S'il n'y a plus suffisamment de place sur le support magnétique :



### 2.3.5 Enregistrer sous

**Type :** Commande étendue.

**Mnémo. :** R

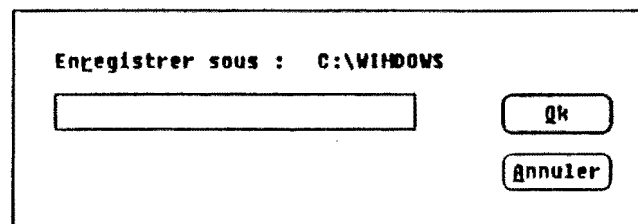
**Accél. :** Aucun.

**Actif :** Dès que la fenêtre principale est associée à une simulation.

**Action :** Sauver le rapport courant en l'enregistrant sur un support magnétique. On utilisera cette commande de préférence lors du premier sauvetage, ou pour produire une copie de l'enregistrement du rapport.

**Boîte de dialogue :** Il s'agit de la boîte de dialogue modale d'enregistrement classique. Elle comprend :

- Un champ d'édition uni-linéaire permettant de spécifier un nom de fichier éventuellement précédé d'un chemin d'accès. Par défaut, ce champ est vide.
- Un bouton de commande *Ok*.
- Un bouton de commande *Annuler*.



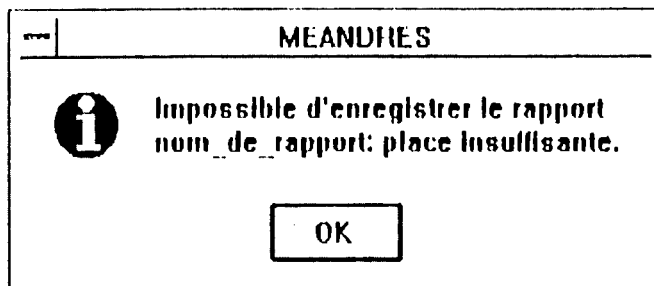
**Illustration 13 :** Boîte de dialogue de la commande **Enregistrer sous**

Effets : Cette commande a pour effet d'écrire le rapport courant dans le fichier dont le nom et l'emplacement ont été spécifiés dans la boîte de dialogue.

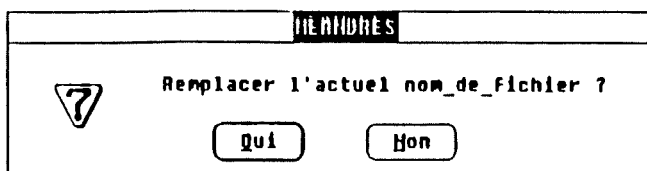
Remarque : Il n'est pas obligatoire que le rapport courant soit le document courant.

Messages : Transformation du curseur en sablier durant toute la durée de l'enregistrement.

S'il n'y a plus suffisamment de place sur le support magnétique :



Si le fichier existe déjà :



### 2.3.6 Fermer

Type : Commande.

Mnémo. : F

Accél. : Aucun.

Actif : Dès qu'une simulation a été associée à la fenêtre principale.

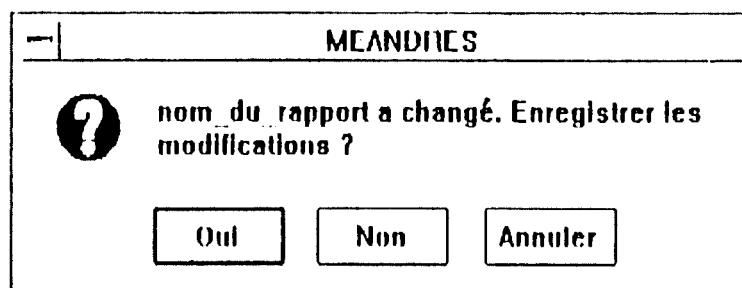
Action : Fermer la fenêtre du document courant.

Effets : 1) Si le document courant est une fiche, alors cette commande a pour effet de fermer sa fenêtre. Cette fiche n'est plus disponible via la liste des documents du menu **Ecran**. Si l'utilisateur souhaite la refaire apparaître, il devra utiliser la commande **Sélectionner un objet**.

2) Si le document courant est un rapport, alors cette commande a pour effet "d'effacer sa fenêtre", c'est-à-dire de fermer le fichier correspondant. Si le rapport n'a jamais été enregistré, ou s'il a été modifié depuis son dernier enregistrement, un message avertira l'utilisateur de la situation. Si le rapport fermé est le seul rapport créé ou ouvert, alors MEANDRES crée automatiquement un nouveau rapport (*Rapport n*). Sinon, le rapport affiché dans la zone de travail de la fenêtre principale est le rapport qui suit, dans la liste des documents du menu **Ecran**, le rapport fermé.

Dans ces deux cas, la liste des documents du menu **Ecran** est mise à jour (cfr menu **Ecran**).

Messages : Si le document courant est un rapport qui n'a jamais été enregistré, ou qui a été modifié depuis son dernier enregistrement :



### 2.3.7 Aperçu avant impression

Type : Commande étendue.

Mnémo. : A

Accél. : Aucun.

Actif : Dès que la fenêtre principale est associée à une simulation.

Action : Offrir une vision globale des pages du rapport courant telles qu'elles seront imprimées. Tous les éléments de la page situés dans la zone imprimable de la page apparaissent dans ce mode.

Boîte de dialogue : La boîte de dialogue est de type modale. Elle comprend :

- Une zone graphique munie d'une barre de scrolling verticale présentant une réduction de la page sélectionnée. On retrouve dans cette réduction non seulement les informations collées dans la zone de travail de la fenêtre principale, mais aussi l'en-tête et le pied de page.
- Une indication du numéro de la page visualisée.
- Un bouton de commande *Annuler*.

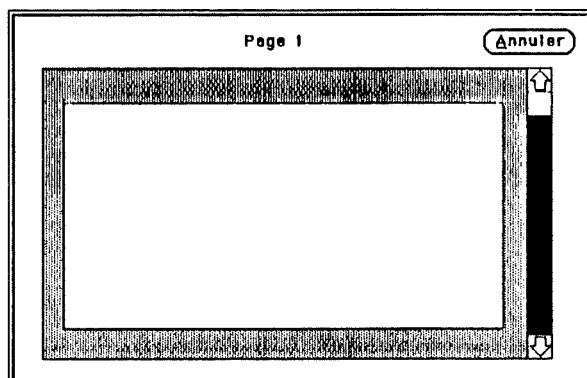


Illustration 14 : Boîte de dialogue de la commande **Aperçu avant impression**

**Effets :** Quand l'utilisateur sélectionne la commande **Aperçu avant impression**, la zone graphique de la boîte de dialogue affiche la réduction de la page courante, c'est-à-dire de la page dont le numéro est spécifié dans la zone d'information de la fenêtre principale. L'utilisateur peut choisir de visualiser une autre page en se servant de la barre de scrolling verticale.

Quand l'utilisateur clique sur le bouton *Annuler*, la page courante de la zone de travail de la fenêtre principale correspondra à la page présente dans la zone graphique au moment de la fermeture de la boîte de dialogue.

Remarque : L'illustration ci-dessus représente une page en mode paysage (cfr ci-après). Si le rapport est en mode portrait, la zone graphique autorise la visualisation de deux pages à la fois (la page courante et la page suivante).

**Messages :** Aucun.

### 2.3.8 Effectuer la mise en page

**Type :** Commande étendue.

**Mnémo. :** M

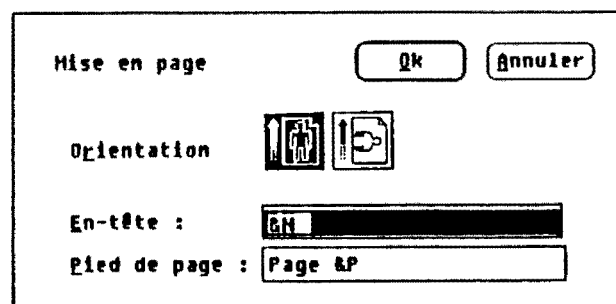
**Accél. :** Aucun.

**Actif :** Dès qu'une simulation a été associée à la fenêtre principale.

**Action :** Permettre de fixer les paramètres de mise en page du rapport courant : orientation des feuilles, en-tête, pied de page.

**Boîte de dialogue :** La boîte de dialogue modale comprend :

- Deux boutons graphiques représentant les deux modes d'orientation des feuilles : mode portrait (impression des lignes sur la largeur de la feuille) et mode paysage (impression des lignes sur la longueur de la feuille).
- Un champ d'édition uni-linéaire permettant de spécifier le texte et la présentation de l'en-tête.
- Un champ d'édition uni-linéaire permettant de spécifier le texte et la présentation du pied de page.
- Un bouton de commande *Ok*.
- Un bouton de commande *Annuler*.



**Illustration 15:** Boîte de dialogue de la commande **Effectuer la mise en page**

Effets : Nous insistons sur le fait que les différents paramètres spécifiés à l'aide de la boîte de dialogue ne sont valables que pour le rapport courant.

Les effets de cette commande sont suffisamment explicites pour se passer d'une description complète. Décrivons plutôt la manière de spécifier le contenu de l'en-tête et du pied de page.

Pour utiliser l'en-tête ou le pied de page, il suffit d'entrer le texte souhaité dans le champ correspondant. L'utilisateur a la possibilité de demander de justifier des parties de l'en-tête ou du pied de page à gauche, à droite ou au centre; d'inclure le numéro de la page, la date, l'heure de l'impression et le nom du rapport. Pour cela, il doit utiliser un certain nombre de codes :

- &G Aligner à gauche les caractères suivants.
- &C Centrer les caractères suivants.
- &D Aligner à droite les caractères suivants.
- &P Imprimer le numéro de la page.
- &H Imprimer l'heure en cours.
- &J Imprimer la date en cours.
- &N Imprimer le nom du rapport (sans le suffixe RAP).

Ainsi, par exemple, *&GRapport : &N&CPage &P&D&J* produira

<i>Rapport : nom du rapport</i>	<i>Page 1</i>	<i>16/1/91</i>
---------------------------------	---------------	----------------

Par défaut, le texte est centré, le champ d'édition de l'en-tête contient *&N*, et le champ d'édition du pied de page *Page &P*. Les codes peuvent être indistinctement entrés en majuscule ou en minuscule.

#### Remarques :

- Il n'est pas possible de spécifier une numérotation particulière : les pages seront toujours numérotées à partir de 1.
- Les marges ont une grandeur prédéterminée et non paramétrisable.
- Le texte de l'en-tête (ou du pied de page) tiendra sur une seule ligne. Par conséquent, l'utilisateur doit faire attention de ne pas entrer des textes trop longs afin d'éviter des recouvrements ou des troncatures.

Messages : Aucun.

### 2.3.9 Imprimer

Type : Commande étendue.

Mnémo. : I

Accél. : Aucun

Actif : Dès que la fenêtre principale est associée à une simulation.

Action : Imprimer le rapport courant.

Boîte de dialogue : Il s'agit d'une boîte de dialogue modale extensible. Elle comprend :

- Un champ d'édition uni-linéaire permettant de spécifier le nombre de copies à imprimer.
- Deux boutons radios permettant de choisir entre imprimer tout le rapport ou seulement une partie.



- Deux champ d'édition uni-linéaire permettant de spécifier, le cas échéant, les bornes d'impression.
- Un bouton de commande *Ok*.
- Un bouton de commande *Annuler*.

Imprimer

Copies: 1

Nombre de pages:

Toutes

De: A:

Ok

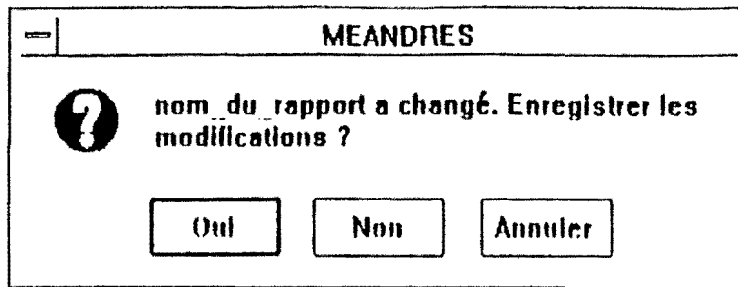
Annuler

Illustration 16 : Boîte de dialogue de la commande *Imprimer*.

- Effets : Le lecteur est habitué de rencontrer cette boîte de dialogue. Aussi estimons-nous inutile de décrire ses effets.
- Messages : Les messages sont gérés par le gestionnaire d'impression intégré à Ms-Windows.

### 2.3.10 Quitter

- Type : Commande.
- Mnémo. : Q
- Accél. : Aucun.
- Actif : Toujours.
- Action : Quitter l'application MEANDRES.
- Effets : Cette commande a pour effet de quitter l'occurrence de l'application MEANDRES associée à la fenêtre principale. Autrement dit, si une ou plusieurs autres simulations sont encore exploitées, cette commande est sans effet sur ces occurrences.
- Avant de fermer la fenêtre principale de l'occurrence de MEANDRES abandonnée, il faut vérifier qu'aucune modification des rapports associés n'a eu lieu depuis leur dernier enregistrement. Si c'est le cas, la fenêtre principale (et ses fenêtres filles) est effacée. Si ce n'est pas le cas, il y a affichage d'un message d'avertissement permettant à l'utilisateur de procéder au sauvetage des fichiers modifiés. Il y aura autant de messages que de rapports modifiés. L'enregistrement des rapports ne se fera qu'après avoir passé en revue tous les messages d'avertissement. Si l'utilisateur choisit, pour un des messages, le bouton *Annuler*, aucun enregistrement ne se produira.
- Messages : Si un rapport n'a jamais été enregistré, ou si un rapport a été modifié depuis son dernier enregistrement :



## 2.4 MENU EDITION

Comme toutes les applications qui laissent l'utilisateur déplacer ou copier du texte ou des graphiques par l'intermédiaire du Presse-papiers, MEANDRES possède un menu **Edition**. Ce menu offre toutes les commandes nécessaires à la gestion du Presse-papiers.

Le menu **Edition** comporte six items.

<b>Edition</b>	
<b>Annuler</b>	<b>Alt+Ret arr</b>
<b>Couper</b>	<b>Maj+Supp</b>
<b>Copier</b>	<b>Ctrl+Ins</b>
<b>Coller</b>	<b>Maj+Ins</b>
<b>Effacer</b>	<b>Supp</b>
<b>Tout sélectionner</b>	

Illustration 17 : Menu **Edition**

### 2.4.1 Annuler

Type : Commande.

Mnémo. : A

Accél. : Alt+Ret arr

Actif : Dès que le contenu du document courant a été modifié.

Action : Restaurer le document courant dans l'état qui était le sien avant la dernière action de modification de son contenu.

Effets : Cette action peut être occasionnée suite à l'utilisation des commandes suivantes :

**Couper**  
**Coller**  
**Effacer**  
**St. globales**  
**St. périodiques**  
**St. chronologiques**  
**Commentaires**  
**Calendrier**  
**Processus requérants**  
**Ressources requises**

Cette action peut être également le déplacement, le redimensionnement d'une ou plusieurs informations du rapport courant, ou encore leur effacement par l'utilisation de la touche *Supprimer (Delete)*.

La commande **Annuler** est elle-même une commande qui peut être annulée. Par conséquent, si l'utilisateur sélectionne cet item, celui-ci se transforme en l'item **Rétablir**. Inversement, le choix de l'action **Rétablir** modifie cet item au profit de **Annuler**. D'autre part, la commande **Rétablir** se transforme automatiquement en **Annuler** dès que l'utilisateur effectue une action.

Messages : Aucun.

### 2.4.2 Couper

Type : Commande.

Mnémo. : C

Accél. : Maj+Supp

Actif : Dès qu'une ou plusieurs informations du document courant ont été sélectionnées.

Action : Copier la sélection dans le Presse-papiers et effacer cette sélection du document courant.

Effets : Outre l'action décrite ci-dessus, cette commande a également pour effet de modifier le contenu de la boîte de dialogue de certaines commandes.

En effet, si cette commande est utilisée pour effacer certaines informations d'une fiche, la boîte de dialogue des commandes qui gèrent le contenu de cette fiche doit refléter les modifications effectuées : désélection de boîtes à cocher, de boutons radios, ...

Les informations restantes d'une fiche sont automatiquement repositionnées, alors que celles d'un rapport sont inchangées. Par conséquent, si l'utilisateur coupe une information du rapport courant, il devra éventuellement déplacer lui-même certaines informations pour combler l'espace vierge laissé par l'information coupée.

Messages : Aucun.

### 2.4.3 Copier

Type : Commande.

Mnémo. : P

Accél. : Ctrl+Ins

Actif : Dès qu'une ou plusieurs informations du document courant ont été sélectionnées.

Action : Copier la sélection dans le Presse-papiers sans l'effacer du document courant.

Effets : Il n'y a pas d'autre effet que celui décrit ci-dessus, si ce n'est que des informations non contiguës sur le document courant le deviennent dans le Presse-papiers : il y a suppression des espaces vides.

Messages : Aucun.

#### 2.4.4 Coller

Type : Commande.

Mnémo. : O

Accél. : Maj+Ins

Actif : Si le Presse-papiers n'est pas vide, et si le document courant est un rapport.

Action : Copier le contenu du Presse-papiers dans le rapport courant.

Effets : Pour que cette commande soit active, l'utilisateur doit donc rendre le rapport courant actif. Pour cela, il lui suffit de cliquer quelque part sur la zone de travail de la fenêtre principale. Par défaut, le contenu du Presse-papiers sera inséré à cet endroit (symbolisé par une petite croix). Plus précisément, le coin supérieur gauche du convexe délimitant le contenu du Presse-papiers coïncidera avec le point d'insertion.

Bien entendu, l'utilisateur a la possibilité de choisir un autre point d'insertion. Pour l'aider dans cette tâche, il peut utiliser les règles affichables par la commande **Afficher les règles** du menu **Ecran**.

Si l'utilisateur a coupé ou copié des informations non contiguës, celles-ci seront collées les unes en dessous des autres. Il est possible que cette opération superpose une information à une autre déjà présente dans le rapport. L'utilisateur devra donc souvent déplacer certaines informations collées.

La commande **Coller** ne modifie pas le Presse-papiers. Dès lors, l'utilisateur a la possibilité de coller de multiples fois les mêmes informations.

Messages : Aucun.

#### 2.4.5 Effacer

Type : Commande

Mnémo. : E

Accél. : Supp

Actif : Dès qu'une ou plusieurs informations du document courant ont été sélectionnées.

Action : Effacer la sélection du document courant sans modifier le Presse-papiers.

Effets : Comme la commande **Couper**, cette commande a également pour effet de modifier le contenu de la boîte de dialogue de certaines commandes, et dans le cas d'une fiche, de réarranger son contenu.

Messages : Aucun.

## 2.4.6 Tout sélectionner

Type : Commande.

Mnémo. : T

Accél. : Aucun.

Actif : Dès qu'une simulation a été associée à la fenêtre principale.

Action : Sélectionner tout le contenu du document courant.

Effets : Apparition d'un rectangle plein autour de toutes les informations du document courant. Si le document courant est vide, cette commande est sans effet.

Si le document courant est un rapport, alors les rectangles pleins entourant chacune des informations sélectionnées sont munis de trois poignées : une poignée de déformation horizontale, une poignée de déformation verticale, et une poignée de déformation homothétique. De plus, la zone d'information de la fenêtre principale est modifiée : elle indique les dimensions de la première information du rapport (celle située en haut à gauche de la première page).

L'utilisateur a aussi la possibilité de ne sélectionner que certaines informations. Pour de plus amples renseignements sur les mécanismes de déformation ou de sélection, reportez-vous à la section 2.9 "Sélection et déformation".

Messages : Aucun.

## 2.5 MENU PROCESSUS

Ce menu regroupe toutes les commandes de gestion des informations des fiches associées aux processus. Il n'est actif que si le document courant est une fiche associée à un processus.

Nous faisons ici l'hypothèse que le lecteur connaît bien les différentes mesures statistiques issues de DSL/SIM. Si ce n'est pas le cas, nous lui conseillons vivement de se référer au manuel avant de poursuivre la lecture.

Le menu **Processus** comporte sept items.

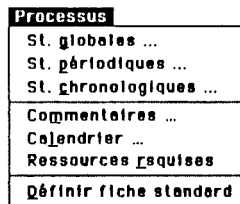


Illustration 18 : Menu Processus

### 2.5.1 St. globales

Type : Commande étendue.

Mnémo. : G

Accél. : Aucun.

Actif : Quand le document courant est une fiche associée à un processus.

Action : Sélectionner et afficher les informations concernant les statistiques globales du processus courant.

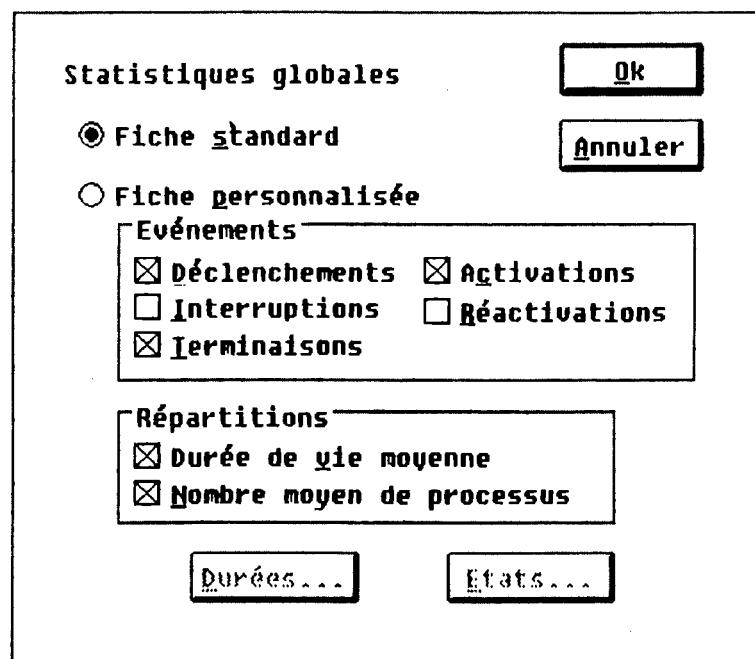


Illustration 19 : Boîte de dialogue de la commande St. globales

Boîte de dialogue : Il s'agit d'une boîte de dialogue modale extensible. Elle comprend, entre autres :

- Un bouton de commande *Durées*.
- Un bouton de commande *Etats*.

Chacun de ces deux boutons engendre une deuxième boîte de dialogue.

Durées	Ok		Annuler	
	Min.	Moy.	Max.	Ecart
Durée d'act <u>i</u> vité	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Durée d'att <u>e</u> nte d'act.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Durée d'inte <u>r</u> ruption	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Durée moy. d'inter./proc.		<input type="checkbox"/>		
Durée de re <u>p</u> os	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Durée moy. de re <u>p</u> os/proc.		<input type="checkbox"/>		
Durée moy. de v <u>i</u> e		<input type="checkbox"/>		

Illustration 20 : Boîte de dialogue engendrée par le bouton *Durées*

Etats	Ok		Annuler	
	Min.	Moy.	Max.	Ecart
Nombre de proc. en att <u>e</u> nte d'act.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Nombre de proc. i <u>n</u> terrompus	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Nombre de proc. en act <u>i</u> vité	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Illustration 21 : Boîte de dialogue engendrée par le bouton *Etats*

Effets : La boîte de dialogue principale s'utilise de deux manières. L'utilisateur peut choisir d'afficher une fiche standard, ou de consulter une fiche personnalisée. Dans ce deuxième cas, il lui incombe de sélectionner les différentes informations qu'il souhaite voir apparaître dans la fiche. Celles-ci sont rassemblées en quatre groupes en fonction de leur nature :

#### 1) Evénements

Cette classe regroupe les informations concernant le nombre d'occurrences des différents types d'événements : déclenchement, activation, interruption, réactivation, terminaison.



La sélection de l'une de ces cases à cocher provoquera l'affichage d'un texte indiquant le nombre d'occurrences des événements du type sélectionné. Ces différents textes constituent une seule unité graphique.

## 2) Durées

Cette classe regroupe les informations concernant les valeurs minimale, moyenne et maximale, ainsi que l'écart type de certaines durées :

- Durée d'activité.
- Durée d'attente d'activation.
- Durée d'interruption.
- Durée moyenne d'interruption / processus.
- Durée de repos.
- Durée moyenne de repos / processus.
- Durée moyenne de vie.

Pour chacune de ces durées, l'utilisateur peut donc demander d'afficher la valeur moyenne, et pour la plupart d'entre elles, leurs minimum, maximum et/ou écart type.

Pour cela, il lui suffit de cocher les cases qui l'intéressent. Pour faciliter cette tâche, un certain nombre de raccourcis lui sont offerts. En effet, s'il souhaite afficher toutes les mesures (min, moy., ...) associées à une durée, il n'a qu'à enfoncer la touche correspondant au mnémonique de cette durée. De même, s'il désire consulter la valeur moyenne (par exemple) de toutes les durées, il n'a qu'à enfoncer la touche correspondant au mnémonique de ce type de mesure. Enfin, s'il souhaite sélectionner toutes les informations, il lui suffit d'appuyer sur la touche *D* (mnémonique de *Durées*).

Quand l'utilisateur emploie un raccourci, si toutes les cases associées sont cochées, alors le raccourci les désélectionne. Sinon, il les sélectionne toutes.

Les différentes mesures sélectionnées apparaîtront dans la fiche sous forme d'un tableau similaire à celui présenté dans les listings produits par DSL/SIM<sup>1</sup>. Toute modification ultérieure de la boîte de dialogue modifiera ce tableau. Autrement dit, la fiche ne contient, au plus, qu'un seul tableau de durées. Ce tableau constitue une seule unité graphique.

## 3) Etats

Cette classe regroupe les informations concernant les valeurs minimale, moyenne et maximale, ainsi que l'écart type du nombre de processus dans un état déterminé :

- Nombre de processus en attente d'activation.
- Nombre de processus interrompus.
- Nombre de processus en activité.

La façon de sélectionner les informations est la même que celle utilisée dans la boîte de dialogue *Durées*.

De même, les différentes mesures sélectionnées apparaîtront également dans la fiche sous forme d'un tableau similaire à celui présenté dans les listings produits par DSL/SIM. Toute modification ultérieure de la boîte de dialogue modifiera ce tableau.

---

<sup>1</sup> Rappelons que la structure de ces listings est reproduite en annexe.

Ce tableau constitue une seule unité graphique.

#### 4) Répartitions

Cette classe regroupe les informations concernant la répartition de certaines données :

- Durée de vie moyenne.
- Nombre moyen de processus.

##### 4.1) Durée de vie moyenne

La sélection de la répartition de la durée de vie moyenne provoque l'apparition d'un graphique en forme de camembert mettant en évidence l'importance relative de trois mesures :

- La durée moyenne d'attente (= durée moy. d'attente d'activation + durée moy. d'interruption).
- La durée moyenne de repos.
- La durée moyenne d'activité.

Ce camembert est accompagné d'une légende et d'un titre. Le camembert et la légende, et le titre forment une unité graphique indissociable.

Répartition de la durée de vie moyenne

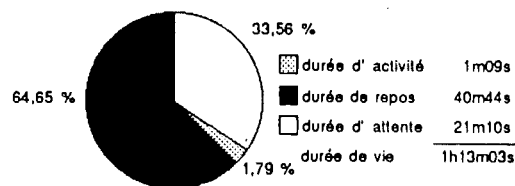


Illustration 22 : Répartition de la durée de vie moyenne

##### 4.2) Nombre moyen de processus

De même, la sélection de la répartition du nombre moyen de processus provoque également l'apparition d'un graphique en forme de camembert mettant en évidence l'importance relative de deux mesures :

- Le nombre moyen de processus en attente (= nbre moy. de processus en attente d'activation + nbre moy. de processus interrompus).
- Le nombre moyen de processus en activité.

Ce camembert est accompagné d'une légende et d'un titre. Le camembert et la légende, et le titre forment une unité graphique indissociable.

Répartition du nombre moyen de processus

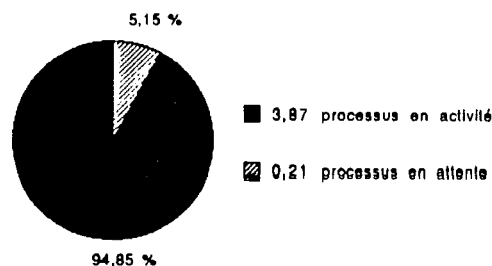


Illustration 23 : Répartition du nombre moyen de processus

Par défaut, lors de la création d'une nouvelle fiche, la boîte de dialogue propose une fiche standard. Les mesures statistiques globales présentes dans cette fiche sont :

- Le nombre de déclenchements.
- Le nombre d'activations.
- Le nombre de terminaisons.
- La répartition de la durée de vie moyenne.
- La répartition du nombre moyen de processus.

Ces informations sont cochées dans la boîte de dialogue.

Pour composer une fiche personnalisée, l'utilisateur peut soit sélectionner explicitement le bouton radio *Fiche personnalisée*, auquel cas toutes les cases cochées sont décochées (il n'a plus qu'à composer sa fiche personnalisée), soit cocher directement les informations, auquel cas le bouton radio *Fiche personnalisée* est automatiquement activé.

Il va sans dire que si l'utilisateur resélectionne les cinq informations composant la fiche standard, le bouton radio *Fiche standard* est activé au détriment du bouton radio *Fiche personnalisée*.

Messages : Aucun.

## 2.5.2 St. périodiques

Statistiques périodiques

Fiche standard

Fiche personnalisée

OK

Annuler

Evénements

Déclenchements  Activations

Interruptions  Réactivations

Terminaisons

Durées

Durée moy. d'attente d'act.

Durée moy. d'interruption

Durée moy. de repos

Répartitions

Nombre d'événements

Durée moy. d'attente et de repos

Illustration 24 : Boîte de dialogue de la commande St. périodiques

Type : Commande étendue.

Mnémo. : P

- Accél. : Aucun.
- Actif : Quand le document courant est une fiche associée à un processus.
- Action : Sélectionner et afficher les informations concernant les statistiques périodiques du processus courant.
- Boîte de dialogue : Il s'agit d'une boîte de dialogue modale.
- Effets : Cette boîte de dialogue s'utilise de deux manières. L'utilisateur peut choisir d'afficher une fiche standard, ou de consulter une fiche personnalisée. Dans ce deuxième cas, il lui incombe de sélectionner les différentes informations qu'il souhaite voir apparaître dans la fiche. Celles-ci sont rassemblées en trois groupes en fonction de leur nature :

#### 1) Evénements

Cette classe regroupe les informations concernant le nombre d'occurrences des différents types d'événements : déclenchement, activation, interruption, réactivation, terminaison.

La sélection de l'une de ces cases à cocher provoquera l'affichage d'un tableau analogue à celui produit par DSL/SIM, indiquant pour chaque cellule d'observation, le nombre d'occurrences des événements du type sélectionné.

#### 2) Durées

Cette classe regroupe les informations concernant la durée moyenne de certains états :

- Durée moyenne d'attente d'activation.
- Durée moyenne d'interruption.
- Durée moyenne de repos.

La sélection de l'une de ces cases à cocher provoquera l'affichage d'un tableau analogue à celui produit par DSL/SIM, indiquant les durées sélectionnées pour chaque cellule d'observation.

Le tableau des événements et le tableau des durées ne forment qu'un seul tableau (cfr DSL/SIM). Ce tableau constitue une seule unité graphique.

#### 3) Répartitions

Cette classe regroupe les informations concernant la répartition de certaines données :

- Le nombre d'événements.
- La durée moyenne d'attente et de repos.

##### 3.1) Nombre d'événements

La sélection de cette case provoque l'affichage d'un histogramme (c'est-à-dire un diagramme à bâtons) reflétant l'évolution dans le temps du nombre d'occurrences des événements.

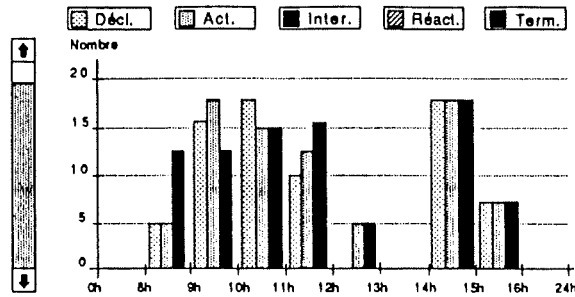


Illustration 25 : Histogramme des événements

L'axe des abscisses est divisé en cellules de temps. Cette division dépend bien sûr des spécifications apportées par l'utilisateur lors de l'exécution de DSL/SIM. Toutefois, il n'y a pas nécessairement correspondance parfaite entre cette division et la définition des cellules d'observation. En effet, nous avons choisi de regrouper toutes les cellules non significatives contiguës. Ainsi, par exemple, si l'utilisateur avait choisi de découper les journées en vingt-quatre cellules d'observation horaires, et qu'aucun événement n'est recensé de minuit à huit heures du matin, alors les huit premières cellules sont regroupées sur le schéma pour former la cellule [0h,8h[

L'axe des ordonnées est gradué de manière optimale en fonction de la valeur maximale des bâtons. A gauche de cet axe, une barre de défilement vertical permet de modifier l'échelle de cet axe. La graduation de l'axe s'adapte à l'échelle choisie<sup>1</sup>.

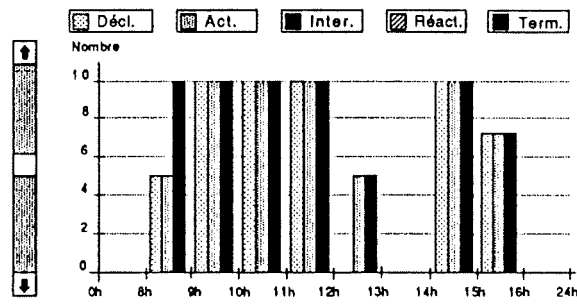


Illustration 26 : Changement d'échelle de l'histogramme des événements

Si l'échelle ne permet pas de visualiser l'entièreté d'un bâton, l'utilisateur a la possibilité de connaître la valeur exacte de la mesure représentée. Il suffit de positionner le curseur de la souris au dessus de ce bâton : la valeur s'affiche au sommet du bâton tronqué. Dès que le curseur quitte le bâton, la mesure est effacée. Si l'utilisateur souhaite garder cette valeur quand le curseur quitte le bâton (utile, par exemple, pour conserver la valeur sur l'histogramme copié dans un rapport), il lui suffit de maintenir enfoncée la touche *Majuscule* quand le curseur passe sur le bâton. Pour effacer la mesure, il doit recommencer cette manœuvre. Nous insistons sur le fait que l'affichage des valeurs au dessus des bâtons n'est prévue que pour les bâtons tronqués : c'est la manière de savoir si un bâton est tronqué ou non.

<sup>1</sup> Le lecteur soucieux d'avoir plus de précisions quant au changement d'échelle en fonction de la barre de défilement trouvera en annexe un algorithme proposant une manière d'implémenter ce "zooming"

Par défaut, l'histogramme n'affiche que les représentations du nombre de déclenchements, du nombre d'activations, et du nombre de terminaisons. La fiche ne contient donc qu'un seul histogramme.

Au dessus du graphique, se trouve une légende reprenant le nom des cinq types d'événement ainsi qu'une case colorée permettant de faire le lien avec les bâtons correspondants. Cette légende est toujours présente même si l'histogramme n'affiche pas la représentation des valeurs des cinq types d'événement. La raison est que l'utilisateur peut cliquer sur les cases de la légende pour afficher (ou effacer) les bâtons correspondants. Ce mécanisme est un raccourci qui évite de passer par la boîte de dialogue de la commande **St. périodiques**.

Il n'y a pas isomorphisme entre l'histogramme de la fiche et celui obtenu par copier/coller dans le rapport. En effet, le graphique collé dans le rapport ne possède pas de barre de défilement vertical. De plus, la légende reflète la présence, ou non, de bâtons. Ainsi, par exemple, si l'utilisateur choisit de ne visualiser que le nombre de déclenchements et le nombre de terminaisons, la légende ne contiendra que le nom de ces deux types d'événement.

L'histogramme et la légende constituent une seule unité graphique.

### 3.2) La durée moyenne d'attente et de repos

La sélection de cette case à cocher provoquera l'affichage d'un histogramme reflétant l'évolution dans le temps de ces mesures.

Cet histogramme est analogue à celui des événements.

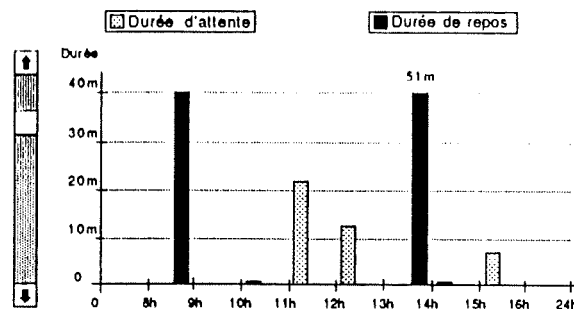


Illustration 27 : Histogramme des durées

Par défaut, lors de la création d'une nouvelle fiche, la boîte de dialogue propose une fiche standard. Les mesures statistiques périodiques présentes dans cette fiche sont :

- La répartition du nombre d'événements.
- La répartition de la durée moyenne d'attente et de repos.

Messages : Aucun.

### 2.5.3 St. chronologiques

Type : Commande étendue.

Mnémo. : C

Accél. : Aucun.

- Actif : Quand le document courant est une fiche associée à un processus.
- Action : Sélectionner et afficher les informations concernant les statistiques chronologiques du processus courant.
- Boîte de dialogue : Il s'agit d'une boîte de dialogue modale extensible. Elle comprend, entre autres :
- Un bouton de commande *Durées*.
  - Un bouton de commande *Etats*.
- Chacun de ces deux boutons engendre une deuxième boîte de dialogue.

Statistiques chronologiques

Fiche standard

Fiche personnalisée

Evénements

Déclenchements     Activations

Interruptions     Réactivations

Terminaisons

Répartitions

Nombre d'événements

Durée d'attente

Nombre de proc. en att. ou en act.

Durées...    Etats...

Ok    Annuler

Illustration 28 : Boîte de dialogue de la commande St. chronologiques

Durées

Ok    Annuler

Durée moy. d'attente d'act.

Durée moy. d'interruption

Illustration 29 : Boîte de dialogue engendrée par le bouton *Durées*

Etats	<input type="button" value="OK"/>	<input type="button" value="Annuler"/>
<input type="checkbox"/> Nombre de proc. en attente d'act.		
<input type="checkbox"/> Nombre min. de proc en attente d'act.		
<input type="checkbox"/> Nombre de proc. en attente d'act. à la fin de la cellule		
<input type="checkbox"/> Nombre de proc. interrompus		
<input type="checkbox"/> Nombre de proc. en activité		

Illustration 30 : Boîte de dialogue engendrée par le bouton *Etats*

Effets : Cette boîte de dialogue s'utilise de deux manières. L'utilisateur peut choisir d'afficher une fiche standard, ou de consulter une fiche personnalisée. Dans ce deuxième cas, il lui incombe de sélectionner les différentes informations qu'il souhaite voir apparaître dans la fiche. Celles-ci sont rassemblées en quatre groupes en fonction de leur nature :

#### 1) Evénements

Cette classe regroupe les informations concernant le nombre d'occurrences des différents types d'événements : déclenchement, activation, interruption, réactivation, terminaison.

La sélection de l'une de ces cases à cocher provoquera l'affichage d'un tableau analogue à celui produit par DSL/SIM, indiquant pour chaque cellule d'observation de chaque période, le nombre d'occurrences des événements du type sélectionné.

#### 2) Durées

Cette classe regroupe les informations concernant la durée moyenne de certains états :

- Durée moyenne d'attente d'activation.
- Durée moyenne d'interruption.

La sélection de l'une de ces cases à cocher provoquera l'affichage d'un tableau analogue à celui produit par DSL/SIM, indiquant les durées sélectionnées pour chaque cellule d'observation de chaque période.

#### 3) Etats

Cette classe regroupe les informations concernant le nombre de processus dans un état déterminé :

- Nombre de processus en attente d'activation.
- Nombre minimum de processus en attente d'activation.
- Nombre de processus en attente d'activation à la fin de la cellule
- Nombre de processus interrompus.
- Nombre de processus en activité.



La sélection de l'une de ces cases à cocher provoquera l'affichage d'un tableau analogue à celui produit par DSL/SIM, indiquant le nombre de processus dans l'état sélectionné pour chaque cellule d'observation de chaque période.

Le tableau des événements, le tableau des durées, et celui des états ne forment qu'un seul tableau (cfr DSL/SIM). Ce tableau constitue une seule unité graphique.

#### 4) Répartitions

Cette classe regroupe les informations concernant la répartition de certaines données :

- Le nombre d'événements.
- La durée d'attente.
- Le nombre de processus en attente ou en activité.

##### 4.1) Le nombre d'événements

La sélection de cette case provoque l'affichage d'un histogramme reflétant l'évolution dans le temps du nombre d'occurrences des événements.

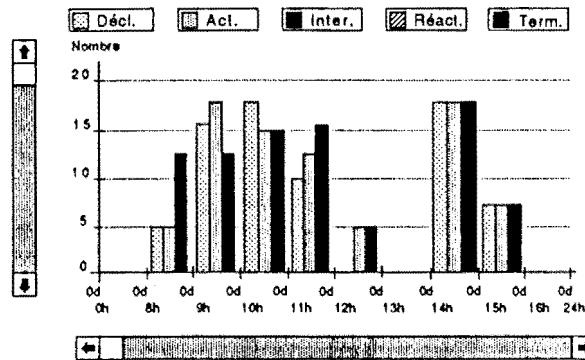


Illustration 31 : Histogramme des événements

Il s'agit d'un histogramme analogue à celui des statistiques périodiques. Quelques différences dans la présentation et dans la gestion sont à noter.

Puisque, contrairement aux statistiques périodiques, les statistiques chronologiques ne cumulent pas les mesures en les ramenant à une seule période, une barre de défilement horizontal est nécessaire pour parcourir l'ensemble de l'intervalle de temps de la simulation.

Par défaut, l'histogramme affiche les mesures concernant la première période. En cliquant sur les flèches de défilement, il est possible d'avancer (ou de reculer) d'une cellule d'observation. En cliquant dans la barre de défilement, il est possible d'avancer (ou de reculer) d'une période (par rapport à la période de la cellule d'observation la plus à gauche le long de l'axe des abscisses).

Rappelons que les cellules d'observation non significatives contiguës sont regroupées en une seule cellule. Toutes les périodes sont divisées de la même façon. Le choix du regroupement des cellules est donc fonction de la période la plus "riche" (c'est-à-dire celle possédant le plus de cellules significatives).

Si l'utilisateur clique rapidement deux fois sur un des carrés de la légende, alors une boîte de dialogue modale apparaît afin de le renseigner sur la valeur minimale, la valeur moyenne et la valeur maximale du nombre d'événements du type ainsi sélectionné, pour la période de la cellule d'observation la plus à gauche le long de l'axe des abscisses.

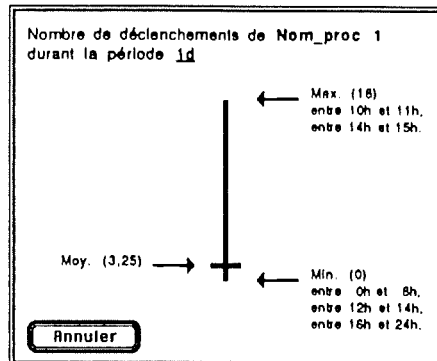


Illustration 32 : Visualisation des valeurs min., moy., et max.

#### 4.2) La durée d'attente

La sélection de cette case à cocher provoquera l'affichage d'un histogramme reflétant l'évolution dans le temps de la durée d'attente.

Cet histogramme est analogue à celui des événements.

#### 4.3) Le nombre de processus en attente ou en activité.

La sélection de cette case à cocher provoquera l'affichage d'un histogramme reflétant l'évolution dans le temps de ces deux mesures.

Cet histogramme est analogue à celui des événements.

Par défaut, lors de la création d'une nouvelle fiche, la boîte de dialogue propose une fiche standard. Les mesures statistiques chronologiques présentes dans cette fiche sont :

- La répartition du nombre d'événements.
- La répartition de la durée d'attente.
- La répartition du nombre de processus en attente ou en activité.

Messages : Aucun.

### 2.5.4 Commentaires

Type : Commande étendue.

Mnémo. : M

Accél. : Aucun.

Actif : Quand le document courant est une fiche associée à un processus.

Action : Sélectionner et afficher des informations générales concernant la fiche courante.

Boîte de dialogue : Il s'agit d'une boîte de dialogue modale :

Illustration 33 : Boîte de dialogue de la commande **Commentaires**

Effets : Les différentes informations que l'utilisateur peut ajouter dans la fiche courante sont :

- Le titre de la simulation.
- La date d'exécution de la simulation.
- Le type de statistiques de la fiche courante.
- La période de prise en compte, c'est-à-dire la période durant laquelle les statistiques ont été calculées.
- Le nom du processus.
- Le calendrier du processus, c'est-à-dire sa période d'activité.

Si l'utilisateur sélectionne ces informations, elles apparaîtront au début de la fiche, sous forme de texte.

Illustration 34 : Exemple de commentaires sur une fiche

Signalons que la case *Calendrier* n'est pas redondante avec la commande **Calendrier**, puisque la première affiche la période d'activité du processus sous forme de texte, alors que la deuxième l'affiche sous forme graphique.

Si, lors de la création d'une fiche, l'utilisateur choisit l'option *Fiche standard*, alors toutes les informations de la boîte de dialogue de la commande **Commentaires** sont affichées.

Chacune de ces informations constitue une unité graphique en soi. Dans le cas où les deux informations d'une boîte de regroupement sont sélectionnées, ces deux commentaires ne forment qu'une seule même unité graphique.

Messages : Aucun.

### 2.5.5 Calendrier

Type : Commande étendue.

Mnémo. : L

Accél. : Aucun.

Actif : Si le document courant est une fiche présentant des mesures statistiques chronologiques d'un processus.

Action : Afficher le calendrier d'activité du processus courant ainsi que le calendrier de disponibilité des ressources requises par ce processus.

Boîte de dialogue : Il s'agit d'une boîte de dialogue modale :

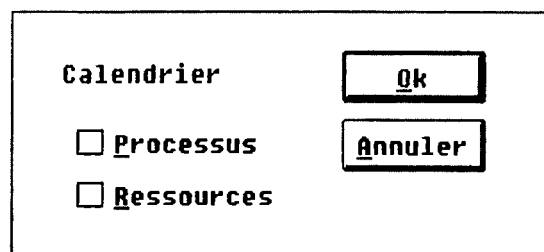


Illustration 35 : Boîte de dialogue de la commande **Calendrier**

Effets : Si l'utilisateur choisit le bouton *Processus*, alors le calendrier d'activité du processus courant s'affiche sous forme de segments de droites. Si plusieurs histogrammes sont présents dans la fiche, alors le calendrier s'affiche en dessous du dernier histogramme. Si aucun histogramme n'est présent, alors une barre de défilement horizontal s'affiche au dessus des calendriers.

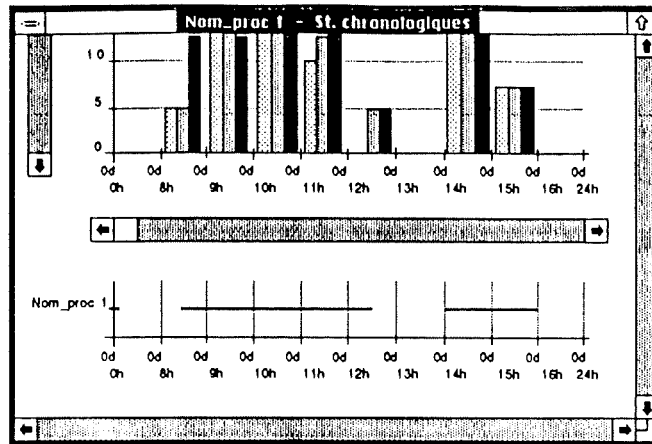


Illustration 36 : Calendrier d'activité du processus

Si l'utilisateur sélectionne *Ressources*, alors le calendrier de disponibilité des ressources requises par le processus courant s'affiche sous forme de segments de droites exactement en dessous du calendrier du processus. Son axe des abscisses est identique à celui du calendrier d'activité du processus. Le bouton *Ressources* n'est disponible que si le processus courant requiert des ressources et si l'utilisateur a coché le bouton *Processus*.

Chacun des deux graphiques constitue une seule unité.

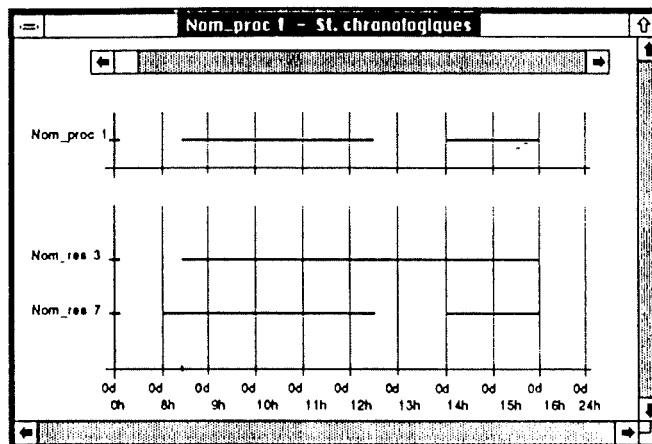


Illustration 37 : Calendrier de disponibilité des ressources

Bien que le calendrier d'activité du processus, le calendrier de disponibilité des ressources, et le dernier histogramme soient des entités distinctes (et donc sélectionnables séparément), ils ne sont pas indépendants. En effet, les trois axes des abscisses présentent à tout moment les mêmes cellules. Ainsi, non seulement la division des périodes en cellules est identiques sur les trois graphiques, mais le fait d'utiliser la barre de défilement horizontal modifie simultanément les trois axes.

Si l'utilisateur désélectionne le bouton *Processus*, ou s'il coupe (ou efface) le calendrier d'activité du processus, alors le bouton *Ressources* est automatiquement désélectionné, et le calendrier de disponibilité des ressources est supprimé de la fiche.

Si l'utilisateur coupe (ou efface) l'histogramme auquel sont attachés les calendriers, alors ceux-ci sont associés au dernier histogramme encore présent dans la fiche. S'il n'y a plus d'histogramme présent, alors une barre de défilement horizontal apparaît au dessus du calendrier d'activité du processus. Ce que nous avons signalé concernant la dépendance des calendriers reste valable.

Messages : Aucun.

### 2.5.6 Ressources requises

Type : Descripteur d'état.

Mnémo. : R

Accél. : Aucun.

Actif : Quand le document courant est une fiche présentant des mesures statistiques globales d'un processus qui requiert des ressources.

Action : Afficher des informations sur les ressources requises par le processus courant.

Effets : Cette commande a pour effet d'afficher un tableau récapitulatif des ressources requises par le processus courant. Ce tableau, analogue à celui présenté par DSL/SIM, reprend pour chaque type de ressources :

- La capacité maximale et la capacité moyenne utilisée.
- Les nombres maximum et moyen de points d'entrée utilisés.
- Le nombre moyen de processus en attente devant cette ressource.

Ce tableau constitue une seule unité graphique.

Par défaut, lors de la création de la fiche de mesures statistiques globales du processus courant, ce descripteur d'état est décoché.

Messages : Aucun.

### 2.5.7 Définir fiche standard

Type : Commande.

Mnémo. : D

Accél. : Aucun.

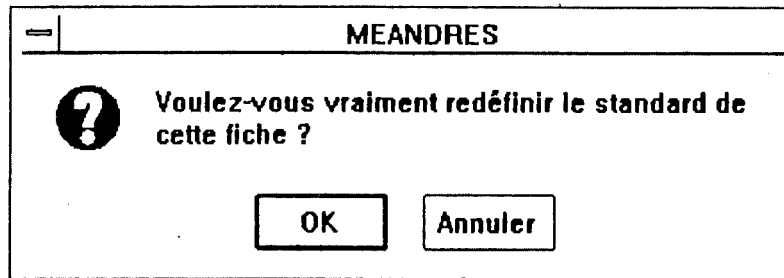
Actif : Quand le document courant est une fiche associée à un processus.

Action : Redéfinir le standard des fiches de même type que celui de la fiche courante.

Effets : Cette commande a pour effet de modifier l'option *Fiche standard* des commandes **St. globales**, **St. périodiques**, **St. chronologiques**. Les fiches standards de même type que celui de la fiche courante seront dorénavant similaires à la fiche courante.

Cette commande modifie donc la présentation des boîtes de dialogue de sélection des informations (changement des options par défaut), et, éventuellement, le menu **Processus** (présence d'une coche devant l'item **Ressources requises**).

Messages : Cette commande est potentiellement dangereuse. Un message de confirmation s'avère nécessaire.



## 2.6 MENU RESSOURCES

Ce menu regroupe toutes les commandes de gestion des informations des fiches associées aux processus. Il n'est actif que si le document courant est une fiche associée à une ressource.

Ce menu comprend six items.

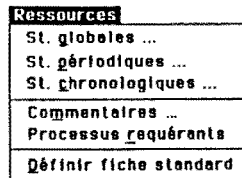


Illustration 38 : Menu Ressources

### 2.6.1 St. globales

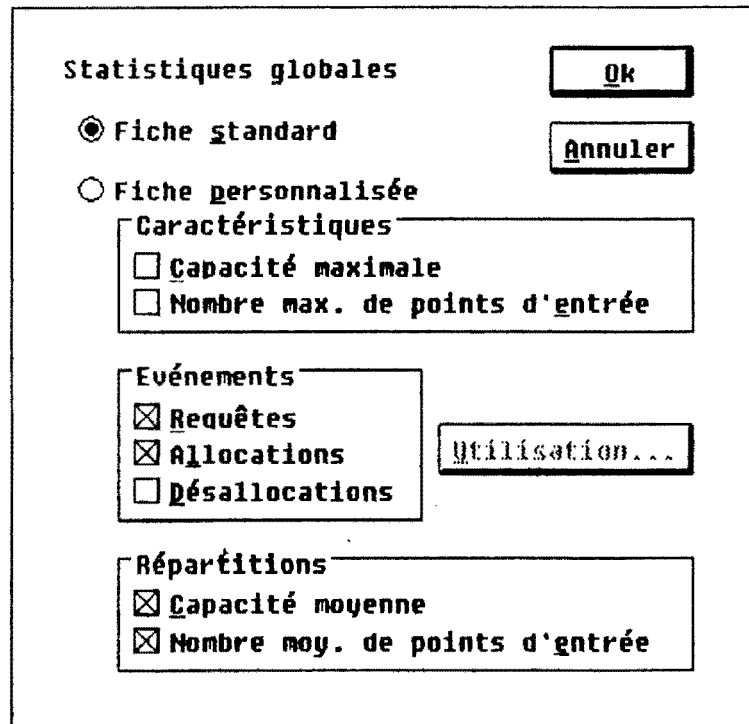


Illustration 39 : Boîte de dialogue de la commande St. globales

- Type : Commande étendue.
- Mnémo. : G
- Accél. : Aucun.
- Actif : Quand le document courant est une fiche associée à une ressource.
- Action : Sélectionner et afficher les informations concernant les statistiques globales de la ressource courante.



Boîte de dialogue : Il s'agit d'une boîte de dialogue modale extensible qui comprend, entre autres, un bouton *Utilisation*. Ce bouton engendre une deuxième boîte de dialogue.

<u>Utilisation</u>	Ok		Annuler	
	Min.	Moy.	Max.	Ecart
<u>C</u> apacité utilisée	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Nombre de <u>r</u> equêtes satisfaites simultanément	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Nombre de <u>p</u> roc. en attente devant la ressource	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Illustration 40 : Boîte de dialogue engendrée par le bouton *Utilisation*

Effets : La boîte de dialogue principale s'utilise de deux manières. L'utilisateur peut choisir d'afficher une fiche standard, ou de consulter une fiche personnalisée. Dans ce deuxième cas, il lui incombe de sélectionner les différentes informations qu'il souhaite voir apparaître dans la fiche. Celles-ci sont rassemblées en quatre groupes en fonction de leur nature :

### 1) Caractéristiques

Cette classe regroupe les informations concernant les caractéristiques de la ressource courante : sa capacité maximale et/ou son nombre maximum de points d'entrée.

La sélection de l'une de ces cases à cocher provoquera l'affichage d'un texte indiquant ces caractéristiques. Ces différents textes constituent une seule unité graphique.

Notez qu'il est possible qu'une de ces deux cases ne soit pas disponible : toutes les ressources ne sont pas nécessairement quantifiées à la fois en terme de capacité et en terme de point d'entrée.

### 2) Événements

Cette classe regroupe les informations concernant le nombre d'occurrences des différents types d'événements : requête, allocation, désallocation.

La sélection de l'une de ces cases à cocher provoquera l'affichage d'un texte indiquant le nombre d'occurrences des événements du type sélectionné. Ces différents textes constituent une seule unité graphique.

### 3) Utilisation

Cette classe regroupe les informations concernant le taux d'utilisation de la ressource courante :

- La capacité utilisée.
- Le nombre de requêtes satisfaites simultanément.
- Le nombre de processus en attente devant la ressource.

Pour chacune de ces mesures, l'utilisateur peut demander d'afficher la valeur moyenne, la valeur minimale, la valeur maximale et/ou l'écart type.

Rappelons que toutes les ressources ne sont pas nécessairement quantifiées à la fois en terme de capacité et en terme de point d'entrée. Par conséquent, certaines des mesures exposées ci-dessus ne seront peut-être pas disponibles.

La façon de sélectionner les informations est la même que celle utilisée dans la boîte de dialogue *Durées* de la commande **St. globales** du menu **Processus**.

Les différentes mesures sélectionnées apparaîtront dans la fiche sous forme d'un tableau similaire à celui présenté dans les listings produits par DSL/SIM. Toute modification ultérieure de la boîte de dialogue modifiera ce tableau. Autrement dit, la fiche ne contient, au plus, qu'un seul tableau de mesures d'utilisation. Ce tableau constitue une seule unité graphique.

#### 4) Répartitions

Cette classe regroupe les informations concernant la répartition de certaines données :

- La capacité moyenne.
- Le nombre moyen de points d'entrée.

Pour les raisons déjà évoquées, il est possible que l'une de ces deux options soit indisponible.

##### 4.1) La capacité moyenne

La sélection de la répartition de la capacité provoque l'apparition d'un graphique en forme de camembert mettant en évidence l'importance relative de deux mesures :

- La capacité moyenne utilisée.
- La capacité moyenne résiduelle.

Ce camembert est accompagné d'une légende et d'un titre. Le camembert et la légende, et le titre forment une unité graphique indissociable.

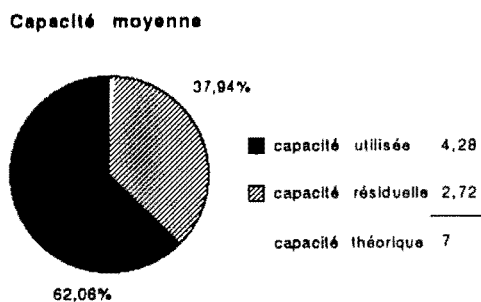


Illustration 41 : Répartition de la capacité

##### 4.2) Le nombre moyen de points d'entrée

De même, la sélection de la répartition du nombre de points d'entrée provoque également l'apparition d'un graphique en forme de camembert mettant en évidence l'importance relative de deux mesures :

- Le nombre moyen de points d'entrée utilisés.
- Le nombre moyen de points d'entrée non utilisés.

Ce camembert est accompagné d'une légende et d'un titre. Le camembert et la légende, et le titre forment une unité graphique indissociable.

Nombre moyen de points d'entrée

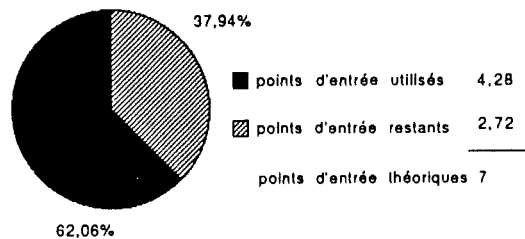


Illustration 42 : Répartition du nombre de points d'entrée

Par défaut, lors de la création d'une nouvelle fiche, la boîte de dialogue propose une fiche standard. Les mesures statistiques globales présentes dans cette fiche sont :

- Le nombre de requêtes.
- Le nombre d'allocations.
- La répartition de la capacité moyenne.
- La répartition du nombre moyen de points d'entrée.

Messages : Aucun.

## 2.6.2 St. périodiques

**Statistiques périodiques**

**Fiche standard**

**Fiche personnalisée**

**Evénements**

Re~~q~~uêtes

Allocations

Désallocations

**Répartitions**

Nombre d'événements

Illustration 43 : Boîte de dialogue de la commande St. périodiques

Type : Commande étendue.

Mnémo. : P

- Accél. : Aucun.
- Actif : Quand le document courant est une fiche associée à une ressource.
- Action : Sélectionner et afficher les informations concernant les statistiques périodiques de la ressource courante.
- Boîte de dialogue : Il s'agit d'une boîte de dialogue modale :
- Effets : La boîte de dialogue principale s'utilise de deux manières. L'utilisateur peut choisir d'afficher une fiche standard, ou de consulter une fiche personnalisée. Dans ce deuxième cas, il lui incombe de sélectionner les différentes informations qu'il souhaite voir apparaître dans la fiche. Celles-ci sont rassemblées en deux groupes en fonction de leur nature :
- 1) Evénements
- Cette classe regroupe les informations concernant le nombre d'occurrences des différents types d'événements : requête, allocation, désallocation.
- La sélection de l'une de ces cases à cocher provoquera l'affichage d'un tableau analogue à celui produit par DSL/SIM, indiquant pour chaque cellule d'observation. Ce tableau constitue une seule unité graphique, le nombre d'occurrences des événements du type sélectionné.
- 2) Répartitions
- Cette "classe" ne regroupe qu'un seul type d'information : la répartition du nombre d'occurrences des différents types d'événements.
- La sélection de cette case à cocher provoquera l'affichage d'un histogramme reflétant l'évolution dans le temps du nombre d'occurrences des différents types d'événements.
- Cet histogramme est analogue à ceux décrits dans le chapitre concernant la commande **St. périodiques** du menu **Processus**.
- Par défaut, lors de la création d'une nouvelle fiche, la boîte de dialogue propose une fiche standard. Les seules mesures statistiques périodiques présentes dans cette fiche sont l'histogramme de répartition des événements.
- Messages : Aucun.

### 2.6.3 St. chronologiques

- Type : Commande étendue.
- Mnémo. : C
- Accél. : Aucun.
- Actif : Quand le document courant est une fiche associée à une ressource.
- Action : Sélectionner et afficher les informations concernant les statistiques chronologiques du processus courant.

Boîte de dialogue : Il s'agit d'une boîte de dialogue modale :

**Statistiques chronologiques**

**Fiche standard**

**Fiche personnalisée**

**Evénements**

Requête

Allocations

Désallocations

**Répartitions**

Nombre d'occ. des événements

Taux moyen d'utilisation

Nombre moy. de proc. en attente

**Utilisation**

Capacité moyenne

Nombre moy. de requêtes satisfaites simultanément

Nombre moy. de proc. en attente devant la ressource

Nombre max. de proc. en attente devant la ressource

Nombre de proc. en attente devant la ressource à la fin de la cellule d'observation

Illustration 44 : Boîte de dialogue de la commande St. chronologiques

Effets : La boîte de dialogue principale s'utilise de deux manières. L'utilisateur peut choisir d'afficher une fiche standard, ou de consulter une fiche personnalisée. Dans ce deuxième cas, il lui incombe de sélectionner les différentes informations qu'il souhaite voir apparaître dans la fiche. Celles-ci sont rassemblées en trois groupes en fonction de leur nature :

#### 1) Evénements

Cette classe regroupe les informations concernant le nombre d'occurrences des différents types d'événements : requête, allocation, désallocation.

La sélection de l'une de ces cases à cocher provoquera l'affichage d'un tableau analogue à celui produit par DSL/SIM, indiquant pour chaque cellule d'observation de chaque période, le nombre d'occurrences des événements du type sélectionné.

#### 2) Utilisation

Cette classe regroupe les informations concernant le taux d'utilisation de la ressource courante :

- La capacité moyenne utilisée.
- Le nombre moyen de requêtes satisfaites simultanément.
- Le nombre moyen de processus en attente devant la ressource.
- Le nombre maximum de processus en attente devant la ressource.
- Le nombre de processus en attente devant la ressource à la fin de la cellule d'observation.

La sélection de l'une de ces cases à cocher provoquera l'affichage d'un tableau analogue à celui produit par DSL/SIM, indiquant le nombre de processus dans l'état sélectionné pour chaque cellule d'observation de chaque période.

Le tableau des événements, et celui des utilisations ne forment qu'un seul tableau (cfr DSL/SIM). Ce tableau constitue une seule unité graphique.

### 3) Répartitions

Cette classe regroupe les informations concernant la répartition de certaines données :

- Le nombre d'occurrences des différents types d'événements.
- Le taux moyen d'utilisation.
- Le nombre moyen de processus en attente devant la ressource courante.

La sélection de ces cases à cocher provoquera l'affichage d'un histogramme reflétant l'évolution dans le temps des mesures sélectionnées.

Ces histogrammes sont analogues à ceux décrits dans le chapitre concernant la commande **St. chronologiques** du menu **Processus**.

Par défaut, lors de la création d'une nouvelle fiche, la boîte de dialogue propose une fiche standard. Les mesures statistiques chronologiques présentes dans cette fiche sont :

- La répartition du nombre d'occurrences des différents types d'événements.
- La répartition de la capacité moyenne utilisée et/ou du nombre moyen de points d'entrée utilisés.
- La répartition du nombre moyen de processus en attente devant la ressource courante.

Messages : Aucun.

## 2.6.4 Commentaires

Type : Commande étendue.

Mnémo. : M

Accél. : Aucun.

Actif : Quand le document courant est une fiche associée à une ressource.

Action : Sélectionner et afficher des informations générales concernant la fiche courante.

Boîte de dialogue : Il s'agit d'une boîte de dialogue modale :

**Commentaires**

Simulation

Titre

Date d'exécution

Mesures statistiques

Type

Période de prise en compte

Ressources

Nom

Calendrier

Ok

Annuler

Illustration 45 : Boîte de dialogue de la commande **Commentaires**

- Effets :
- Les différentes informations que l'utilisateur peut ajouter dans la fiche courante sont :
- Le titre de la simulation.
  - La date d'exécution de la simulation.
  - Le type de statistiques de la fiche courante.
  - La période de prise en compte, c'est-à-dire la période durant laquelle les statistiques ont été calculées.
  - Le nom de la ressource.
  - Le calendrier de disponibilité général de la ressource.

Si l'utilisateur sélectionne ces informations, elles apparaîtront au début de la fiche, sous forme de texte.

Si, lors de la création d'une fiche, l'utilisateur choisit l'option *Fiche standard*, alors toutes les informations de la boîte de dialogue de la commande **Commentaires** sont affichées dans la fiche.

Chacune de ces informations constitue une unité graphique en soi. Dans le cas où les deux informations d'une boîte de regroupement sont sélectionnées, ces deux commentaires ne forment qu'une seule même unité graphique.

Messages : Aucun.

### 2.6.5 Processus requérants

Type : Descripteur d'état.

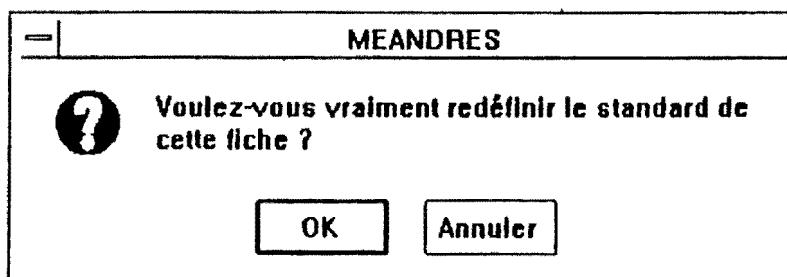
Mnémo. : R

Accél. : Aucun.

- Actif :** Quand le document courant est une fiche présentant des mesures statistiques globales d'une ressource requise par au moins un processus.
- Action :** Afficher des informations sur les processus qui requièrent la ressource courante.
- Effets :** Cette commande a pour effet d'afficher un tableau récapitulatif des processus qui requièrent la ressource courante. Ce tableau, analogue à celui présenté par DSL/SIM, reprend pour chaque type de processus :
- La durée moyenne d'attente d'activation.
  - La durée moyenne d'interruption.
  - La durée moyenne d'activité.
  - Le nombre moyen de processus en attente d'activation.
  - Le nombre moyen de processus interrompus.
  - Le nombre moyen de processus en activité.
- Comme cet item est un descripteur d'état, rappelons qu'il sera précédé d'une coche s'il est actif. Par défaut, lors de la création de la fiche de mesures statistiques globales du processus courant, il est inactif.
- Messages :** Aucun.

## 2.6.6 Définir fiche standard

- Type :** Commande.
- Mnémo. :** D
- Accél. :** Aucun.
- Actif :** Quand le document courant est une fiche associée à une ressource.
- Action :** Redéfinir le standard des fiches de même type que celui de la fiche courante.
- Effets :** Cette commande a pour effet de modifier l'option *Fiche standard* des commandes **St. globales**, **St. périodiques**, **St. chronologiques**. Les fiches standards de même type que celui de la fiche courante seront dorénavant similaires à la fiche courante.
- Cette commande modifie donc la présentation des boîtes de dialogue de sélection des informations (changement des options par défaut), et, éventuellement, le menu **Ressources** (présence d'une coche devant l'item **Processus requérants**).
- Messages :** Cette commande est potentiellement dangereuse. Un message de confirmation s'avère nécessaire.





## 2.7 MENU PT. DE SYNCHR.

Ce menu regroupe toutes les commandes de gestion des informations des fiches associées aux points de synchronisation. Il n'est actif que si le document courant est une fiche associée à un point de synchronisation.

Le menu **Processus** comporte trois items.

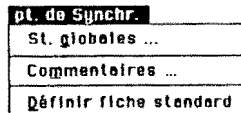


Illustration 46 : Menu Pt. de synchr.

### 2.7.1 St. globales

Type : Commande étendue.

Mnémo. : G

Accél. : Aucun.

Actif : Quand le document courant est une fiche associée à un point de synchronisation.

Action : Sélectionner et afficher les informations concernant les statistiques globales du point de synchronisation courant.

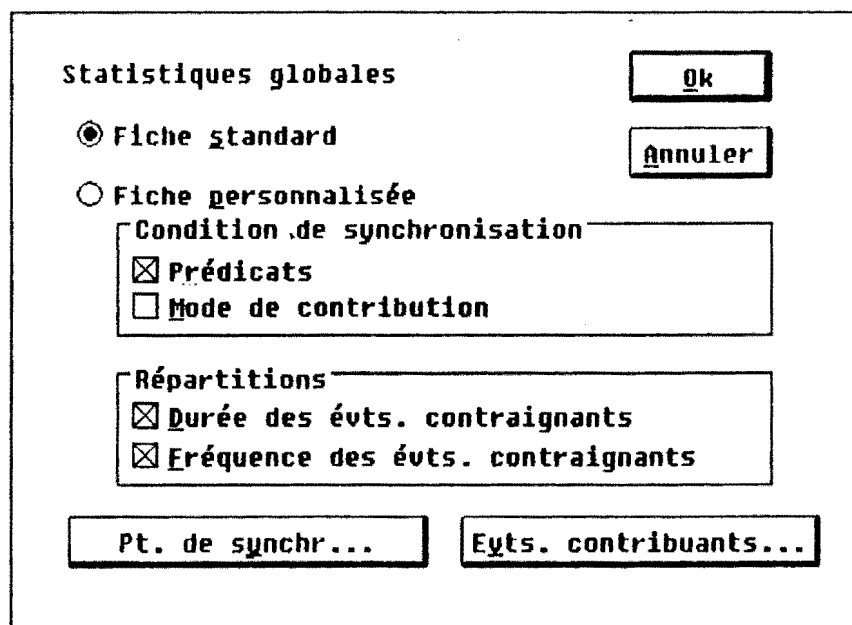


Illustration 47 : Boîte de dialogue de la commande St. globales

- Boîte de dialogue : Il s'agit d'une boîte de dialogue modale extensible. Elle comprend, entre autres :
- Un bouton de commande *Pt. de synchr.*
  - Un bouton de commande *Evs. contributeurs.*
- Chacun de ces deux boutons engendre une deuxième boîte de dialogue.

Point de synchronisation

Nombre de réalisations

Temps de réalisation

Minimum

Moyenne

Maximum

Ecart type

Ok

Annuler

Illustration 48 : Boîte de dialogue engendrée par le bouton *Pt. de synchr.*

Evénements contributeurs

Nombre de contributions

Nombre de participations

Temps de participation

Minimum

Moyenne

Maximum

Ecart type

Ok

Annuler

Illustration 49 : Boîte de dialogue engendrée par le bouton *Evs. contributeurs*

Effets : Cette boîte de dialogue s'utilise de deux manières. L'utilisateur peut choisir d'afficher une fiche standard, ou de consulter une fiche personnalisée. Dans ce deuxième cas, il lui incombe de sélectionner les différentes informations qu'il souhaite voir apparaître dans la fiche. Celles-ci sont rassemblées en quatre groupes en fonction de leur nature :

1) Condition de synchronisation

Cette classe regroupe les informations concernant les prédicats de la condition de synchronisation :

- Prédicats.
- Mode de synchronisation de ces prédicats.

La sélection de la première option provoque l'affichage des prédicats précédés d'un numéro qui sera utilisé par la suite pour identifier ces prédicats. Par exemple,

```

Réalisé quand
  et
  [1] terminaison de P1
      et
  [2] réalisation de S2
  [3] génération de M1

```

Ce texte constitue une seule unité graphique.

Notez que, contrairement à DSL/SIM, nous ne numérotions que les opérandes des prédicats. La numérotation de celles-ci commence à 1.

La sélection de la deuxième option complète le texte ci-dessus en ajoutant entre parenthèses, à côté de chaque prédicat, son mode de contribution : simple ou multiple. Cette option n'est bien sûr disponible qu'après avoir sélectionné la première option.

## 2) Point de synchronisation

Cette classe regroupe les informations concernant le point de synchronisation proprement dit :

- Le nombre de réalisation.
- Les temps minimum, moyen, maximum de réalisation, ainsi que l'écart type.

La sélection de la première option provoque l'affichage d'un texte spécifiant le nombre de réalisation du point de synchronisation. Ce texte constitue une unité graphique :

```

Nombre de réalisations : 7

```

La deuxième option permet à l'utilisateur de choisir entre quatre renseignements sur le temps de réalisation. S'il clique dans la case *Temps de réalisation*, alors les quatre mesures sont automatiquement sélectionnées. Il peut aussi ne sélectionner que certaines de ces mesures (dans ce cas, la case *Temps de réalisation* est automatiquement cochée).

La sélection d'une ces quatre mesures provoque l'affichage d'un texte qui constitue une seule unité graphique.

```

                Temps de réalisation
           Min.      Moy.      Max.      Ecart
 [i]          ...      ...      ...      ...

```

## 3) Événements contributants

Cette classe regroupe les informations concernant les événements qui contribuent à la réalisation du point de synchronisation :

- Le nombre de contribution de chaque type d'événement.
- Le nombre de participation de chaque type d'événement.
- Les temps minimum, moyen, maximum de participation, ainsi que l'écart type.

Le bouton *Evts contributants* qui affiche la boîte de dialogue permettant de sélectionner ces informations, n'est disponible que si la case *Prédicats* a été cochée.

La sélection de la première option provoque l'affichage d'un texte reprenant, pour chaque événement (représenté par son numéro), le nombre de contributions correspondant. Ce texte constitue une unité graphique.

```

Nombre de contributions :
  [1] 7
  [2] 7

```

La sélection de la deuxième option provoque l'affichage d'un texte reprenant, pour chaque événement (représenté par son numéro), le nombre de participations correspondant. Ce texte constitue une unité graphique.

Nombre de participations :

[1]	7
[2]	7

Si deux au moins des options *Nombre de réalisation*, *Nombre de contribution*, *Nombre de participation* sont sélectionnées, alors leurs textes respectifs sont remplacés par un tableau récapitulatif :

	Réal.	Nombre de Contrib.	Partic.
[0]	...		
[i]		...	...

Le point de synchronisation est représenté par le numéro 0 mis entre crochets. Ce tableau constitue une seule unité graphique.

La troisième option permet à l'utilisateur de choisir entre quatre renseignements sur le temps de participation. S'il clique dans la case *Temps de participation*, alors les quatre mesures sont automatiquement sélectionnées. Il peut aussi ne sélectionner que certaines de ces mesures (dans ce cas, la case *Temps de participation* est automatiquement cochée).

Le choix de l'une, au moins, de ces quatre mesures provoque l'affichage d'un tableau reprenant cette (ces) mesure(s) pour chaque événement contribuant :

	Temps de participation			
	Min.	Moy.	Max.	Ecart
[i]	...	...	...	...

Si l'utilisateur souhaite aussi afficher les mesures relatives au temps de réalisation du point de synchronisation, alors le texte de ces mesures est intégré dans le tableau ci-dessus.

	Temps de réal. / partic.			
	Min.	Moy.	Max.	Ecart
[0]	...	...	...	...
[i]	...	...	...	...

Condition de synchronisation			
Réalisé quand			
et			
[1]	terminaison de Nom_proc 7 (simple)		
et			
[2]	réalisation de Nom_pt_synch 1 (simple)		
et			
[3]	génération de Nom_mes 13 (simple)		
Nombre de réal. / contr.	Temps moyen de réal. / part.		
[0]	7	[0]	1h18m30s
[1]	7	[1]	30m10s
[2]	7	[2]	66m45s
[3]	7	[3]	1h15m30s

Illustration 50 : Exemple de fiche

#### 4) Répartitions

Cette classe regroupe les informations concernant la répartition des événements contraignant :

- En durée.
- En fréquence.

Cette classe n'est disponible que si la case *Prédicats* a été cochée.

##### 4.1) Répartition de la durée des événements contraignants

La sélection de la répartition de la durée des événements contraignants provoque l'apparition d'un graphique en forme de camembert mettant en évidence l'importance relative des événements.

Ce camembert est accompagné d'un titre. Chaque quartier est associé au numéro identifiant l'événement. Le camembert et le titre forment une unité graphique indissociable.

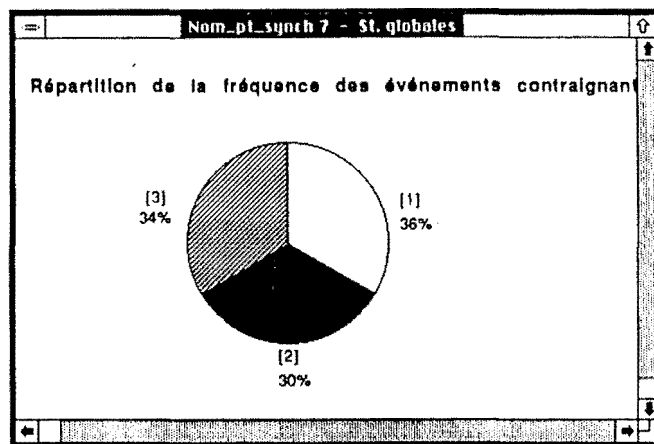


Illustration 51 : Répartition de la durée des événements contraignants

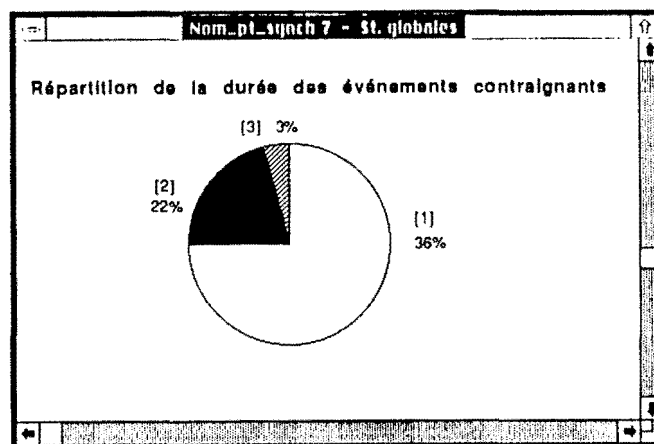


Illustration 52 : Répartition de la fréquence des événements contraignants

#### 4.2) Répartition de la fréquence des événements contraignants

La sélection de la répartition de la fréquence des événements contraignants provoque également l'apparition d'un graphique en forme de camembert mettant en évidence l'importance relative des événements.

Ce camembert est accompagné d'un titre. Chaque quartier est associé au numéro identifiant l'événement. Le camembert et le titre forment une unité graphique indissociable.

Par défaut, lors de la création d'une nouvelle fiche, la boîte de dialogue propose une fiche standard. Les mesures statistiques globales présentes dans cette fiche sont :

- Les prédicats.
- Le nombre de réalisations.
- Le temps moyen de réalisation.
- Le nombre de contributions.
- Le temps moyen de participation.
- La répartition de la durée des événements contraignants.
- La répartition de la fréquence des événements contraignants.

Messages : Aucun.

### 2.7.2 Commentaires

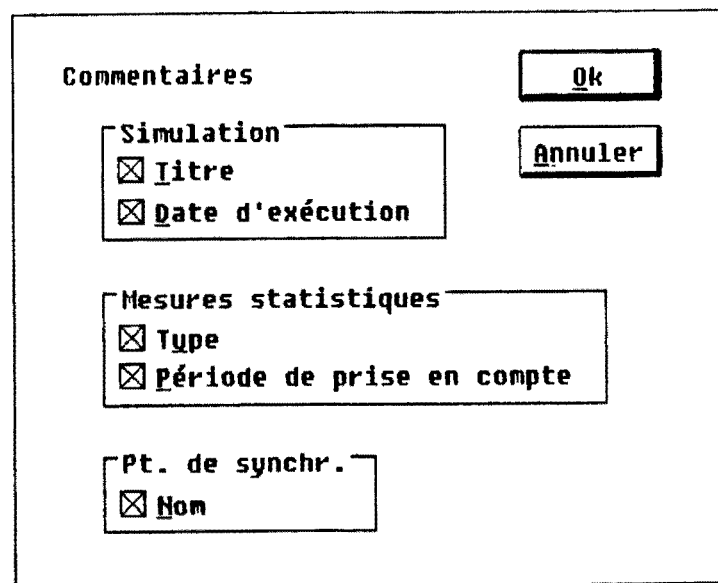


Illustration 53 : Boîte de dialogue de la commande Commentaires

Type : Commande étendue

Mnémo. : M

Accél. : Aucun.

Actif : Quand le document courant est une fiche associée à un point de synchronisation.

- Action : Sélectionner et afficher des informations générales concernant la fiche courante.
- Boîte de dialogue : Il s'agit d'une boîte de dialogue modale :
- Effets : Les différentes informations que l'utilisateur peut ajouter dans la fiche courante sont :
- Le titre de la simulation.
  - La date d'exécution de la simulation.
  - Le type de statistiques de la fiche courante.
  - La période de prise en compte, c'est-à-dire la période durant laquelle les statistiques ont été calculées.
  - Le nom du point de synchronisation.
- Si l'utilisateur sélectionne ces informations, elles apparaîtront au début de la fiche, sous forme de texte.
- Si, lors de la création d'une fiche, l'utilisateur choisit l'option *Fiche standard*, alors toutes les informations de la boîte de dialogue de la commande **Commentaires** sont affichées.
- Chacune de ces informations constitue une unité graphique en soi. Dans le cas où les deux informations d'une boîte de regroupement sont sélectionnées, ces deux commentaires ne forment qu'une seule même unité graphique.
- Messages : Aucun.

### 2.7.3 Définir fiche standard

- Type : Commande.
- Mnémo. : D
- Accél. : Aucun.
- Actif : Quand le document courant est une fiche associée à un point de synchronisation.
- Action : Redéfinir le standard des fiches de même type que celui de la fiche courante.
- Effets : Cette commande a pour effet de modifier l'option *Fiche standard* de la commande **St. globales**. Les fiches standards de même type que celui de la fiche courante seront dorénavant similaires à la fiche courante.
- Cette commande modifie donc la présentation des boîtes de dialogue de sélection des informations (changement des options par défaut).
- Messages : Cette commande est potentiellement dangereuse. Le message de confirmation habituel s'avère nécessaire.

## 2.8 MENU ECRAN

Le menu **Ecran** a pour fonctionnalité première d'offrir à l'utilisateur les outils et les informations nécessaires pour gérer efficacement l'affichage des différentes fenêtres.

On peut considérer ce menu comme étant composé de trois parties. Une première partie offre la possibilité d'afficher une fenêtre d'aide ou une fenêtre d'informations concernant la simulation exploitée. Une deuxième partie présente trois items de gestion de fenêtres. Enfin une troisième partie affiche la liste des documents (fiches, ou rapports) présents sur le bureau, masqués ou non.

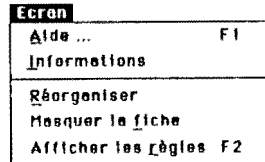


Illustration 54 : Menu Ecran

### 2.8.1 Aide

Type : Commande étendue.

Mnémo. : A

Accél. : F1

Actif : Toujours.

Action : Offrir à l'utilisateur des indications, des renseignements concernant la manière de se servir de l'application MEANDRES.

Boîte de dialogue : Non encore déterminée<sup>1</sup>.

Effets : Non encore déterminés.

Messages : Non encore déterminés.

### 2.8.2 Informations

Type : Commande.

Mnémo. : I

Accél. : Aucun.

Actif : Dès qu'une simulation a été associée à la fenêtre principale.

Action : Afficher un certain nombre d'informations concernant la simulation.

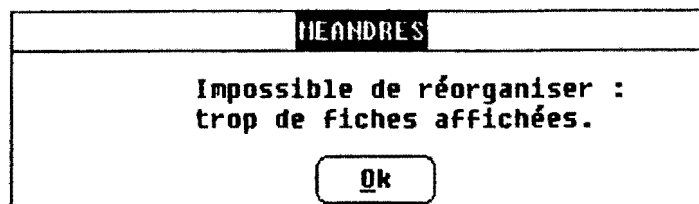
<sup>1</sup> Pour plus de renseignements, se référer au chapitre suivant.



- Boîte de dialogue :** Il ne s'agit pas à proprement parler d'une boîte de dialogue, mais plutôt d'une fenêtre de type logo. Outre un bouton de commande *Ok*, cette fenêtre contient les indications suivantes :
- Nom de la simulation.
  - Date de la simulation.
  - Nombre de processus.
  - Nombre de ressources.
  - Nombre de points de synchronisation.
- Effets :** Afficher la fenêtre décrite ci-dessus. Ces informations ne sont pas transférables sur un rapport.
- Messages :** Aucun.

### 2.8.3 Réorganiser

- Type :** Commande.
- Mnémo. :** R
- Accél. :** Aucun.
- Actif :** Si une fiche, au moins, est affichée à l'écran.
- Action :** Réorganiser l'affichage des fiches présentes à l'écran afin de pouvoir toutes les visualiser sans qu'elles ne se recouvrent.
- Effets :** Cette commande a pour effet de recalculer les paramètres de positionnement et de taille de chacune des fiches présentes à l'écran (c'est-à-dire des fiches dont le nom repris dans la liste des documents du menu **Ecran** n'est pas mis entre parenthèses; cfr ci-dessous). Ces calculs visent à déterminer une taille et une position optimale afin de pouvoir afficher toutes ces fiches sans qu'elles ne se recouvrent (configuration en mosaïque ou en damier).
- Rappelons que les fiches ne peuvent être affichées en dehors de la zone de travail de la fenêtre principale. Par conséquent, plus il y a de fiches présentes à l'écran, et plus la zone de travail de la fenêtre principale est petite, moins la nouvelle taille des fiches sera grande. Il pourrait même arriver qu'il ne soit pas possible de procéder à la réorganisation si la nouvelle taille des fiches devait être inférieure à la taille minimale autorisée.
- Messages :** Si le nombre de fiches présentes à l'écran et la taille de la zone de travail de la fenêtre principale sont tels que la nouvelle taille des fiches devrait être inférieure à la taille minimale autorisée :



## 2.8.4 Masquer la fiche

Type : Commande.

Mnémo. : F

Accél. : Aucun.

Actif : Si le document courant est une fiche.

Action : Masquer la fiche courante.

Effets : Outre masquer la fiche courante, cette commande a pour effets de :

- Supprimer la coche devant le nom de la fiche : une fiche masquée ne saurait être le document courant.
- Mettre le nom de cette fiche entre parenthèses (symbole du masquage).
- Mettre une coche devant le nom de la fiche qui suit la fiche masquée dans la liste des documents, ou devant le nom du rapport courant s'il n'existe pas d'autre fiche (cfr ci-dessous).

Pour réafficher la fiche masquée, l'utilisateur devra la sélectionner dans la liste des documents du menu **Ecran**.

Il est utile d'offrir à l'utilisateur la possibilité de masquer simultanément toutes les fiches. C'est intéressant si, par exemple, l'utilisateur souhaite travailler la présentation d'un rapport et donc visualiser complètement la zone de travail de la fenêtre principale. Pour masquer toutes les fiches, l'utilisateur doit enfoncer la touche *Majuscule* tout en sélectionnant le menu **Ecran**. Dans ce cas, la commande **Masquer la fiche** est remplacée par la commande **Masquer toutes les fiches**. Le caractère mnémonique de cette commande reste F.

Messages : Aucun.

## 2.8.5 Afficher les règles

Type : Commande.

Mnémo. : R

Accél. : F2

Actif : Dès qu'une simulation a été associée à la fenêtre principale.

Action : Afficher les deux règles sur la zone de travail de la fenêtre principale.

Effets : Cette commande modifie la présentation de la zone de travail : affichage de la règle horizontale et de la règle verticale, décalage du contenu. De plus, si le rapport courant est le document courant, alors le curseur est prolongé par deux lignes fines qui permettent de déterminer sa position avec précision sur les règles.

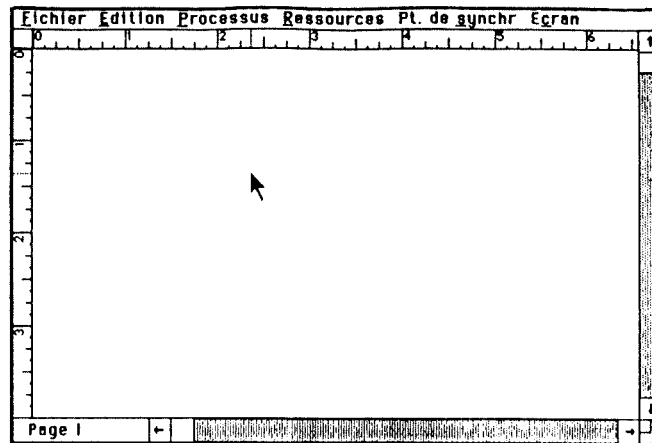


Illustration 55 : Règles

Les règles sont graduées en centimètres. La règle horizontale est située au sommet de la zone de travail de la fenêtre principale; la règle verticale est située contre le bord gauche de la zone de travail. Elles ont leur extrémité initiale (0 cm) commune; la valeur de leur extrémité finale dépend de la taille du papier sélectionné. L'utilisateur n'a pas la possibilité de changer les graduations des règles.

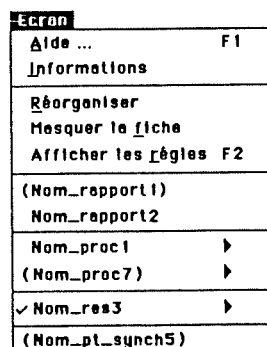
Les règles constituent le moyen mis à la disposition de l'utilisateur pour lui permettre d'insérer, de déplacer des informations avec précision sur le rapport courant.

La sélection de cette commande modifie également le menu **Ecran** : l'item **Afficher les règles** devient **Masquer les règles**. Le mnémonique et le raccourci de cet item sont inchangés. Le choix de l'item **Masquer les règles** efface les règles de la zone de travail de la fenêtre principale et provoque le retour de l'item **Afficher les règles**.

Messages : Aucun.

## 2.8.6 Liste des documents

Pour bien utiliser la commande **Masquer la fiche**, il faut savoir quelle est la fiche courante. De même, lors de la définition de certaines commandes, nous avons maintes fois utilisé les concepts de fiche courante, de rapport courant, de document courant. Pour utiliser au mieux toutes ces commandes, il est donc essentiel de fournir à l'utilisateur une liste lui renseignant, à tout moment, l'ensemble des documents présents sur le bureau, ainsi que leur état (courant, ou non; masqué, ou non).

Illustration 56 : Menu **Ecran** avec une liste de documents

Cette liste est en fait composée de deux listes : la liste des rapports et la liste des fiches, elle-même divisée en trois sous-listes, une liste par type d'objets. Nous avons donc, par ordre, la liste des rapports, la liste des fiches associées à un processus, celle des fiches associées à une ressource et celle associée à un point de synchronisation.

### Liste des rapports

Cette liste contient le nom de tous les rapports présents sur le bureau, c'est-à-dire de tous les rapports créés ou ouverts mais non encore fermés. Parmi tous ces rapports, un seul est visible : c'est le rapport courant, c'est-à-dire celui qui est affiché dans la zone de travail de la fenêtre principale. La liste des rapports fournit le moyen de distinguer le rapport courant (affiché) des rapports non courants (masqués) : le nom des rapports masqués est mis entre parenthèses.

Si l'utilisateur souhaite afficher un rapport masqué, il lui suffit de le sélectionner dans la liste. Cette sélection aura également pour effet de modifier la liste : le nom de l'ancien rapport courant est mis entre parenthèses; le nom du nouveau rapport courant est débarrassé de ses parenthèses. De plus, comme le rapport courant devient le document courant, son nom est précédé d'une coche.

### Liste des fiches

Cette liste contient le nom de toutes les fiches présentes sur le bureau, c'est-à-dire de toutes les fiches actuellement ouvertes. Nous avons vu que chaque fiche correspond à un objet et à un type de statistiques. Ainsi, il peut arriver d'avoir à l'écran trois fiches correspondant à un même objet (fiches sœurs) : une fiche pour les statistiques globales, une pour les statistiques périodiques, et une pour les statistiques chronologiques.

Pour éviter que la liste ne s'allonge outre mesure, nous proposons de n'y reprendre que le nom de l'objet (par exemple, **nom\_de\_processus**). Cet item est en réalité un menu déroulant dont la sélection provoque l'apparition d'items : **Glob.**, **Périod.**, **Chron.** Ces items indiquent quelles sont les fiches associées à cet objet actuellement présentes. Précisons que le nom d'un point de synchronisation n'est pas un menu déroulant, puisque les points de synchronisation ne possèdent qu'un seul type de statistiques (globales).

Nom_proc1	▶	Glob.
(Nom_proc7)	▶	(Périod.) Chron.

Illustration 57 : Fiche présente

Si une fiche est masquée, le nom de l'objet (dans le cas d'un point de synchronisation) ou le type de statistiques (dans les autres cas) est mis entre parenthèses. De plus, si toutes les fiches sœurs d'un objet sont masquées, alors son nom est mis entre parenthèses dans le menu **Ecran**.

Nom_proc1	▶	(Glob.)
(Nom_proc7)	▶	(Chron.)

Illustration 58 : Fiches sœurs masquées

Si une fiche affichée devient courante, une coche apparaît devant le nom de l'objet correspondant. De plus, si l'objet possède plusieurs fiches sœurs présentes, alors le type de la fiche courante sera également précédé d'une coche.

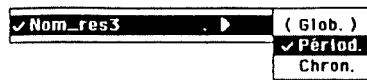


Illustration 59 : Fiche courante

Si une fiche masquée devient courante, non seulement ce qui vient d'être décrit dans le cas de la fiche affichée reste valable, mais les parenthèses qui entouraient le nom de l'objet, et éventuellement celles qui entouraient le type de statistiques disparaissent.

Nous venons de décrire le mécanisme de sélection de fiches dont nous avons laissé entendre l'existence quand nous avons défini la commande **Sélectionner un objet**. Dans la remarque faite à l'époque, nous disions également qu'il existe une méthode pour afficher une nouvelle fiche qui possède une fiche sœur déjà présente. Il suffit de rendre courante une des fiches sœurs et d'utiliser la commande de sélection des informations statistiques désirées (**St. globales**, **St. périodiques**, **St. chronologiques**).

## 2.9 SELECTION ET DEFORMATION

### 2.9.1 Mécanismes de sélection

Nous avons vu que la commande **Tout sélectionner** du menu **Edition** permet de sélectionner en une seule fois toutes les informations présentes dans le document courant. Bien que cette commande soit utile, il est nécessaire d'offrir à l'utilisateur des moyens pour ne sélectionner que des informations précises.

Rappelons que quand une information est sélectionnée, un rectangle plein l'entoure et que, dans le cas d'un rapport, trois poignées apparaissent : une poignée de déformation horizontale, une poignée de déformation verticale, et une poignée de déformation homothétique. De plus, si l'information sélectionnée fait partie d'un rapport, alors la zone d'information de la fenêtre principale affiche les dimensions de cette information.

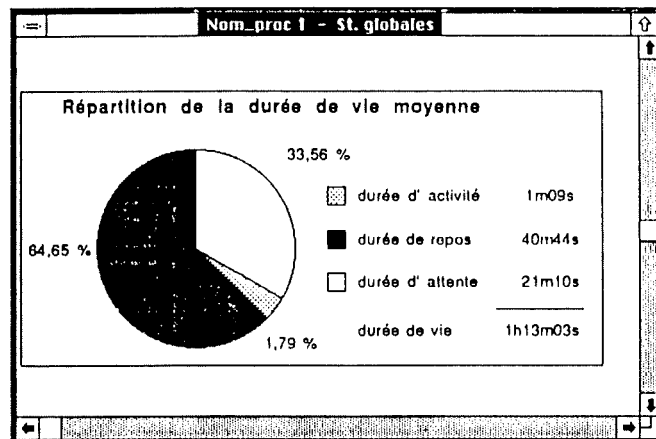


Illustration 60 : Sélection sur une fiche

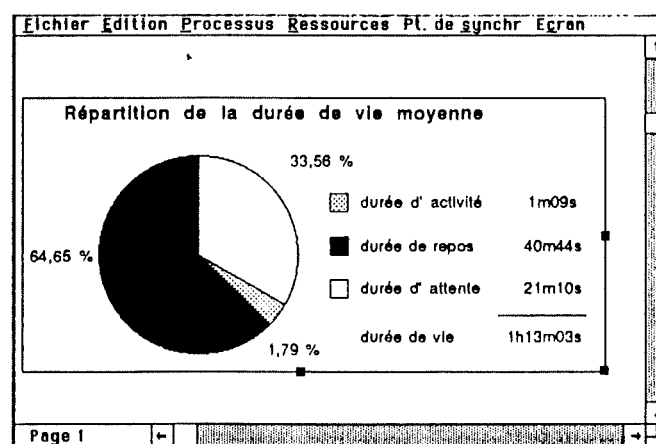


Illustration 61 : Sélection sur un rapport

## Sélectionner avec la souris

### Pour sélectionner une information

- Cliquer à l'intérieur de l'information.

ou

- Glisser le pointeur flèche (qui se transforme en pointeur main au cours de l'opération) de façon à entourer l'information d'une zone pointillée. Quand le bouton de la souris est relâché, l'information est sélectionnée.

### Pour sélectionner plusieurs informations

- Cliquer sur une première information, puis en maintenant la touche *Contrôle* enfoncée, cliquer sur les autres informations à sélectionner. La zone d'information de la fenêtre principale affiche les dimensions de la dernière information sélectionnée (dans le cas d'un rapport).

ou

- Cliquer sur la première information, puis en maintenant la touche *Majuscule* enfoncée, cliquer sur la dernière information à sélectionner. Toutes les informations comprises entre ces deux bornes sont automatiquement sélectionnées. La zone d'information de la fenêtre principale affiche les dimensions de la dernière information sélectionnée (dans le cas d'un rapport).

ou

- Glisser le pointeur flèche (qui se transforme en pointeur main au cours de l'opération) de façon à entourer toutes les informations à sélectionner d'une zone pointillée. Si les informations à sélectionner ne sont pas contiguës, procéder à plusieurs glissements tout en maintenant enfoncée la touche *Contrôle*. La zone d'information de la fenêtre principale affiche les dimensions de la première information du groupe sélectionné, c'est-à-dire l'information située le plus en haut à gauche de toutes celles sélectionnées (dans le cas d'un rapport).
- Glisser le pointeur flèche (qui se transforme en pointeur main au cours de l'opération) de façon à entourer toutes les informations à sélectionner d'une zone pointillée. S'il reste des informations contiguës au groupe à sélectionner, cliquer sur la dernière information (c'est-à-dire la plus éloignée du groupe déjà sélectionné), en maintenant la touche *Majuscule* enfoncée<sup>1</sup>. La zone d'information de la fenêtre principale affiche les dimensions de la dernière information sélectionnée (dans le cas d'un rapport).

### Pour désélectionner

Cliquer en dehors d'une information. La zone d'information de la fenêtre principale affiche de nouveau le numéro de la page courante (dans le cas d'un rapport).

Remarque : il n'est pas nécessaire d'entourer complètement l'information à sélectionner d'une zone pointillée : toute information comprise en partie, ou en totalité, dans la zone pointillée est sélectionnée.

## Sélectionner avec le clavier

Le clavier nécessite de faire une distinction entre l'information courante et l'information sélectionnée. En effet, les touches doivent permettre d'une part de passer d'une

---

<sup>1</sup> Cette manière d'étendre la sélection d'un groupe est valable quelle que soit la méthode utilisée pour sélectionner le groupe initial.

information à l'autre, et d'autre part de (dé)sélectionner l'information courante s'il y a lieu.

L'information courante se distingue des autres par le rectangle pointillé qui l'entoure. Les informations sélectionnées se repèrent par la présence du rectangle plein. En outre, un rectangle pointillé ne possède jamais de poignées de déformation.

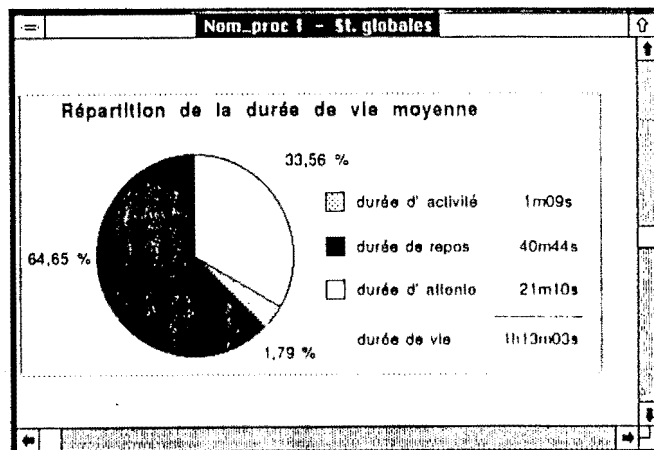


Illustration 62 : Information courante

**TAB** Rend courante l'information qui suit l'information courante. Le sens de parcours est de gauche à droite et de haut en bas. Si aucune information n'était courante, alors la première information du document devient courante.

**MAJ+TAB** Rend courante l'information qui précède l'information courante. Le sens de parcours est de droite à gauche et de bas en haut. Si aucune information n'était courante, alors la dernière information du document devient courante.

**BARRE D'ESPACEMENT** Sélectionne l'information courante si elle n'était pas sélectionnée et réciproquement. Lors de la désélection d'une information d'un rapport, la zone d'information de la fenêtre principale affiche les dimensions de la dernière (selon l'ordre chronologique) information sélectionnée.

La sélection d'une information ne provoque pas la désélection des autres informations sélectionnées.

## 2.9.2 Modifier la taille des informations

La taille des informations du rapport courant peuvent ne pas convenir à l'utilisateur. Par conséquent, il est utile de lui permettre de les redimensionner.

Pour ce faire, il doit préalablement sélectionner l'information dont il souhaite modifier la taille. Nous avons vu que la zone d'information de la fenêtre principale affiche les dimensions de cette information.



Illustration 63 : Zone d'information de la fenêtre principale



Cette zone donne trois renseignements : la longueur et la hauteur (en centimètres), et le pourcentage de surface occupée actuellement par l'information par rapport à sa surface initiale (c'est-à-dire au moment du collage).

Pour modifier la longueur (la hauteur), il suffit que l'utilisateur glisse la poignée de déformation horizontale<sup>1</sup> (verticale). Pour modifier l'information en gardant les proportions, l'utilisateur doit utiliser la poignée de déformation homothétique. Au fur et à mesure de ces glissements, la zone d'information reflète le redimensionnement.

Si plusieurs informations sont sélectionnées, l'utilisateur peut procéder de la même manière pour modifier la taille de l'information dont les dimensions sont affichées dans la zone d'information de la fenêtre principale. Attention ! Les modifications seront portées de manière proportionnelle sur chacune des informations sélectionnées. Ainsi, par exemple, si la longueur de l'information dont les dimensions sont affichées dans la zone d'information de la fenêtre principale passe de 10 cm à 5 cm, alors la longueur de toutes les autres informations sélectionnées est réduite également de moitié (et non pas nécessairement de 5 cm). Cet artifice n'est pas dénué d'intérêt dans la mesure où il arrivera fréquemment que l'utilisateur souhaite modifier isomorphiquement des graphiques semblables.

Si l'utilisateur ne dispose pas de souris, il peut utiliser les flèches de direction pour modifier la taille des informations :

- GAUCHE* : Réduire la longueur d'un millimètre.
- DROITE* : Augmenter la longueur d'un millimètre.
- HAUT* : Réduire la hauteur d'un millimètre.
- BAS* : Augmenter la hauteur d'un millimètre.
- CTRL+GAUCHE* : Réduire proportionnellement la longueur et la hauteur d'une unité.
- CTRL+DROITE* : Augmenter proportionnellement la longueur et la hauteur d'une unité.

---

<sup>1</sup> Le terme "poignée de déformation horizontale" peut prêter à confusion. C'est en fait la déformation qui est horizontale, et non la poignée de sélection. La poignée de déformation horizontale se trouve sur le côté vertical du rectangle.

---

## Commentaires et justifications

### 3.1 REGLES ERGONOMIQUES

Nous ne comptons pas justifier toute l'ergonomie de notre l'interface : beaucoup de justifications sont inutiles parce qu'évidentes. Cette remarque concerne essentiellement la conception formelle des objets interactifs. Sachez que nous avons essayé de suivre un certain nombre de règles ergonomiques élémentaires dont nous dressons ci-après une liste non exhaustive. Nous n'avons pas la prétention de croire que nous les avons toujours respectées tant cela demande de la vigilance et de la rigueur.

Cette liste est inspirée des recommandations émises par IBM [CUA89].

#### Barre des menus et menus déroulants

- Bannir les nombres.
- Donner à chaque menu de la barre et à chaque item de chaque menu déroulant un mnémonique. Ce mnémonique doit être un seul caractère qui identifie de façon unique l'item associé. Le caractère rappelant le mnémonique doit être souligné.
- Mettre en majuscule uniquement la première lettre, à l'exception des noms propres, des abréviations et des acronymes.
- Pour la barre de menus,
  - Utiliser de préférence un seul mot.
  - Placer les menus dans l'ordre suivant : **F**ichier, **E**dition, les menus propres à l'application, **A**ide.
- Pour les menus déroulants,
  - Placer les choix les plus fréquents en tête.
  - Regrouper les items sémantiquement semblables à l'aide d'une ligne de séparation.

- Faire suivre par trois points (...) les items qui provoquent l'affichage d'une boîte de dialogue.
- Associer un accélérateur aux items les plus fréquemment choisis.

#### Boîtes de dialogue

- Regrouper les contrôles sémantiquement semblables en une seule ligne ou en une seule colonne à partir du coin supérieur gauche de la boîte de dialogue.
- Aligner les champs d'édition ou de sélection verticalement et horizontalement.
- Disposer les contrôles en fonction de l'ordre (s'il existe) suivant lequel les utilisateurs doivent les parcourir.
- Regrouper les contrôles évidents pour les utilisateurs dans des boîtes de regroupement correctement espacées.
- Aligner les boîtes de regroupement lorsque c'est possible. On peut étendre ces boîtes vers la droite de manière à obtenir des largeurs uniformes.
- Utiliser des énoncés concis, explicites et identifiants pour décrire le sens des groupes de choix.
- Mettre en majuscule uniquement la première lettre, à l'exception des noms propres, des abréviations et des acronymes.
- Proposer des choix par défaut (présélectionnés) afin de minimiser le travail de l'utilisateur (diminuer la charge informationnelle).

#### Textes

- Un texte placé au dessus d'un champ doit être aligné sur le bord gauche de ce champ.
- Un texte placé à gauche d'un champ doit être aligné avec le début des autres textes.
- Il faut utiliser les règles typographiques (majuscule, ponctuation, ...) courantes.

## **3.2 FENETRES**

### **3.2.1 Fenêtre principale**

Il nous a été signalé qu'utiliser la zone de travail de la fenêtre principale pour afficher le rapport courant favorise l'aspect "rapport" vis-à-vis de l'aspect "consultation". C'est tout à fait exact, mais telle n'était pas notre intention.

Ce choix découle de la force de l'habitude. En effet, la zone de travail de tous les logiciels (multi-fenêtrés) que nous utilisons habituellement est toujours occupée par l'affichage d'informations : textes, tableaux, dessins, champs de données, ... Par conséquent, il nous semblait peu conventionnel de laisser la zone de travail vide, d'autant plus que nous avons besoin d'un espace de visualisation pour les rapports.

Il existe au moins une alternative à la zone de travail pour présenter les rapports. Elle est présentée dans le chapitre suivant.

### **3.2.2 Fiche**

a) Nous interdisons qu'une fiche sorte de la zone de travail de la fenêtre principale car le concept de fiche est assujéti à l'application MEANDRES. Par conséquent, nous estimons que déplacer la fenêtre d'une fiche en dehors de celle de l'application n'a pas de sens.

On pourrait objecter que pour des raisons de lisibilité, il serait pourtant intéressant que la fenêtre d'une fiche ait sa propre liberté de déplacement et de "dimensionnement" de sorte que quand l'utilisateur examine une fiche, il puisse la visualiser pleinement sans devoir au préalable modifier la taille de la fenêtre principale.

Nous sommes d'accord avec cet argument, mais nous lui préférons un autre. En effet, toujours pour des raisons de lisibilité, on est en droit craindre qu'une profusion de fiches pouvant être déplacées en toute liberté n'entraîne une anarchie visuelle. La question "à quelle application appartient cette fenêtre ?" risque de devenir monnaie courante, surtout si l'utilisateur lance simultanément plusieurs exécutions de MEANDRES. En interdisant aux fiches de sortir de la fenêtre principale, le problème ne se pose plus.

Les partisans de la liberté des fenêtres filles pourraient alors prétendre résoudre le problème en ajoutant explicitement, sur la barre de titre par exemple, le nom de l'application qui a créé ces fenêtres filles. Nous pourrions objecter à ce moment que cela nuit à la concision de la présentation ...

Nous pensons que les deux solutions se valent, bien que nous préférons la première en raison de sa cohérence conceptuelle. Nous tenons à signaler au lecteur que Ms-Windows 3 adopte également cette solution pour gérer ses fenêtres filles<sup>1</sup>.

b) Lorsqu'on modifie la taille d'une fiche, faut-il que les informations contenues dans cette fenêtre reflètent ce changement de taille ? Voilà une question qui suscite beaucoup de polémiques.

Les partisans de cette solution avancent que cet artifice fournit un mécanisme de gestion de la taille de données très souple. Nous reconnaissons la véracité de cet argument, mais uniquement dans le cas où la fenêtre ne contient qu'une seule information : un graphique. Dans les autres cas, nous partageons l'opinion des détracteurs qui estiment que le "redimensionnement" d'une fenêtre et la gestion de la taille d'informations sont deux actions qui doivent être distinctes. En effet, on peut fort bien "redimensionner" une fenêtre pour éviter le recouvrement d'une autre fenêtre, et vouloir conserver la taille de données estimée optimale. Ainsi, à un mécanisme qui lie deux actions au nom de la rapidité, nous préférons un mécanisme qui scinde deux actions au nom de la liberté. Bien sûr, un rétrécissement de la fenêtre diminue la quantité d'informations visualisables, mais mieux vaut visualiser moins de données lisibles que plus de données illisibles.

Il va de soi qu'au moment d'implémenter notre outil, il va falloir choisir rigoureusement la taille des informations affichées sur les fiches en fonction de la taille moyenne de ces fenêtres de façon à minimiser le nombre de scrolling.

---

<sup>1</sup> Nous avons élaboré cette gestion des fenêtres filles avant de prendre connaissance de l'environnement Ms-Windows 3.

### 3.3 MENU FICHIER

Le lecteur a sans doute remarqué la rareté des accélérateurs. Nous devons bien avouer notre perplexité en ce domaine. En effet, contrairement au Macintosh, Windows permet facilement d'activer un item d'un menu à partir du clavier : dans un premier temps, l'utilisateur déroule le menu en tapant simultanément sur *Alt* et sur la touche correspondant au caractère qui est souligné dans le nom du menu; ensuite, l'utilisateur tape sur la touche correspondant à la lettre qui est soulignée dans le nom de l'item choisi.

Bien que cette manipulation nécessite une étape de plus que dans le cas d'un véritable accélérateur, il nous semble qu'elle peut faire office d'accélérateur honorable. Bien sûr, un véritable accélérateur est plus performant, mais il nécessite d'être mémorisé. De plus, imaginez la lisibilité des menus si on associait un accélérateur à chaque item !

Alors, peu ou beaucoup d'accélérateurs ?

Pour répondre objectivement à cette question, il faudrait déterminer les items les plus couramment utilisés et associer un accélérateur à ces seuls items. Devant la difficulté de déterminer actuellement cette fréquence, nous nous sommes contentés de suivre les recommandations en la matière, en étant bien conscients qu'il serait sans doute préférable d'offrir plus d'accélérateurs.

#### 3.3.1 Nouveau

a) Nous pensons que le terme "nouveau" est suffisamment précis dans la mesure où l'application MEANDRES n'est capable de créer qu'un seul type de fichiers, à savoir des fichiers contenant des rapports. Par conséquent, nous n'avons pas jugé utile de préciser le type de l'objet que cette commande crée (nous n'avons donc pas retenu la formulation **Nouveau rapport**).

b) L'item **Nouveau** est toujours actif, même si aucune simulation n'est associée à la fenêtre principale. L'utilisateur peut donc créer autant de rapports qu'il veut avant d'ouvrir le fichier contenant les données d'une simulation, un peu comme s'il achetait des cahiers avant de savoir ce qu'il va y inscrire. On imagine qu'un utilisateur sensé associera d'abord une simulation à la fenêtre principale avant de créer un nouveau rapport d'autant plus que, par défaut, il dispose toujours d'un rapport, mais en vertu du principe de liberté, nous ne pouvons pas interdire de procéder autrement.

#### 3.3.2 Ouvrir

a) La définition que nous avons donnée d'une boîte de dialogue modale n'est pas assez précise. Nous avons dit que l'utilisateur est incapable de faire quoi que ce soit tant qu'il n'a pas introduit les informations demandées par la boîte de dialogue. Dans un environnement multi-tâches tel que Ms-Windows, cette restriction ne concerne que les actions associées à l'occurrence de l'application qui a engendré la boîte de dialogue. En effet, bien qu'une boîte de dialogue modale soit affichée, l'utilisateur peut passer à une autre application et travailler avec cette dernière sans que la boîte de dialogue ne manifeste son mécontentement.

La contrainte induite par ce type de boîtes de dialogue semble être en contradiction avec notre souci d'offrir le plus de liberté possible aux utilisateurs et de leur laisser

le contrôle explicite des actions. Il ne faut toutefois pas confondre liberté et laxisme : l'absence d'une directivité minimale peut être aussi néfaste qu'une directivité envahissante. Il nous semble que lorsqu'une boîte de dialogue est utilisée pour préciser une action, elle forme un tout indivisible avec cette action. On peut supposer qu'un utilisateur "fréquent" est capable de gérer à son avantage l'éventuelle décomposition de l'action, mais certainement pas un utilisateur novice. Comme tout utilisateur est un jour novice ...

L'utilisation de boîtes de dialogue non modales doit donc être minimisée. Nous reconnaissons que dans certaines circonstances, de telles boîtes de dialogue sont bienvenues. Ainsi en est-il de la boîte de dialogue qui gère dans certains logiciels la recherche d'un mot dans un texte : la recherche étant un processus itératif, on conçoit aisément qu'il serait fastidieux d'avoir à passer par le menu déroulant à chaque itération. Ainsi en est-il également de la boîte de dialogue qui gère les styles de Ms-Word 4 (Macintosh) : elle évite la redondance de certaines actions (couper, ouvrir, ...) en permettant d'avoir accès aux items de menus plutôt qu'en créant des boutons superflus.

b) Il existe une autre boîte de dialogue pour la commande **Ouvrir**. Celle-ci scinde en deux listes les fichiers et les répertoires. Notre préférence va à cette solution pour des raisons de clarté, mais les normes de Ms-Windows 2 recommandent la boîte de dialogue présentée dans le chapitre précédent. Heureusement, Ms-Windows 3 recommande maintenant l'autre boîte de dialogue.

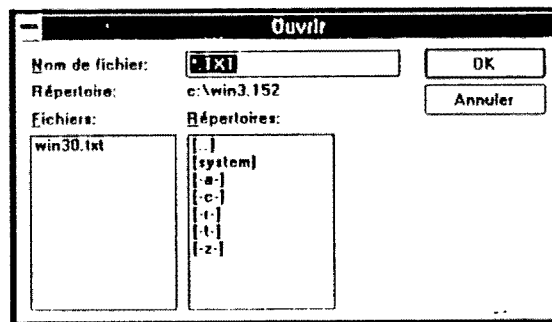


Illustration 64 : Autre boîte de dialogue pour la commande **Ouvrir**

c) La commande **Ouvrir** permet d'ouvrir deux types de fichiers. Peut-être aurait-il fallu souligner explicitement cette dichotomie en proposant deux items : **Ouvrir une simulation** et **Ouvrir un rapport** ? Nous ne l'avons pas fait pour deux raisons :

- Etablir cette distinction au niveau du menu **Fichier** introduit des termes qui sont étrangers à ce menu. En effet, le menu **Fichier** gère des fichiers en tant que structures de données quelle que soit la signification qu'on leur associe. Par conséquent les termes "rapport" et "simulation" sont incongrus dans ce contexte.
- Nous estimons que les suffixes SIM et RAP sont suffisamment explicites pour permettre à un utilisateur d'utiliser correctement la commande **Ouvrir**. Bien sûr, il s'agit là d'une hypothèse délicate qui suppose que les utilisateurs sont capables de percevoir une simulation en tant que fichier. Etant donné le public auquel s'adresse notre outil, nous pensons que nous pouvons faire cette hypothèse.

Nous devons bien reconnaître que nous ne sommes pas entièrement satisfaits puisque nous basons une différence conceptuelle propre à la tâche du concepteur sur des notions de syntaxe étrangères à cette même tâche. Par conséquent, on peut nous adresser les mêmes reproches que ceux invoqués ci-dessus pour rejeter la scission en deux items !

d) La possibilité de pouvoir ouvrir une autre simulation à partir de l'application MEANDRES est une nouveauté importante dans le monde de Ms-Windows. En effet,

quand l'utilisateur décide de consulter une nouvelle simulation, il lance implicitement l'exécution d'une nouvelle occurrence de MEANDRES. Ceci est contraire aux habitudes qui forcent un utilisateur à revenir au niveau de l'Exécutif Ms-Dos pour lancer l'exécution d'un programme.

En conception d'interfaces homme/machine, le respect des habitudes est très important afin de ne pas déstabiliser les utilisateurs, mais nous pensons que dans notre cas, ce non-respect n'est pas un handicap dans la mesure où nous croyons que l'utilisateur a plus une vision d'une simulation en tant que tâche plutôt qu'en tant que logiciel. Par conséquent quand il ouvre une nouvelle simulation, ce qui lui importe c'est de disposer d'un nouvel espace de travail. Que ce dernier découle de l'exécution d'une nouvelle occurrence du programme plutôt que de celui en cours n'est sans doute pas une préoccupation prépondérante.

Les propos que nous venons de tenir n'induisent pas que nous pensons que l'utilisateur ne doit pas avoir conscience de la notion de programme. Ce serait une erreur puisque tôt ou tard il sera confronté à cette réalité : il doit gérer la fenêtre (place, dimension,...), quitter l'application,... Nous avons seulement essayé d'intégrer les notions de fenêtre et d'occurrence d'application dans l'univers conceptuel de la tâche, au nom de la réduction du fossé séparant l'univers psychologique de l'utilisateur et celui physique de l'outil.

Notre tentative est sans doute maladroite pour plusieurs raisons :

- Elle est redondante avec l'Exécutif Ms-Dos.
- Elle peut gêner des utilisateurs habitués à l'environnement de Ms-Windows 2.
- Elle mêle le concept de fichier à celui d'application.
- Elle renforce le cloisonnement entre les consultations de simulations différentes.

Malgré toutes ces raisons, nous avons adopté cette technique non seulement parce qu'elle offre un raccourci utile (on ne quitte pas l'univers de la tâche), mais aussi parce qu'elle a le mérite de provoquer un débat sur le bien-fondé du respect des habitudes.

### **3.3.3 Sélectionner un objet**

a) Quand une boîte de dialogue permet d'afficher une deuxième boîte de dialogue, l'annulation de cette dernière fait retourner l'utilisateur à la première boîte de dialogue.

Cette règle est évidente dans le cas d'une sous-boîte de dialogue (c'est-à-dire d'une boîte de dialogue qui permet d'afficher des informations qu'on n'a pas su ou voulu mettre sur la boîte de dialogue principale; cette boîte de dialogue s'affiche au dessus de la première boîte de dialogue; cfr boîte de dialogue de la commande **St. globales** du menu **Processus**).

Elle l'est peut-être moins dans le cas de l'enchaînement de deux boîtes de dialogue de même niveau. On pourrait décider que l'annulation de la deuxième boîte de dialogue entraîne l'annulation de l'action; ainsi l'annulation de chacune des deux boîtes aurait le même effet.

Cette solution ne nous semble pas opportune car elle ne tient pas compte de l'histoire de l'action. Dans le cas de la commande **Sélectionner un objet** la première boîte de dialogue permet de choisir un objet. Cela fait, la deuxième boîte permet de choisir les résultats statistiques d'un certain type concernant cet objet. Dès lors, l'annulation de cette deuxième boîte de dialogue doit renvoyer à la première boîte de dialogue de sorte que l'utilisateur puisse choisir un autre type de statistiques, ou un autre objet, ou enfin annuler l'action de sélection. Autrement dit, l'annulation pure et

simple du dialogue obligerait inutilement l'utilisateur à recommencer l'étape de la sélection d'un objet.

b) La boîte de dialogue spécifique à l'item **Sélectionner un objet** peut paraître complexe. Nous sommes pourtant persuadés qu'elle se révélera très pratique, et plus simple à utiliser qu'il n'y paraît de prime abord. Elle constitue l'outil privilégié de navigation : c'est un véritable échangeur autoroutier !

C'est dans cette optique que nous avons souhaité lier la liste des processus à celle des ressources. En effet, quand on examine la tâche de consultation, il ressort rapidement qu'il est indispensable que l'utilisateur puisse passer rapidement de l'examen d'un processus à celui des ressources qu'il requiert (et inversement). Nous estimons que notre boîte de dialogue permet d'atteindre cet objectif.

Initialement, nous pensions également lier la liste des processus à celle des points de synchronisation, mais devons-nous lier les processus aux points de synchronisation auxquels ils contribuent ou aux points de synchronisation qui les déclenchent ? Quel est le point de vue à favoriser ? Dans la mesure où le schéma de la dynamique tranche ce nœud gordien, nous avons décidé de ne favoriser aucune des deux alternatives et donc de ne pas lier la liste des processus à celle des points de synchronisation.

Nous avons également pensé distinguer les ressources requises directement par un processus de celles requises indirectement (c'est-à-dire celles requises par une ressource plutôt que directement par un processus). Des utilisateurs d'IDA nous ont convaincus que cette information ne leur est pas très utile.

Nous avons élaboré la boîte de dialogue de la commande **Sélectionner un objet** en fonction du type d'utilisateurs de notre outil. Comme nous l'avons précisé dans le premier chapitre, les utilisateurs seront essentiellement des personnes qui privilégient la performance. C'est pourquoi, nous avons opté pour une boîte de dialogue puissante qui permet d'atteindre l'objectif fixé en un nombre minimum d'actions.

c) Nous tenons à attirer l'attention du lecteur sur le fait qu'il n'est pas certain que la boîte de dialogue de l'item **Sélectionner un objet** pourra être implémentée. En effet, dans la mesure où les trois listes de sélection sont placées côte à côte, elles seront relativement étroites. Par conséquent, si on ne fixe pas le nombre maximum de caractères que le concepteur de S.I. peut utiliser pour nommer un processus, une ressource ou un point de synchronisation, les titres de certains de ces objets risquent d'être tronqués, ce qui est interdit par la recommandation [CUA89,74] stipulant qu'une colonne d'informations doit être aussi large que son item le plus long. On ne peut évidemment pas imposer une limitation de la longueur des noms des objets. Par conséquent, ou bien on espère que la partie non tronquée des titres sera toujours suffisante pour distinguer ces titres, ou bien on ajoute une barre de défilement horizontal en bas de chacune des listes de sélection pour permettre de prendre connaissance de la partie tronquée. Cette dernière solution renforce la complexité de la boîte de dialogue.

### **3.3.4 Enregistrer**

a) Nous aurions pu préciser le type de fichiers enregistrables grâce à l'item **Enregistrer un rapport** afin que l'utilisateur garde bien en tête que seuls les rapports peuvent être sauvegardés. Pour des raisons analogues à celles évoquées pour la commande **Ouvrir** nous avons conservé la dénomination standard.

b) Nous avons pensé donner la possibilité à l'utilisateur de sauver les fiches qu'il estime susceptibles de révéler par la suite des pathologies du comportement du S.I. analysé afin de garder une trace de sa session de travail. Cette proposition n'a pas rencontré beaucoup de succès auprès des utilisateurs d'IDA.



c) Le fait que l'item **Enregistrer** soit actif dès que la fenêtre principale est associée à une simulation signifie que nous laissons à l'utilisateur la possibilité de sauver un rapport vide, même si ce n'est pas fort utile.

d) Puisque seul un rapport peut être enregistré, il n'est pas obligatoire que le rapport courant soit le document courant. Cette façon de procéder n'est pas très orthodoxe dans la mesure où une action ne peut concerner que l'objet (au sens large) courant. Ainsi, normalement, l'item **Enregistrer** ne peut être actif que si le document courant est un rapport. Nous croyons que cette entorse aux normes en vigueur n'est pas gênante puisqu'il ne peut pas y avoir de confusion quant à ce qui fait l'objet de l'action : le rapport courant. Nous justifions surtout notre décision par la facilité qu'elle induit. En effet, l'essence même de la commande **Enregistrer** est de pouvoir être exécutée à tout moment. Obliger l'utilisateur à faire du rapport courant le document courant est en contradiction avec ce principe d'efficacité. Ce principe nous pousserait aussi à proposer un accélérateur pour cet item (comme sur le Macintosh), mais nous respectons ici les normes de Ms-Windows 2 qui n'en recommandent pas.

### 3.3.5 Enregistrer sous

Dans la mesure où le chemin d'accès peut être très long (par exemple, *c:\windows\simulation\meandres*), nous avons descendu les boutons de commande *Ok* et *Annuler* afin d'autoriser la visualisation d'un plus grand nombre de caractères.

Notez que nous sommes quand même obligés de fixer une borne supérieure au nombre de caractères puisque la boîte de dialogue a une largeur fixe.

### 3.3.6 Fermer

La zone de travail de la fenêtre principale doit toujours contenir un rapport, fut-il vide. C'est pourquoi, lors de la création de cette fenêtre, nous lui associons quand cela s'avère nécessaire un rapport vide : *Rapport 1*. Que se passe-t-il si l'utilisateur souhaite fermer le seul rapport présent sur le bureau ? Plusieurs alternatives s'offrent à nous :

- Quitter l'application MEANDRES.
- Interdire la fermeture du rapport.
- Créer automatiquement un rapport vide : *Rapport n*.

La première solution a pour effet d'accorder beaucoup trop de poids à l'élaboration de rapports vis-à-vis de la consultation de statistiques, et de renforcer ainsi une tendance déjà trop marquée. Nous la rejetons.

La deuxième solution obligerait l'utilisateur à créer un nouveau rapport avant de fermer le rapport souhaité. C'est une contrainte inutile. A rejeter donc.

Il ne reste plus que la dernière solution. Elle n'est pas parfaite car si l'utilisateur n'a pas compris que MEANDRES impose la présence d'un rapport, il risque de se demander pourquoi, quand il ferme un rapport, dans certain cas l'ordinateur lui crée automatiquement un nouveau rapport, et dans d'autres cas non. L'utilisateur n'a plus le sentiment de contrôler l'ordinateur. Bien sûr, on pourrait envisager un message qui informerait l'utilisateur profane de cette particularité, mais cela risque de gêner la plupart des autres utilisateurs. En vertu du profil de l'utilisateur que nous avons dégagé dans le premier chapitre, nous ne souhaitons pas fournir des feed-backs intempestifs. Par conséquent, nous préférons que cette information soit indiquée dans le fichier d'aide et dans le guide de l'utilisateur plutôt qu'affichée à l'écran sous forme d'un message d'information.

### 3.3.7 Aperçu avant impression

a) On pourrait offrir plus de fonctionnalités au niveau de cette boîte de dialogue : imprimer le rapport, couper, coller, ... Cela rendrait sans doute la boîte de dialogue non modale.

b) Nous avons longuement hésité sur le nom à donner au bouton qui efface la boîte de dialogue. *Annuler* ne convient pas parfaitement car comme son nom l'indique, ce bouton doit annuler l'action de la boîte de dialogue. Or si l'utilisateur clique sur ce bouton, la page courante de la zone de travail de la fenêtre principale correspondra à la page présente dans la boîte de dialogue au moment de la fermeture de celle-ci. On peut donc objecter que le bouton *Annuler* induit une action.

Choisir *Ok* ne convient pas non plus car cela laisserait sous-entendre que la boîte de dialogue a explicitement pour objectif d'offrir un moyen de navigation à travers un rapport.

Dans l'incapacité de nous décider sur base d'arguments ergonomiques, nous avons choisi d'hurler avec les loups et d'adopter la solution communément choisie.

c) Nous avons indiqué que la page courante désigne la page dont le numéro est spécifié dans la zone d'information de la fenêtre principale. Mais que représente ce numéro de page ? Plus précisément, quand deux pages sont visibles sur la zone de travail de la fenêtre principale, quelle page désigne le numéro ? Nous proposons que le numéro désigne la page qui occupe visiblement la plus grande partie de la zone de travail. S'il est difficile de déterminer cette page, alors l'ordinateur ne modifie pas le numéro.

Ainsi, supposons que la quatrième page occupe toute la zone de travail de la fenêtre principale. Le numéro apparaissant dans la zone d'information est facile à déterminer (4). Supposons que l'on utilise la barre de scrolling pour visualiser ce qui suit. Tant que la quatrième page occupe la majeure partie de la zone de travail, le numéro reste 4. Mais dès que la cinquième page devient plus importante, le numéro change (5).

### 3.3.8 Effectuer la mise en page

Les possibilités de mise en page proposées sont minimales. On peut imaginer beaucoup d'enrichissements. La relative pauvreté de notre proposition tient au fait qu'initialement il était prévu que nous ayons à implémenter notre outil. Par conséquent, nous étions soucieux de proposer une interface correcte mais relativement simple.

### 3.3.9 Imprimer

a) A l'origine nous pensions offrir la possibilité à l'utilisateur de choisir une imprimante à partir de la boîte de dialogue. Nous avons renoncé à proposer cette facilité puisque l'utilisateur a le loisir de sélectionner une imprimante par l'intermédiaire d'utilitaires spécifiques intégrés à Ms-Windows. Peut-être faudrait-il réexaminer la question ?

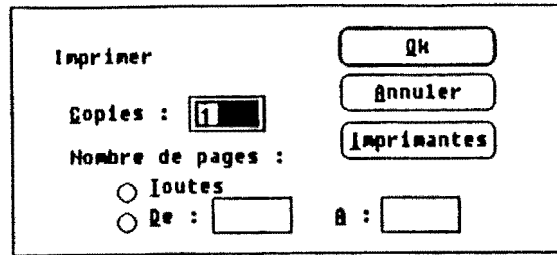


Illustration 65 : Ancienne boîte de dialogue de la commande *Imprimer*

Le bouton *Imprimantes* de cette boîte affiche une deuxième boîte de dialogue permettant de choisir une imprimante.

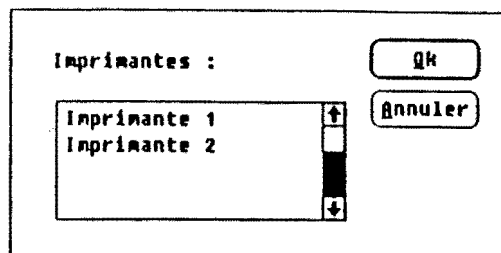


Illustration 66 : Boîte de dialogue du bouton *Imprimantes*

b) Le lecteur méticuleux a sans doute remarqué un problème au niveau des mnémoniques de la boîte de dialogue. En effet, le bouton de commande *Annuler* et le champ *A* possèdent tous les deux le même mnémonique : *a*. Pour éviter ce conflit, nous proposons en accord avec certaines normes de ne pas assigner de mnémonique aux champs *DE* et *A*.

### 3.4 MENU EDITION

Le choix des mnémoniques se base sur les normes de Ms-Windows 2.

#### 3.4.1 Annuler

On pourrait améliorer cette commande en modifiant sa terminologie afin de préciser le contexte : **Annuler Coller**, **Annuler modification fiche**, ... Cette technique souvent employée permet à l'utilisateur d'être sûr de l'effet de l'annulation envisagée. En effet, il arrive fréquemment que dans le feu de l'action on ne se souvienne plus avec certitude de la dernière action effectuée.

Bien entendu, cette amélioration concerne également l'action réciproque **Rétablir** : **Rétablir Coller**, **Rétablir modification fiche**, ...

#### 3.4.2 Couper

Les commandes **Copier** et **Coller** représentent le mécanisme par lequel l'utilisateur transfère des informations sur un rapport. La commande **Couper** sert plutôt

à réorganiser les rapports en permettant de déplacer des informations<sup>1</sup>. Bien que cette commande peut servir à effacer des données d'une fiche, la commande **Effacer** est plus appropriée.

Par conséquent, on pourrait objecter que la commande **Couper** est inutile, ou tout au moins redondante dans le cas des fiches. Comme la commande **Coller** est inaccessible dans le cas des fiches, on pourrait se poser des questions sur le bien-fondé de ne pas avoir établi une distinction de ces commandes en fonction du contexte (fiche ou rapport).

Nous pensons que les utilisateurs d'environnements multi-fenêtrés (Ms-Windows, Macintosh, ...) sont tellement habitués à utiliser ces commandes qu'une distinction fine de leur contexte d'utilisation s'avère inutile.

### **3.4.3 Copier**

Nous avons précisé que si les informations copiées sont non contiguës, c'est-à-dire séparées par une ou plusieurs autres informations, alors l'espace qui les sépare est enlevé dans le Presse-papiers. Par conséquent, la commande **Coller** aura pour effet de restituer des données contiguës.

Nous avons opté pour cet artifice afin de faciliter la tâche de l'utilisateur. En effet, quand celui-ci colle sur le rapport courant les informations glanées çà et là sur une fiche, il aura à cœur de les redresser de manière ergonomique. Il nous semble qu'il est souhaitable que ces informations à déplacer soient regroupées afin d'éviter trop de scrolling.

### **3.4.4 Coller**

Nous devons préciser que si l'information est collée de telle manière qu'elle chevauche deux pages, un mécanisme de saut de page automatique agira pour empêcher qu'elle ne soit effectivement coupée en deux parties lors de l'impression. Le passage à la page suivante est indiqué par une ligne pointillée horizontale qui traverse toute la largeur de la zone de travail de la fenêtre principale.

Dans le même ordre d'idée, une information collée trop à droite sera automatiquement redéplacée vers la gauche (son bord droit sera placé contre le bord droit de la partie imprimable du rapport). De manière générale, l'utilisateur est dans l'incapacité de déplacer (par glissement) une information en dehors de la partie imprimable.

### **3.4.5 Effacer**

Cette commande semble faire double emploi avec la commande **Couper** plus riche. Elle est pourtant utile pour supprimer des informations sans modifier le Presse-papiers. En effet, le Presse-papiers ne conservant que les dernières informations coupées ou copiées, il faut particulièrement faire preuve de vigilance afin de ne pas perdre ces informations surtout si elles y ont été mises à la suite de la commande **Couper**.

---

<sup>1</sup> Bien entendu, l'utilisateur peut aussi déplacer les informations en les glissant.

### 3.4.6 Tout sélectionner

Après un temps de tâtonnement, il est fort probable que l'utilisateur déterminera un standard de fiches reprenant toutes les informations jugées pertinentes. Dès lors, on peut fort bien imaginer que l'utilisateur transférera fréquemment l'intégralité du contenu de certaines fiches sur un rapport. Par conséquent, il faudrait prévoir un accélérateur pour faciliter cette tâche.

## 3.5 MENU PROCESSUS

a) Le lecteur trouvera dans le mémoire de B. Sacré et de J. Lefèvre [Lef87] beaucoup de renseignements qui expliquent certains de nos choix : standards proposés, mesures présentées sous forme de camembert ou d'histogramme, ...

b) Nous avons voulu que notre interface reflète le plus possible le monde de la tâche du concepteur de S.I. C'est pourquoi, nous proposons trois menus qui correspondent aux trois types d'objets pour lesquels DSL/SIM produit des mesures. Pour chacun de ces types d'objets, l'utilisateur peut choisir le type de mesures à consulter. Remarquez que cette structure est en adéquation avec le mécanisme de sélection d'une fiche (commande **Sélectionner un objet**) : l'utilisateur sélectionne d'abord un objet, et ensuite il choisit un type de statistiques.

Cette structure induit que la notion de type d'objets prime sur celle de type de statistiques.

Nous estimons qu'il doit en être ainsi : les concepts de processus, de ressource et de point de synchronisation dépassent le cadre des simulations. Ce n'est pas le cas des notions de statistiques globales, périodiques et chronologiques. De plus, l'objet de la tâche est bien d'analyser le comportement des processus, des ressources et des points de synchronisation, et pas de vérifier la pertinence des types de mesures produites par DSL/SIM.

c) Les boîtes de dialogue permettant de choisir les résultats statistiques (et autres) à consulter illustrent notre volonté de laisser à l'utilisateur la possibilité d'obtenir toutes les informations que livrent les listings. Nous avons expliqué dans le premier chapitre la raison de cette décision. Précisons-la à l'aide d'un exemple.

Comme il n'est pas nécessaire, dans la majorité des cas, d'entrer dans des distinctions subtiles telles que, par exemple, différencier l'attente avant activation de l'attente avant réactivation, MEANDRES propose parfois des informations cumulées (par exemple, le temps d'attente d'un processus). Nous obéissons en cela à notre volonté de ne pas surcharger l'utilisateur (respect de la charge informationnelle).

Toutefois, nous ne pouvons pas imposer notre vision de l'analyse du comportement d'un S.I. Par conséquent, nous devons tenir compte de l'éventuelle préférence de l'utilisateur pour des valeurs non cumulées.

d) Lorsque les boîtes de dialogue doivent gérer trop d'informations, il est impossible de tout afficher sur une seule fenêtre. Nous avons alors recours à des boîtes de dialogue secondaires activées à l'aide de boutons de commande de la boîte de dialogue principale. Pour des raisons évidentes, nous avons associé à ces boîtes de dialogue les informations moins importantes ou consultées moins fréquemment. Inversement, nous avons toujours associé les informations faisant l'objet de camemberts ou d'histogrammes aux boîtes de dialogue principales. Les boîtes de dialogue secondaires s'affichent au dessus des boîtes de dialogue principales.

e) Les boîtes de dialogue (principales) du menu **Processus** reprennent en haut à gauche le nom de l'item qui a provoqué son affichage. Nous obéissons en cela au *principe de concision* qui repose sur l'existence de limites de la mémoire à court terme et qui vise à minimiser la charge mnésique de l'utilisateur.

Ainsi, nous avons voulu que l'utilisateur ne doive pas fournir d'effort pour se souvenir de la nature de la boîte de dialogue avec laquelle il travaille. Il est en effet réaliste de penser qu'un utilisateur puisse parfois éprouver quelques difficultés à se repérer : les boîtes de dialogue sont complexes, présentent d'évidentes ressemblances entre elles (par souci d'homogénéité), ...

Nous aurions dû aller plus loin en rappelant également le type d'objets dont l'utilisateur consulte les mesures statistiques. Nous ne l'avons pas fait, d'une part parce que nous craignons de surcharger les boîtes de dialogue et, d'autre part, parce que estimions qu'il est peu probable que l'utilisateur oublie le type d'objet qu'il est occupé d'étudier. Après avoir réalisé une maquette du fonctionnement des boîtes de dialogue, nous pensons qu'il est préférable d'ajouter cette information.

Nous avons manqué de cohérence en ayant parfois omis le titre de certaines boîtes de dialogue d'autres menus (**Aperçu avant impression**, ...).

Les boîtes de dialogue secondaires portent également un titre : le nom du bouton de commande de la boîte de dialogue principale qui a provoqué leur affichage. Il est utile de préciser que ces boîtes de dialogue sont légèrement décalées (à droite et en bas) de la boîte de dialogue principale afin que l'utilisateur puisse toujours lire le titre de la boîte de dialogue principale.

### 3.5.1 St. globales

a) Le choix des mesures présentées par défaut repose sur les propositions de B. Sacré et de J. Lefèvre [Lef87].

b) Nous avons signalé que les mesures de durées et d'états apparaissent dans une fiche sous forme d'un tableau similaire à celui présenté dans les listings produits par DSL/SIM. Précisons toutefois que pour être cohérent avec la langue utilisée dans l'interface, ces mesures seront exprimées en français, et non pas en anglais comme c'est le cas dans les listings.

Cette remarque vaut également pour toutes les autres mesures pour lesquelles nous précisons qu'elles apparaissent sous une forme similaire à celle présentée dans les listings.

c) Idéalement, il aurait fallu concevoir la boîte de dialogue de la commande **St. Globales** (et celles qui gèrent les autres fiches) de manière à ce que sa structure soit la proche possible de celle des fiches. Autrement dit, il aurait fallu établir un isomorphisme entre boîte de dialogue et fiche générée afin de minimiser l'effort d'évaluation de l'action de création de fiches (principe de continuité entre l'input et l'output).

Malheureusement, le nombre d'informations affichables dans une fiche est trop élevé pour que nous puissions disposer leurs contrôles dans une seule boîte de dialogue. Une solution serait d'imaginer une boîte de dialogue extensible, c'est-à-dire une boîte de dialogue dont certains boutons de commande agrandissent sa taille, au lieu d'afficher une deuxième boîte de dialogue.

Nous n'avons pas proposé cette technique parce qu'elle ne résout pas entièrement le problème : il est toujours impossible de tout visualiser sur une seule boîte de

dialogue. De plus, elle n'est pas courante en Ms-Windows 2 (elle le devient en Ms-Windows 3).

La véritable solution consisterait en une boîte de dialogue munie d'une barre de défilement vertical ! Voilà un nouvel objet interactif qui demanderait un examen plus approfondi.

d) Notre réflexion sur la représentation de données sous forme de camembert est incomplète. En effet, comment orienter les différents quartiers ? Où placer les valeurs de ces quartiers ? Dans quel ordre faut-il écrire les éléments composant la légende ? Quelles couleurs adopter ? ...

Nous n'avons donc pas creusé les aspects ergonomiques de la création de graphiques en forme de camembert. Pourtant ceux-ci sont très importants. En particulier, l'utilisation de la couleur est fondamentale dans la mesure où elle est visuellement attrayante et où elle permet de mettre des informations en évidence. Son usage s'avère toutefois très délicat.

Nous nous souvenons qu'au cours d'"Interface homme/machine", un ergonomiste, G. Karnas, a attiré notre attention sur quelques règles qui permettent de tirer profit de l'utilisation de la couleur :

- Utiliser peu de couleurs. Une surabondance de couleurs n'aide pas les utilisateurs à retenir la signification associée à ces couleurs. Au contraire, elle les amène à rechercher des significations qui n'existent pas.
- Utiliser la même couleur pour mettre en évidence la même fonctionnalité.
- Permettre aux utilisateurs de choisir les couleurs. La sélection des couleurs est très subjective et les préférences en ce domaine varient souvent.

Parce que nous avons omis de mener une réflexion sur l'ergonomie de la couleur, l'implémentation de notre outil risque de ne pas respecter les deux premières règles. Plus grave encore, à l'heure actuelle notre interface ne respecte pas la troisième règle.

### **3.5.2 St. périodiques**

a) Le problème des cellules d'observation non significatives nous plaçait devant deux alternatives :

- Refléter totalement le choix de l'utilisateur en découpant strictement l'axe des abscisses sans tenir compte des cellules vides.
- Regrouper les cellules contiguës non significatives.

La première alternative a le mérite de présenter un histogramme stable : l'axe des abscisses est toujours découpé de la même manière, quel que soit le processus analysé. Par contre, l'axe des abscisses est composé de plus de cellules, ce qui diminue la largeur des bâtonnets. Cette minceur peut nuire à la lisibilité. C'est d'autant plus regrettable que cette réduction de la lisibilité de mesures significatives est due à l'amélioration de la lisibilité de mesures non significatives !

Nous avons donc opté pour la deuxième alternative tout en reconnaissant que nous aurions peut-être dû utiliser un artifice quelconque (la couleur, par exemple) pour signaler que certains intervalles proviennent de la concaténation d'intervalles vides.

b) Nous aurions peut-être dû offrir un moyen qui permet de faire défiler l'histogramme dans le sens de l'axe des ordonnées, mais les utilisateurs d'IDA nous ont convaincu du peu d'utilité de ce mécanisme.

c) Nous avons préféré regrouper les mesures concernant le nombre d'occurrences des différents types d'événements en un seul histogramme. Cette décision est peut-être critiquable dans la mesure où l'utilisateur risque d'éprouver des difficultés pour visualiser simultanément les cinq types d'informations. Ce risque existe, c'est vrai; mais d'après les discussions que nous avons eues avec des utilisateurs d'IDA, il est rare que l'on souhaite visualiser simultanément ces cinq types d'informations.

Par contre, il est certain que l'utilisateur devra souvent comparer deux ou trois types de mesures entre eux. Associer un histogramme à chacun des types de statistiques aurait nettement moins facilité cette tâche.

d) Nous avons longuement hésité à mettre sur la fiche la barre de défilement vertical qui gère le changement d'échelle. En effet, nous étions soucieux d'offrir le plus de concordance possible entre les informations d'une fiche et celles transférées sur un rapport. Nous avons pensé gérer le changement d'échelle par l'intermédiaire d'une boîte de dialogue (cfr annexe 7 : *Première maquette*). Après réflexion, nous sommes d'avis que le désagrément provoqué par l'utilisation d'une boîte de dialogue supplémentaire dépasse celui du manque d'isomorphisme.

### 3.5.3 St. chronologiques

a) Le lecteur a sans doute remarqué que nous utilisons deux syntaxes différentes pour désigner minuit : 0h au début de la période et 24h à la fin de la période. Cette distinction se justifie par le souci de refléter l'écoulement du temps mais occasionne un problème quand l'utilisateur avance cellule d'observation par cellule d'observation. En effet, si l'utilisateur visualise deux périodes, faut-il considérer minuit comme la fin de la première période (24h) ou comme le début de la deuxième période (0h) ?

Nous avons privilégié la solution 0h, mais l'inverse aurait sans doute été préférable.

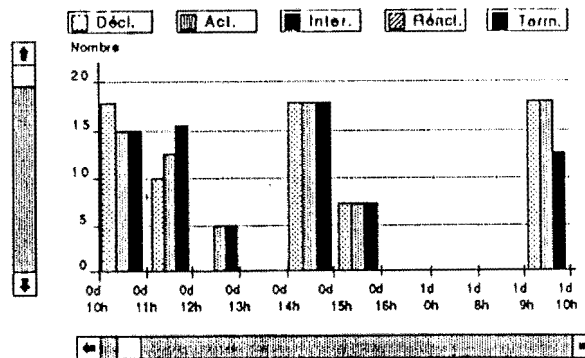


Illustration 67 : Histogramme montrant deux périodes

b) La double graduation de l'axe des abscisses est assez lourde. Il n'est sans doute pas nécessaire de spécifier l'indication de la période à chaque cellule d'observation (redondance, bruit). On pourrait éventuellement ne donner cette indication que pour le temps charnière entre deux périodes, ou imaginer deux cases situées aux extrémités de l'axe des abscisses qui contiendraient le nom la période correspondant aux deux cellules extrêmes. On pourrait aussi utiliser la couleur pour représenter de manière différente les cellules appartenant à des périodes différentes.



c) Nous permettons à l'utilisateur de consulter les valeurs minimale, moyenne et maximale des mesures reproduites sur l'histogramme. Ces valeurs<sup>1</sup> sont calculées en fonction de la période d'observation la plus à gauche de l'axe des abscisses. Nous aurions tout aussi bien pu choisir la période la plus à droite, ou la période possédant le plus de cellules d'observation représentées sur l'histogramme. La multiplicité des choix nous a convaincu de la nécessité de préciser à l'utilisateur la période qu'il a choisie. C'est pourquoi nous indiquons ce renseignement dans le titre de la boîte de dialogue.

Remarquez que les renseignements contenus dans la boîte de dialogue ne peuvent pas être transférés sur un rapport. C'est une lacune à laquelle il faudra remédier.

Nous tenons à signaler au lecteur que nous nous sommes écartés des recommandations de B. Sacré et de J. Lefèvre. En effet, ces derniers proposaient de spécifier les valeurs minimale, moyenne et maximale des mesures à l'aide d'un trait sur l'axe des ordonnées. Nous avons rejeté cette suggestion car nous ne voulions pas encombrer l'histogramme.

d) Le fait que les légendes déclenchent des actions à la fois à partir d'un simple clic et à partir d'un double clic n'est peut-être pas très heureux. L'expérience du gestionnaire de fichiers de Ms-Windows 3 nous apprend que ce mécanisme est perturbant surtout pour des utilisateurs novices qui ne maîtrisent pas le double-clic.

e) Les valeurs minimale, moyenne et maximale sont produites en comparant les différentes mesures de chaque cellule d'observation à l'intérieur d'une période déterminée. On aurait également pu proposer les valeurs minimale, moyenne et maximale en comparant les mesures d'une cellule d'observation déterminée sur différentes périodes. Par exemple, on aurait pu s'intéresser à l'évolution du nombre de déclenchements pour la cellule d'observation [10,11[ sur toutes les périodes. Les utilisateurs d'IDA ne jugent pas ces renseignements indispensables.

### 3.5.4 Commentaires

a) Pourquoi avoir placé cet item, ainsi que les items **Calendrier** et **Ressources requises** sur le même pied que les items **St. globales**, **St. périodiques** et **St. Chronologiques** ? La question est pertinente puisqu'on pourrait nous reprocher d'avoir manqué de cohérence en n'ayant pas homogénéisé le mécanisme de sélection des informations consultables sur une fiche.

Nous retournons la question à nos détracteurs : fallait-il homogénéiser ces informations ? Nous ne le pensons pas car nous nous trouvons en présence de deux types de données : d'une part des mesures statistiques et d'autres part des informations sur ces mesures statistiques ou sur l'objet que caractérisent ces mesures statistiques (de quelle simulation proviennent-elles ? Quel objet décrivent-elles ?...). Les renseignements que fournissent les trois items incriminés se situent à un méta-niveau par rapport aux mesures statistiques proprement dites : ils en parlent.

On pourrait alors nous objecter que ce raisonnement n'est exact que pour la commande **Commentaires**. Par contre les deux autres items ont la même fonction que les mesures statistiques : caractériser un objet. C'est vrai, mais la nature des caractéristiques est réellement différente. En effet, les renseignements fournis par l'item **Calendrier** ne sont pas issus à proprement parler de DSL/SIM, ils n'ont pas été calculés; ce ne sont pas des mesures statistiques mais des paramètres qui influencent la simulation. Quant à la commande **Ressources requises**, elle traite de ressources, c'est-à-dire

---

<sup>1</sup> Nous signalons au lecteur que ces valeurs ne sont pas fournies explicitement dans les rapports DSL/SIM. C'est une particularité de notre outil par rapport à la situation existante.

d'un type d'objets étranger au menu **Processus**. L'avoir mise au niveau du menu souligne le rôle spécial de cette commande : servir de trait d'union conceptuel entre deux menus.

b) L'exemple de commentaires que nous proposons (illustration 34) ne respecte pas totalement la logique de la boîte de dialogue. En effet, nous avons regroupé en une seule phrase les commentaires concernant le type de mesures statistiques, la période de prise en compte et le nom du processus étudié. Or dans la boîte de dialogue, cette dernière information est reprise dans une boîte de regroupement séparée. Voilà un manque de cohérence à corriger :

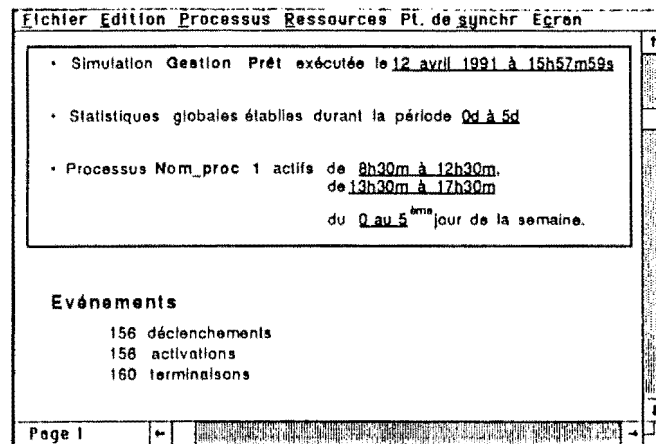


Illustration 68 : Commentaires revus et corrigés

### 3.5.5 Calendrier

a) Un problème survient si certaines périodes d'activité (de disponibilité) coïncident avec des cellules d'observation non significatives contiguës. Imaginons le cas de trois cellules non significatives contiguës dont la deuxième coïncide avec une période d'activité. Comme les trois cellules sont regroupées en une seule, comment signaler que la période d'activité se limite à une partie de cette "cellule" ? Deux possibilités :

- Ne pas représenter les périodes d'activité (de disponibilité) qui tombent dans de telles cellules.
- Placer un segment de droite d'une longueur proportionnelle à la période d'activité (de disponibilité) juste au dessus de cette partie de la "cellule".

Nous sommes indécis.

b) Avoir choisi d'afficher les calendriers en dessous du dernier histogramme n'est pas totalement heureux. En effet, la hauteur des fiches ne permet pas de visualiser simultanément tous les histogrammes et les calendriers. Or il faudrait que l'utilisateur puisse consulter ces calendriers chaque fois qu'il examine un histogramme. A l'heure actuelle, notre choix implique que l'utilisateur doit effacer tous les histogrammes de la fiche qui se situent entre l'histogramme à consulter et les calendriers. Voilà un surcroît de travail bien inutile.

Comme nous tenons à éviter qu'une même information ne soit reproduite en multiples exemplaires au sein d'une même fiche (isomorphisme avec la boîte de dialogue), nous rejetons la solution qui consisterait à associer systématiquement les calendriers à chaque histogramme. De plus, cette solution risque d'encombrer inutile-

ment les fiches et de forcer l'utilisateur à avoir (trop) fréquemment recours au scrolling (c'est justement ce que nous cherchons à éviter !).

Une autre solution serait d'associer à chaque histogramme un bouton de commande qui permettrait d'afficher le(s) calendrier(s) en dessous du graphique. Cette solution a le mérite d'être plus souple que la précédente mais présente les mêmes inconvénients.

Enfin, une troisième solution consisterait à offrir la possibilité de décomposer la zone de travail de la fiche en deux volets munis chacun d'une barre de défilement vertical. Ces volets seraient indépendants dans la mesure où le scrolling de l'un ne modifierait pas la zone de travail de l'autre. Ce mécanisme se rencontre de plus en plus souvent (Ms-Word, Ms-Excel, ...) et se révèle réellement pratique. L'utilisateur n'a plus qu'à afficher dans un volet l'histogramme voulu, et dans le second les calendriers. Il pourrait également avoir recours à cet artifice pour comparer d'autres informations non contiguës de la fiche. Ce mécanisme pallie donc la rigidité qu'impose le fait d'avoir déterminé un ordre fixe pour chaque information des fiches. Nous pourrions étendre cet artifice à la fenêtre principale afin d'offrir les mêmes facilités de navigation à travers le rapport courant.

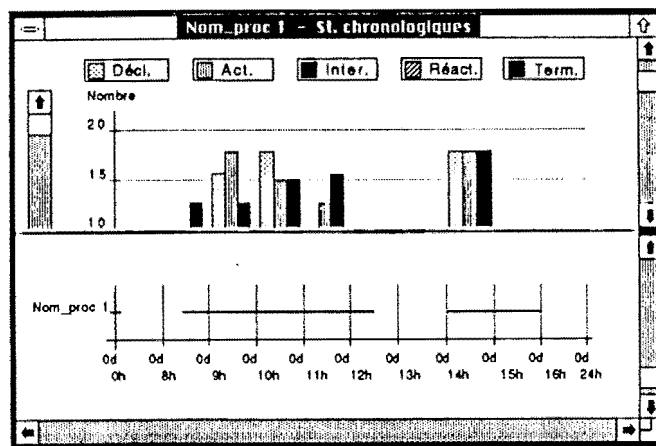


Illustration 69 : Fiche scindée en deux volets

### 3.5.6 Ressources requises

a) On pourrait profiter de l'affichage de la liste des ressources requises pour offrir un raccourci de navigation intéressant. On peut en effet imaginer qu'en cliquant deux fois sur le nom d'une des ressources reprises dans le tableau, l'utilisateur puisse avoir directement accès à la boîte de dialogue de la commande **Sélectionner un objet** dûment mise à jour pour refléter le contexte de navigation. Cette technique renforcerait le rôle de trait d'union de la commande **Ressources requises** précédemment évoqué.

b) Peut-être faudrait-il offrir à l'utilisateur de pouvoir également obtenir à partir du menu **Processus**, un résumé des statistiques globales des points de synchronisation qui déclenchent les processus étudiés.

### 3.5.7 Définir fiche standard

a) Nous pensons que la notion de fiche standard est très importante. En effet, comme nous l'avons déjà signalé, nous croyons qu'après un temps de tâtonnement, il est fort probable que l'utilisateur déterminera un standard de fiches reprenant toutes les

informations jugées pertinentes afin de faciliter sa tâche d'analyse. La notion de fiche standard est donc une façon de personnaliser le logiciel.

Autoriser une personnalisation est intéressant car cela permet de gérer souplement l'hétérogénéité des utilisateurs.

On pourrait prévoir d'autres artifices de personnalisation. Le plus courant est la possibilité de modifier les menus (suppression d'items peu usités, ajouts d'items particuliers, modification d'accélérateurs, ...). On pourrait offrir plus d'options : permettre de choisir la forme des graphiques, leurs couleurs; de modifier la taille et l'ordre des données sur les fiches; ...

b) L'item **Définir fiche standard** se retrouve également dans les menus **Ressources** et **Pt. de synchr.** Ces trois items remplissent strictement la même fonctionnalité. Idéalement, il aurait fallu éliminer cette redondance. Le problème réside dans le choix du menu qui accueillerait l'item unique **Définir fiche standard**.

Nous ne pouvons évidemment pas choisir l'un des trois menus gérant les résultats d'une simulation. Le menu **Fichier** ne convient pas car nous ne sommes pas en présence de fichiers, du moins pas explicitement. Le menu **Edition** ignore totalement la notion de fiche en tant qu'ensemble logique d'informations. Quant au menu **Ecran**, il gère bien des fiches de toutes natures, mais uniquement en tant que fenêtres, c'est-à-dire en termes de contenants et non en termes de contenus. Il ne nous semble donc impossible, dans la structure actuelle de la barre des menus, d'adopter un item unique pour redéfinir les standards des fiches.

Plutôt que d'essayer d'insérer l'item au niveau d'un menu, ne serait-il pas avisé de prévoir sur chaque boîte de dialogue un bouton de commande *Option prédéfinie* qui permettrait de redéfinir les standards liés à la boîte de dialogue affichée ? Cette solution est fréquemment utilisée (Ms-Word) mais présente, dans notre cas, plusieurs inconvénients :

- A l'heure actuelle, redéfinir un standard s'avère très rapide : dès que l'utilisateur est satisfait du contenu de la fiche courante, il lui suffit de choisir l'item **Définir fiche standard** du menu correspondant au type de l'objet analysé dans la fiche. Par contre la solution envisagée impose à l'utilisateur d'afficher chacune des boîtes de dialogue et de cliquer chaque fois sur le bouton **Option prédéfinie**.
- Non seulement la solution envisagée demande plus de manipulations et donc plus de temps, mais elle demande aussi plus d'efforts cognitifs à l'utilisateur qui doit décomposer son intention en plusieurs actions (il doit notamment se souvenir de l'item qui a provoqué l'affichage de telle information).
- Enfin cette solution ne tient pas compte de l'item **Ressources requises** qui ne possède pas de boîte de dialogue.

Cette réflexion sur la notion de standard illustre bien la difficulté de concevoir une interface : tout est lié, et le moindre détail peut remettre en question l'édifice construit.

### 3.6 MENU RESSOURCES

Les commentaires et justifications du menu **Ressources** sont essentiellement semblables à ceux du menu **Processus**.

### 3.7 MENU PT. DE SYNCHR.

a) Nous avons hésité à appeler ce menu **Pt. de synchr.** plutôt que **Pt. de synchro.** En effet, lorsque des utilisateurs d'IDA parlent entre eux, ils utilisent volontiers l'expression "point de synchro.". Mais comme nous avons partout ailleurs suivi les règles typographiques en matière d'abréviation, nous étions tenu de les respecter dans le choix du nom de ce menu. Or celles-ci stipulent que l'abréviation doit se faire "par retranchement, *toujours avant une voyelle*<sup>1</sup>, des lettres finales que l'on remplace par un point" [Cod89,3].

b) De même que le menu **Processus** dispose de l'item **Ressources requises**, de même que le menu **Ressources** bénéficie de l'item **Processus requerants**, ne faudrait-il pas prévoir un item **Processus contributeurs** qui permettrait à l'utilisateur de consulter un résumé des principales statistiques globales des processus qui contribuent à la réalisation des points de synchronisation analysés ?

#### 3.7.1 St. globales

a) Contrairement aux listings de DSL/SIM, nous ne numérotions que les opérandes des prédicats dans un but de simplification. Ainsi, cela nous permet d'obtenir une numérotation continue des quartiers des camemberts et des différentes lignes des tableaux proposés.

b) Les raccourcis qui permettent de sélectionner les options des boîtes de dialogue secondaires de la commande **St. globales** du menu **Processus** sont différents de ceux qui facilitent la sélection des différents temps de réalisation et de contribution. En effet, nous avons utilisé ici une boîte à cocher plutôt qu'un mnémonique pour (dé)sélectionner un ensemble d'options.

Cette légère inhomogénéité nous a été dictée par la structure des boîtes de dialogue secondaires. En effet, il nous semblait que, dans le cas de la boîte de dialogue gérant les mesures concernant les réalisations, nous étions en présence de deux types d'informations de même importance : le nombre de réalisations et le temps de réalisation, ce dernier faisant l'objet de mesures plus fines. Par conséquent, si nous associons une boîte à cocher au nombre de réalisations, nous devons également en associer une au temps de réalisation.

Notez que l'utilisateur peut avoir recours au mnémonique de l'option *Temps de réalisation* pour cocher la boîte. Par conséquent, la cohérence est garantie.

Peut-être faudrait-il revoir les boîtes de dialogue secondaires de la commande **St. globales** du menu **Processus** et leur associer une boîte à cocher à chacune des options.

### 3.8 MENU ECRAN

Ce menu n'existe généralement pas dans les applications exécutées dans l'environnement Ms-Windows 2. Nous pensons que c'est une lacune importante. Apparemment, nous ne sommes pas les seuls à le croire, puisque la nouvelle version de Ms-Windows propose un menu **Fenêtre** qui ressemble très fort à notre menu **Ecran**.

---

<sup>1</sup> Souligné dans l'ouvrage.

Nous avons conçu notre interface avant d'avoir pris connaissance de Ms-Windows 3. C'est pourquoi le nom de notre menu reflète plus le monde Macintosh que celui de Ms-Windows 3. Bien sûr, nous suggérons que d'adapter le titre de notre menu et de le rebaptiser **Fenêtre**.

### 3.8.1 Aide

a) La conception d'un système d'aide est une véritable tâche en soi, non pas tant pour sa réalisation que pour sa conception. En effet, nous devons porter un soin tout particulier au choix des renseignements accessibles, à leur formulation, à la manière d'y accéder. Nous n'avons pas eu le temps de nous consacrer à la rédaction du système d'aide, ni d'ailleurs à celle d'un manuel d'utilisation.

A l'heure actuelle, nous pouvons seulement faire part au lecteur de notre souhait de mettre à la disposition de l'utilisateur une aide indexée sur des items tels que les commandes des différents menus, les principales fonctionnalités de MEANDRES, les touches du clavier, ....

Idéalement, il faudrait compléter l'aide en offrant des recommandations d'analyse des résultats statistiques.

b) La seule information que nous pouvons livrer avec certitude à l'heure actuelle concerne l'accélérateur : *F1*. Ce choix découle des recommandations de Ms-Windows.

c) L'absence fréquente d'aide est une caractéristique malheureuse des logiciels tournant sous Ms-Windows 2. Heureusement, Ms-Windows 3 pallie cette lacune en proposant une gestion automatique des systèmes d'aide. Les logiciels conformes à ce nouvel environnement disposent systématiquement d'un menu **Aide** situé à l'extrême droite de la barre des menus. Cette localisation remet en question la conception de notre item **Aide** qui devrait se transformer en menu à part entière.

### 3.8.2 Informations

La fenêtre utilisée n'est pas non plus à proprement parler une fenêtre logo. Une fenêtre logo contient généralement :

- le logo du programme ou celui de la société qui a conçu le programme,
- le nom de l'application,
- le nom des auteurs,
- le numéro et la date de la version de l'application,
- les droits de reproduction.

L'affichage d'une fenêtre logo est normalement provoqué par la commande **A propos de** placée dans le menu **Fichier**.

Notre application n'étant pas commerciale, nous n'avons pas jugé utile de disposer d'une fenêtre identifiant notre outil. Nous avons toutefois gardé l'idée d'une identification du contexte d'utilisation, c'est-à-dire, dans notre cas, de la simulation associée à la fenêtre principale.

Bien que certaines des informations données par la commande **Informations** peuvent être visualisées sur les fiches (cfr items **Commentaires**), nous croyons utile de fournir à l'utilisateur un moyen rapide pour vérifier s'il analyse les résultats de la simulation souhaitée.

Les informations données spécifient donc le contexte de travail de l'utilisateur. Elles lui fournissent ainsi une sorte d'aide. C'est pourquoi nous avons regroupé cette commande avec la commande **Aide**.

Afin d'être plus explicite, peut-être faudrait-il modifier le nom de l'item au profit de **A propos de la simulation**. Idéalement, cette commande devrait être placée dans le menu **Aide**.

### **3.8.3 Réorganiser**

Nous avons déjà signalé au lecteur qu'à l'origine, il était prévu que nous devions implémenter notre outil. Dans cette optique, nous n'avons présenté qu'une seule méthode de réorganisation bien qu'il en existe au moins une autre : disposer les fiches en cascade, c'est-à-dire superposer les fiches en les décalant légèrement à droite et en bas de façon à visualiser leur titre.

Idéalement, il faudrait offrir la possibilité à l'utilisateur de choisir l'un des deux types de configuration puisque ces deux stratégies présentent des avantages (et des inconvénients) complémentaires. La réorganisation en mosaïque offre l'avantage de permettre d'obtenir des fenêtres de taille agréable. Par contre, seule la zone de travail de la dernière fiche est visible. Par conséquent, si l'utilisateur ne conserve que deux, ou trois fiches sur le bureau, il aura tout intérêt à réorganiser leur affichage en mosaïque. Par contre, s'il souhaite travailler avec de multiples fiches, la réorganisation en cascade se révélera nettement plus pratique car elle lui permet de passer rapidement d'une fiche à l'autre.

### **3.8.4 Masquer la fiche**

L'inconvénient du multi-fenêtrage, c'est la prolifération des fenêtres présentes à un moment donné à l'écran. Quand, en plus, ces fenêtres sont cantonnées dans une zone réduite de l'écran, l'utilisateur est rapidement confronté au problème de leur gestion. C'est pourquoi nous avons mis à sa disposition les deux commandes de gestion de fenêtres : **Réorganiser** et **Masquer la fiche**.

Masquer une fiche permet de faire disparaître momentanément une fiche. Mais pourquoi ne pas la fermer à l'aide de la commande **Fermer** du menu **Fichier** ? La réponse à cette question légitime est que pour ouvrir une fiche fermée, il faut de nouveau passer par les boîtes de dialogue de sélection de mesures statistiques. Même si respecifier les différentes mesures à consulter ne prend pas beaucoup de temps (surtout dans le cas d'une fiche standard), cette contrainte risque de devenir rapidement fastidieuse : on n'aime pas devoir faire deux fois la même chose. D'autre part, l'utilisateur pressé est susceptible de se tromper dans le choix des statistiques et de ne pas obtenir la même fiche que celle qu'il avait précédemment.

### **3.8.5 Afficher les règles**

a) Nous aurions pu prévoir d'autres mécanismes d'aide au placement d'informations sur un rapport : alignement automatique ou manuel (système d'axes magnétiques), grilles magnétiques, boîte de dialogue permettant de spécifier avec précision la position d'une information,...

b) La règle verticale n'est pas continue en ce sens qu'à chaque nouvelle page, elle est remise à zéro.

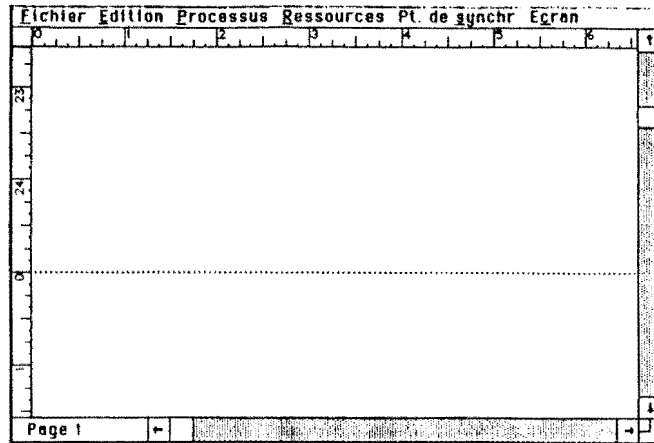


Illustration 70 : Réinitialisation de la règle verticale

### 3.8.6 Liste des documents

a) Nous avons hésité à utiliser la technique des menus en cascade dans la mesure où elle n'est pas implémentée en Ms-Windows 2<sup>1</sup>. De plus, l'utilisation des menus en cascade dérouté parfois les utilisateurs novices qui ne maîtrisent pas bien les mouvements de la souris.

Malgré ces arguments, nous proposons cette technique car elle offre des avantages certains : elle permet d'avoir une vision concise, simplifiée et structurée de la liste des documents.

b) Nous avons omis de spécifier les mnémoniques des items de la liste des documents et ceux des items des menus en cascade. Pour remédier à cette lacune nous proposons de faire précéder le nom de chaque document d'un nombre indiquant la place du document dans la liste. Quant au mnémonique des items **Glob.**, **Périod.**, et **Chron.**, nous choisissons respectivement les lettres G, P, C.

Choisir des nombres pour mnémoniques se justifie par le fait qu'on ne connaît pas a priori le nom des items apparaissant dans la liste des documents. Non seulement, on ne peut donc pas les spécifier maintenant, mais il est probable qu'on soit à un moment donné dans l'incapacité de trouver des caractères non encore utilisés.

Les nombres permettent de contourner cette incertitude. De plus, ils offrent un aspect de continuité qui contrebalance quelque peu l'instabilité du menu **Ecran** dont le contenu varie en fonction des besoins de l'utilisateur. Enfin, il nous paraît naturel d'avoir recours à des nombres pour caractériser les éléments d'une liste ordonnée.

<sup>1</sup> Par contre, elle est implémentée en Ms-Windows 3.



## **3.9 SELECTION ET DEFORMATION**

### **3.9.1 Mécanismes de sélection**

a) Initialement, nous avons également muni le rectangle de sélection des informations d'une fiche des trois poignées de déformation. Malgré le fait que ces poignées ne servent à rien dans le cas des fiches, nous estimions nécessaire d'adopter un mode de représentation unique afin de ne pas troubler l'utilisateur. Ensuite, nous nous sommes rendus compte que nous mélangions deux réalités différentes : sélection et déformation, et que l'utilisateur risque d'être plus troublé par des poignées de déformation inopérantes que par des représentations différentes. D'autre part, ces représentations différentes offrent l'avantage de rappeler à l'utilisateur occasionnel qu'il ne peut pas modifier la taille que des données d'un rapport.

b) Nous avons opté pour une utilisation réduite mais simple du clavier. En effet, nous aurions pu prévoir les mêmes possibilités de sélection que celles offertes par la souris. Toutefois, cela impliquerait une complexité d'utilisation du clavier nettement accrue<sup>1</sup>. D'autre part, nous faisons l'hypothèse que l'utilisateur emploie plus fréquemment la souris.

De manière générale, il est nécessaire de faire en sorte que la plupart des opérations utilisant une souris soient réalisables à partir du clavier. En effet, il est possible (mais peu probable) que l'utilisateur ne dispose pas de souris. De plus, le clavier se révèle parfois plus pratique car plus rapide mais demande une certaine mémorisation des commandes. Par conséquent, l'utilisateur fréquent aura plus souvent recours au clavier que l'utilisateur novice.

### **3.9.2 Modifier la taille des informations**

a) Nous n'avons pas jugé utile de prévoir un mécanisme de modification de la taille des informations d'une fiche. Nous aurions pu le faire en proposant l'artifice des poignées de sélection, ou celui de la modification automatique de la taille en fonction des dimensions de la fenêtre.

Il incombe à l'implémenteur de déterminer une taille optimale pour représenter les informations sur les fiches. Le choix de cette taille doit être pondéré par les exigences d'une bonne lisibilité (surtout pour les histogrammes) et par le souci de réduire le scrolling (surtout horizontal). Il doit également tenir compte du fait que les informations peuvent être transférées sur un rapport pour être imprimées. Il doit donc veiller à ce que les fiches ne soient pas plus larges que les rapports afin d'éviter à l'utilisateur de modifier systématiquement les tableaux et histogrammes trop larges.

b) La modification de la taille des informations d'un rapport peut paraître secondaire. Il est exact que cette action est peu importante en regard de l'ensemble des tâches de consultation des fiches et d'élaboration d'un rapport. Toutefois, sa présence a eu une conséquence prépondérante dans la conception de notre outil.

En effet, nous aurions très bien pu ne pas gérer la partie "élaboration de rapports" en laissant le soin à l'utilisateur de transférer les informations copiées dans un logiciel de traitement de texte. Cette solution est fort avantageuse :

---

<sup>1</sup> Pour s'en convaincre, il suffit de prendre connaissance des recommandations du choix des touches [CUA89,35].

- Elle représente moins de travail pour nous et pour l'implémenteur.
- Elle permet à l'utilisateur d'employer toute la puissance de mise en page d'un logiciel de traitement de texte, et en particulier la possibilité d'insérer du texte libre (nous reviendrons sur cette lacune dans le prochain chapitre).
- Dans la mesure où les logiciels de traitement de texte sont plus diffusés que MEANDRES (sic), les rapports pourraient être plus facilement et plus universellement communiqués et utilisés.

Malheureusement, nous ne connaissons pas de logiciel de traitement de texte qui autorise un changement de la taille des graphiques de qualité satisfaisante. Les graphiques modifiés sont réellement déformés. Nous ne sommes pas les seuls à déplorer cette lacune qui réduit souvent à néant les efforts, parfois importants, de mise en page.

Par conséquent, nous avons préféré gérer nous-mêmes l'élaboration des rapports. MEANDRES possédant toutes les informations concernant la réalisation des différents graphiques, nous sommes en droit d'espérer obtenir un changement de taille qui préserve la qualité des graphiques.

Rien n'empêche de prévoir, lors d'une version ultérieure, la possibilité d'ajouter du texte. On pourrait même imaginer des techniques d'hypertextes au niveau des rapports.

---

## **Critiques et améliorations**

Nous avons déjà fait état d'une bonne partie des avis de tiers dans le chapitre précédent, dans la mesure où ils apportaient d'importantes précisions, suggestions et nuances aux commentaires émis.

Nous ne reprenons donc dans le présent chapitre que les critiques qui remettent en question certains choix fondamentaux sur lesquels repose l'interface proposée, ainsi que celles qui nécessitent d'être détaillées, et enfin toutes celles que nous n'avons pas pu intégrer adéquatement dans le précédent chapitre.

### **4.1 CHANGEMENTS IMPORTANTS**

#### **4.1.1 Scinder la gestion des fiches et celle des rapports**

Les discussions du chapitre précédent révèlent le principal point faible de notre interface : l'absence d'une gestion claire et séparée des concepts de fiche et de rapport. En effet, le menu **Fichier** est bancal dans la mesure où certaines actions se rapportent à des fiches, d'autres à des rapports, d'autres enfin aux simulations elles-mêmes. Ce mélange conceptuel, qui se retrouve ailleurs, ne nous a pas paru gênant mais nous reconnaissons volontiers qu'il peut handicaper un utilisateur novice.

Pour remédier aux différentes ambiguïtés de notre interface, il nous a été proposé de mettre explicitement sur un pied d'égalité les concepts de fiches et de rapports en affichant le contenu des rapports dans une fenêtre fille plutôt que dans la zone de travail de la fenêtre principale. Il nous semble que cette suggestion est excellente et qu'elle a le mérite d'ouvrir de nouvelles perspectives très intéressantes (cfr ci-dessous la section concernant la gestion de la liste des documents).

Nous sommes par contre plus réticents vis-à-vis de la proposition visant à associer une barre de menus à chacune des fenêtres filles dans le but de scinder explicitement le menu général en un menu propre aux fiches et en un menu propre aux rapports. Notre réticence vient du fait que les conventions stipulent que seules les fenêtres d'applications (c'est-à-dire ici la fenêtre principale) disposent d'une barre de menus. Nous estimons que ces conventions ont raison d'interdire la présence d'une barre de menus sur les fenêtres documents (c'est-à-dire ici les fiches et les rapports) parce que cela augmenterait la confusion entre les deux types de fenêtres, et que cela diminuerait la lisibilité des fenêtres documents en présentant une information somme toute redondante.

Nous sommes plutôt partisans de disposer d'une barre de menu variable selon que le document courant est une fiche ou un rapport. Voici, à titre d'illustration, une proposition qui demanderait une réflexion beaucoup plus poussée :

Si le document courant est une fiche

**Fichier, Edition, Processus, Ressources, Pt. de synchr., Fenêtre, Aide**

**Fichier :** Nouveau, Ouvrir, Fermer, Modifier le standard, Quitter.

**Edition :** Annuler, Copier, Effacer, Tout sélectionner.

**Processus, Ressources et Pt. de synchr. :** inchangés à l'exception de **Définir fiche standard** remis dans le menu **Fichier**.

**Fenêtre :** Cascade, Mosaïque + la liste de tous les documents non masqués présents sur le bureau.

**Aide :** ?, A propos de la simulation.

L'item **Nouveau** remplacerait la commande **Sélectionner un objet**. L'item **Ouvrir** servirait à ouvrir une nouvelle simulation. La commande **Modifier le standard** est explicitée dans une autre section ci-dessous. Le menu **Edition** ne propose pas les items **Couper** et **Coller**, inutiles dans le contexte d'une fiche. Comme nous le verrons plus loin, la commande **Masquer la fiche** n'est plus nécessaire.

Si le document courant est un rapport

**Fichier, Edition, ?, Fenêtre, Aide**

**Fichier :** Nouveau, Ouvrir, Enregistrer, Enregistrer sous, Fermer, Aperçu avant impression, Mise en page, Imprimer, Quitter.

**Edition :** Annuler, Couper, Copier, Coller, Effacer, Tout sélectionner.

**?** : ce point d'interrogation signale qu'il faudrait envisager au moins un menu supplémentaire qui permettrait par exemple de gérer la frappe d'un texte libre dans des rapports (cfr ci-dessous).

**Fenêtre :** Cascade, Mosaïque, Afficher les règles + la liste de tous les documents non masqués présents sur le bureau.

**Aide :** ?, A propos de la simulation.

L'item **Ouvrir** affiche une première boîte de dialogue demandant de préciser si l'utilisateur souhaite ouvrir un rapport ou une simulation.

Il faudrait également prévoir un moyen pour permettre de basculer du mode fiche en mode rapport et réciproquement. Le moyen le plus simple est de rendre courant une fiche ou un rapport, mais que se passe-t-il s'il n'y a pas encore de fiche présente à l'écran ? On pourrait ajouter à cette fin un item dans le menu **Fenêtre**, par exemple.

Avec du recul, ces propositions semblent évidentes. Il n'en va pas de même quand on est plongé dans la conception d'une interface en suivant une logique particulière. Nous avons adopté une logique d'actions, c'est-à-dire une stratégie de conception d'interfaces qui se base sur les actions à proposer alors que nous aurions dû suivre une logique d'objets !

### **4.1.2 Simulations et rapports**

Nous avons précédemment évoqué qu'en général, les personnes consultées préfèrent lancer l'exécution d'une nouvelle occurrence de MEANDRES à partir de l'Exécutif Ms-Dos. Toutefois, nous sommes persuadés de l'intérêt d'ouvrir une nouvelle simulation à partir de MEANDRES. Nous pourrions adopter un compromis en proposant que cette nouvelle simulation remplace l'ancienne. Ainsi, l'utilisateur ne crée pas de nouvelles occurrences de MEANDRES, il réactualise l'occurrence courante.

Pour d'évidentes raisons de simplicité, nous proposons qu'à tout moment, une fenêtre ne soit associée qu'à une seule simulation. Par conséquent, nous ne comptons pas offrir la possibilité d'ouvrir simultanément plusieurs simulations au sein d'une même occurrence de MEANDRES.

Concrètement, le fait de pouvoir changer de simulation implique que :

- Toutes les fiches correspondant à l'ancienne simulation seront automatiquement fermées. Un message d'avertissement devra donc être affiché lors de l'ouverture de la nouvelle simulation.
- Le titre de la fenêtre principale reflétera ce changement de simulation en affichant le nom de la nouvelle simulation.
- Normalement, tous les rapports devraient également être automatiquement fermés. Des messages devraient demander à l'utilisateur s'il souhaite enregistrer les modifications apportées à ces rapports.

En réalité, il n'est peut-être pas nécessaire de fermer les rapports. Jusqu'à présent, en ne permettant d'insérer dans un rapport que les résultats d'une seule simulation, nous avons étroitement lié les rapports à la simulation qui a produit ces résultats. La raison essentielle de ce lien était de permettre d'exécuter directement MEANDRES à partir de l'Exécutif Ms-Dos en cliquant deux fois sur le nom d'un rapport. Nous n'avons pas remis en question ce lien car nous pensions qu'il aurait permis d'enrichir ultérieurement les rapports par des mécanismes d'hypertextes. Or, à court terme, les utilisateurs estiment essentielle la possibilité d'élaborer un rapport qui reprenne des mesures provenant de diverses simulations.

Le fait de pouvoir ouvrir une nouvelle simulation offre l'opportunité de concevoir des rapports tout à fait généraux. Par conséquent, l'ouverture d'une simulation ne doit pas entraîner la fermeture systématique des rapports présents sur le bureau.

### **4.1.3 Texte libre**

Le troisième changement important concerne la possibilité d'ajouter un texte quelconque à un rapport. Cette facilité est réclamée par les utilisateurs d'IDA qui jugent insuffisants les différents textes accompagnant les mesures statistiques.

Il est indéniable que cette possibilité doit être offerte. Nous ne l'avons pas fait parce qu'elle implique un surcroît de travail d'implémentation. En effet, offrir la possibilité de taper du texte revient à offrir, au minimum, la possibilité de choisir la police, le style, la taille, la position (exposant, indice) des caractères et la justification des lignes.

Nous aurions dû ajouter un menu supplémentaire dans la barre des menus afin de gérer tous ces attributs. Ce menu aurait alourdi notre interface (peut-être nous aurait-il permis de nous rendre compte qu'il fallait scinder les menus). Par contre, il s'intègre aisément dans la nouvelle configuration proposée ci-dessus.

## **4.2 LISTE DES DOCUMENTS**

Nous avons vu que le menu **Ecran** affiche la liste des documents présents sur le bureau. Nous avons omis d'analyser le comportement dynamique de cette liste. En effet, nous avons décrit ce menu en supposant implicitement que l'utilisateur ne manipule, à un moment donné, qu'un nombre réduit de documents et que, par conséquent, la liste des documents reste de taille modeste. Les utilisateurs d'IDA nous ont convaincus que cette hypothèse est erronée dans la mesure où la tâche de navigation est un processus non structuré. En effet, on peut imaginer que dans un premier temps l'utilisateur va examiner tous les processus, et dans un deuxième temps les ressources se rapportant à certains processus pathologiques. De ce fait, si l'utilisateur ne ferme pas de temps à autre des fiches, la liste des documents risque d'être fort longue au point de ne pas pouvoir être entièrement affichée.

Que faire pour éviter ce problème ?

Une solution, mise en œuvre dans Ms-Word, serait de faire coulisser la liste des documents. Autrement dit, si la liste est trop grande, l'utilisateur peut glisser le pointeur de la souris vers le bas du menu et faire monter progressivement la liste tronquée. Bien entendu, au fur et à mesure que l'utilisateur accède au bas de la liste, il n'a plus accès au sommet de la liste. L'expérience montre que ce mécanisme est peu agréable à utiliser. Nous ne le retenons pas.

La solution évidente pour éviter que la liste des documents ne grandisse outre mesure serait d'interdire à l'utilisateur d'ouvrir trop de documents. Il faudrait donc fixer un nombre maximum pour chaque type de documents (rapports, fiches se rapportant à un processus, fiches se rapportant à une ressource et fiches se rapportant à un point de synchronisation). Ces bornes supérieures seront certainement différentes (on élabore moins de rapports qu'on ne consulte de fiches sur les processus) et devront être déterminées empiriquement. Ensuite, il faudrait déterminer le comportement à adopter quand l'utilisateur souhaite ouvrir un nouveau document alors qu'il a déjà atteint le nombre maximum autorisé. Deux alternatives :

- Rendre inactifs les menus permettant de créer un nouveau document.
- Fermer automatiquement le document le plus ancien de même type que celui que l'utilisateur souhaite ouvrir.

La première alternative peut être implémentée en rendant réellement indisponibles ces menus (les mettre en gris), ou en affichant un message signalant à l'utilisateur qu'il est momentanément dans l'impossibilité de créer un nouveau document à moins de fermer un ancien document. La deuxième proposition est de loin plus conviviale (volonté de renseigner le plus possible l'utilisateur et de lui laisser l'impression qu'il pilote le système).

La deuxième alternative doit afficher un message d'avertissement signalant à l'utilisateur que s'il poursuit son action, le document un tel sera fermé.

Cette dernière alternative présente l'avantage d'éviter à l'utilisateur d'avoir à fermer lui-même un document, mais ne lui laisse pas le choix du document à fermer. Par conséquent, nous préférons la première solution.

Une troisième solution peut être envisagée pour réduire le nombre de documents affichés dans le menu **Ecran**. Elle consiste à éliminer de cette liste les documents masqués. Cette solution a le mérite de rencontrer le souhait de certaines personnes qui préfèrent que les documents masqués soient réduits sous forme d'une icône placée en bas de la zone de travail de la fenêtre principale. Ce souhait est légitime puisqu'il reflète plus la philosophie de Ms-Windows (Ms-Windows 3 propose en effet ce mécanisme) tandis que notre mécanisme découle plus d'une vision Macintosh du masquage. Ce sujet nous a valu des discussions homériques tant les utilisateurs de systèmes informatisés tiennent à conserver leurs habitudes de travail.

Cette solution ne peut être mise en œuvre que si nous effectuons les changements évoqués plus haut. En effet, elle nécessite que les rapports soient visualisés à partir de fenêtres filles, et cela pour deux raisons :

- Par convention, une icône est la représentation réduite d'une fenêtre, et pas de la zone de travail d'une fenêtre.
- La zone de travail de la fenêtre principale ne peut à la fois contenir un rapport et des icônes de fenêtres réduites.

Signalons que l'item **Masquer la fiche** n'est plus nécessaire. On lui préférera la case de réduction icônique standard que nous ajouterons aux fenêtres filles.

Enfin, il serait utile de prévoir des icônes différentes pour distinguer facilement les différents types de documents.

### **4.3 MODIFIER LE STANDARD DES FICHES**

Le mécanisme de définition d'une fiche standard que nous avons proposé élude deux aspects importants :

- La redéfinition d'un standard est-elle permanente, c'est-à-dire sera-t-elle conservée lors d'une prochaine exécution de MEANDRES ?
- La redéfinition d'un standard est-elle universelle, c'est-à-dire influence-t-elle le standard des fiches des autres occurrences de MEANDRES présentes au moment de cette redéfinition ?

Nous pensons que la réponse à la première question doit être positive. En effet, la redéfinition d'un standard doit être permanente afin d'éviter à l'utilisateur d'avoir à recommencer cette tâche chaque fois qu'il exécute MEANDRES.

Cette permanence doit toutefois être gérée convenablement. En effet, d'une part, l'utilisateur n'est peut-être pas seul à utiliser MEANDRES. Par conséquent, il faut tenir compte du fait qu'un autre utilisateur souhaite sans doute disposer d'un autre standard. D'autre part, un même utilisateur peut vouloir définir plusieurs standards afin d'utiliser celui qui lui convient le mieux à un moment donné.

Nous devons donc offrir la possibilité de définir plusieurs standards et d'adopter à tout moment n'importe lequel de ces standards. Il faut également prévoir la possibilité de revenir au standard prédéfini de MEANDRES.

Dans la mesure où nous désirons conserver une certaine indépendance entre deux occurrences de MEANDRES, nous ne souhaitons pas qu'une redéfinition d'un standard modifie systématiquement le standard des autres occurrences. Toutefois, en vertu de ce qui précède, nous offrons la liberté à l'utilisateur d'associer explicitement ce nouveau standard à une autre occurrence de MEANDRES.

Il nous reste à lever une dernière ambiguïté. Parlons-nous ici de la même notion de "standard" que précédemment ? Pas tout à fait. Dans les chapitres précédents, nous traitons du concept de fiche standard, en ce sens que nous gérons séparément le standard de chaque type de fiches (par exemple, celui concernant les fiches contenant les statistiques globales des processus). Par contre, notre réflexion actuelle nous pousse à envisager un regroupement des anciens standards.

En effet, si nous voulons qu'un utilisateur puisse conserver les caractéristiques des standards et les associer à tout moment aux fiches qu'il consulte, nous devons lui permettre de mémoriser ces standards au sein d'un fichier. Il serait toutefois impensable d'imposer à l'utilisateur d'avoir à gérer autant de fichiers qu'il n'y a de types de fiches. Par conséquent, nous proposons de considérer les standards de chaque type de fiches comme définissant un seul standard (enregistrable dans un seul fichier).

Cette modification de point de vue impose que nous changions le nom de l'item **Définir fiche standard** en **Modifier le standard des fiches**<sup>1</sup>. De plus, nous ne pouvons plus associer cet item aux menus **Processus**, **Ressources**, **Pt. de synchr.** Cette redondance ne se justifie plus dans la mesure où le concept de "standard" n'est plus vraiment lié à un type de fiches. Par conséquent, nous proposons de placer cet item dans le menu **Fichier** puisque modifier un standard revient à modifier, créer ou lire un fichier. Certains logiciels placent plutôt cette commande dans le menu **Edition**.

Concrètement, ces propos nous amènent à associer la boîte de dialogue modale suivante à l'item **Modifier le standard des fiches** :

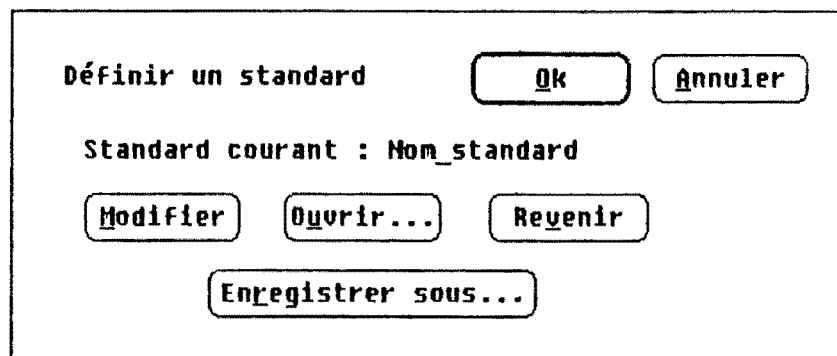


Illustration 71 : Boîte de dialogue de la commande **Modifier le standard des fiches**

Passons en revue les principaux composants de cette boîte de dialogue.

- Comme chaque boîte de dialogue, la boîte de dialogue de la commande **Modifier le standard des fiches** possède un titre rappelant l'item qui l'a créée.
- Nous rappelons également le nom du standard en cours. Ce nom est celui du fichier contenant les caractéristiques des standards des différentes fiches. Cette information est cruciale dans la mesure où l'utilisateur souhaite modifier ces standards de façon durable, c'est-à-dire réaliser une action potentiellement destructrice.
- Le bouton *Modifier* permet d'enregistrer les caractéristiques de tous les types de fiches dans le fichier du standard courant. Peut-être aurait-il fallu préférer le terme "enregistrer" plutôt que "modifier".
- Le bouton *Enregistrer sous* permet d'enregistrer les caractéristiques de tous les types de fiches dans un fichier différent de celui du standard courant. Si l'utilisateur clique sur ce bouton, il verra apparaître la boîte de dialogue d'enregistrement habituelle.

<sup>1</sup> Nous ne tenons pas compte ici des idées émises dans la section 4.1.1.



- Le bouton *Ouvrir* permet d'ouvrir un fichier contenant un autre standard que le standard courant. Ce bouton affiche la boîte de dialogue d'ouverture de fichier habituelle.
- Le bouton *Revenir* permet de revenir au standard prédéfini. Contrairement aux standards définis par l'utilisateur, le standard prédéfini ne possède pas de fichier. Le texte indiquant le nom du standard courant sera *Standard prédéfini*.

#### 4.4 DIVERS

a) Il nous a été reproché de ne pas avoir scindé les items **Nouveau**, **Ouvrir**, **Fermer** des items **Enregistrer** et **Enregistrer sous**. Pourtant la plupart des logiciels conformes à l'environnement Ms-Windows 2 (et Ms-Windows 3) regroupent ces cinq items. Il en va de même pour le monde Macintosh. A notre connaissance, seul IBM recommande de scinder ces items en deux groupes [CUA89].

Nous pensons qu'il vaut mieux les laisser ensembles car ils ont une sémantique analogue : ils régissent le cycle de vie des fichiers : création/ouverture, modification et fermeture. Si on souhaite vraiment séparer ces cinq items, alors il faudra plutôt les séparer en trois groupes.

b) L'expression **Sélectionner un objet** n'est pas heureuse dans la mesure où elle ne correspond pas tout à fait à l'effet de cette commande. Nous proposons de changer le nom de cet item en **Créer une fiche**<sup>1</sup>. Cette dénomination reflète mieux la spécificité de cette commande.

En effet, le but fondamental de cet item n'est pas vraiment de sélectionner un objet (c'est une étape intermédiaire), mais plutôt d'afficher une fiche particulière. De plus, comme nous l'avons signalé dans le deuxième chapitre, nous ne conseillons pas à l'utilisateur d'avoir recours à cette commande pour sélectionner une fiche présente sur le bureau : il existe d'autres méthodes plus rapides. Par contre, cette commande est le seul moyen dont dispose l'utilisateur pour afficher une nouvelle fiche, c'est-à-dire pour créer une fiche.

L'inconvénient de cette dénomination est qu'elle introduit le concept de fiche dans le menu **Fichier**, ce que nous nous sommes toujours interdits de faire.

c) La boîte de dialogue de la commande **Sélectionner un objet** nécessite quelques améliorations ergonomiques. En effet, le bouton de commande *Schéma* est regroupé avec les boutons *Globales*, *Périodiques* et *Chronologiques*, ce qui induit l'idée, fautive, que ces quatre contrôles sont similaires. De plus, nous utilisons les boutons *Processus*, *Ressources* et *Pt. de synchr.* de manière non conventionnelle : contrairement à HyperCard, Ms-Windows n'autorise pas qu'un bouton de commande reflète explicitement le résultat d'un choix. Ce rôle est plutôt celui des boîtes à cocher, ou dans notre cas, des boutons-radios.

---

<sup>1</sup> Nous ne tenons pas compte ici des idées émises dans la section 4.1.1.

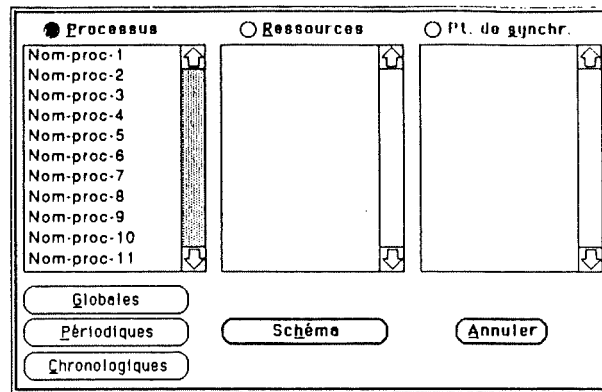


Illustration 72 : Nouvelle boîte de dialogue de l'item **Sélectionner un objet**

Le lecteur notera que le fait d'avoir choisi des boutons-radios oblige que l'une des trois listes soit sélectionnée et affichée. Quelle doit être cette liste ? Deux alternatives s'offrent à nous :

- Choisir toujours la même liste (par exemple, la liste des processus).
- Choisir la liste en fonction du contexte (par exemple, choisir la liste correspondant au type d'objets de la fiche courante<sup>1</sup>).

La première solution a le mérite d'offrir une boîte de dialogue stable. La deuxième solution évitera souvent d'avoir à sélectionner une autre liste. Il nous semble que cette dernière alternative est plus intéressante.

Enfin, nous avons utilisé le même mnémonique pour les boutons *Processus* et *Périodiques* : p. En soi, un mnémonique ne doit pas nécessairement être unique. Il est toutefois plus facile d'utiliser un mnémonique unique : si un mnémonique est unique, sa sélection active directement le bouton de commande associé; sinon, sa sélection rend courant le prochain bouton de commande sans l'activer.

Nous pourrions choisir un autre mnémonique pour l'une de ces deux expressions. Mais quelle devrait être l'expression à sacrifier ? En effet, "Processus" et "Périodiques" se rencontrent ailleurs que dans la boîte de dialogue de la commande **Sélectionner un objet**, et chaque fois, leur mnémonique est p. Par conséquent, nous sommes réticents à modifier ce mnémonique, non seulement pour des raisons d'homogénéité, mais surtout pour éviter à un utilisateur fréquent (mais distrait) de commettre une erreur en employant par habitude la lettre p.

Nous souhaitons donc garder le double mnémonique, à condition toutefois d'en proposer une gestion plus "intelligente". En effet, plutôt que de passer au bouton *Processus* ou au bouton *Périodiques* selon un critère de proximité par rapport au contrôle courant, nous proposons que le mnémonique se rapporte prioritairement au bouton *Processus*, sauf si un objet (processus, ressource) est sélectionné dans l'une des listes. Dans ce dernier cas, le mnémonique rendra courant le bouton *Périodiques*.

d) On nous a proposé d'améliorer la boîte de dialogue de la commande **Imprimer** en offrant la possibilité à l'utilisateur d'incrémenter (décrémenter) le nombre de copies à imprimer simplement en cliquant sur une flèche.

<sup>1</sup> Pour être complets, nous devons préciser que si le document courant est un rapport, alors la liste sélectionnée sera la liste des processus.



**Copies :**  

Illustration 73 : Ascenseur facilitant la spécification du nombre de copies

e) Certaines personnes ont émis le désir de disposer d'une meilleure boîte de dialogue pour la commande **Effectuer la mise en page**. En particulier, cette boîte de dialogue devrait permettre de choisir les différentes options de mise en page à l'aide de contrôles explicites plutôt qu'à l'aide d'un système de codes quelque peu archaïque.

f) Par souci d'homogénéité, certaines corrections seraient à apporter :

- Les items du menu **Fichier** devraient tous être des verbes<sup>1</sup>. Par conséquent, une personne nous a proposé de changer **Aperçu avant impression** au profit de **Prévisualiser**. Signalons que cette légère incohérence ne gêne pas d'autres personnes qui souhaitent plutôt diminuer la longueur de l'item **Effectuer la mise en page** en remplaçant cet item par **Mise en page** afin d'être cohérent avec **Aperçu avant impression**.
- L'item **Calendrier** du menu **Processus** devrait se voir ajouter un s. En effet, l'item **Calendriers** est plus approprié dans la mesure où il permet de consulter deux types de calendriers.
- Dans la boîte de dialogue gérant les statistiques globales des processus, nous proposons les options suivantes :

Durée de vie moyenne  
Nombre moyen de processus

Il serait souhaitable d'uniformiser ces expressions :

Durée moyenne de vie  
Nombre moyen de processus

- ...

g) Nous avons précédemment signalé qu'il faudrait compléter le titre des boîtes de dialogue permettant de sélectionner les mesures statistiques à consulter en ajoutant le nom du type de l'objet analysé (processus, ressource, point de synchronisation). Certaines personnes à qui nous avons parlé de cette amélioration se sont montrées réticentes car elles estiment que cette indication supplémentaire ne ferait qu'alourdir les boîtes de dialogue. Elles reconnaissent toutefois la nécessité de proposer cette indication, et avancent une solution intéressante : placer le nom du type de l'objet analysé dans une barre d'état.

Cette solution pourrait être généralisée pour offrir plus de renseignements sur l'action que l'utilisateur est occupé de réaliser. Bien sûr, il serait impératif de permettre à l'utilisateur expérimenté de masquer cette barre s'il ne souhaite pas encombrer son champ de vision avec ce système d'aide.

h) On nous a fait la suggestion de prévoir un bouton d'aide sur les boîtes de dialogue les plus complexes.

i) Les histogrammes présentent généralement plusieurs types de mesures en même temps (par exemple, l'histogramme illustrant l'évolution du nombre d'occurrences des événements d'un processus présente les mesures concernant les nombres de déclenchements, d'activations, d'interruptions, de réactivations et de terminaisons). Nous n'avons pas précisé si, par défaut, ces histogrammes reprennent toutes les mesures, ou seulement certaines, jugées plus utiles (par exemple, les nombres de déclenchements, d'activations et de terminaisons).

---

<sup>1</sup> Pourtant l'item **Nouveau** est universellement utilisé.

On peut regretter cet oubli car il aurait peut-être été judicieux de laisser la possibilité à l'utilisateur de gérer les choix par défaut des mesures présentées par un histogramme (souci de personnalisation). Notre mémoire devrait donc être complété par une réflexion sur ce sujet, et, en particulier, sur la manière dont l'utilisateur pourrait spécifier ces options (par exemple, menu déroulant dans la boîte de dialogue à côté de l'item de sélection de l'histogramme).

j) La boîte de dialogue gérant les mesures globales concernant les durées des processus contient beaucoup d'options. Certaines de ces options avaient un texte descriptif trop long. Nous avons donc dû les abréger.

Durées	Ok		Annuler	
	Min.	Moy.	Max.	Ecart
Durée d'activité	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Durée d'attente d'act.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Durée d'interruption	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Durée moy. d'inter./proc.		<input type="checkbox"/>		
Durée de repos	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Durée moy. de repos/proc.		<input type="checkbox"/>		
Durée moy. de vie		<input type="checkbox"/>		

Illustration 74 : Boîte de dialogue gérant les durées des processus

Certaines personnes nous ont fait remarquer que notre stratégie d'abréviation était malheureuse. En effet, pourquoi abréger les parties significatives des textes, alors que pour gagner de la place, il suffisait d'éliminer le mot "durée" redondant dans le contexte de la boîte de dialogue ?

A cette remarque pertinente, ajoutons celle qui met en cause la cohérence du mécanisme de raccourcis. En effet, nous utilisons le mnémonique du titre pour sélectionner toutes les mesures offertes par la boîte de dialogue. En procédant ainsi, nous conférons au titre un rôle différent non seulement du rôle habituel d'un titre, mais surtout du rôle des titres des autres boîtes de dialogue de MEANDRIS<sup>1</sup>. Incohérence donc ! De plus, dans le chapitre précédent, nous avons déjà fait remarquer que ce mécanisme était en léger porte à faux avec celui adopté dans les boîtes de dialogue de la commande **St. globales** du menu **Pt. de synchr.**

Pour tenir compte de ces critiques, nous proposons cette nouvelle boîte de dialogue<sup>2</sup> :

<sup>1</sup> A l'exception toutefois de celui de la boîte de dialogue qui gère les mesures globales des états d'un processus.

<sup>2</sup> Il faudrait également corriger la boîte de dialogue qui gère les mesures globales des états d'un processus.

<b>Durées</b>		<input type="button" value="Ok"/>		<input type="button" value="Annuler"/>	
<input type="radio"/>	Toutes les <u>durées</u>				
<input type="radio"/>	Durées...	<b>Min.</b>	<b>Moy.</b>	<b>Max.</b>	<b>Ecart</b>
	d' <u>act</u> ivité	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	d' <u>att</u> ente d'activation	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	d' <u>int</u> erruption	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	moy. d' <u>inter.</u> /processus		<input type="checkbox"/>		
	de <u>re</u> pos	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	moy. de <u>re</u> pos/processus		<input type="checkbox"/>		
	moy. de <u>vie</u>		<input type="checkbox"/>		

Illustration 75 : Boîte de dialogue revue et corrigée

k) Certaines personnes ont regretté l'absence d'une facilité que nous avons initialement pensé offrir (cfr annexe 7 : "Première maquette"). Il s'agit de la possibilité d'obtenir l'éclatement d'un histogramme présentant des mesures chronologiques en plusieurs sous-histogrammes correspondant chacun à une période d'observation.

L'éclatement en plusieurs histogrammes permet de visualiser, de manière statique, l'ensemble des périodes d'observation. Si, dans le cas d'une fiche, nous lui avons préféré le mécanisme du scrolling, il n'en demeure pas moins que l'éclatement reste la seule solution pour visualiser toutes les périodes d'un histogramme sur une feuille de papier.

Jusqu'à présent, nous laissons le soin à l'utilisateur de déterminer, à l'aide du scrolling, les périodes qui l'intéressaient, et de coller sur un rapport autant de fois l'histogramme qu'il n'y a de périodes intéressantes. Certains utilisateurs d'IDA estiment qu'ils auront souvent à comparer des périodes différentes et souhaitent disposer d'un mécanisme qui leur évite ces nombreux "copier/coller".

A l'heure actuelle, nous envisageons l'utilisation d'un bouton de commande placé à côté de l'histogramme sur la fiche. Cliquer sur un bouton provoque soit l'éclatement de l'histogramme en plusieurs sous-histogrammes placé l'un en dessous de l'autre (et la disparition de la barre de défilement horizontal), soit le retour à l'histogramme initial. Tous ces sous-histogrammes ne forment qu'une entité graphique transférable sur un rapport.

Ce mécanisme n'est pas entièrement satisfaisant puisque, d'une part il nous force à ajouter un objet interactif sur les fiches (le bouton de commande), et que d'autre part, il ne permet de transférer que l'ensemble des sous-histogrammes : l'utilisateur devra effacer les sous-histogrammes superflus dans le rapport.

l) Le nom de l'item **Commentaires** n'est pas heureux puisque les informations qu'il donne ne sont pas vraiment des commentaires.

m) De manière générale, il faudrait que l'interface tiennent mieux compte des différents niveaux des utilisateurs : novices et experts.

---

## A propos de la conception d'interfaces

### 5.1 ITINERAIRE

Comme la plupart des étudiants, nous avons commencé notre mémoire en parcourant la littérature se rapportant au sujet étudié. En particulier, nous avons été séduits par le mémoire de B. Sacré et de J. Lefèvre [Lef87] qui émettaient des suggestions très avancées en vue d'améliorer la consultation des résultats statistiques produits par DSL/SIM.

Afin d'améliorer notre connaissance de la spécificité des actions que doit réaliser le concepteur de S.I., nous souhaitons dialoguer avec des utilisateurs d'IDA. Pour que les discussions soient fructueuses, il était impératif que nous basions notre discours sur des propositions concrètes. Ces propositions n'avaient pour objectifs que de susciter les réactions de tiers. Parce qu'il nous semblait que les suggestions de B. Sacré et de J. Lefèvre pouvaient constituer ce point de départ recherché, nous avons voulu les concrétiser, les affiner voire les compléter.

Malheureusement, nous nous sommes rapidement heurtés à une difficulté que nous allions encore rencontrer : comment exprimer les idées en matière d'interfaces ? Pour que le lecteur se fasse une idée de ces difficultés, nous lui proposons un petit exercice : essayer de décrire oralement la boîte de dialogue de la commande **Sélectionner un objet**. Le lecteur se rendra compte que l'attention de son interlocuteur diminue rapidement. Ensuite, le lecteur peut demander à son interlocuteur de décrire à son tour la boîte de dialogue à une troisième personne qui devra la dessiner. Le résultat risque d'être étonnant.

Pour faciliter la communication de nos premières idées, nous avons eu recours à HyperCard afin de réaliser assez rapidement une petite maquette (cfr annexe 7 "Première maquette"). Cette maquette nous a été très précieuse et nous a permis

d'atteindre notre but, à savoir connaître les souhaits de certains utilisateurs d'IDA en matière de consultation des résultats de simulations et d'élaboration de rapports.

Parallèlement à ce travail, nous nous étions lancés dans l'apprentissage du langage C et de la programmation en Ms-Windows 2 puisqu'il était convenu que nous devions implémenter notre futur outil. Il est généralement connu que la maîtrise de la programmation en Windows demande énormément de temps. Nous n'avons pas fait exception à cette règle, au point que nous nous sommes rendus compte que nous n'aurions pas le temps de programmer notre interface. Non seulement, au mois de février, nous n'étions arrivés qu'à la moitié de notre apprentissage, mais il nous apparaissait clairement que la programmation de notre interface allait nous demander une maîtrise de Windows que seule une expérience incompatible avec le temps qui nous restait pouvait nous donner. A ce propos, il nous semble extrêmement difficile pour un étudiant ne connaissant préalablement ni le langage C ni la programmation sous Windows de pouvoir implémenter dans cet environnement un programme conséquent dans le cadre d'un mémoire.

Voyant le temps fuir plus certainement qu'une poignée de sable dans une main, nous avons voulu proposer un brouillon d'interface afin de fixer quelque peu le cadre de nos recherches. A ce moment, il n'était nullement question de ne pas analyser la tâche. Nous ressentions seulement le besoin d'avoir quelque chose de concret à proposer, de faire le point. En quelque sorte, nous souhaitions recommencer ce que nous avions fait avec la première maquette, mais à une échelle et avec une qualité supérieures.

Nous nous sommes donc centrés sur la conception d'une interface en nous basant sur le mémoire de B. Sacré et de J. Lefèvre et sur les renseignements que nous avons pu obtenir grâce à notre première maquette. Une nouvelle surprise nous attendait : le temps et l'investissement que nous avons demandés ce "brouillon" ont été prohibitifs. Nous avons été pris au piège : il n'était plus possible de faire marche arrière, de trouver le temps de réaliser une analyse de la tâche sérieuse.

Nous disposions encore moins de temps que nous rencontrions de nouveau notre vieux problème : la communication. Pour permettre à des tiers de se faire une bonne idée de nos propositions, il s'avérait nécessaire de concevoir une nouvelle maquette. Cependant, étant donnée la complexité de notre brouillon, cette maquette a demandé un investissement non négligeable, et cela d'autant plus que nous ne disposions pas d'outils permettant de la réaliser rapidement.

Ainsi, par le force des choses, nous n'avons pas pu donner un cadre théorique rigoureux à notre interface. Heureusement, la présentation que nous avons faite de notre maquette à différentes personnes nous a assurés que malgré de réelles imperfections, notre interface n'est pas dénuée de qualités et qu'elle satisfait la plupart des besoins des utilisateurs d'IDA.

Pour convaincre le lecteur que la conception d'une interface n'est pas une tâche négligeable, nous reprenons ci-dessous le temps (en heures) qu'il nous a fallu pour mener à bien le présent mémoire. Il s'agit d'une estimation minimale.

Lectures "théoriques" :	40
Réalisation de la première maquette :	20
Apprentissage du C et de Windows <sup>1</sup> :	200
Elaboration du "brouillon" :	120
Réalisation de la deuxième maquette :	60

---

<sup>1</sup> Nous ne sommes arrivés qu'à la moitié de notre apprentissage.

Présentations de cette maquette :	10
Conception de l'algorithme de gestion d'histogrammes <sup>1</sup> :	20
Conception d'une barre de défilement <sup>2</sup> :	20
Rédaction finale :	160
<i>Total :</i>	<i>650</i>

## **5.2 SUGGESTIONS POUR FACILITER LE TRAVAIL DU CONCEPTEUR D'INTERFACES**

Le lecteur a sans doute éprouvé quelques difficultés à lire le deuxième chapitre, tant il était long et dense. Voilà un premier problème : comment détailler de manière claire, précise et conviviale une proposition d'interface ? Peut-on imaginer disposer un jour d'un formalisme, d'une grammaire qui simplifierait la tâche de description d'une interface ?

L'absence de ce formalisme nous a forcé à avoir recours à la langue française. Cette dernière, comme tout langage naturel, est ambiguë. Par conséquent, décrire avec précision une interface demande des trésors de rigueur et de concision. Nous retirons beaucoup de cette phase de rédaction.

Nous pouvons également déplorer la quantité de temps nécessaire à cette activité de description. Nous ne remettons pas en cause la nécessité d'une bonne description d'une interface, mais nous estimons que cela exige trop de temps en regard de l'ensemble de la tâche à effectuer (analyse, conception, implémentation). Le formalisme évoqué ci-dessus pourrait apporter une solution à ce problème.

Le lecteur a sans doute remarqué que l'étape de justification est également conséquente. Elle nous semble intéressante car elle constitue l'occasion d'exprimer les silences, c'est-à-dire les choix que nous avons été amenés à écarter. De même que le bord d'un trou caractérise autant un trou que le vide qui le compose, nous pensons que les options non retenues donne également une bonne idée de la nature de l'interface.

Nous formons le vœux que les futurs concepteurs d'interfaces disposent d'une taxonomie détaillée, complète et pratique des règles ergonomiques afin de leur faciliter l'étape de justification. En effet, non seulement une telle taxonomie leur permettrait de justifier leurs choix de manière rigoureuse et homogène, mais elle leur offrirait une structure qui garantirait la complétude de leur démarche.

Un texte, aussi bien écrit soit-il, ne permet pas de se faire facilement une idée d'une interface. En effet, un texte est inévitablement linéaire et statique, alors qu'une interface est dynamique en ce sens qu'elle se modifie en fonction du contexte d'utilisation. Par conséquent, ou bien on rédige la description d'une interface sous forme d'un hypertexte, ou bien on est obligé d'en faire une maquette. La première solution est loin d'être évidente : nous ne connaissons pas de théories de conception d'hypertextes. La deuxième nécessite des outils appropriés. Nous ne disposons pas de tels outils. Nous avons utilisé HyperCard qui est peut-être l'outil le plus adapté à l'heure actuelle<sup>3</sup>, mais ce logiciel (du moins la version que nous avons utilisée) ne permet pas de simuler aisément une gestion des boîtes de dialogue. En effet, il faut

---

<sup>1</sup> Cfr annexe 3 "Algorithme de gestion d'histogrammes".

<sup>2</sup> Cfr annexe 4 "Créer une barre de défilement en HyperCard".

<sup>3</sup> Il est arrivé plus d'une fois que des personnes, voyant notre maquette réalisée avec HyperCard, nous avouent qu'elles ne savaient que Ms-Windows tournait également sur le Macintosh !!



savoir qu'en HyperCard, tout doit être dessiné<sup>1</sup>. Par conséquent, la combinatoire induite par tous les choix des options des boîtes de dialogue étant explosive, il serait ahurissant de simuler toutes les combinaisons possibles. C'est pourquoi nous avons dû réellement programmer le comportement des boîtes de dialogue en Ms-Windows 2<sup>2</sup>.

Nous rêvons d'un outil de prototypage d'interfaces qui permettrait au concepteur de construire facilement une maquette très fidèle, mais surtout qui lui permettrait d'éviter d'avoir à décrire longuement sa proposition d'interface. En effet, pourquoi ne pas imaginer un outil d'aide à la conception d'interfaces qui permettrait de se passer de la description sur papier, et d'éliminer ainsi une redondance d'autant plus inutile qu'elle consomme beaucoup de temps. Nous ne remettons pas en question l'utilité d'avoir une description formelle d'une interface, mais il nous semble inutile de disposer d'une telle description pour chaque "brouillon" d'interfaces.

En effet, il faut bien être conscient du fait que la conception d'une interface est un travail itératif dans la mesure où il est impératif de soumettre toute proposition aux critiques de personnes possédant des profils très divers et de corriger ces propositions en tenant compte des critiques et suggestions émises. Cette vision cybernétique de la conception d'interfaces trouve ses justifications dans le fait que la conception d'interfaces est un processus créatif reflétant les limites intrinsèques des individus, inévitablement liés par leur expérience (nous avons vu l'influence souvent prépondérante, parfois contraignante, des habitudes d'utilisation de systèmes informatisés). D'autre part, le but d'une interface étant de faciliter la tâche des utilisateurs, il faut constamment vérifier qu'elle rencontre les souhaits de ces derniers. Il est inutile de rappeler la difficulté de cerner ces souhaits si ce n'est pour signaler qu'ici cette difficulté est particulièrement aiguë puisque la subjectivité humaine entre en ligne de compte (des goûts et des couleurs, il faut discuter !).

Pour ces différentes raisons, il est clair qu'il faut souvent remettre en question certains aspects, parfois fondamentaux, d'une interface. Si à chaque refonte de l'interface, le concepteur doit passer une centaine d'heure à décrire la nouvelle mouture, il n'aura certainement pas envie de soumettre trop souvent ses propositions à la critique, d'autant plus qu'il n'est jamais agréable de voir remettre en question son labeur (phénomène d'identification et d'appropriation). Par conséquent, plus souple sera la méthode de description d'une interface, plus nombreux seront les contrôles et meilleur sera le produit final.

Nous avons attiré plus d'une fois l'attention du lecteur sur l'instabilité inhérente à la conception d'une interface. En effet, il est arrivé plus d'une fois qu'un détail remette en question tout l'édifice construit. Concevoir une interface est un travail ingrat : tant que l'on n'a pas passé en revue tous les détails, rien ne garantit que ce que l'on a déjà réalisé est valable. Au début, le concepteur se trouve devant une liberté totale et angoissante; au fur et à mesure qu'il opère des choix, il restreint la liberté originelle; il peut alors arriver qu'il ne dispose plus de liberté du tout, au point que ses choix ultérieurs soient entièrement déterminés par l'état actuel de l'interface ("L'homme passe la première moitié de sa vie à se forger des chaînes, et la seconde moitié à se plaindre d'avoir à les porter" Mirabeau); pire, ces "choix" forcés peuvent être en contradiction avec d'autres. A ce moment, il ne reste plus au concepteur qu'à recommencer sur de nouvelles bases en espérant avoir cette fois plus de chance.

---

<sup>1</sup> Le lecteur peut se rendre compte en annexe 6 du travail que demande la conception d'une maquette avec HyperCard.

<sup>2</sup> Le lecteur trouvera les sources de ce programme dans l'annexe 5. Nous attirons l'attention du lecteur sur la quantité de lignes de code nécessaires à la gestion des boîtes de dialogues et sur la difficulté pour un profane de les décrypter. Quand on sait que la gestion des boîtes de dialogue et des menus constitue certainement la partie de programmation la plus aisée de Windows, ...

La conception d'une interface nous fait penser à ce "gag" quelque peu cynique de Franquin :

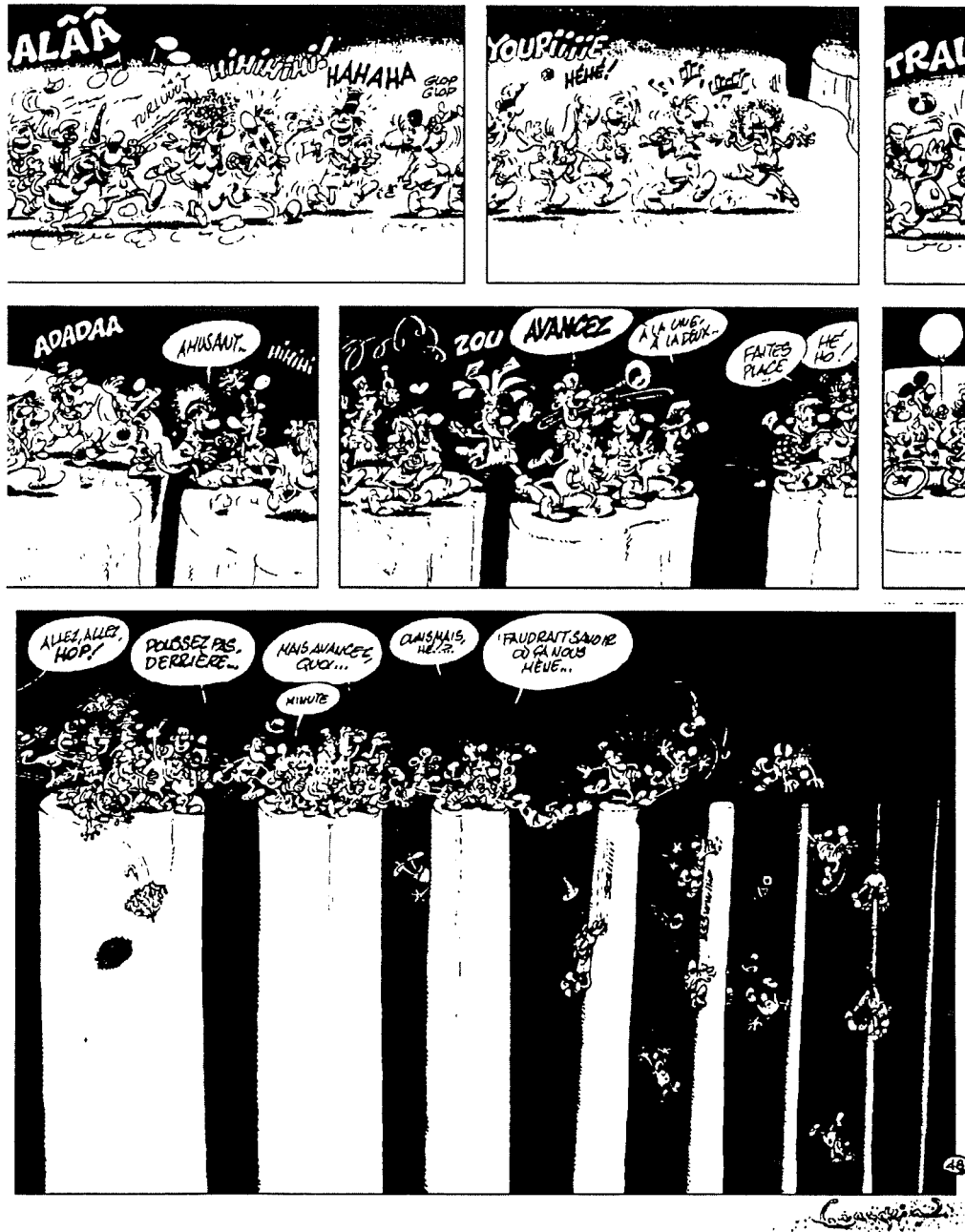


Illustration 76 : Métaphore ?

Bien sûr, nous avons caricaturé la situation. Il n'en demeure pas moins vrai qu'il est très difficile, voire impossible, à un concepteur même expérimenté d'imaginer toutes les conséquences des choix qu'il opère. C'est pourquoi, il nous semble qu'il faudrait lui fournir un outil qui lui permette de déterminer avec exactitude les contraintes de ses choix. Un tel système expert rendrait des services précieux et éviterait de nombreux efforts inutiles.

---

## Conclusion

Dans ce mémoire, nous avons essayé de concevoir une interface pour un outil de consultation de résultats statistiques et d'élaboration de rapports. Pour atteindre cet objectif, nous avons adopté une démarche pragmatique essentiellement basée sur notre expérience des logiciels de bureautique. L'absence d'une analyse de la tâche du concepteur de S.I. et d'une réflexion théorique globale ne permet pas de garantir que nous avons satisfait les besoins des utilisateurs d'IDA. Toutefois, bien que présentant des lacunes et de réelles faiblesses, nous pensons que notre outil améliore sensiblement la situation existante. Nous formons le souhait que l'utilisation de notre interface aide à l'amélioration de la connaissance des besoins des utilisateurs et serve de base pour une future analyse de la tâche.

Nous avons essayé de faire sentir au lecteur que le travail de conception d'une interface constitue une de ces occasions privilégiées où se mélangent l'imagination créatrice et la rigueur opérationnelle. Concevoir une interface est très différent de critiquer une interface. Bien que ces deux tâches demandent une rigueur et des connaissances similaires, la conception reste un de ces domaines où la liberté est reine. C'est une grande chance pour un étudiant de pouvoir faire acte de créativité.

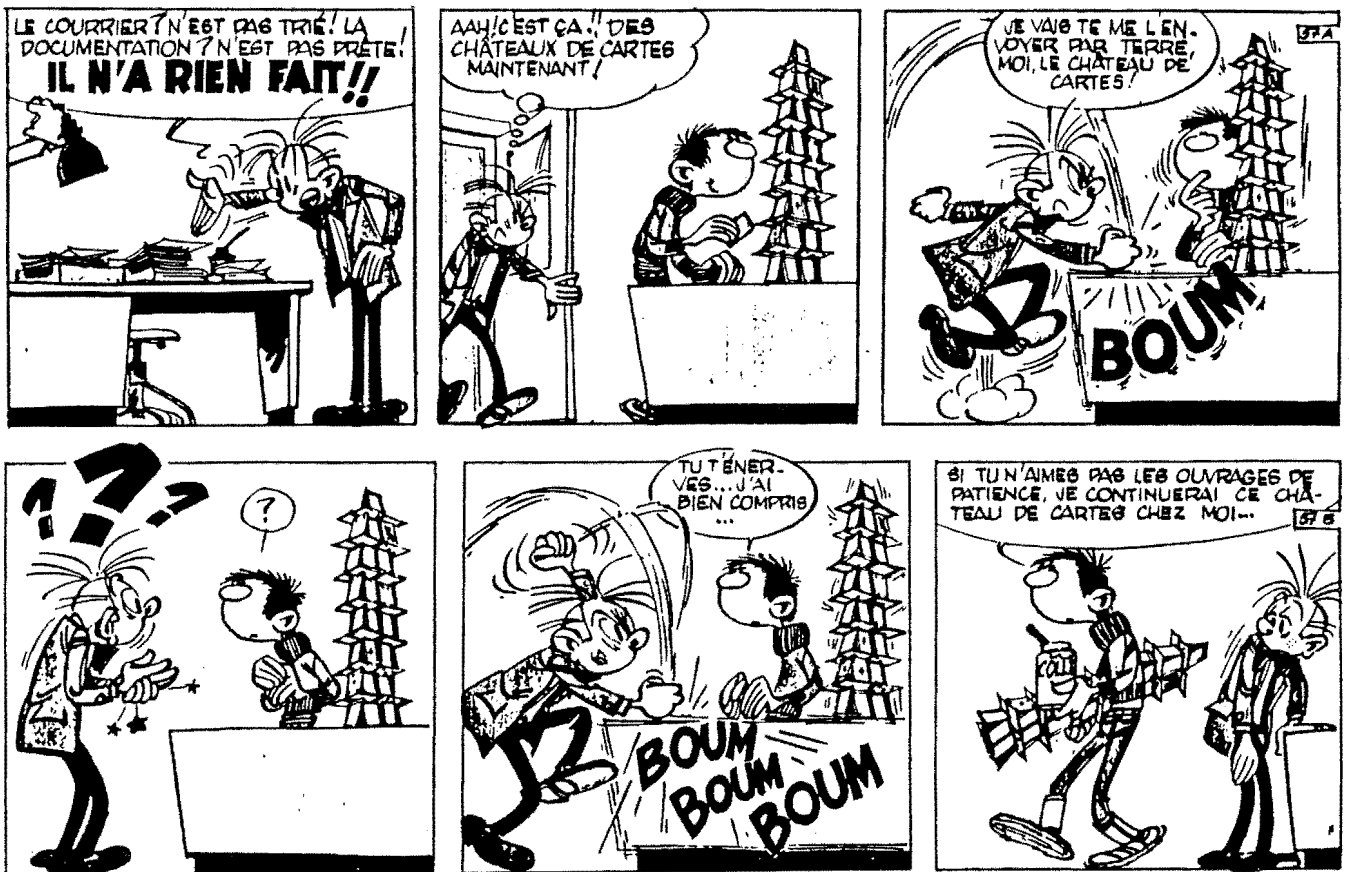
Ce mémoire nous a permis d'aller au bout d'une préoccupation qui nous était apparue lors de notre projet personnel : comment concevoir un mode de dialogue convivial entre l'homme et la machine. Si le cours d'"interface homme/machine" nous a fourni un cadre de réflexion, ce mémoire nous a donné l'occasion de répondre concrètement aux questions que nous nous étions posées.

Nous pensons avoir beaucoup appris lors de ce mémoire. Non seulement nous avons pu parfaire notre connaissance de l'environnement de Ms-Windows et en tirer profit lors des cours que nous assurons, non seulement nous avons pu parfaire nos capacités de rédaction et de communication, mais surtout nous avons pu toucher certaines de nos limites, et ainsi mieux nous connaître.

Bien sûr, il reste beaucoup à faire. Nous avons souligné certains changements importants à apporter, quelques améliorations possibles. Nous engageons vivement les personnes concernées par la conception d'interfaces à réfléchir sur les difficultés auxquelles nous nous sommes heurtés et sur les suggestions que nous proposons en vue de les éliminer. Nous souhaitons rester à leur disposition pour leur faire part de notre modeste expérience.

Enfin, nous aimerions terminer en résumant notre impression globale de la conception d'interfaces à l'aide d'une métaphore :

*Concevoir une interface, c'est construire un château de cartes. Il est facile de placer les éléments de base; mais plus l'édifice se construit, plus il devient fragile à tel point que le souffle d'un détail provoque l'anéantissement du fruit d'un labeur passionné.*



## Bibliographie

- [Blo89] BLOEM D., VIRGA, *Aide-mémoire d'HyperTalk*, Marabout, Allleur (Belgique), 1989.
- [Bod89] BODART F., PIGNEUR Y., *Conception assistée des systèmes d'information : méthode - modèles - outils*, Masson, 1989.
- [Bod90] BODART F., *Cours introductif aux interfaces homme-machine*, notes de cours, Institut d'Informatique, F.U.N.D.P., Namur, 1990.
- [Cod89] *Code typographique*, Fédération C.G.C. de la communication, 1989.
- [CUA89] *Common User Access - Advanced Interface Design Guide*, International Business Machines Corp., Boca Raton (Etats-Unis), 1989.
- [Den90] DENIS J., *MEANDRES*, rapport réalisé dans le cadre d'un projet personnel, Institut d'Informatique, F.U.N.D.P., Namur, 1990.
- [Exc89] *Manuel de référence de Microsoft Excel version 2.2 pour le Macintosh d'Apple*, Microsoft Corporation, 1989.
- [Hyp87] *Guide de l'utilisateur HyperCard*, Apple Computer France, 1987.
- [Hyp88] *Guide du langage HyperTalk*, Apple Computer France, 1988.
- [IDA88] - Manuel de référence DSL, version 2.1.  
- Manuel de référence de DSL/SIM, version 2.1.
- [Jou89] JOURQUIN PH., VANDERDONCKT J., *Evaluation de l'outil de simulation DSL-SIM*, rapport réalisé dans le cadre du cours "Conception des systèmes d'information, matières approfondies", Institut d'Informatique, F.U.N.D.P., Namur, 1989.
- [Kar83] KARNAS G., SALENGROS P., *L'ergonomie : adapter ?*, in La Revue Nouvelle, n°3, 1983, pp 272-281.
- [Ker84] KERNIGHAN B. W., RITCHIE D. M., *Le langage C*, Masson, 1984.
- [Lam91] LAMBERT J.-M., *Initiation à HyperCard*, Centre de Calcul, F.U.N.D.P., Namur, 1991.
- [Lef87] LEFEVRE J., SACRE B., *Interprétation graphique des mesures de performance d'un système*, mémoire de fin d'études, Institut d'Informatique, F.U.N.D.P., Namur, 1987.
- [Leo89] LEONARD C., *HyperCard facile*, Marabout, Allleur (Belgique), 1989.
- [Pet88] PETZOLD CH., *Programming Windows - The Microsoft Guide for the MS-DOS Presentation Manager : Windows 2.0 and Windows/386*, Microsoft Press, Redmond, Washington, 1988.
- [Pet90] PETZOLD CH., *Programming Windows - The Microsoft guide to writing applications for Windows 3*, Microsoft Press, Redmond, Washington, 1990.

- [Pro90a] PROVOT I., SACRE B., *Description of the architecture model for a highly interactive business-oriented application*, Projet TRIDENT, Institut d'Informatique, F.U.N.D.P., Namur, 1990.
- [Pro90b] PROVOT I., SACRE B., *Description of TRIDENT integrated environment*, Projet TRIDENT, Institut d'Informatique, F.U.N.D.P., Namur, 1990.
- [Pro90c] PROVOT I., VANDERDONCKT J., *Les objets interactifs : classification et typologie*, rapport IHM/Ergo/5, Institut d'Informatique, F.U.N.D.P., Namur, 1990.
- [Reu87] REUVIAUX D., al., *Rapport de simulation DSL-SIM*, rapport réalisé dans le cadre du cours "Conception des systèmes d'information, matières approfondies", Institut d'Informatique, F.U.N.D.P., Namur, 1987.
- [Sac88] SACRE B., *Programmation sous Windows*, Institut d'Informatique, F.U.N.D.P., Namur, 1988.
- [Sca86] SCAPIN D., *Guide ergonomique de conception des interfaces homme-ordinateur*, Institut National de Recherche en Informatique et en Automatique, Rocquencourt (France), 86.
- [Shn87] SHNEIDERMAN B., *Designing the User Interface : Strategies for effective Human-Computer Interaction*, Addison-Wesley Publishing Company, Inc., 1987.
- [Van90a] VANDERDONCKT J., *Generating presentation for a highly interactive business-oriented application*, Projet TRIDENT, Institut d'Informatique, F.U.N.D.P., Namur, 1990.
- [Van90b] VANDERDONCKT J., *Modèle de la présentation d'une application hautement interactive : concepts et règles ergonomiques*, rapport IHM/Ergo/6, Institut d'Informatique, F.U.N.D.P., Namur, 1990.
- [Van91] VANDERDONCKT J., *Conseils et directives concernant les travaux pratiques du cours d'Interface Homme-Machine*, Institut d'Informatique, F.U.N.D.P., Namur, 1991.
- [Win86] *Microsoft Windows - Software Development Kit : Application Style Guide version 1.03*, Microsoft Corporation, 1986.
- [Win90] *Microsoft Windows - Guide de l'utilisateur version 3.0*, Microsoft Corporation, 1990.
- [Wor89] *Manuel de référence de Microsoft Word version 4 pour le Macintosh d'Apple*, Microsoft Corporation, 1989.

FACULTES UNIVERSITAIRES  
NOTRE-DAME DE LA PAIX



NAMUR

INSTITUT D'INFORMATIQUE

**MEANDRES**

**ou**

**Proposition d'une interface pour un Module  
d'Evaluation et d'ANalyse De RESULTats  
Statistiques**

*Annexes*

X. Gillo

Promoteur : F. Bodart

Mémoire présenté en vue de  
l'obtention du titre de licencié  
et maître en informatique

Année académique 1990-1991

RUE GRANDGAGNAGE, 21, B - 5000 NAMUR (BELGIUM)

## Table des matières

<b>1 Structure des listings produits par DSL/SIM</b> .....	1
1.1 Processus .....	1
1.2 Processeurs .....	3
1.3 Points de synchronisation.....	4
<b>2 Utilisation d'une boîte de dialogue</b> .....	5
<b>3 Algorithme de gestion d'histogrammes</b> .....	13
3.1 Exposé du problème .....	13
3.2 Algorithme .....	14
3.3 Prototype .....	15
3.3.1 Script du bouton <i>Flèche Haut</i> .....	16
3.3.2 Script du bouton <i>Barre verticale</i> .....	16
3.3.3 Script du bouton <i>Flèche Bas</i> .....	18
3.3.4 Procédure <i>Histogramme</i> .....	19
3.3.5 Exemples .....	24
<b>4 Créer une barre de défilement en HyperCard</b> .....	27
4.1 Exposé du problème .....	27
4.2 Algorithme .....	28
4.3 Code .....	29
<b>5 Maquette : partie Windows</b> .....	35
5.1 Introduction .....	35
5.2 meandres.c.....	37
5.3 fichier.h .....	43
5.4 processu.h.....	44
5.5 ressour.h .....	67
5.6 synchr.h .....	78
5.7 meandres.h.....	88
5.8 meandres.rc .....	90
5.9 meandres.dlg .....	92
<b>6 Maquette : partie HyperCard</b> .....	113
6.1 Pile <i>Simulation</i> .....	113
6.2 Pile <i>Sélection</i> .....	119
<b>7 Première maquette</b> .....	121
7.1 Objectifs .....	121
7.2 Concepts .....	121



7.3 Présentation sommaire de l'interface .....	122
7.3.1 Sélectionner une fiche .....	122
7.3.2 Ranger une fiche.....	124
7.3.3 Passer à une autre fiche présente sur le bureau .....	124
7.3.4 Visualiser une autre partie d'un graphique.....	125
7.3.5 Effectuer un changement d'échelle .....	126
7.3.6 Imprimer .....	126

---

## Structure des listings produits par DSL/SIM

### 1.1 PROCESSUS

#### Statistiques globales

PROCESS nom-de-processus

REPORTED BETWEEN t-debut AND t-fin

CALENDAR RELATED TO nom-de-processus  
SPANS FROM ...

Triggering Number  
Inception Number  
Interruption Number  
Resumption Number  
Termination Number

	Min	Mean	Max	Std Dev
Active Time	...	(1)	...	...
Waiting Time	...	(2)	...	...
Intr Time by Intr	...	(3)	...	...
Intr Time by Process		(4)		
Idle Time by Incp/Rsmp	...	(5)	...	...
Idle Time by Process		(6)		
Elapse Time		(7)		

*Structure des listings produits par DSL/SIM*

Numb of Proc in Wait State	...	(8)	...	...
Numb of Proc in Inter State	...	(9)	...	...
Numb of Proc in Actv State	...	(10)	...	...

Required Resource :	Capacity		Sharability		Queue Size
	Max	Used	Max	Used	
nom-de-processeur-1	...	...	...	...	...
...	...	...	...	...	...
...	...	...	...	...	...
nom-de-processeur-n	...	...	...	...	...

- (1) : durée moyenne d'activité.
- (2) : durée moyenne d'attente d'activation.
- (3) : durée moyenne d'interruption.
- (4) : durée moyenne d'interruption par processus.
- (5) : durée moyenne de repos.
- (6) : durée moyenne de repos par processus.
- (7) : durée moyenne de vie.
- (8) : nombre moyen de processus en attente d'activation.
- (9) : nombre moyen de processus interrompus.
- (10) : nombre moyen de processus en activité.

**Statistiques périodiques**

[Low Bnd	Upp Bnd[	Number of					Wait	Intr Time	Idle Time
t1	t2	Trig	Incp	Intr	Resm	Term	Time	by Intr	by Incp/Intr
		(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)

- (1) : nombre de déclenchements durant toutes les cellules [t1,t2[.
- (2) : nombre d'activations durant toutes les cellules [t1,t2[.
- (3) : nombre d'interruptions durant toutes les cellules [t1,t2[.
- (4) : nombre de réactivations durant toutes les cellules [t1,t2[.
- (5) : nombre de terminaisons durant toutes les cellules [t1,t2[.
- (6) : durée moyenne d'attente d'activation durant toutes les cellules [t1,t2[.
- (7) : durée moyenne d'interruption durant toutes les cellules [t1,t2[.
- (8) : durée moyenne de repos durant toutes les cellules [t1,t2[.

**Statistiques chronologiques**

PROCESS nom-de-processus

REPORTED BETWEEN t-debut AND t-fin

		Number of			Wait time	Int time	Nb of proc in			Wait Numb	
Trig	Incp	Intr	Resm	Term	by intr	by intr	Wait	Intr	Actv	Min	Last
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)

- (1) : nombre de déclenchements durant la cellule [t1,t2[.
- (2) : nombre d'activations durant la cellule [t1,t2[.
- (3) : nombre d'interruptions durant la cellule [t1,t2[.
- (4) : nombre de réactivations durant la cellule [t1,t2[.
- (5) : nombre de terminaisons durant la cellule [t1,t2[.
- (6) : durée moyenne d'attente d'activation durant la cellule [t1,t2[.
- (7) : durée moyenne d'interruption durant la cellule [t1,t2[.
- (8) : nombre moyen de processus en attente d'activation durant la cellule [t1,t2[.
- (9) : nombre moyen de processus interrompus durant la cellule [t1,t2[.

- (10) : nombre moyen de processus en activité durant la cellule [t1,t2[.  
 (11) : nombre minimum de processus en attente d'activation durant la cellule [t1,t2[.  
 (12) : nombre de processus en attente d'activation à la fin de la cellule.

## 1.2 PROCESSEURS

### Statistiques globales

RESOURCE nom-de-processeur

REPORTED BETWEEN t-début AND t-fin

CALENDAR RELATED TO nom-de-processeur  
 SPANS FROM ...

Maximum Capacity  
 Maximum Sharability

Request Number  
 Allocate Number  
 Deallocate Number

	Min	Mean	Max	Std Dev
Used Capacity	...	(1)	...	...
Used Sharability	...	(2)	...	...
Queue Size	...	(3)	...	...

Process which requires this resource	Wait-Time	Intr-Time by intr	Actv-Time	Numb of proc. in state		
				wait	Intr	Actv
nom-de-processus-1	...	...	...	...	...	...
...	...	...	...	...	...	...
...	...	...	...	...	...	...
nom-de-processus-n	...	...	...	...	...	...

- (1) : moyenne de la capacité utilisée.  
 (2) : moyenne du nombre de requêtes satisfaites simultanément.  
 (3) : moyenne du nombre de processus en attente.

### Statistiques périodiques

[Low Bnd	Upp Bnd[	Request (1)	Number of Alloc (2)	Dealloc (3)
t1	t2			

- (1) : nombre de requêtes durant toutes les cellules [t1,t2[.  
 (2) : nombre d'allocations durant toutes les cellules [t1,t2[.  
 (3) : nombre de désallocations durant toutes les cellules [t1,t2[.

**Statistiques chronologiques**

RESOURCE nom-de-processeur

REPORTED BETWEEN t-début AND t-fin

Req	All	Dal	Mean	Queue Size Max	Last	Mean Cap	Mean Sha
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)

- (1) : nombre de requêtes durant la cellule [t1,t2[.
- (2) : nombre d'allocations durant la cellule [t1,t2[.
- (3) : nombre de désallocations durant la cellule [t1,t2[.
- (4) : nombre moyen de processus en attente à un instant donné dans la cellule [t1,t2[.
- (5) : nombre maximum de processus en attente à un instant donné dans la cellule [t1,t2[.
- (6) : nombre de processus en attente au temps t2.
- (7) : moyenne de la capacité utilisée durant la cellule [t1,t2[.
- (8) : moyenne du nombre de requêtes satisfaites simultanément durant la cellule [t1,t2[.

**1.3 POINTS DE SYNCHRONISATION**

**Statistiques globales**

SYNCHRONIZATION POINT nom-de-pt-de-synchro

REPORTED BETWEEN t-debut AND t-fin

REALIZED-WHEN Text (and contribution mode)

[1] opérande ( ONE ou MANY ) ou opérateur  
 ...  
 [n] opérande ( ONE ou MANY ) ou opérateur

times	Number of			Realization or Participation		
	Realiz. Std Dev	Contrib.	Partic.	Min	Mean	Max
[1]	...	...	...	...	(1)	...
[i]	...	...	...	...	(2)	...
	...					

Constraining Event	In Time	Per Freq
[i]	...	...

- (1) : moyenne de la durée de réalisation.
- (2) : moyenne de la durée de participation d'un événement.

## 2

---

# Utilisation d'une boîte de dialogue

Cette annexe vous propose d'examiner de plus près l'utilisation de la boîte de dialogue de la commande **Sélectionner un objet** du menu **Edition**.

### CAS 1

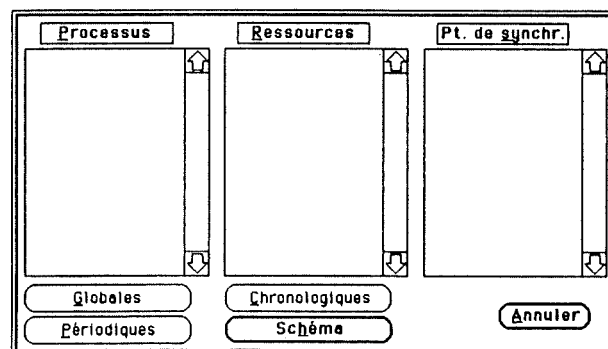


Illustration 1 : cas 1

Quand l'utilisateur choisit la commande **Sélectionner un objet** cette boîte de dialogue apparaît. Remarquez que, d'une part, cinq boutons de commandes sont actifs : *Annuler*, *Schéma*, *Processus*, *Ressources* et *Pt. de synchr.*, et, d'autre part, trois boutons sont inactifs : *Globales*, *Périodiques* et *Chronologiques*. A ce stade, l'utilisateur se trouve devant trois alternatives :

- Cliquer sur le bouton *Processus*, *Ressources* ou le bouton *Pt. de Synchr.*
- Cliquer sur le bouton *Schéma*.

- Annuler la commande **Sélectionner un objet**.

S'il clique sur le bouton *Processus*, *Ressources* ou le bouton *Pt. de Synchr.*, alors se référer au cas 2, 5 ou 8.

S'il clique sur le bouton *Schéma*, comme aucun objet n'est sélectionné, par défaut, la fenêtre d'affichage du schéma montre la partie supérieure gauche du schéma. Rappelons que si lors de l'ouverture des fichiers sources de la simulation, MEANDRES n'a pas trouvé le fichier contenant les informations nécessaires à l'élaboration du schéma de la dynamique, le bouton *Schéma* est grisé.

## CAS 2

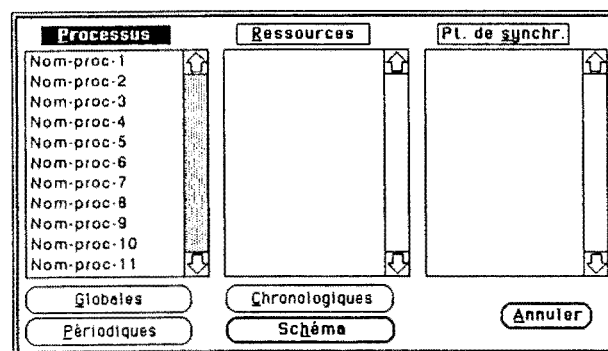


Illustration 2 : cas 2

L'utilisateur a cliqué sur le bouton *Processus*. Par conséquent, la liste de sélection des processus contient l'ensemble des processus présents dans la simulation. A ce stade, l'utilisateur se trouve devant quatre alternatives :

- Sélectionner un processus dans la liste de sélection.
- Cliquer sur le bouton *Ressources* ou le bouton *Pt. de Synchr.*
- Cliquer sur le bouton *Schéma*.
- Annuler la commande **Sélectionner un objet**.

S'il sélectionne un des processus, alors se référer au cas 3.

S'il clique sur le bouton *Ressources* ou le bouton *Pt. de Synchr.*, alors se référer au cas 5 ou 8.

S'il clique sur le bouton *Schéma*, alors le schéma de la dynamique apparaît. Sa fenêtre montre la partie supérieure gauche du schéma.

## CAS 3

L'utilisateur a sélectionné un processus. Remarquez que la liste de sélection des ressources contient la liste des ressources requises, directement ou non, par le processus sélectionné. A ce stade, l'utilisateur se trouve devant six alternatives :

- Sélectionner un autre processus dans la liste de sélection.
- Sélectionner une des ressources requises.
- Cliquer sur le bouton *Ressources* ou le bouton *Pt. de Synchr.*

- Cliquer sur le bouton *Schéma*.
- Cliquer sur le bouton *Globales, Périodiques, Chronologiques*.
- Annuler la commande **Sélectionner un objet**.

S'il sélectionne une des ressources requises, alors se référer au cas 4.

S'il clique sur le bouton *Ressources* ou le bouton *Pt. de Synchr.*, alors se référer au cas 5 ou 8.

S'il clique sur le bouton *Schéma*, alors le schéma de la dynamique apparaît. Sa fenêtre est centrée sur le processus sélectionné.

S'il clique sur le boutons *Globales, Périodiques, ou Chronologiques*, alors la boîte de dialogue habituelle de sélection des informations à consulter apparaît (cfr commandes *St. globales, St. périodiques, St. chronologiques*).

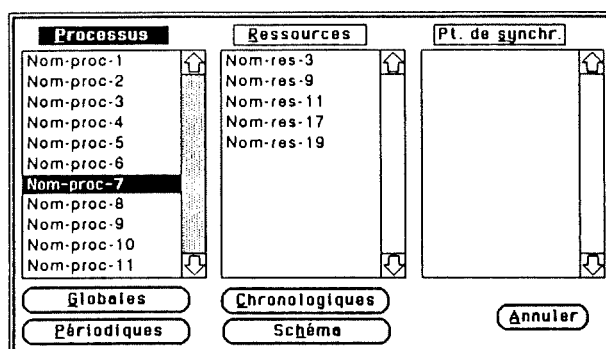


Illustration 3 : cas 3

#### CAS 4

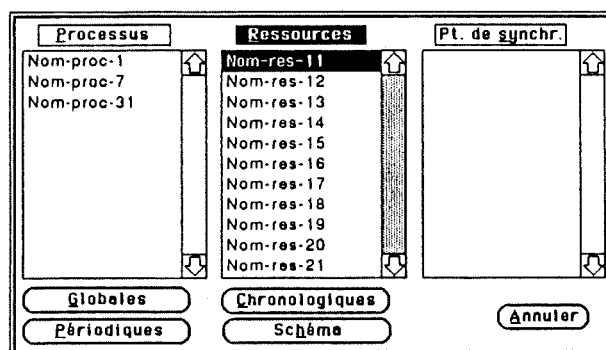


Illustration 4 : cas 4

L'utilisateur a sélectionné une des ressources requises par le processus précédemment sélectionné. Remarquez que la liste de sélection des ressources contient maintenant l'ensemble des ressources présentes dans la simulation, et que la liste de sélection des processus contient l'ensemble des processus qui requièrent la ressource sélectionnée. A ce stade, l'utilisateur se trouve devant six alternatives :

- Sélectionner une autre ressource dans la liste de sélection.
- Sélectionner un des processus requérants.



- Cliquer sur le bouton *Processus* ou le bouton *Pt. de Synchr.*
- Cliquer sur le bouton *Schéma*.
- Cliquer sur le bouton *Globales, Périodiques, Chronologiques*.
- Annuler la commande **Sélectionner un objet**.

S'il sélectionne une autre ressource, alors se référer au cas 6.

S'il sélectionne un des processus requérants, alors se référer au cas 7.

S'il clique sur le bouton *Processus* ou le bouton *Pt. de Synchr.*, alors se référer au cas 2 ou 8.

S'il clique sur le bouton *Schéma*, alors le schéma de la dynamique apparaît. Sa fenêtre montre la partie supérieure gauche du schéma.

S'il clique sur les boutons *Globales, Périodiques, ou Chronologiques*, alors la boîte de dialogue habituelle de sélection des informations à consulter apparaît.

## CAS 5

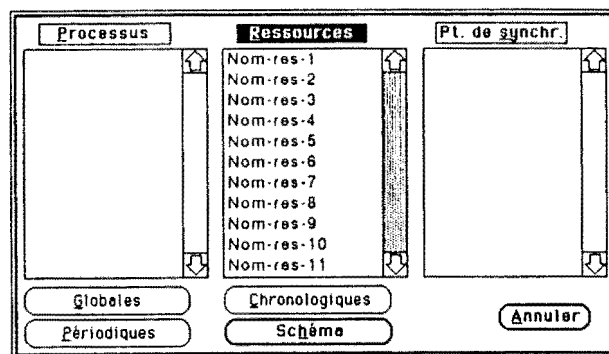


Illustration 5 : cas 5

L'utilisateur a cliqué sur le bouton *Ressources*. Par conséquent, la liste de sélection des ressources contient l'ensemble des ressources présentes dans la simulation. A ce stade, l'utilisateur se trouve devant quatre alternatives :

- Sélectionner une ressource dans la liste de sélection.
- Cliquer sur le bouton *Processus* ou le bouton *Pt. de Synchr.*
- Cliquer sur le bouton *Schéma*.
- Annuler la commande **Sélectionner un objet**.

S'il sélectionne une des ressources, alors se référer au cas 6.

S'il clique sur le bouton *Processus* ou le bouton *Pt. de Synchr.*, alors se référer au cas 2 ou 8.

S'il clique sur le bouton *Schéma*, alors le schéma de la dynamique apparaît. Sa fenêtre montre la partie supérieure gauche du schéma.

## CAS 6

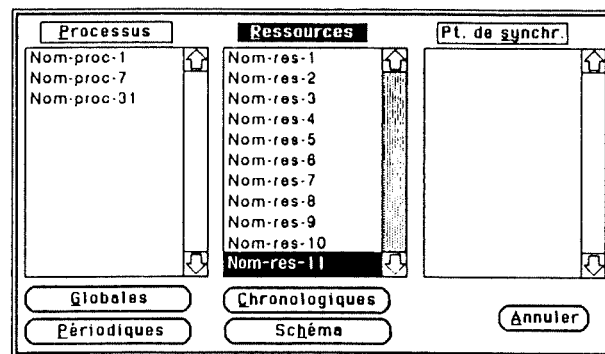


Illustration 6 : cas 6

L'utilisateur a sélectionné une ressource. Remarquez que la liste de sélection des processus contient la liste des processus qui requièrent, directement ou non, la ressource sélectionnée. A ce stade, l'utilisateur se trouve devant six alternatives :

- Sélectionner une autre ressource dans la liste de sélection.
- Sélectionner un des processus requérants.
- Cliquer sur le bouton *Processus* ou le bouton *Pt. de Synchr.*
- Cliquer sur le bouton *Schéma*.
- Cliquer sur le bouton *Globales, Périodiques, Chronologiques*.
- Annuler la commande **Sélectionner un objet**.

S'il sélectionne un des processus requérants, alors se référer au cas 7.

S'il clique sur le bouton *Processus* ou le bouton *Pt. de Synchr.*, alors se référer au cas 2 ou 8.

S'il clique sur le bouton *Schéma*, alors le schéma de la dynamique apparaît. Sa fenêtre est centrée sur le processus sélectionné.

S'il clique sur le boutons *Globales, Périodiques, ou Chronologiques*, alors la boîte de dialogue habituelle de sélection des informations à consulter apparaît.

## CAS 7

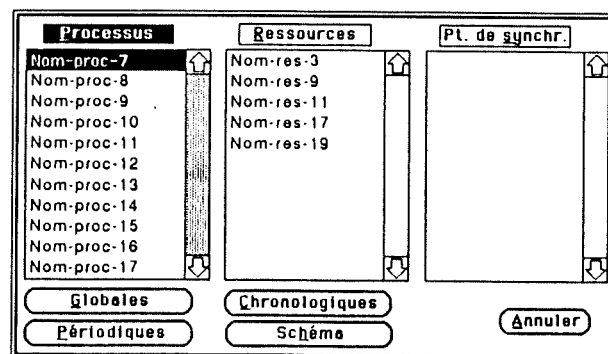


Illustration 7 : cas 7

L'utilisateur a sélectionné un des processus requérant la ressource précédemment sélectionnée. Remarquez que la liste de sélection des processus contient maintenant l'ensemble des processus présents dans la simulation, et que la liste de sélection des ressources contient l'ensemble des ressources requises par le processus sélectionné. A ce stade, l'utilisateur se trouve devant six alternatives :

- Sélectionner un autre processus dans la liste de sélection.
- Sélectionner une des ressources requises.
- Cliquer sur le bouton *Ressources* ou le bouton *Pt. de Synchr.*
- Cliquer sur le bouton *Schéma*.
- Cliquer sur le bouton *Globales, Périodiques, Chronologiques*.
- Annuler la commande **Sélectionner un objet**.

S'il sélectionne un autre processus, alors se référer au cas 3.

S'il sélectionne une des ressources requises, alors se référer au cas 4.

S'il clique sur le bouton *Ressources* ou le bouton *Pt. de Synchr.*, alors se référer au cas 5 ou 8.

S'il clique sur le bouton *Schéma*, alors le schéma de la dynamique apparaît. Sa fenêtre est centrée sur le processus sélectionné.

S'il clique sur le boutons *Globales, Périodiques, ou Chronologiques*, alors la boîte de dialogue habituelle de sélection des informations à consulter apparaît.

## CAS 8

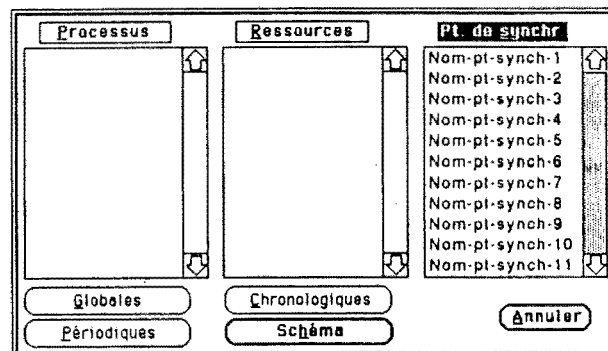


Illustration 8 : cas 8

L'utilisateur a cliqué sur le bouton *Pt. de synchr.* Par conséquent, la liste de sélection des points de synchronisation contient l'ensemble des points de synchronisation présents dans la simulation. A ce stade, l'utilisateur se trouve devant quatre alternatives :

- Sélectionner un point de synchronisation dans la liste de sélection.
- Cliquer sur le bouton *Processus* ou le bouton *Ressources*.
- Cliquer sur le bouton *Schéma*.
- Annuler la commande **Sélectionner un objet**.

S'il sélectionne un des points de synchronisation, alors se référer au cas 9.

S'il clique sur le bouton *Processus* ou le bouton *Ressources*, alors se référer au cas 2 ou 5.

S'il clique sur le bouton *Schéma*, alors le schéma de la dynamique apparaît. Sa fenêtre montre la partie supérieure gauche du schéma.

## CAS 9

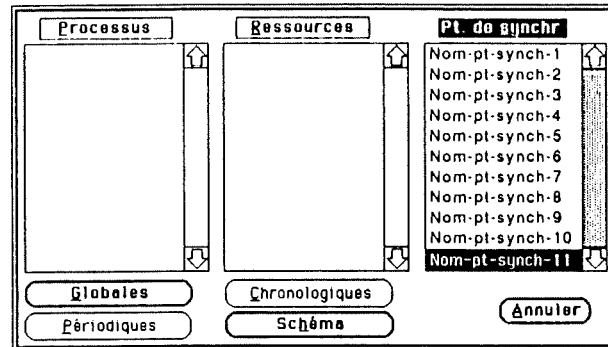


Illustration 9 : cas 9

L'utilisateur a sélectionné un point de synchronisation. A ce stade, il se trouve devant cinq alternatives :

- Sélectionner un autre point de synchronisation dans la liste de sélection.
- Cliquer sur le bouton *Processus* ou le bouton *Ressources*.
- Cliquer sur le bouton *Schéma*.
- Cliquer sur le bouton *Globales*.
- Annuler la commande **Sélectionner un objet**.

S'il clique sur le bouton *Processus* ou le bouton *Ressources*, alors se référer au cas 2 ou 5.

S'il clique sur le bouton *Schéma*, alors le schéma de la dynamique apparaît. Sa fenêtre est centrée sur le point de synchronisation sélectionné.

S'il clique sur le bouton *Globales*, alors la boîte de dialogue habituelle de sélection des informations à consulter apparaît.

---

## **Algorithme de gestion d'histogrammes**

### **3.1 EXPOSE DU PROBLEME**

Les différents histogrammes présentés dans MEANDRES sont munis, à gauche de l'axe des ordonnées, d'une barre de défilement vertical. Cette barre a pour fonction de modifier l'échelle de l'axe des ordonnées.

Ainsi, lors de la création de l'histogramme, l'ascenseur est positionné au sommet de la barre permettant à l'utilisateur de visualiser l'entièreté de tous les bâtonnets. Le fait de déplacer l'ascenseur, disons à la moitié de la barre de défilement, a pour effet de modifier l'échelle de l'axe de moitié (environ). Bien entendu, les graduations de l'axe des ordonnées reflètent ce changement d'échelle. Autrement dit, si à l'origine la graduation maximale était 100, le déplacement de l'ascenseur fixe la nouvelle graduation maximale à 50.

Nous avons élaboré un algorithme qui détermine le nombre optimal d'intervalles (c'est-à-dire de graduations), ainsi que leur valeur, permettant de visualiser une certaine valeur. Mais que veut dire optimal ?

Nous avons lié le critère d'optimalité au nombre maximum d'intervalles représentables. En effet, étant donné un axe des ordonnées de hauteur fixe, il est clair que plus nous permettons d'intervalles, plus nous offrons un découpage précis de la valeur à représenter. Toutefois, le mieux étant l'ennemi du bien, il n'est pas recommandé d'afficher une graduation trop fine : l'utilisateur risque d'être saturé par cette information trop riche et non nécessairement pertinente. Par conséquent, nous devons fixer un nombre maximal d'intervalles. Ce nombre dépend de plusieurs paramètres : la hauteur de l'axe des ordonnées, la hauteur de la police de caractères choisie (afin d'éviter le recouvrement des valeurs des graduations), et la charge informationnelle (à

déterminer empiriquement selon des critères ergonomiques). Idéalement, ce nombre maximal devrait constituer le nombre optimal.

Deux alternatives s'offrent à nous : privilégier le nombre d'intervalles ou privilégier la valeur des intervalles. La première alternative revient à dire que quelle que soit la valeur à représenter, le nombre d'intervalles est constant et égal au nombre optimal fixé une fois pour toutes. Dans ce cas, la valeur unitaire d'un intervalle est égale à la valeur à représenter divisée par le nombre d'intervalles. L'inconvénient de cette méthode est que les intervalles peuvent prendre n'importe quelle valeur, en ce y compris des nombres irrationnels, ce qui est contraire aux habitudes. En effet, généralement, les gens préfèrent travailler avec des valeurs entières ou rationnelles usuelles (0.5, 1, 2, 5, 10, 100 ...) qu'ils maîtrisent mieux. C'est pourquoi, nous privilégions la deuxième alternative, à savoir *déterminer la valeur des intervalles qui permettent d'obtenir un nombre d'intervalles le plus proche du nombre optimal.*

Pour ce faire, nous nous donnons un ensemble de bases, c'est-à-dire de valeurs d'intervalles potentielles : par exemple, 0.5, 1, 2, 5, 10, 20, 50, 100, 200, 500. Le choix de cet ensemble dépend de la moyenne des valeurs à représenter. En effet, si l'algorithme est appelé à représenter des valeurs faibles, le concepteur a tout intérêt à disposer des bases décimales. Inversement, il doit prévoir des bases élevées pour pouvoir tenir compte des valeurs importantes. C'est là une limite de la méthode proposée. Par exemple, si le nombre maximal d'intervalles autorisé est 15, et que la base la plus élevée est 10, alors l'algorithme ne fonctionne pas pour des valeurs supérieures à 150 ! Le choix des bases est donc crucial. En termes de performances, plus l'ensemble de bases est restreint, plus l'algorithme est rapide, mais, en fonction de ce que nous venons de dire, plus il est limité.

### 3.2 ALGORITHME

Soit  $b_i$  la  $i^{\text{ème}}$  base avec  $i=1, \dots, n$  tq  $b_{i-1} < b_i$  avec  $1 < i \leq n$ .

Soit  $max$  le nombre maximal d'intervalles,  $max \in \mathbb{N}$ ,  $max > 0$ .

Soit  $x$  la valeur à représenter où  $x \leq b_n * max$

1° Calculer le nombre d'intervalles  $n_i$  de base  $b_i$  ( $i=1, \dots, n$ ),  $n_i \in \mathbb{N}$

- $n_i \leftarrow x / b_i$
- si  $n_i = \text{Ent}(n_i)$  alors  $n_i \leftarrow \text{Ent}(n_i)$   
sinon  $n_i \leftarrow \text{Ent}(n_i) + 1$

2° Choisir la meilleure base :  $b_j$  tq  $1 \leq j \leq n$ ,  $n_j \leq max$ ,  $\forall i$   $1 \leq i < j$   $n_i > max$

### 3.3 PROTOTYPE

Pour nous rendre compte de l'exactitude ergonomique de l'algorithme, nous l'avons implémenté en HyperTalk : la pile *Histogrammes* permet à l'utilisateur de spécifier la hauteur de trois bâtonnets, et de modifier l'échelle de l'axe des ordonnées grâce à une barre de défilement vertical.

Cette pile n'est composée que d'une carte.

Le fond de la carte comprend tous les dessins "statiques" : la barre de défilement (à l'exception de l'ascenseur), les flèches de défilement, les cadres, les axes et les graduations de l'axe des abscisses.

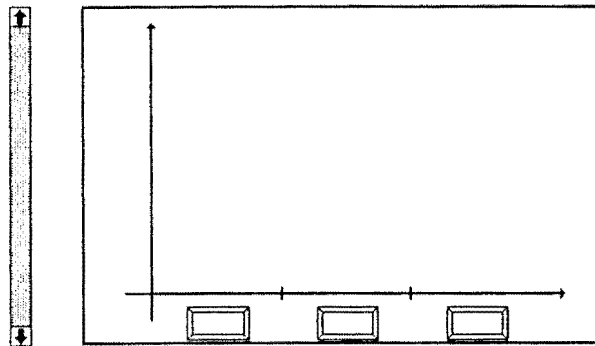


Illustration 10 : Fond de la carte

La carte proprement dite comprend le dessin de l'ascenseur, les boutons et champs de gestion de la barre de défilement<sup>1</sup>, les champs de carte ("1", "2", "3") contenant la valeur des bâtonnets, les dessins des bâtonnets, les graduations et les valeurs de l'axe des ordonnées, le dessin de la petite figure, le bouton *Prêt* et le bouton *Base*.

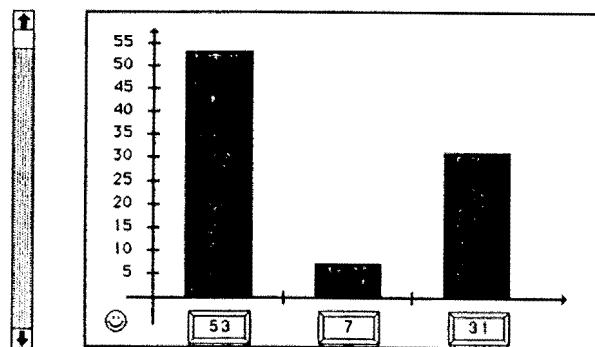


Illustration 11 : Carte

<sup>1</sup> Pour plus de renseignements concernant la barre de défilement, se reporter à l'annexe suivante. La barre de défilement de la pile *Histogrammes* a été créée grâce au bouton *Créer Barre Déf. Vert.* Ses scripts ont été ensuite modifiés pour s'adapter à l'application.

Le bouton *Base* sert à revenir à la carte de base.

Le dessin de la petite figure et le bouton opaque *Prêt* servent à signaler à l'utilisateur qu'il peut continuer.

Les boutons de gestion de la barre de défilement font appel à la procédure *Histogramme* définie au niveau de la carte. C'est cette procédure qui renferme l'algorithme expliqué ci-dessus. Elle prend également en charge le calcul du "zooming" induit par la barre de défilement (c'est-à-dire le calcul de la valeur sur laquelle va agir l'algorithme expliqué précédemment), et le dessin des trois bâtonnets.

Nous reprenons ci-dessous les différents scripts, ainsi que quelques exemples.

### 3.3.1 Script du bouton *Flèche Haut*

```
on mouseUp
  set visible of card button "Prêt" to true

  put line 1 of card field "Coordonnées" into PosHS
  put line 2 of card field "Coordonnées" into PosVS
  put line 3 of card field "Coordonnées" into PosVI
  put line 4 of card field "Coordonnées" into PosAsc

  if PosAsc > PosVS +15 then
    set lockscreen to true

    choose select tool
    drag from PosHS,PosAsc to PosHS+16,PosAsc+15
    doMenu "Cut Picture"

    subtract 1 from PosAsc
    put PosAsc into line 4 of card field Coordonnées

    choose rect tool
    drag from PosHS,PosAsc to PosHS+16,PosAsc+15
    choose bucket tool
    set pattern to 1
    click at PosHS+1,PosAsc+1

    choose browse tool

    Histogramme PosVS,PosVI,PosAsc

    set lockScreen to false
  end if

  set visible of card button "Prêt" to false
end mouseUp
```

### 3.3.2 Script du bouton *Barre verticale*

```
on mouseDown
  global TempsClick

  put the ticks into TempsClick
```



## Algorithme de gestion d'histogrammes

```
-- TempsClick : détermine le moment où l'utilisateur a enfoncé le bouton de la souris
end mouseDown
on mouseStillDown
  set visible of card button "Prêt" to true
  -- Cacher le dessin indiquant que l'utilisateur peut continuer

  -- 1° Gestion de la barre de défilement

  global TempsClick

  put the ticks into TempsEcoulé
  -- TempsEcoulé contient le temps présent

  put second item of the ClickLoc into PosClick
  -- PosClick contient la coordonnée verticale de la position courante de la souris

  subtract TempsClick from TempsEcoulé
  -- TempsEcoulé contient maintenant le temps écoulé entre le moment où l'utilisateur a enfoncé
  -- le bouton de la souris, et le moment présent

  put line 1 of card field "Coordonnées" into PosHS
  -- PosHS contient la coordonnée horizontale du coin supérieur gauche
  -- de la case de la flèche de défilement vers le haut

  put line 2 of card field "Coordonnées" into PosVS
  -- PosVS contient la coordonnée verticale du coin supérieur gauche
  -- de la case de la flèche de défilement vers le haut

  put line 3 of card field "Coordonnées" into PosVI
  -- PosVI contient la coordonnée verticale du coin inférieur droit
  -- de la case de la flèche de défilement vers le bas

  put line 4 of card field "Coordonnées" into PosAsc
  -- PosAsc contient la coordonnée verticale du coin supérieur gauche de l'ascenseur

  set lockscreen to true

  choose select tool -- Effacer l'ascenseur
  drag from PosHS,PosAsc to PosHS+16,PosAsc+15 --
  doMenu "Cut Picture" --

  if TempsEcoulé < 4 then -- cas où l'utilisateur a cliqué
    if PosClick < PosAsc then -- au dessus de l'ascenseur

      subtract 15 from PosAsc
    else
      if PosClick > PosAsc+15 then -- en dessous de l'ascenseur
        add 15 to PosAsc
      end if
    end if
  else -- cas où l'utilisateur a "draggé" l'ascenseur
    put second item of the mouseLoc into PosAsc
  end if
  -- PosAsc contient la nouvelle position de l'ascenseur

  -- Traitement des cas limites
  if PosAsc < PosVS+15 then
    put PosVS+15 into PosAsc
```

```
else
  if PosAsc > PosVI-30 then
    put PosVI-30 into PosAsc
  end if
end if

put PosAsc into line 4 of card field Coordonnées

choose rect tool -- Dessiner l'ascenseur
drag from PosHS,PosAsc to PosHS+16,PosAsc+15 -- à sa nouvelle
choose bucket tool -- position
set pattern to 1 --
click at PosHS+1,PosAsc+1 --

choose browse tool

-- 2° Gestion de l'histogramme

Histogramme PosVS,PosVI,PosAsc

set lockScreen to false

set visible of card button "Prêt" to false
  -- Refaire apparaître le dessin indiquant que l'utilisateur
  -- peut continuer
end mouseStillDown
```

### 3.3.3 Script du bouton *Flèche Bas*

```
on mouseUp
  set visible of card button "Prêt" to true

  put line 1 of card field "Coordonnées" into PosHS
  put line 2 of card field "Coordonnées" into PosVS
  put line 3 of card field "Coordonnées" into PosVI
  put line 4 of card field "Coordonnées" into PosAsc

  if PosAsc < PosVI-30 then
    set lockscreen to true

    choose select tool
    drag from PosHS,PosAsc to PosHS+16,PosAsc+15
    doMenu "Cut Picture"

    add 1 to PosAsc
    put PosAsc into line 4 of card field Coordonnées

    choose rect tool
    drag from PosHS,PosAsc to PosHS+16,PosAsc+15
    choose bucket tool
    set pattern to 1
    click at PosHS+1,PosAsc+1

    choose browse tool
```

Histogramme PosVS,PosVI,PosAsc

```
set lockScreen to false
end if
```

```
set visible of card button "Prêt" to false
end mouseUp
```

### 3.3.4 Procédure *Histogramme*

on Histogramme PosVS,PosVI,PosAsc

```
-- Procédure de gestion des bâtonnets
-- PosVS contient la coordonnée verticale du coin supérieur gauche
-- de la case de la flèche de défilement vers le haut
-- PosVI contient la coordonnée verticale du coin inférieur droit
-- de la case de la flèche de défilement vers le bas
-- PosAsc contient la coordonnée verticale du coin supérieur gauche de l'ascenseur
```

```
put line 1 of card field "1" into Val1 -- Lire la valeur
put line 1 of card field "2" into Val2 -- des bâtonnets
put line 1 of card field "3" into Val3 --
```

```
put 15 into NbreMaxInterv
-- NbreMaxInterv : paramètre fixant le nombre maximum d'intervalles
-- (i.e. de graduations) sur l'axe des ordonnées
```

```
put Val1 into ValMax -- Déterminer la valeur maximale des bâtonnets
if Val2 > ValMax then
    put Val2 into ValMax
end if
if Val3 > ValMax then
    put Val3 into ValMax
end if
```

-- 1° Déterminer le "zooming" induit par la barre de défilement

```
put PosVI into HauteurBarre -- Calculer la hauteur, en pixels,
subtract PosVS from HauteurBarre -- de la barre de défilement
subtract 45 from HauteurBarre --
```

```
put ValMax into ValUnit -- Calculer la valeur (au sens des
divide ValUnit by HauteurBarre -- des bâtonnets) d'un pixel
```

```
put PosVI into Hauteur -- Calculer la hauteur de
subtract 30 from Hauteur -- l'ascenseur (en pixels)
subtract PosAsc from Hauteur --
```

```
Put ValUnit into Valeur -- Calculer la valeur (au sens des bâtonnets) déterminée
multiply Valeur by Hauteur -- par la position de l'ascenseur
```

-- 2° Déterminer la meilleure base, c'est-à-dire celle qui permet d'afficher le plus grand

```
-- nombre d'intervalles (<= NbreMaxInterv)
```

```
put 0.5 into B1 -- Initialisation de la première base
```

```
put 1 into B2      -- Initialisation de la deuxième base
put 2 into B3      -- etc
put 5 into B4      --
put 10 into B5     --
put 20 into B6     --
put 50 into B7     --
put 100 into B8    --
put 200 into B9    --
put 500 into B10   --

put Valeur into Div1
put Valeur into Div2
put Valeur into Div3
put Valeur into Div4
put Valeur into Div5
put Valeur into Div6
put Valeur into Div7
put Valeur into Div8
put Valeur into Div9
put Valeur into Div10

divide Div1 by B1
divide Div2 by B2
divide Div3 by B3
divide Div4 by B4
divide Div5 by B5
divide Div6 by B6
divide Div7 by B7
divide Div8 by B8
divide Div9 by B9
divide Div10 by B10

put trunc(Div1) into DivEnt1
put trunc(Div2) into DivEnt2
put trunc(Div3) into DivEnt3
put trunc(Div4) into DivEnt4
put trunc(Div5) into DivEnt5
put trunc(Div6) into DivEnt6
put trunc(Div7) into DivEnt7
put trunc(Div8) into DivEnt8
put trunc(Div9) into DivEnt9
put trunc(Div10) into DivEnt10

if (Div1 <> DivEnt1) then add 1 to DivEnt1
    -- Déterminer le nombre d'intervalles nécessaires pour atteindre
    -- le multiple de la base B1 supérieur à Valeur
if (Div2 <> DivEnt2) then add 1 to DivEnt2
if (Div3 <> DivEnt3) then add 1 to DivEnt3
if (Div4 <> DivEnt4) then add 1 to DivEnt4
if (Div5 <> DivEnt5) then add 1 to DivEnt5
if (Div6 <> DivEnt6) then add 1 to DivEnt6
if (Div7 <> DivEnt7) then add 1 to DivEnt7
if (Div8 <> DivEnt8) then add 1 to DivEnt8
if (Div9 <> DivEnt9) then add 1 to DivEnt9
if (Div10 <> DivEnt10) then add 1 to DivEnt10
```

```
if (Valeur >= B10) then
  if (DivEnt10 <= NbreMaxInterv) then
    put DivEnt10 into NbreInterv
    put B10 into Base
  end if
else
  put 1 into NbreInterv
  put B10 into Base
end if
```

```
if (Valeur >= B9) then
  if (DivEnt9 <= NbreMaxInterv) then
    put DivEnt9 into NbreInterv
    put B9 into Base
  end if
else
  put 1 into NbreInterv
  put B9 into Base
end if
```

```
if (Valeur >= B8) then
  if (DivEnt8 <= NbreMaxInterv) then
    put DivEnt8 into NbreInterv
    put B8 into Base
  end if
else
  put 1 into NbreInterv
  put B8 into Base
end if
```

```
if (Valeur >= B7) then
  if (DivEnt7 <= NbreMaxInterv) then
    put DivEnt7 into NbreInterv
    put B7 into Base
  end if
else
  put 1 into NbreInterv
  put B7 into Base
end if
```

```
if (Valeur >= B6) then
  if (DivEnt6 <= NbreMaxInterv) then
    put DivEnt6 into NbreInterv
    put B6 into Base
  end if
else
  put 1 into NbreInterv
  put B6 into Base
end if
```

```
if (Valeur >= B5) then
  if (DivEnt5 <= NbreMaxInterv) then
    put DivEnt5 into NbreInterv
    put B5 into Base
  end if
else
  put 1 into NbreInterv
```

```

    put B5 into Base
end if

if (Valeur >= B4) then
    if (DivEnt4 <= NbreMaxInterv) then
        put DivEnt4 into NbreInterv
        put B4 into Base
    end if
else
    put 1 into NbreInterv
    put B4 into Base
end if

if (Valeur >= B3) then
    if (DivEnt3 <= NbreMaxInterv) then
        put DivEnt3 into NbreInterv
        put B3 into Base
    end if
else
    put 1 into NbreInterv
    put B3 into Base
end if

if (Valeur >= B2) then
    if (DivEnt2 <= NbreMaxInterv) then
        put DivEnt2 into NbreInterv
        put B2 into Base
    end if
else
    put 1 into NbreInterv
    put B2 into Base
end if

if (Valeur >= B1) then
    if (DivEnt1 <= NbreMaxInterv) then
        put DivEnt1 into NbreInterv
        put B1 into Base
    end if
else
    put 1 into NbreInterv
    put B1 into Base
end if
    -- NbreInterv contient le nombre d'intervalles de base Base nécessaires pour atteindre Valeur

-- 3° Redessiner les bâtonnets suivant la nouvelle échelle

choose select tool          -- Effacer les bâtonnets
drag from 110,30 to 470,250 --
doMenu "Cut Picture"       --

put 200 into HauteurBase    -- La hauteur maximale d'un bâtonnet est fixée à 200
divide HauteurBase by NbreInterv -- HauteurBase contient la hauteur (en pixel) d'un
intervalle
repeat with i=1 to NbreInterv
    choose line tool          -- Dessiner les graduations
    set lineSize to 2        --

```

## Algorithme de gestion d'histogrammes

```
put HauteurBase into HauteurTrait --
multiply HauteurTrait by i --
put 250 into PosTrait -- Postrait est la position
subtract HauteurTrait from PosTrait -- verticale du trait n° i
put trunc (PosTrait) into PosTrait --
drag from 146,PosTrait to 154,PosTrait --

choose text tool -- Ecrire les valeurs des graduations alignées
put Base into Graduation -- à droite
multiply Graduation by i --
if Graduation < 10 then --
  click at 126,PosTrait+3 --
else --
  if Graduation < 100 then --
    click at 118,PosTrait+3 --
  else --
    click at 110,PosTrait+3 --
  end if --
end if --
type Graduation --
end repeat

put Base into ValSup -- Déterminer le nombre de pixels correspondant
Multiply ValSup by NbreInterv -- à une unité (au sens des bâtonnets)
Put 200 into HauteurUnit --
Divide HauteurUnit by ValSup --

put Val1 into Hauteur1 -- Déterminer la position verticale
put Val2 into Hauteur2 -- du sommet de chaque bâtonnet
put Val3 into Hauteur3 --
multiply Hauteur1 by HauteurUnit --
multiply Hauteur2 by HauteurUnit --
multiply Hauteur3 by HauteurUnit --
put 250 into Pos1 --
put 250 into Pos2 --
put 250 into Pos3 --
subtract Hauteur1 from Pos1 --
subtract Hauteur2 from Pos2 --
subtract Hauteur3 from Pos3 --
put trunc (Pos1) into Pos1 --
put trunc (Pos2) into Pos2 --
put trunc (Pos3) into Pos3 --

choose rect tool -- Dessiner les trois bâtonnets
if Pos1 < PosTrait then -- PosTrait est la position
  drag from 175,PosTrait to 225,250 -- verticale de la dernière
  choose text tool -- graduation.
  click at 195,40 -- Si la bâtonnet est trop grand,
  type Val1 -- on le tronque, et on inscrit
  choose rect tool -- sa valeur au sommet.
else
  drag from 175,Pos1 to 225,250
end if
if Pos2 < PosTrait then
  drag from 275,PosTrait to 325,250
  choose text tool
  click at 295,40
```

## Algorithme de gestion d'histogrammes

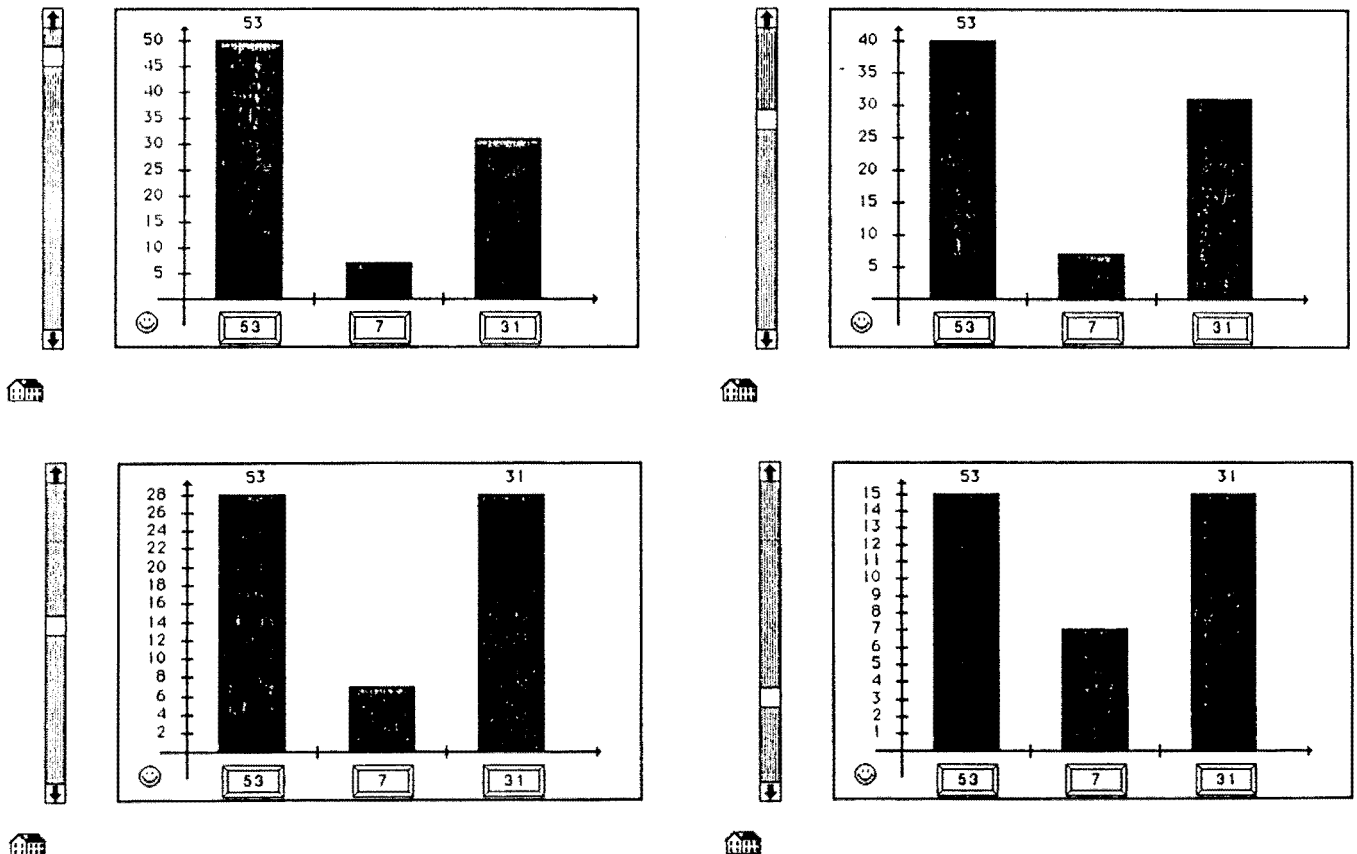
```

type Val2
choose rect tool
else
  drag from 275,Pos2 to 325,250
end if
if Pos3 < PosTrait then
  drag from 375,PosTrait to 425,250
  choose text tool
  click at 395,40
  type Val3
  choose rect tool
else
  drag from 375,Pos3 to 425,250
end if

choose bucket tool          -- Mise en couleur des trois bâtonnets.
set pattern to 22           --
click at 200,248           --
click at 300,248           --
click at 400,248           --

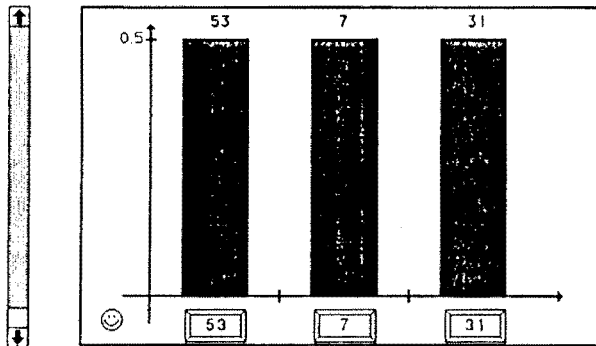
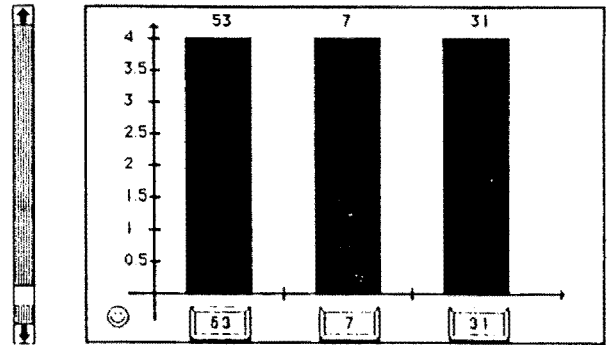
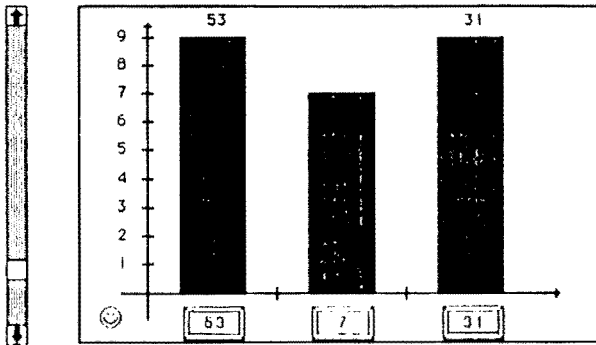
set lineSize to 1          -- Réinitialisation des paramètres d'HyperCard
choose browse tool         --
end Histogramme
  
```

### 3.3.5 Exemples





Algorithme de gestion d'histogrammes



## 4

---

# Créer une barre de défilement en HyperCard

### 4.1 EXPOSE DU PROBLEME

Lorsque nous avons voulu construire la maquette de MEANDRES avec HyperCard (version 1.2.5), nous avons regretté l'absence d'objets interactifs prédéfinis, tels qu'une barre de défilement. Tout ce que nous pouvons faire, c'est *imiter* ces objets interactifs : dessiner ces objets et les munir d'un certain nombre de boutons qui permettent d'aller à des cartes illustrant des états différents de ces objets.

Face à cette lacune, nous avons voulu établir un mini atelier logiciel qui offrirait la possibilité de créer de véritables objets interactifs. Nous vous proposons ci-dessous un algorithme qui permet de créer une véritable barre de défilement vertical.

Nous avons rapidement abandonné l'idée de rédiger les algorithmes des autres objets interactifs. Deux raisons expliquent cet arrêt :

- Tout d'abord, bien que la barre de défilement obtenue se comporte comme une véritable barre de défilement, le mouvement de l'ascenseur est beaucoup trop lent pour qu'il soit tolérable. Cela est dû au nombre relativement élevé d'instructions de l'algorithme de gestion de la barre, et au fait que le langage HyperTalk est interprété et non compilé.
- Ensuite, cette solution n'est pas rentable : il nous a fallu autant de temps pour réaliser cette seule barre de défilement que pour élaborer la maquette de MEANDRES en simulant cet objet interactif !

## 4.2 ALGORITHME

Pour créer une barre de défilement vertical, il suffit de cliquer sur le bouton *Créer Barre Déf. Vert.* Le script de ce dernier demande à l'utilisateur différents renseignements :

- la position horizontale du coin supérieur gauche de la barre (plus précisément, il s'agit de la position horizontale du coin supérieur gauche de la case contenant la flèche vers le haut);
- la position verticale du coin supérieur gauche de la barre;
- la position verticale du coin inférieur droit de la barre (plus précisément, il s'agit de la position verticale du coin inférieur droit de la case contenant la flèche vers le bas).

Ces trois renseignements suffisent à déterminer les dimensions de la barre compte tenu que sa largeur est fixée à 16 pixels et que la hauteur des cases des flèches et celle de l'ascenseur sont fixées à 15 pixels.

Après vérification de l'exactitude des renseignements donnés, le script crée un champ de carte appelé *Coordonnées* qui va servir à mémoriser les trois renseignements spécifiés ci-dessus, ainsi que la position courante de l'ascenseur. Ce champ invisible est superposé à la barre. Il est nécessaire pour réinitialiser les paramètres du script des boutons de la barre de défilement lorsque l'utilisateur a fermé la pile contenant la barre de défilement.

Ensuite, le script dessine la barre de défilement : les flèches de défilement et la barre proprement dite sur le fond, l'ascenseur sur la carte. On pourrait objecter qu'un tel choix est nuisible dans la mesure où toutes les cartes partageant le fond de la carte sur laquelle l'utilisateur crée la barre de défilement seront munies du dessin de la barre de défilement. Notre algorithme n'est effectivement pas valable dans ce cas. Nous avons sacrifié la généralité au profit de la performance. En effet, puisque seul l'ascenseur doit être effacé et redessiné pour refléter le dynamisme, le fait de mettre la barre de défilement proprement dite au niveau du fond permet de ne pas la redessiner à chaque mouvement de l'ascenseur.

L'étape suivante est la plus intéressante. Il s'agit de la création des boutons qui gèrent le dynamisme de la barre de défilement. Ces boutons sont au nombre de trois : *Barre verticale*, *Flèche Haut* et *Flèche Bas*. Ces boutons de carte sont transparents et sont exactement superposés à la partie de la barre de défilement qu'ils gèrent. Remarquez la manière de créer le script de ces boutons.

### 4.3 CODE

```
on mouseUp
  global PosAsc
  global PosHS
  global PosVS
  global PosVI

  put 1 into Vrai
  put 0 into Faux

  -- Demande des coordonnées de la barre de défilement
  Put Faux into Correct
  repeat until Correct is Vrai
    ask "Position horizontale du coin supérieur gauche"
    if it is empty then exit repeat -- L'utilisateur annule le dialogue
    else
      if (it ≥ 0) and (it ≤ 495) then
        put it into PosHS
        put Vrai into Correct
      else
        answer "La position horizontale doit être comprise" →
          &&"entre 0 et 495 !"
      end if
    end if
  end repeat

  if it is not empty then -- L'utilisateur n'a pas annulé le dialogue
    Put Faux into Correct
    repeat until Correct is Vrai
      ask "Position verticale du coin supérieur gauche"
      if it is empty then exit repeat -- L'utilisateur annule le dialogue
      else
        if (it ≥ 0) and (it ≤ 296) then
          put it into PosVS
          put Vrai into Correct
        else
          answer "La position verticale doit être comprise" →
            &&"entre 0 et 296 !"
        end if
      end if
    end repeat

  if it is not empty then -- L'utilisateur n'a pas annulé le dialogue
    Put Faux into Correct
    repeat until Correct is Vrai
      ask "Position verticale du coin inférieur gauche"
      if it is empty then exit repeat -- L'utilisateur annule le dialogue
      else
        if (it ≥ PosVS+45) and (it ≤ 341) then
          put it into PosVI
          put Vrai into Correct
        else
          answer "La position verticale doit être comprise" →
```

## Créer une barre de défilement en HyperCard

```
&&"entre"&&PosVS&&"et 341 !"  
end if  
end if  
end repeat
```

```
if it is not empty then  
-- L'utilisateur n'a pas annulé et a fourni des coordonnées correctes
```

```
set lockScreen to true
```

```
-- Sauver les coordonnées de manière permanente  
put "Coordonnées" into NomChamp
```

```
doMenu "New Field"  
set the name of card field (the number of card fields)→  
to NomChamp  
set the rect of card field NomChamp to PosHS,PosVS+15,→  
PosHS+16,PosVI-15  
set the style of card field NomChamp to transparent  
set the locktext of card field NomChamp to true  
set the visible of card field NomChamp to false
```

```
put PosHS into line 1 of card field NomChamp  
put PosVS into line 2 of card field NomChamp  
put PosVI into line 3 of card field NomChamp
```

```
-- Dessiner la barre de défilement vertical :  
-- Sur le fond : les flèches de défilement et la barre proprement dite  
-- Sur la carte : l'ascenseur
```

```
doMenu "Background"
```

```
-- 1° : dessiner la flèche vers le haut  
choose rect tool  
drag from PosHS,PosVS to PosHS+16,PosVS+15  
choose pencil tool  
put PosHS+8 into MilieuFlèche  
repeat with i = 1 to 6  
  repeat with j = MilieuFlèche+1-i to MilieuFlèche-1+i  
    click at j,PosVS+1+i  
  end repeat  
  repeat with j= PosHS+6 to PosHS+10  
    click at j,PosVS+7+i  
  end repeat  
end repeat
```

```
-- 2° : dessiner et colorier la barre même  
choose rect tool  
drag from PosHS,PosVS+15 to PosHS+16,PosVI-15  
choose bucket tool  
set pattern to 13  
click at PosHS+1,PosVS+16
```

## Créer une barre de défilement en HyperCard

```
-- 3° : dessiner la flèche vers le bas
choose rect tool
drag from PosHS,PosVI-15 to PosHS+16,PosVI
choose pencil tool
repeat with i = 1 to 6
  repeat with j = MilieuFlèche+1-i to MilieuFlèche-1+i
    click at j,PosVi-1-i
  end repeat
  repeat with j= PosHS+6 to PosHS+10
    click at j,PosVi-7-i
  end repeat
end repeat

-- 4° : dessiner l'ascenseur
doMenu "Background" -- Revenir au niveau de la carte
choose rect tool
drag from PosHS,PosVS+15 to PosHS+16,PosVS+30
choose bucket tool
set pattern to 1
click at PosHS+1,PosVS+16
put PosVS+15 into PosAsc

put PosAsc into line 4 of card field NomChamp

choose browse tool

-- Créer le bouton qui gère le dynamisme de la barre de défilement
put "Barre verticale" into NomBouton

put "on mouseDown" into line 1 of ScriptBouton
put "global TempsClick" into line 2 of ScriptBouton
put "put the ticks into TempsClick" into line 3 of ScriptBouton
put "end mouseDown" into line 4 of ScriptBouton
put "on mouseStillDown" into line 8 of ScriptBouton
put "global TempsClick" into line 9 of ScriptBouton
put "put the ticks into TempsEcoulé"↵
into line 11 of ScriptBouton
put "put second item of the ClickLoc into PosClick"↵
into line 12 of ScriptBouton
put "subtract TempsClick from TempsEcoulé"↵
into line 13 of ScriptBouton
put "put line 1 of card field"&&quote&NomChamp&quote&&↵
"into PosHS" into line 15 of ScriptBouton
put "put line 2 of card field"&&quote&NomChamp&quote&&↵
"into PosVS" into line 16 of ScriptBouton
put "put line 3 of card field"&&quote&NomChamp&quote&&↵
"into PosVI" into line 17 of ScriptBouton
put "put line 4 of card field"&&quote&NomChamp&quote&&↵
"into PosAsc" into line 18 of ScriptBouton
put "set lockscreen to true" into line 20 of ScriptBouton
put "choose select tool" into line 21 of ScriptBouton
put "drag from PosHS,PosAsc to PosHS+16,PosAsc+15"↵
into line 22 of ScriptBouton
put "doMenu"&&quote&"Cut Picture"&quote into line 23↵
of ScriptBouton
```

```
put "if TempsEcoulé < 4 then" into line 25 of ScriptBouton
put "if PosClick < PosAsc then" into line 26 of ScriptBouton
put "subtract 15 from PosAsc" into line 27 of ScriptBouton
put "else" into line 28 of ScriptBouton
put "if PosClick > PosAsc+15 then" into line 29 of ScriptBouton
put "add 15 to PosAsc" into line 30 of ScriptBouton
put "end if" into line 31 of ScriptBouton
put "end if" into line 32 of ScriptBouton
put "else" into line 33 of ScriptBouton
put "put second item of the mouseLoc into PosAsc"→
into line 34 of ScriptBouton
put "end if" into line 35 of ScriptBouton
put "if PosAsc < PosVS+15 then" into line 37 of ScriptBouton
put "put PosVS+15 into PosAsc" into line 38 of ScriptBouton
put "else" into line 39 of ScriptBouton
put "if PosAsc > PosVI-30 then" into line 40 of ScriptBouton
put "put PosVI-30 into PosAsc" into line 41 of ScriptBouton
put "end if" into line 42 of ScriptBouton
put "end if" into line 43 of ScriptBouton
put "put PosAsc into line 4 of card field"&&NomChamp→
into line 44 of ScriptBouton
put "choose rect tool" into line 45 of ScriptBouton
put "drag from PosHS,PosAsc to PosHS+16,PosAsc+15"→
into line 46 of ScriptBouton
put "choose bucket tool" into line 47 of ScriptBouton
put "set pattern to 1" into line 48 of ScriptBouton
put "click at PosHS+1,PosAsc+1" into line 49 of ScriptBouton
put "choose browse tool" into line 50 of ScriptBouton
put "set lockscreen to false" into line 51 of ScriptBouton
put "end mouseStillDown" into line 52 of ScriptBouton
```

```
doMenu "New Button"
set the name of card button (the number of card buttons)→
to NomBouton
set the rect of button NomBouton to PosHS,PosVS+15,→
PosHS+16,PosVI-15
set the style of button NomBouton to transparent
set the showName of button NomBouton to false
set the script of button NomBouton to ScriptBouton
```

```
put "Flèche Haut" into NomBouton
put empty into ScriptBouton
put "on mouseUp" into line 1 of ScriptBouton
put "put line 1 of card field"&&quote&NomChamp&quote&&→
"into PosHS" into line 2 of ScriptBouton
put "put line 2 of card field"&&quote&NomChamp&quote&&→
"into PosVS" into line 3 of ScriptBouton
put "put line 3 of card field"&&quote&NomChamp&quote&&→
"into PosVI" into line 4 of ScriptBouton
put "put line 4 of card field"&&quote&NomChamp&quote&&→
"into PosAsc" into line 5 of ScriptBouton
put "if PosAsc > PosVS +15 then" into line 7 of ScriptBouton
put "set lockscreen to true" into line 8 of ScriptBouton
put "choose select tool" into line 9 of ScriptBouton
put "drag from PosHS,PosAsc to PosHS+16,PosAsc+15"→
```

```
into line 10 of ScriptBouton
put "doMenu"&&quote&"Cut Picture"&quote into line 11→
of ScriptBouton
put "subtract 1 from PosAsc" into line 12 of ScriptBouton
put "put PosAsc into line 4 of card field"&&NomChamp→
into line 13 of ScriptBouton
put "choose rect tool" into line 14 of ScriptBouton
put "drag from PosHS,PosAsc to PosHS+16,PosAsc+15"→
into line 15 of ScriptBouton
put "choose bucket tool" into line 16 of ScriptBouton
put "set pattern to 1" into line 17 of ScriptBouton
put "click at PosHS+1,PosAsc+1" into line 18 of ScriptBouton
put "choose browse tool" into line 19 of ScriptBouton
put "set lockscreen to false" into line 20 of ScriptBouton
put "end if" into line 21 of ScriptBouton
put "end mouseUp" into line 22 of ScriptBouton
```

```
doMenu "New Button"
set the name of card button (the number of card buttons)→
to NomBouton
set the rect of button NomBouton to PosHS,PosVS,→
PosHS+16,PosVS+15
set the style of button NomBouton to transparent
set the showName of button NomBouton to false
set the script of button NomBouton to ScriptBouton
```

```
put "Flèche Bas" into NomBouton
put empty into ScriptBouton
put "on mouseUp" into line 1 of ScriptBouton
put "put line 1 of card field"&&quote&NomChamp&quote&&→
"into PosHS" into line 2 of ScriptBouton
put "put line 2 of card field"&&quote&NomChamp&quote&&→
"into PosVS" into line 3 of ScriptBouton
put "put line 3 of card field"&&quote&NomChamp&quote&&→
"into PosVI" into line 4 of ScriptBouton
put "put line 4 of card field"&&quote&NomChamp&quote&&→
"into PosAsc" into line 5 of ScriptBouton
put "if PosAsc < PosVI-30 then" into line 7 of ScriptBouton
put "set lockscreen to true" into line 8 of ScriptBouton
put "choose select tool" into line 9 of ScriptBouton
put "drag from PosHS,PosAsc to PosHS+16,PosAsc+15"→
into line 10 of ScriptBouton
put "doMenu"&&quote&"Cut Picture"&quote into line 11→
of ScriptBouton
put "add 1 to PosAsc" into line 12 of ScriptBouton
put "put PosAsc into line 4 of card field"&&NomChamp→
into line 13 of ScriptBouton
put "choose rect tool" into line 14 of ScriptBouton
put "drag from PosHS,PosAsc to PosHS+16,PosAsc+15"→
into line 15 of ScriptBouton
put "choose bucket tool" into line 16 of ScriptBouton
put "set pattern to 1" into line 17 of ScriptBouton
put "click at PosHS+1,PosAsc+1" into line 18 of ScriptBouton
put "choose browse tool" into line 19 of ScriptBouton
put "set lockscreen to false" into line 20 of ScriptBouton
put "end if" into line 21 of ScriptBouton
```



*Créer une barre de défilement en HyperCard*

put "end mouseUp" into line 22 of ScriptBouton

doMenu "New Button"

set the name of card button (the number of card buttons)→  
to NomBouton

set the rect of button NomBouton to PosHS,PosVI-15,→  
PosHS+16,PosVI

set the style of button NomBouton to transparent

set the showName of button NomBouton to false

set the script of button NomBouton to ScriptBouton

choose browse tool

set lockScreen to False

end if

end if

end if

end mouseUp

## 5

---

# Maquette : partie Windows

## 5.1 INTRODUCTION

Le lecteur sera peut-être étonné par l'absence de commentaires dans les sources ci-dessous. Il faut qu'il se rende compte que les algorithmes proposés, quoique verbeux, constituent réellement l'ABC de la programmation en Ms-Windows. Par conséquent, nous croyons que toute personne connaissant un minimum de ce langage de programmation complexe comprendra relativement aisément nos algorithmes. Par ailleurs, nous invitons les personnes n'ayant aucune notion de Ms-Windows à parcourir le document de Benoît Sacré, "*Programmation sous Windows*" [Sac88], ou, dans une autre mesure, l'excellent ouvrage de Charles Petzold, "*Programming Windows*" [Pet88], qui nous a servi de guide. Les différentes conventions d'écriture que nous avons suivies dans la rédaction et la structuration des algorithmes sont directement inspirées de cet ouvrage.

Pour faciliter la tâche du lecteur, nous avons commenté la procédure *PGLOBDlgProc* qui gère la boîte de dialogue affichée suite au choix de l'item **St. globales...** du menu **Processus**. Toutes les autres procédures sont bâties sur le même modèle, au sens propre du terme, car nous les avons élaborées par copier/coller à partir de *PGLOBDlgProc* et en remplaçant le nom de certaines variables.

A titre d'indication, il nous a fallu environ une vingtaine d'heures pour réaliser la procédure *PGLOBDlgProc* et ses deux sous-procédures *PGLTDlgProc*, *PGLDRDlgProc*. Ses trois procédures mises au point, il ne nous a fallu de nouveau qu'une vingtaine d'heures pour élaborer les ??? autres procédures.

Six fichiers se partagent les sources du programme :

meandres.c

Ce fichier contient le code du programme principal *WinMain* tout à fait classique (définition de la classe de la fenêtre de l'application MEANDRES et "routage" des messages), et le code de la procédure *WndProc* qui gère tous les messages destinés à l'application. *WndProc* fait appel aux procédures définies dans les fichiers *fichier.h*, *processu.h*, *ressour.h* et *synchr.h*.

fichier.h

Ce fichier contient le code de la procédure de gestion de la boîte de dialogue de l'item **Imprimer...** du menu **Fichier**.

processu.h

Ce fichier contient le code des procédures de gestion des boîtes de dialogue des items du menu **Processus**.

ressour.h

Ce fichier contient le code des procédures de gestion des boîtes de dialogue des items du menu **Ressources**.

synchr.h

Ce fichier contient le code des procédures de gestion des boîtes de dialogue des items du menu **Pt. de Synchr.**

meandres.h

Ce fichier contient la liste des constantes utilisées dans la définition des items des menus et dans celles des contrôles des différentes boîtes de dialogue.

Le bon fonctionnement du programme nécessite trois fichiers contenant les différentes ressources :

meandres.rc

Ce fichier contient la définition de toutes les ressources. Il décrit explicitement les menus. Il fait référence au fichier *meandres.ico* et au fichier *meandres.dlg*.

meandres.ico

Ce fichier contient la définition de l'icône de l'application. Rappelons que cette icône apparaît quand l'utilisateur réduit la fenêtre de l'application sous forme icônique. Ce fichier a été produit grâce à l'utilitaire *iconedit.exe* qui permet de créer interactivement l'icône de l'application.



Illustration 12 : Icône de l'application MEANDRES

meandres.dlg

Ce fichier contient la définition des boîtes de dialogue. Il a été produit grâce à l'utilitaire *dialog.exe* qui permet de construire interactivement des boîtes de dialogues.

Nous vous présentons maintenant le contenu des ces différents fichiers (à l'exception du fichier *meandres.ico*) précédé par une petite table des matières. Ne soyez pas étonnés par certains caractères incongrus : ils sont la traduction Macintosh de caractères PC eux-mêmes abscons, mais tout à fait compréhensibles par Ms-Windows qui les restitue correctement à l'exécution. Pour votre facilité, voici une table de correspondance :

Mac	PC	Exécution	Code
‡	α	à	224
Á	γ	ç	321
Ë	Φ	è	232
È	Θ	é	233
Í	Ω	ê	235

## 5.2 MEANDRES.C

### Liste des procédures

WinMain .....	37
WndProc .....	38

### Contenu du fichier

```
#include <windows.h>
#include <stdio.h>
#include "meandres.h"
#include "fichier.h"
#include "processu.h"
#include "ressour.h"
#include "synchr.h"

long FAR PASCAL WndProc (HWND, unsigned, WORD, LONG);

char szAppName[] = "Meandres";

int PASCAL WinMain (hInstance, hPrevInstance, lpszCmdLine,
                   nCmdShow)
{
    HANDLE      hInstance, hPrevInstance;
    LPSTR       lpszCmdLine;
    int         nCmdShow;

    HANDLE      hAccel;
    HWND        hWnd;
    MSG         msg;
    WNDCLASS    wndclass;

    if (!hPrevInstance)
    {
        wndclass.style           = CS_HREDRAW | CS_VREDRAW;
        wndclass.lpfnWndProc     = WndProc;
        wndclass.cbClsExtra      = 0;
        wndclass.cbWndExtra      = 0;
    }
}
```

```

    wndclass.hInstance      = hInstance;
    wndclass.hIcon          = LoadIcon (hInstance, szAppName);
    wndclass.hCursor        = LoadCursor(NULL, IDC_ARROW);
    wndclass.hbrBackground = GetStockObject(WHITE_BRUSH);
    wndclass.lpszMenuName   = szAppName;
    wndclass.lpszClassName  = szAppName;

    if (!RegisterClass (&wndclass))
        return FALSE;
}

hWnd = CreateWindow (szAppName, "MEANDRES - (sans titre)",
                    WS_OVERLAPPEDWINDOW | WS_HSCROLL |
                    WS_VSCROLL,
                    CW_USEDEFAULT, 0,
                    CW_USEDEFAULT, 0,
                    NULL, NULL, hInstance, NULL);

ShowWindow (hWnd, nCmdShow);
UpdateWindow (hWnd);

hAccel = LoadAccelerators (hInstance, szAppName);

while (GetMessage(&msg, NULL, 0, 0))
{
    if (!TranslateAccelerator (hWnd, hAccel, &msg))
    {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
}
return msg.wParam;
}

```

**long FAR PASCAL WndProc (hWnd, iMessage, wParam, lParam)**

```

HWND      hWnd;
unsigned  iMessage;
WORD      wParam;
LONG      lParam;
{
    static FARPROC lpfnIMPRDlgProc;
    static FARPROC lpfnPGLOBDlgProc;
    static FARPROC lpfnPPERDlgProc;
    static FARPROC lpfnPCHRDlgProc;
    static FARPROC lpfnPCOMDlgProc;
    static FARPROC lpfnRGLOBDlgProc;
    static FARPROC lpfnRPERDlgProc;
    static FARPROC lpfnRCHRDlgProc;
    static FARPROC lpfnRCOMDlgProc;
    static FARPROC lpfnPCALDlgProc;
    static FARPROC lpfnSGLOBDlgProc;
    static FARPROC lpfnSCOMDlgProc;
    static HWND  hInstance;
    HMENU       hMenu;

```

```

switch (iMessage)
{
    case WM_CREATE :
        hInstance = ((LPCREATESTRUCT) lParam) -> hInstance;
        hInst = hInstance;

        lpfnIMPRDlgProc = MakeProcInstance (IMPRDlgProc,
                                           hInstance);
        lpfnPGLOBDlgProc = MakeProcInstance (PGLOBDlgProc,
                                           hInstance);
        lpfnPPERDlgProc = MakeProcInstance (PPERDlgProc,
                                           hInstance);
        lpfnPCHRDlgProc = MakeProcInstance (PCHRDlgProc,
                                           hInstance);
        lpfnPCOMDlgProc = MakeProcInstance (PCOMDlgProc,
                                           hInstance);
        lpfnPCALDlgProc = MakeProcInstance (PCALDlgProc,
                                           hInstance);
        lpfnRGLOBDlgProc = MakeProcInstance (RGLOBDlgProc,
                                           hInstance);
        lpfnRPERDlgProc = MakeProcInstance (RPERDlgProc,
                                           hInstance);
        lpfnRCHRDlgProc = MakeProcInstance (RCHRDlgProc,
                                           hInstance);
        lpfnRCOMDlgProc = MakeProcInstance (RCOMDlgProc,
                                           hInstance);
        lpfnSGLOBDlgProc = MakeProcInstance (SGLOBDlgProc,
                                           hInstance);
        lpfnSCOMDlgProc = MakeProcInstance (SCOMDlgProc,
                                           hInstance);

        break;

    case WM_COMMAND :
        switch (wParam)
        {
            case IDM_NOUVEAU :
            case IDM_SELECT :
            case IDM_ENREGSOUS :
            case IDM_APERCU :
            case IDM_MISEPG :

            case IDM_ANNULER :
            case IDM_COUPER :
            case IDM_COPIER :
            case IDM_COLLER :
            case IDM_EFFACER :
            case IDM_TSELECT :

            case IDM_PRES :

            case IDM_RPROC :

            case IDM_AIDE :
            case IDM_INFO :
            case IDM_REORG :
            case IDM_MASQUER :
            case IDM_AFFICHER :
        }
    }
}

```

```
MessageBox (hWnd, "Cette commande n'est pas encore  
implémentée.", "MEANDRES",  
MB_ICONASTERISK | MB_OK);  
break;  
  
case IDM_OUVRIR :  
    MessageBox (hWnd,  
        "Impossible de trouver le fichier  
        nom_de_fichier dans le répertoire  
        nom_de_répertoire.",  
        "MEANDRES", MB_ICONASTERISK |  
        MB_OK);  
    MessageBox (hWnd,  
        "Impossible de trouver le fichier  
        nom_de_fichier dans le répertoire ...  
        \nVous ne pourrez pas utiliser le schéma  
        de la dynamique.",  
        "MEANDRES", MB_ICONQUESTION |  
        MB_OKCANCEL);  
    break;  
  
case IDM_ENREG :  
    MessageBox (hWnd, "Impossible d'enregistrer le rapport  
        nom_de_rapport: place insuffisante.",  
        "MEANDRES", MB_ICONASTERISK |  
        MB_OK);  
    break;  
  
case IDM_FERMER :  
    MessageBox (hWnd, "nom_du_rapport a changé.  
        Enregistrer les modifications ?",  
        "MEANDRES", MB_ICONQUESTION |  
        MB_YESNOCANCEL);  
    break;  
  
case IDM_IMPRIMER :  
    DialogBox (hInstance, "IMPR", hWnd,  
        lpfnIMPRDlgProc);  
    break;  
  
case IDM_PGLOB :  
    DialogBox (hInstance, "PGLOB", hWnd,  
        lpfnPGLOBDlgProc);  
    break;  
  
case IDM_PPER :  
    DialogBox (hInstance, "PPER", hWnd,  
        lpfnPPERDlgProc);  
    break;  
  
case IDM_PCHR :  
    DialogBox (hInstance, "PCHR", hWnd,  
        lpfnPCHRDlgProc);  
    break;
```

```
case IDM_PCOM :
    DialogBox (hInstance, "PCOM", hWnd,
              lpfnPCOMDlgProc);
    break;

case IDM_PCAL :
    DialogBox (hInstance, "PCAL", hWnd,
              lpfnPCALDlgProc);
    break;

case IDM_PDEF :
case IDM_RDEF :
case IDM_SDEF :
    MessageBox (hWnd,
                "Voulez-vous vraiment redÉfinir le standard
                de cette fiche ?", "MEANDRES",
                MB_ICONQUESTION |
                MB_OKCANCEL);
    break;

case IDM_RGLOB :
    DialogBox (hInstance, "RGLOB", hWnd,
              lpfnRGLOBDlgProc);
    break;

case IDM_RPER :
    DialogBox (hInstance, "RPER", hWnd,
              lpfnRPERDlgProc);
    break;

case IDM_RCHR :
    DialogBox (hInstance, "RCHR", hWnd,
              lpfnRCHRDlgProc);
    break;

case IDM_RCOM :
    DialogBox (hInstance, "RCOM", hWnd,
              lpfnRCOMDlgProc);
    break;

case IDM_SGLOB :
    DialogBox (hInstance, "SGLOB", hWnd,
              lpfnSGLOBDlgProc);
    break;

case IDM_SCOM :
    DialogBox (hInstance, "SCOM", hWnd,
              lpfnSCOMDlgProc);
    break;

case IDM_QUITTER :
    MessageBox (hWnd, "nom_du_rapport a changÉ.
                    Enregistrer les modifications ?",
                "MEANDRES", MB_ICONQUESTION |
                MB_YESNOCANCEL);

    SendMessage (hWnd, WM_CLOSE, 0, 0L);
```



```
        break;
    default :
        break;
    }
    break;
case WM_DESTROY :
    PostQuitMessage (0);
    break;
default :
    return DefWindowProc (hWnd, iMessage, wParam, lParam);
}
return 0L;
}
```

## 5.3 FICHIER.H

### Liste des procédures

IMPRDlgProc ..... 43

### Contenu du fichier

#### **BOOL FAR PASCAL IMPRDlgProc (hDlg, iMessage, wParam, lParam)**

```

HWND    hDlg;
unsigned iMessage;
WORD    wParam;
LONG    lParam;
{
char szbuffer[1];

switch (iMessage)
{
case WM_INITDIALOG :
    SendMessage (GetDlgItem(hDlg, IMPR_T),
                BM_SETCHECK, 1, 0L);
    SendMessage (GetDlgItem(hDlg, IMPR_S),
                BM_SETCHECK, 0, 0L);
    sprintf (szbuffer, "1");
    SetWindowText (GetDlgItem(hDlg, IMPR_C),szbuffer);
    break;

case WM_COMMAND :
    switch (wParam)
    {
case IMPR_T :
        CheckRadioButton (hDlg, IMPR_T, IMPR_S, wParam);
        break;

case IMPR_S :
        CheckRadioButton (hDlg, IMPR_T, IMPR_S, wParam);
        break;

case IDOK :
        EndDialog (hDlg, TRUE);
        break;

case IDCANCEL :
        EndDialog (hDlg, FALSE);
        break;

default :
        return FALSE;
    }
    break;

default :
    return FALSE;
}
return TRUE;
}

```

## 5.4 PROCESSU.H

### Liste des procédures

PCALDIgProc .....	65
PCHDRDIgProc .....	57
PCHETDIgProc .....	59
PCHRDIGProc .....	60
PCOMDIgProc .....	64
PGLDRDIgProc .....	47
PGLETDIgProc .....	48
PGLOBDIgProc .....	50
PPERDIgProc .....	55

### Contenu du fichier

```

#define NPGLOBCB (sizeof PGLOBCB / sizeof PGLOBCB[0])
/* NPGLOBCB : nombre d'éléments du tableau PGLOBCB */
#define NPGLDRCB (sizeof PGLDRCB / sizeof PGLDRCB[0])
/* NPGLDRCB: nombre d'éléments du tableau PGLDRCB */
#define NPGETCB (sizeof PGETCB / sizeof PGETCB[0])
/* NPGETCB: nombre d'éléments du tableau PGETCB */

#define NPPERCB (sizeof PPERCB / sizeof PPERCB[0])
#define NPCHRCB (sizeof PCHRCB / sizeof PCHRCB[0])
#define NPCHDRCB (sizeof PCHDRCB / sizeof PCHDRCB[0])
#define NPCHETCB (sizeof PCHETCB / sizeof PCHETCB[0])
#define NPCOMCB (sizeof PCOMCB / sizeof PCOMCB[0])
#define NPCALCB (sizeof PCALCB / sizeof PCALCB[0])

HWND hInst;
struct InitCB
/* InitCB (Init. Check Box) : structure dont le premier membre contient l'identifiant d'une boîte à cocher
(ou d'un bouton radio), et dont le deuxième membre détermine l'état de la boîte (0 = non cochée; 1 =
cochée) */
{
    int nId;
    int Etat;
};
struct InitCB PGLOBCB[] =
/* PGLOBCB[] : tableau de structures; décrit l'état initial des boîtes à cocher (et des boutons radios) de la
boîte de dialogue gérée par la procédure PGLOBDIgProc */
{
    PGLOB_FSTA, 1,
    PGLOB_FPER, 0,
    PGLOB_DEC, 1,
    PGLOB_ACT, 1,
    PGLOB_INT, 0,
    PGLOB_REA, 0,
    PGLOB_TER, 1,
    PGLOB_VIE, 1,
    PGLOB_PRO, 1
},

```

```
PGLDRCB[] =
/* PGLDRCB[] : tableau de structures; décrit l'état initial des boîtes à cocher (et des boutons
radios) de la boîte de dialogue gérée par la procédure PGLDRDlgProc */
{
    PGLDR_CN, 0,
    PGLDR_CY, 0,
    PGLDR_CX, 0,
    PGLDR_CE, 0,
    PGLDR_TN, 0,
    PGLDR_TY, 0,
    PGLDR_TX, 0,
    PGLDR_TE, 0,
    PGLDR_IN, 0,
    PGLDR_IY, 0,
    PGLDR_IX, 0,
    PGLDR_IE, 0,
    PGLDR_PY, 0,
    PGLDR_RN, 0,
    PGLDR_RY, 0,
    PGLDR_RX, 0,
    PGLDR_RE, 0,
    PGLDR_SY, 0,
    PGLDR_VY, 0
},
PGLETCB[] =
/* PGLETCB[] : tableau de structures; décrit l'état initial des boîtes à cocher (et des boutons
radios) de la boîte de dialogue gérée par la procédure PGLETDlgProc */
{
    PGLET_TN, 0,
    PGLET_TY, 0,
    PGLET_TX, 0,
    PGLET_TE, 0,
    PGLET_IN, 0,
    PGLET_IY, 0,
    PGLET_IX, 0,
    PGLET_IE, 0,
    PGLET_CN, 0,
    PGLET_CY, 0,
    PGLET_CX, 0,
    PGLET_CE, 0
},
PPERCB[] =
{
    PPER_FSTA, 1,
    PPER_FPER, 0,
    PPER_DEC, 0,
    PPER_ACT, 0,
    PPER_INT, 0,
    PPER_REA, 0,
    PPER_TER, 0,
    PPER_DRT, 0,
    PPER_DRI, 0,
    PPER_DRR, 0,
    PPER_RPV, 1,
    PPER_RPS, 1
},
```

```

PCHRCB[] =
{
    PCHR_FSTA, 1,
    PCHR_FPER, 0,
    PCHR_DEC, 0,
    PCHR_ACT, 0,
    PCHR_INT, 0,
    PCHR_REA, 0,
    PCHR_TER, 0,
    PCHR_RPV, 1,
    PCHR_RPT, 1,
    PCHR_RPP, 1
},
PCHDRCB[] =
{
    PCHDR_T, 0,
    PCHDR_R, 0,
    PCHDR_I, 0
},
PCHETCB[] =
{
    PCHET_T, 0,
    PCHET_N, 0,
    PCHET_F, 0,
    PCHET_I, 0,
    PCHET_C, 0
},
PCOMCB[] =
{
    PCOM_TITRE, 1,
    PCOM_DATE, 1,
    PCOM_TYPE, 1,
    PCOM_PERIODE, 1,
    PCOM_NOM, 1,
    PCOM_CALENDRIER, 1
},
PCALCB[] =
{
    PCAL_P, 0,
    PCAL_R, 0,
};

```

```
struct InitCB PGLDRCB1[NPGLDRCB];
```

/\* PGLDRCB1[NPGLDRCB] : tableau auxiliaire; à l'origine copie conforme de PGLDRCB; reflète l'état courant validé des boîtes à cocher (des boutons radios) de la boîte de dialogue gérée par la procédure PGLDRDlgProc \*/

```
struct InitCB PGLETCB1[NPGLETCB];
```

/\* PGLDRCB1[NPGLDRCB] : tableau auxiliaire; à l'origine copie conforme de PGLETCB; reflète l'état courant validé des boîtes à cocher (des boutons radios) de la boîte de dialogue gérée par la procédure PGLETDlgProc \*/

```
struct InitCB PCHDRCB1[NPCHDRCB];
```

```
struct InitCB PCHETCB1[NPCHETCB];
```

**BOOL FAR PASCAL PGLDRDlgProc (hDlg, iMessage, wParam, lParam)**

/\* Procédure de gestion de la boîte de dialogue permettant de choisir les statistiques concernant les durées des processus \*/

```

    HWND    hDlg;
    unsigned iMessage;
    WORD    wParam;
    LONG    lParam;
    {
    short    n;
    static struct InitCB PGLDRCB2[NPGLDRCB];
    /* PGLDRCB2[NPGLDRCB] : tableau auxiliaire; à l'origine copie conforme de PGLDRCB1;
    reflète l'état courant non encore validé des boîtes à cocher de la boîte de dialogue gérée par la
    procédure PGLDRDlgProc */

```

```

switch (iMessage)
    {
    case WM_INITDIALOG :
        /* Traitement du message d'initialisation de la boîte de dialogue */
        for (n=0; n<NPGLDRCB; n++)
            /* Initialiser le tableau PGLDRCB2[] */
            {
                PGLDRCB2[n].nId = PGLDRCB1[n].nId;
                PGLDRCB2[n].Etat = PGLDRCB1[n].Etat;
            };

        for (n=0; n<NPGLDRCB; n++)
            /* Cocher les boîtes à cocher qui doivent l'être (c'est-à-dire celles dont l'état vaut
            1); décocher les autres */
            switch (PGLDRCB2[n].Etat)
                {
                case 0 :
                    CheckDlgButton (hDlg,PGLDRCB2[n].nId,0);
                    break;

                case 1 :
                    CheckDlgButton (hDlg,PGLDRCB2[n].nId,1);
                    break;

                default :
                    break;
                };
            break;

    case WM_COMMAND :
        /* Traitement des messages issus des boutons et boîtes à cocher de la boîte de dialogue */
        switch (wParam)
            {
            case PGLDR_CN :
            case PGLDR_CY :
            case PGLDR_CX :
            case PGLDR_CE :
            case PGLDR_TN :
            case PGLDR_TY :
            case PGLDR_TX :
            case PGLDR_TE :
            case PGLDR_IN :
            case PGLDR_IY :
            case PGLDR_IX :

```

```

case PGLDR_IE :
case PGLDR_PY :
case PGLDR_RN :
case PGLDR_RY :
case PGLDR_RX :
case PGLDR_RE :
case PGLDR_SY :
case PGLDR_VY :
/* Cas où le message provient de l'une des boîtes à cocher */
    CheckDlgButton (hDlg, wParam,
                    ! IsDlgButtonChecked (hDlg,wParam));
/* Cocher la boîte si elle est décochée; la décocher sinon */

    PGLDRCB2[wParam-PGLDR_CN].Etat =
    PGLDRCB2[wParam-PGLDR_CN].Etat == 0 ? 1 : 0;
/* Refléter cette action sur le tableau des états courants non encore
validés*/

    break;

case IDOK :
/* Cas où le message provient du bouton OK */
    for (n=0; n<NPGLDRCB; n++)
        PGLDRCB1[n].Etat = PGLDRCB2[n].Etat;
/* L'utilisateur a validé ses choix; il faut donc mettre à jour le tableau
des états courants validés */

    EndDialog (hDlg, TRUE);
    break;

case IDCANCEL :
/* Cas où le message provient du bouton Annuler; dans ce cas, il ne faut pas
mettre à jour le tableau des états courants validés */
    EndDialog (hDlg, FALSE);
    break;

default :
    return FALSE;
}
break;

default :
    return FALSE;
}
return TRUE;
}

```

### **BOOL FAR PASCAL PGLETDlgProc (hDlg, iMessage, wParam, lParam)**

/\* Procédure de gestion de la boîte de dialogue permettant de choisir les statistiques concernant les états des processus \*/

```

HWND    hDlg;
unsigned iMessage;
WORD    wParam;
LONG    lParam;
{
short   n;

```

```
static struct InitCB PGLTCB2[NPGLTCB];
/* PGLTCB2[NPGLDRCB] : tableau auxiliaire; à l'origine copie conforme de PGLTCB1;
reflète l'état courant non encore validé des boîtes à cocher de la boîte de dialogue gérée par la
procédure PGLTDLgProc */
```

```
switch (iMessage)
{
case WM_INITDIALOG :
/* Traitement du message d'initialisation de la boîte de dialogue */
for (n=0; n<NPGLTCB; n++)
/* Initialiser le tableau PGLTCB2[] */
{
PGLTCB2[n].nId = PGLTCB1[n].nId;
PGLTCB2[n].Etat = PGLTCB1[n].Etat;
};

for (n=0; n<NPGLTCB; n++)
/* Cocher les boîtes à cocher qui doivent l'être (c'est-à-dire celles dont l'état vaut
1); décocher les autres */
switch (PGLTCB2[n].Etat)
{
case 0 :
CheckDlgButton(hDlg,PGLTCB2[n].nId,0);
break;

case 1 :
CheckDlgButton(hDlg,PGLTCB2[n].nId,1);
break;

default :
break;
};
break;

case WM_COMMAND :
/* Traitement des messages issus des boutons et boîtes à cocher de la boîte de dialogue */
switch (wParam)
{
case PGLT_TN :
case PGLT_TY :
case PGLT_TX :
case PGLT_TE :
case PGLT_IN :
case PGLT_IY :
case PGLT_IX :
case PGLT_IE :
case PGLT_CN :
case PGLT_CY :
case PGLT_CX :
case PGLT_CE :
/* Cas où le message provient de l'une des boîtes à cocher */
CheckDlgButton (hDlg, wParam,
! IsDlgButtonChecked (hDlg,wParam));
/* Cocher la boîte si elle est décochée; la décocher sinon */

PGLTCB2[wParam-PGLT_TN].Etat =
PGLTCB2[wParam-PGLT_TN].Etat == 0 ? 1 : 0;
};
};
};
```



```

/* Réfléter cette action sur le tableau des états courants non encore
validés*/

break;

case IDOK :
/* Cas où le message provient du bouton OK */
for (n=0; n<NPGLTCB; n++)
    PGLTCB1[n].Etat = PGLTCB2[n].Etat;
/* L'utilisateur a validé ses choix; il faut donc mettre à jour le tableau
des états courants validés */

EndDialog (hDlg, TRUE);
break;

case IDCANCEL :
/* Cas où le message provient du bouton Annuler; dans ce cas, il ne faut pas
mettre à jour le tableau des états courants validés */
EndDialog (hDlg, FALSE);
break;

default :
return FALSE;
}
break;

default :
return FALSE;
}
return TRUE;
}

```

### **BOOL FAR PASCAL PGLOBDlgProc (hDlg, iMessage, wParam, lParam)**

*/\* Procédure de gestion de la boîte de dialogue affichée suite au choix de l'item St. globales... du menu Processus \*/*

```

HWND          hDlg;
unsigned      iMessage;
WORD          wParam;
LONG          lParam;
{
static BOOL    Standard;
/* Standard vaut 1 si la boîte de dialogue est standard, c'est-à-dire si l'état courant des boîtes à
cocher (et des boutons radios) correspond à celui mémorisé dans le tableau PGLOBCB[]; 0 sinon
*/
static BOOL    STPGLDR;
/* STPGLDR vaut 1 si la boîte de dialogue gérée par la procédure PGLDRDlgProc est standard,
c'est-à-dire si l'état courant des boîtes à cocher correspond à celui mémorisé dans le tableau
PGLDRCB[]; 0 sinon */
static BOOL    STPGLET;
/* STPGLDR vaut 1 si la boîte de dialogue gérée par la procédure PGLTDlgProc est standard,
c'est-à-dire si l'état courant des boîtes à cocher correspond à celui mémorisé dans le tableau
PGLTCB[]; 0 sinon */
short          n;
static struct InitCB PGLOBCB2[NPGLOBCB];
/* PGLOBCB2[NPGLDRCB] : tableau auxiliaire; à l'origine copie conforme de PGLOBCB;
reflète l'état courant non encore validé des boîtes à cocher (des boutons radios) de la boîte de
dialogue gérée par la procédure PGLOBDlgProc */

```

```

static FARPROC  lpfnPGLDRDlgProc;
static FARPROC  lpfnPGLETDlgProc;

switch (iMessage)
{
case WM_INITDIALOG :
/* Traitement du message d'initialisation de la boîte de dialogue */
  EnableWindow (GetDlgItem(hDlg,PGLOB_DUREE), FALSE);
  EnableWindow (GetDlgItem(hDlg,PGLOB_ETAT), FALSE);
  /* Rendre indisponibles les boutons Durées et Etats */

  Standard = TRUE;
  STPGLDR  = TRUE;
  STPGLET  = TRUE;

  for (n=0; n<NPGLOBCB; n++)
  /* Initialiser le tableau PGLOBCB2[] */
  {
    PGLOBCB2[n].nId = PGLOBCB[n].nId;
    PGLOBCB2[n].Etat = PGLOBCB[n].Etat;
  };

  for (n=0; n<NPGLDRCB; n++)
  /* Initialiser le tableau PGLDRCB1[] */
  {
    PGLDRCB1[n].nId = PGLDRCB[n].nId;
    PGLDRCB1[n].Etat = PGLDRCB[n].Etat;
  };

  for (n=0; n<NPGLET CB; n++)
  /* Initialiser le tableau PGLET CB1[] */
  {
    PGLET CB1[n].nId = PGLET CB[n].nId;
    PGLET CB1[n].Etat = PGLET CB[n].Etat;
  };

  for (n=0; n<2; n++)
    SendMessage (GetDlgItem(hDlg,PGLOBCB[n].nId),
                  BM_SETCHECK, PGLOBCB[n].Etat, 0L);
  /* Cocher les boutons radios qui doivent l'être (c'est-à-dire ceux dont l'état vaut 1);
  décocher les autres */

  for (n=2; n<NPGLOBCB; n++)
  /* Cocher les boîtes à cocher qui doivent l'être (c'est-à-dire celles dont l'état vaut
  1); décocher les autres */
    switch (PGLOBCB[n].Etat)
    {
    case 0 :
      CheckDlgButton(hDlg,PGLOBCB[n].nId,0);
      break;

    case 1 :
      CheckDlgButton(hDlg,PGLOBCB[n].nId,1);
      break;

    default :
      break;
    };
};

```

```

lpfnPGLDRDlgProc = MakeProcInstance (PGLDRDlgProc,
                                     hInst);
lpfnPGLETDlgProc = MakeProcInstance (PGLETDlgProc, hInst);

break;

case WM_COMMAND :
/* Traitement des messages issus des boutons et boîtes à cocher de la boîte de dialogue */
switch (wParam)
{
case PGLOB_FSTA :
/* Cas où le message provient du bouton Fiche standard; Dans ce cas, il faut
réinitialiser la boîte de dialogue */
SendMessage (hDlg, WM_INITDIALOG,
             wParam, lParam);
break;

case PGLOB_FPER :
/* Cas où le message provient du bouton Fiche personnalisée */
Standard = FALSE;
/* La boîte de dialogue n'est plus standard */
CheckRadioButton (hDlg, PGLOB_FSTA,
                 PGLOB_FPER, wParam);
/* Basculer la sélection des deux boutons radios */
PGLOBCB2[0].Etat = 0;
PGLOBCB2[1].Etat = 1;
/* Refléter cette action sur le tableau des états courants non encore
validés*/
for (n=2; n<NPGLOBCB; n++)
/* Décocher toutes les boîtes à cocher et refléter cette action sur le
tableau des états courants non encore validés*/
{
CheckDlgButton(hDlg,PGLOBCB[n].nId,0);
PGLOBCB2[n].Etat = 0;
};
EnableWindow (GetDlgItem(hDlg,PGLOB_DUREE),
             TRUE);
EnableWindow (GetDlgItem(hDlg,PGLOB_ETAT),
             TRUE);
/* Rendre les boutons Durées et Etats disponibles */
break;

case PGLOB_DEC :
case PGLOB_ACT :
case PGLOB_INT :
case PGLOB_REA :
case PGLOB_TER :
case PGLOB_VIE :
case PGLOB_PRO :
/* Cas où le message provient de l'une des boîtes à cocher */
CheckDlgButton (hDlg, wParam,
                ! IsDlgButtonChecked (hDlg,wParam));
/* Cocher la boîte si elle est décochée; la décocher sinon */

PGLOBCB2[wParam-PGLOB_DEC+2].Etat =
PGLOBCB2[wParam-PGLOB_DEC+2].Etat == 0 ? 1 : 0;

```

```

/* Refléter cette action sur le tableau des états courants non encore
validés*/

Standard = TRUE;
for (n=2; n<NPGLOBCB; n++)
/* Vérifier si la boîte de dialogue est standard */
if (PGLOBCB2[n].Etat != PGLOBCB[n].Etat)
/* Si l'état courant d'au moins une des boîtes à cocher diffère
de l'état initial (standard), la boîte de dialogue n'est plus
standard */
{
Standard = FALSE;
break;
};
if (Standard && STPGLDR && STPGLET)
/* Si la boîte de dialogue est standard et si les boîtes de dialogues
Durées et Etats sont standards */
{
CheckRadioButton (hDlg,
PGLOB_FSTA,
PGLOB_FPER,
PGLOB_FSTA);
/* Remettre les boutons radios dans leur état initial */
EnableWindow (GetDlgItem(hDlg,
PGLOB_DUREE), FALSE);
EnableWindow (GetDlgItem(hDlg,PGLOB_ETAT),
FALSE);
/* Rendre indisponibles les boutons Durées et Etats */
}
else
{
CheckRadioButton (hDlg,
PGLOB_FSTA,
PGLOB_FPER,
PGLOB_FPER);
/* Sélectionner le bouton radio Fiche personnalisée et désélec-
tionner le bouton radio Fiche standard */
EnableWindow (GetDlgItem(hDlg,
PGLOB_DUREE), TRUE);
EnableWindow (GetDlgItem(hDlg,PGLOB_ETAT),
TRUE);
/* Rendre disponibles les boutons Durées et Etats */
};

break;

case PGLOB_DUREE :
/* Cas où le message provient du bouton Durées */
DialogBox (hInst, "PGLDR", hDlg, lpfnPGLDRDlgProc);
/* Afficher la boîte de dialogue permettant de choisir les statistiques
concernant les durées des processus. La procédure PGLDRDlgProc
prend la relève et rend ensuite la main. Le tableau des états
PGLDRCB1[] a été éventuellement modifié. */

STPGLDR = TRUE;
for (n=0; n<NPGLDRCB; n++)
/* Vérifier si la boîte de dialogue des durées est encore standard */

```

```

if (PGLDRCB1[n].Etat != PGLDRCB[n].Etat)
/* Si l'état courant d'au moins une des boîtes à cocher diffère
de l'état initial (standard), la boîte de dialogue n'est plus
standard */
{
    STPGLDR = FALSE;
    break;
};
if (Standard && STPGLDR && STPGLET)
/* Si les trois boîtes de dialogue sont encore standards */
{
    CheckRadioButton (hDlg, PGLOB_FSTA,
                      PGLOB_FPER,
                      PGLOB_FSTA);
/* Remettre les boutons radios dans leur état initial */
EnableWindow (GetDlgItem(hDlg,
                          PGLOB_DUREE), FALSE);
EnableWindow (GetDlgItem(hDlg,PGLOB_ETAT),
              FALSE);
/* Rendre indisponibles les boutons Durées et Etats */
}
else
{
    CheckRadioButton (hDlg, PGLOB_FSTA,
                      PGLOB_FPER,
                      PGLOB_FPER);
/* Sélectionner le bouton radio Fiche personnalisée et désélectionner
le bouton radio Fiche standard */
EnableWindow (GetDlgItem(hDlg,
                          PGLOB_DUREE), TRUE);
EnableWindow (GetDlgItem(hDlg,PGLOB_ETAT),
              TRUE);
/* Rendre disponibles les boutons Durées et Etats */
};

break;

case PGLOB_ETAT :
/* Cas où le message provient du bouton Etats */
DialogBox (hInst, "PGET", hDlg, lpfnPGETDlgProc);
/* Afficher la boîte de dialogue permettant de choisir les statistiques
concernant les états des processus. La procédure PGETDlgProc
prend la relève et rend ensuite la main. Le tableau des états
PGETCB1[] a été éventuellement modifié. */

STPGLET = TRUE;
for (n=0; n<NPGETCB; n++)
/* Vérifier si la boîte de dialogue des états est encore standard */
if (PGETCB1[n].Etat != PGETCB[n].Etat)
/* Si l'état courant d'au moins une des boîtes à cocher diffère
de l'état initial (standard), la boîte de dialogue n'est plus
standard */
{
    STPGLET = FALSE;
    break;
};
if (Standard && STPGLDR && STPGLET)
/* Si les trois boîtes de dialogue sont encore standards */

```

```

    {
        CheckRadioButton (hDlg, PGLOB_FSTA,
                          PGLOB_FPER,
                          PGLOB_FSTA);
        /* Remettre les boutons radios dans leur état initial */
        EnableWindow (GetDlgItem(hDlg,
                                PGLOB_DUREE), FALSE);
        EnableWindow (GetDlgItem(hDlg,PGLOB_ETAT),
                      FALSE);
        /* Rendre indisponibles les boutons Durées et Etats */
    }
    else
    {
        CheckRadioButton (hDlg, PGLOB_FSTA,
                          PGLOB_FPER,
                          PGLOB_FPER);
        /* Sélectionner le bouton radio Fiche personnalisée et désélectionner le bouton radio Fiche standard */
        EnableWindow (GetDlgItem(hDlg,
                                PGLOB_DUREE), TRUE);
        EnableWindow (GetDlgItem(hDlg,PGLOB_ETAT),
                      TRUE);
        /* Rendre disponibles les boutons Durées et Etats */
    };

    break;

    case IDOK :
        EndDialog (hDlg, TRUE);
        break;

    case IDCANCEL :
        EndDialog (hDlg, FALSE);
        break;

    default :
        return FALSE;
    }
    break;

    default :
        return FALSE;
    }
    return TRUE;
}

```

**BOOL FAR PASCAL PPERDlgProc (hDlg, iMessage, wParam, lParam)**

```

    HWND          hDlg;
    unsigned      iMessage;
    WORD          wParam;
    LONG          lParam;
    {
    static BOOL    Standard;
    short          n;
    static struct InitCB PPERCB2[NPPERCB];

```

```

switch (iMessage)
{
case WM_INITDIALOG :
    Standard = TRUE;

    for (n=0; n<NPPERCB; n++)
    {
        PPERCB2[n].nId = PPERCB[n].nId;
        PPERCB2[n].Etat = PPERCB[n].Etat;
    };

    for (n=0; n<2; n++)
        SendMessage (GetDlgItem(hDlg,PPERCB[n].nId),
            BM_SETCHECK, PPERCB[n].Etat, 0L);

    for (n=2; n<NPPERCB; n++)
        switch (PPERCB[n].Etat)
        {
        case 0 :
            CheckDlgButton(hDlg,PPERCB[n].nId,0);
            break;

        case 1 :
            CheckDlgButton(hDlg,PPERCB[n].nId,1);
            break;

        default :
            break;
        };

    break;

case WM_COMMAND :
    switch (wParam)
    {
    case PPER_FSTA :
        SendMessage (hDlg, WM_INITDIALOG, wParam,
            lParam);
        break;

    case PPER_FPER :
        Standard = FALSE;
        CheckRadioButton (hDlg, PPER_FSTA, PPER_FPER,
            wParam);
        PPERCB2[0].Etat = 1;
        PPERCB2[1].Etat = 0;
        for (n=2; n<NPPERCB; n++)
        {
            CheckDlgButton(hDlg,PPERCB[n].nId,0);
            PPERCB2[n].Etat = 0;
        };
        break;

    case PPER_DEC :
    case PPER_ACT :
    case PPER_INT :

```

```

case PPER_REA :
case PPER_TER :
case PPER_DRT :
case PPER_DRI :
case PPER_DRR :
case PPER_RPV :
case PPER_RPS :
    CheckDlgButton (hDlg, wParam,
                    ! IsDlgButtonChecked (hDlg,wParam));

    PPERCB2[wParam-PPER_DEC+2].Etat =
    PPERCB2[wParam-PPER_DEC+2].Etat == 0 ? 1 : 0;

    Standard = TRUE;
    for (n=2; n<NPPERCB; n++)
        if (PPERCB2[n].Etat != PPERCB[n].Etat)
            {
                Standard = FALSE;
                break;
            };
    if (Standard)
        CheckRadioButton (hDlg, PPER_FSTA,
                          PPER_FPER, PPER_FSTA);
    else
        CheckRadioButton (hDlg, PPER_FSTA,
                          PPER_FPER, PPER_FPER);

    break;

case IDOK :
    EndDialog (hDlg, TRUE);
    break;

case IDCANCEL :
    EndDialog (hDlg, FALSE);
    break;

default :
    return FALSE;
}
break;

default :
    return FALSE;
}
return TRUE;
}

```

**BOOL FAR PASCAL PCHDRDlgProc (hDlg, iMessage, wParam, lParam)**

```

HWND    hDlg;
unsigned iMessage;
WORD    wParam;
LONG    lParam;
{
short   n;

```



```

static struct InitCB PCHDRCB2[NPCHDRCB];

switch (iMessage)
{
case WM_INITDIALOG :
    for (n=0; n<NPCHDRCB; n++)
    {
        PCHDRCB2[n].nId = PCHDRCB1[n].nId;
        PCHDRCB2[n].Etat = PCHDRCB1[n].Etat;
    };

    for (n=0; n<NPCHDRCB; n++)
        switch (PCHDRCB2[n].Etat)
        {
            case 0 :
                CheckDlgButton(hDlg,PCHDRCB2[n].nId,0);
                break;

            case 1 :
                CheckDlgButton(hDlg,PCHDRCB2[n].nId,1);
                break;

            default :
                break;
        };
        break;

case WM_COMMAND :
    switch (wParam)
    {
        case PCHDR_T :
        case PCHDR_R :
        case PCHDR_I :
            CheckDlgButton (hDlg, wParam,
                ! IsDlgButtonChecked (hDlg,wParam));

            PCHDRCB2[wParam-PCHDR_T].Etat =
            PCHDRCB2[wParam-PCHDR_T].Etat == 0 ? 1 : 0;

            break;

        case IDOK :
            for (n=0; n<NPCHDRCB; n++)
                PCHDRCB1[n].Etat = PCHDRCB2[n].Etat;

            EndDialog (hDlg, TRUE);
            break;

        case IDCANCEL :
            EndDialog (hDlg, FALSE);
            break;

        default :
            return FALSE;
    }
    break;
}

```

```

        default :
            return FALSE;
    }
    return TRUE;
}

```

**BOOL FAR PASCAL PCHETDlgProc (hDlg, iMessage, wParam, lParam)**

```

    HWND    hDlg;
    unsigned iMessage;
    WORD    wParam;
    LONG    lParam;
    {
    short    n;
    static struct InitCB PCHETCB2[NPCHETCB];

    switch (iMessage)
    {
    case WM_INITDIALOG :
        for (n=0; n<NPCHETCB; n++)
            {
            PCHETCB2[n].nId = PCHETCB1[n].nId;
            PCHETCB2[n].Etat = PCHETCB1[n].Etat;
            };

        for (n=0; n<NPCHETCB; n++)
            switch (PCHETCB2[n].Etat)
            {
            case 0 :
                CheckDlgButton(hDlg,PCHETCB2[n].nId,0);
                break;

            case 1 :
                CheckDlgButton(hDlg,PCHETCB2[n].nId,1);
                break;

            default :
                break;
            };

        break;

    case WM_COMMAND :
        switch (wParam)
        {
        case PCHET_T :
        case PCHET_N :
        case PCHET_F :
        case PCHET_I :
        case PCHET_C :
            CheckDlgButton (hDlg, wParam,
                ! IsDlgButtonChecked (hDlg,wParam));

            PCHETCB2[wParam-PCHET_T].Etat =
            PCHETCB2[wParam-PCHET_T].Etat == 0 ? 1 : 0;

            break;

```

```

        case IDOK :
            for (n=0; n<NPCHETCB; n++)
                PCHETCB1[n].Etat = PCHETCB2[n].Etat;

            EndDialog (hDlg, TRUE);
            break;

        case IDCANCEL :
            EndDialog (hDlg, FALSE);
            break;

        default :
            return FALSE;
    }
    break;

    default :
        return FALSE;
}
return TRUE;
}

```

**BOOL FAR PASCAL PCHRDlgProc (hDlg, iMessage, wParam, lParam)**

```

    HWND          hDlg;
    unsigned      iMessage;
    WORD          wParam;
    LONG          lParam;
    {
        static BOOL      Standard;
        static BOOL      STPCHDR;
        static BOOL      STPCHET;
        short          n;
        static struct InitCB    PCHRCB2[NPCHRCB];
        static FARPROC    lpfnPCHDRDlgProc;
        static FARPROC    lpfnPCHETDlgProc;

        switch (iMessage)
        {
            case WM_INITDIALOG :
                EnableWindow (GetDlgItem(hDlg,PCHR_DUREE), FALSE);
                EnableWindow (GetDlgItem(hDlg,PCHR_ETAT), FALSE);

                Standard = TRUE;
                STPCHDR      = TRUE;
                STPCHET      = TRUE;

                for (n=0; n<NPCHRCB; n++)
                {
                    PCHRCB2[n].nId = PCHRCB[n].nId;
                    PCHRCB2[n].Etat = PCHRCB[n].Etat;
                };
            }
        }

```

```
for (n=0; n<NPCHDRCB; n++)
{
    PCHDRCB1[n].nId = PCHDRCB[n].nId;
    PCHDRCB1[n].Etat = PCHDRCB[n].Etat;
};

for (n=0; n<NPCHETCB; n++)
{
    PCHETCB1[n].nId = PCHETCB[n].nId;
    PCHETCB1[n].Etat = PCHETCB[n].Etat;
};

for (n=0; n<2; n++)
    SendMessage (GetDlgItem(hDlg,PCHRCB[n].nId),
                BM_SETCHECK, PCHRCB[n].Etat, 0L);

for (n=2; n<NPCHRCB; n++)
    switch (PCHRCB[n].Etat)
    {
        case 0 :
            CheckDlgButton(hDlg,PCHRCB[n].nId,0);
            break;

        case 1 :
            CheckDlgButton(hDlg,PCHRCB[n].nId,1);
            break;

        default :
            break;
    };

lpfnPCHDRDdlgProc = MakeProcInstance (PCHDRDdlgProc, hInst);
lpfnPCHETDdlgProc = MakeProcInstance (PCHETDdlgProc, hInst);

break;

case WM_COMMAND :
    switch (wParam)
    {
        case PCHR_FSTA :
            SendMessage (hDlg, WM_INITDIALOG, wParam,
                        lParam);
            break;

        case PCHR_FPER :
            Standard = FALSE;
            CheckRadioButton (hDlg, PCHR_FSTA, PCHR_FPER,
                            wParam);
            PCHRCB2[0].Etat = 1;
            PCHRCB2[1].Etat = 0;
            for (n=2; n<NPCHRCB; n++)
            {
                CheckDlgButton(hDlg,PCHRCB[n].nId,0);
                PCHRCB2[n].Etat = 0;
            };
            EnableWindow (GetDlgItem(hDlg,PCHR_DUREE),
                        TRUE);
    }
}
```

```
EnableWindow (GetDlgItem(hDlg,PCHR_ETAT),
              TRUE);
break;

case PCHR_DEC :
case PCHR_ACT :
case PCHR_INT :
case PCHR_REA :
case PCHR_TER :
case PCHR_RPV :
case PCHR_RPT :
case PCHR_RPP :
    CheckDlgButton (hDlg, wParam,
                   ! IsDlgButtonChecked (hDlg,wParam));

PCHRCB2[wParam-PCHR_DEC+2].Etat =
PCHRCB2[wParam-PCHR_DEC+2].Etat == 0 ? 1 : 0;

Standard = TRUE;
for (n=2; n<NPCHRCB; n++)
    if (PCHRCB2[n].Etat != PCHRCB[n].Etat)
        {
            Standard = FALSE;
            break;
        };
if (Standard && STPCHDR && STPCHET)
    {
        CheckRadioButton (hDlg, PCHR_FSTA,
                          PCHR_FPER, PCHR_FSTA);
        EnableWindow (GetDlgItem(hDlg,
                                PCHR_DUREE), FALSE);
        EnableWindow (GetDlgItem(hDlg,PCHR_ETAT),
                      FALSE);
    }
else
    {
        CheckRadioButton (hDlg, PCHR_FSTA,
                          PCHR_FPER, PCHR_FPER);
        EnableWindow (GetDlgItem(hDlg,
                                PCHR_DUREE), TRUE);
        EnableWindow (GetDlgItem(hDlg,PCHR_ETAT),
                      TRUE);
    };

break;

case PCHR_DUREE :
    DialogBox (hInst, "PCHDR", hDlg, lpfnPCHDRDlgProc);

STPCHDR = TRUE;
for (n=0; n<NPCHDRCB; n++)
    if (PCHDRCB1[n].Etat != PCHDRCB[n].Etat)
        {
            STPCHDR = FALSE;
            break;
        };
```

```
if (Standard && STPCHDR && STPCHET)
{
    CheckRadioButton (hDlg, PCHR_FSTA,
                     PCHR_FPER, PCHR_FSTA);
    EnableWindow (GetDlgItem(hDlg,
                             PCHR_DUREE), FALSE);
    EnableWindow (GetDlgItem(hDlg,PCHR_ETAT),
                 FALSE);
}
else
{
    CheckRadioButton (hDlg, PCHR_FSTA,
                     PCHR_FPER, PCHR_FPER);
    EnableWindow (GetDlgItem(hDlg,
                             PCHR_DUREE), TRUE);
    EnableWindow (GetDlgItem(hDlg,PCHR_ETAT),
                 TRUE);
};

break;

case PCHR_ETAT :
    DialogBox (hInst, "PCHET", hDlg, lpfnPCHETDlgProc);

    STPCHET = TRUE;
    for (n=0; n<NPCHETCB; n++)
        if (PCHETCB1[n].Etat != PCHETCB[n].Etat)
            {
                STPCHET = FALSE;
                break;
            };
    if (Standard && STPCHDR && STPCHET)
    {
        CheckRadioButton (hDlg, PCHR_FSTA,
                         PCHR_FPER, PCHR_FSTA);
        EnableWindow (GetDlgItem(hDlg,
                                 PCHR_DUREE), FALSE);
        EnableWindow (GetDlgItem(hDlg,PCHR_ETAT),
                     FALSE);
    }
    else
    {
        CheckRadioButton (hDlg, PCHR_FSTA,
                         PCHR_FPER, PCHR_FPER);
        EnableWindow (GetDlgItem(hDlg,
                                 PCHR_DUREE), TRUE);
        EnableWindow (GetDlgItem(hDlg,PCHR_ETAT),
                     TRUE);
    };

    break;

case IDOK :
    EndDialog (hDlg, TRUE);
    break;
```

```
        case IDCANCEL :
            EndDialog (hDlg, FALSE);
            break;

        default :
            return FALSE;
    }
    break;

    default :
        return FALSE;
    }
    return TRUE;
}
```

**BOOL FAR PASCAL PCOMDlgProc (hDlg, iMessage, wParam, lParam)**

```
HWND    hDlg;
unsigned iMessage;
WORD    wParam;
LONG    lParam;
{
    short    n;

    switch (iMessage)
    {
        case WM_INITDIALOG :
            for (n=0; n<NPCOMCB; n++)
                switch (PCOMCB[n].Etat)
                {
                    case 0 :
                        CheckDlgButton(hDlg,PCOMCB[n].nId,0);
                        break;

                    case 1 :
                        CheckDlgButton(hDlg,PCOMCB[n].nId,1);
                        break;

                    default :
                        break;
                }
            break;

        case WM_COMMAND :
            switch (wParam)
            {
                case PCOM_TITRE :
                case PCOM_DATE :
                case PCOM_TYPE :
                case PCOM_PERIODE :
                case PCOM_NOM :
                case PCOM_CALENDRIER :
                    CheckDlgButton (hDlg, wParam,
                                     ! IsDlgButtonChecked (hDlg,wParam));
                    break;
            }
    }
}
```

```

        case IDOK :
            EndDialog (hDlg, TRUE);
            break;

        case IDCANCEL :
            EndDialog (hDlg, FALSE);
            break;

        default :
            return FALSE;
    }
    break;

    default :
        return FALSE;
}
return TRUE;
}

```

**BOOL FAR PASCAL PCALDlgProc (hDlg, iMessage, wParam, lParam)**

```

HWND    hDlg;
unsigned iMessage;
WORD    wParam;
LONG    lParam;
{
    short    n;

    switch (iMessage)
    {
        case WM_INITDIALOG :
            for (n=0; n<NPCALCB; n++)
                switch (PCALCB[n].Etat)
                {
                    case 0 :
                        CheckDlgButton(hDlg,PCALCB[n].nId,0);
                        break;

                    case 1 :
                        CheckDlgButton(hDlg,PCALCB[n].nId,1);
                        break;

                    default :
                        break;
                }
            break;

        case WM_COMMAND :
            switch (wParam)
            {
                case PCAL_P :
                case PCAL_R :
                    CheckDlgButton (hDlg, wParam,
                                     ! IsDlgButtonChecked (hDlg,wParam));
                    break;
            }
    }
}

```



```
        case IDOK :  
            EndDialog (hDlg, TRUE);  
            break;  
  
        case IDCANCEL :  
            EndDialog (hDlg, FALSE);  
            break;  
  
        default :  
            return FALSE;  
    }  
    break;  
  
    default :  
        return FALSE;  
    }  
    return TRUE;  
}
```

## 5.5 RESSOUR.H

### Liste des procédures

RCHRDlgProc .....	75
RCOMDlgProc .....	77
RGLOBDlgProc .....	70
RGLUTDlgProc .....	68
RPERDlgProc .....	73

### Contenu du fichier

```
#define NRGLOBCB (sizeof RGLOBCB / sizeof RGLOBCB[0])
#define NRGLUTCB (sizeof RGLUTCB / sizeof RGLUTCB[0])
#define NRPERCB (sizeof RPERCB / sizeof RPERCB[0])
#define NRCHRCB (sizeof RCHRCB / sizeof RCHRCB[0])
#define NRCOMCB (sizeof RCOMCB / sizeof RCOMCB[0])
```

```
struct InitCB RGLOBCB[] =
{
    RGLOB_FSTA, 1,
    RGLOB_FPER, 0,
    RGLOB_CC, 0,
    RGLOB_CE, 0,
    RGLOB_ER, 1,
    RGLOB_EL, 1,
    RGLOB_ED, 0,
    RGLOB_RC, 1,
    RGLOB_RE, 1
},
RGLUTCB[] =
{
    RGLUT_CN, 0,
    RGLUT_CY, 0,
    RGLUT_CX, 0,
    RGLUT_CE, 0,
    RGLUT_RN, 0,
    RGLUT_RY, 0,
    RGLUT_RX, 0,
    RGLUT_RE, 0,
    RGLUT_PN, 0,
    RGLUT_PY, 0,
    RGLUT_PX, 0,
    RGLUT_PE, 0
},
RPERCB[] =
{
    RPER_FSTA, 1,
    RPER_FPER, 0,
    RPER_ER, 0,
    RPER_EL, 0,
    RPER_ED, 0,
    RPER_RV, 1
},
```

```

RCHRCB[] =
{
    RCHR_FSTA, 1,
    RCHR_FPER, 0,
    RCHR_ER, 0,
    RCHR_EL, 0,
    RCHR_ED, 0,
    RCHR_UC, 0,
    RCHR_UQ, 0,
    RCHR_UY, 0,
    RCHR_UX, 0,
    RCHR_UF, 0,
    RCHR_RC, 1,
    RCHR_RU, 1,
    RCHR_RT, 1
},
RCOMCB[] =
{
    RCOM_TITRE, 1,
    RCOM_DATE, 1,
    RCOM_TYPE, 1,
    RCOM_PERIODE, 1,
    RCOM_NOM, 1,
    RCOM_CALENDRIER, 1
};

```

```
struct InitCB RGLUTCB1[NPGLDRCB];
```

**BOOL FAR PASCAL RGLUTDlgProc (hDlg, iMessage, wParam, lParam)**

```

HWND hDlg;
unsigned iMessage;
WORD wParam;
LONG lParam;
{
    short n;
    static struct InitCB RGLUTCB2[NRGLUTCB];

    switch (iMessage)
    {
        case WM_INITDIALOG :
            for (n=0; n<NRGLUTCB; n++)
            {
                RGLUTCB2[n].nId = RGLUTCB1[n].nId;
                RGLUTCB2[n].Etat = RGLUTCB1[n].Etat;
            };

            for (n=0; n<NRGLUTCB; n++)
                switch (RGLUTCB2[n].Etat)
                {
                    case 0 :
                        CheckDlgButton(hDlg,RGLUTCB2[n].nId,0);
                        break;

```

```
        case 1 :
            CheckDlgButton(hDlg,RGLUTCB2[n].nId,1);
            break;

        default :
            break;
    };
    break;

case WM_COMMAND :
    switch (wParam)
    {
        case RGLUT_CN :
        case RGLUT_CY :
        case RGLUT_CX :
        case RGLUT_CE :
        case RGLUT_RN :
        case RGLUT_RY :
        case RGLUT_RX :
        case RGLUT_RE :
        case RGLUT_PN :
        case RGLUT_PY :
        case RGLUT_PX :
        case RGLUT_PE :
            CheckDlgButton (hDlg, wParam,
                ! IsDlgButtonChecked (hDlg,wParam));

            RGLUTCB2[wParam-RGLUT_CN].Etat =
            RGLUTCB2[wParam-RGLUT_CN].Etat == 0 ? 1 : 0;

            break;

        case IDOK :
            for (n=0; n<NRGLUTCB; n++)
                RGLUTCB1[n].Etat = RGLUTCB2[n].Etat;

            EndDialog (hDlg, TRUE);
            break;

        case IDCANCEL :
            EndDialog (hDlg, FALSE);
            break;

        default :
            return FALSE;
    }
    break;

default :
    return FALSE;
}
return TRUE;
}
```

```

BOOL FAR PASCAL RGLOBDlgProc (hDlg, iMessage, wParam, lParam)
    HWND          hDlg;
    unsigned      iMessage;
    WORD          wParam;
    LONG          lParam;
    {
    static BOOL    Standard;
    static BOOL    STRGLUT;
    short         n;
    static struct InitCB  RGLOBCB2[NRGLOBCB];
    static FARPROC  lpfnRGLUTDlgProc;

    switch (iMessage)
    {
    case WM_INITDIALOG :
        EnableWindow (GetDlgItem(hDlg,RGLOB_UTIL), FALSE);

        Standard = TRUE;
        STRGLUT = TRUE;

        for (n=0; n<NRGLOBCB; n++)
            {
            RGLOBCB2[n].nId = RGLOBCB[n].nId;
            RGLOBCB2[n].Etat = RGLOBCB[n].Etat;
            };

        for (n=0; n<NRGLUTCB; n++)
            {
            RGLUTCB1[n].nId = RGLUTCB[n].nId;
            RGLUTCB1[n].Etat = RGLUTCB[n].Etat;
            };

        for (n=0; n<2; n++)
            SendMessage (GetDlgItem(hDlg,RGLOBCB[n].nId),
                        BM_SETCHECK, RGLOBCB[n].Etat, 0L);

        for (n=2; n<NRGLOBCB; n++)
            switch (RGLOBCB[n].Etat)
            {
            case 0 :
                CheckDlgButton(hDlg,RGLOBCB[n].nId,0);
                break;

            case 1 :
                CheckDlgButton(hDlg,RGLOBCB[n].nId,1);
                break;

            default :
                break;
            };

        lpfnRGLUTDlgProc = MakeProcInstance (RGLUTDlgProc, hInst);

        break;
    }
    }

```

```

case WM_COMMAND :
    switch (wParam)
    {
    case RGLOB_FSTA :
        SendMessage (hDlg, WM_INITDIALOG, wParam,
                    lParam);
        break;

    case RGLOB_FPER :
        Standard = FALSE;
        CheckRadioButton (hDlg, RGLOB_FSTA,
                        RGLOB_FPER, wParam);
        RGLOB_C2[0].Etat = 0;
        RGLOB_C2[1].Etat = 1;
        for (n=2; n<NRGLOB_C2; n++)
            {
            CheckDlgButton(hDlg, RGLOB_C2[n].nId, 0);
            RGLOB_C2[n].Etat = 0;
            };
        EnableWindow (GetDlgItem(hDlg, RGLOB_UTIL),
                    TRUE);
        break;

    case RGLOB_CC :
    case RGLOB_CE :
    case RGLOB_ER :
    case RGLOB_EL :
    case RGLOB_ED :
    case RGLOB_RC :
    case RGLOB_RE :
        CheckDlgButton (hDlg, wParam,
                    ! IsDlgButtonChecked (hDlg, wParam));

        RGLOB_C2[wParam-RGLOB_CC+2].Etat =
        RGLOB_C2[wParam-RGLOB_CC+2].Etat == 0 ? 1 : 0;

        Standard = TRUE;
        for (n=2; n<NRGLOB_C2; n++)
            if (RGLOB_C2[n].Etat != RGLOB_C2[n].Etat)
                {
                Standard = FALSE;
                break;
                };
        if (Standard && STRGLUT)
            {
            CheckRadioButton (hDlg, RGLOB_FSTA,
                            RGLOB_FPER,
                            RGLOB_FSTA);
            EnableWindow (GetDlgItem(hDlg, RGLOB_UTIL),
                        FALSE);
            }
        else
            {
            CheckRadioButton (hDlg, RGLOB_FSTA,
                            RGLOB_FPER,
                            RGLOB_FPER);
            }
    }
    }

```

```
        EnableWindow (GetDlgItem(hDlg, RGLOB_UTIL),
                      TRUE);
    };

    break;

case RGLOB_UTIL :
    DialogBox (hInst, "RGLUT", hDlg, lpfnRGLUTDlgProc);

    STRGLUT = TRUE;
    for (n=0; n<NRGLUTCB; n++)
        if (RGLUTCB1[n].Etat != RGLUTCB[n].Etat)
            {
                STRGLUT = FALSE;
                break;
            };
    if (Standard && STRGLUT)
        {
            CheckRadioButton (hDlg, RGLOB_FSTA,
                              RGLOB_FPER,
                              RGLOB_FSTA);
            EnableWindow (GetDlgItem(hDlg, RGLOB_UTIL),
                          FALSE);
        }
    else
        {
            CheckRadioButton (hDlg, RGLOB_FSTA,
                              RGLOB_FPER,
                              RGLOB_FPER);
            EnableWindow (GetDlgItem(hDlg, RGLOB_UTIL),
                          TRUE);
        }
    };

    break;

case IDOK :
    EndDialog (hDlg, TRUE);
    break;

case IDCANCEL :
    EndDialog (hDlg, FALSE);
    break;

default :
    return FALSE;
}
break;

default :
    return FALSE;
}
return TRUE;
}
```

```
BOOL FAR PASCAL RPERDlgProc (hDlg, iMessage, wParam, lParam)
    HWND          hDlg;
    unsigned      iMessage;
    WORD          wParam;
    LONG          lParam;
    {
    static BOOL    Standard;
    short         n;
    static struct InitCB  RPERCB2[NRPERCB];

    switch (iMessage)
    {
    case WM_INITDIALOG :
        Standard = TRUE;

        for (n=0; n<NRPERCB; n++)
            {
            RPERCB2[n].nId = RPERCB[n].nId;
            RPERCB2[n].Etat = RPERCB[n].Etat;
            };

        for (n=0; n<2; n++)
            SendMessage (GetDlgItem(hDlg,RPERCB[n].nId),
                BM_SETCHECK, RPERCB[n].Etat, 0L);

        for (n=2; n<NRPERCB; n++)
            switch (RPERCB[n].Etat)
            {
            case 0 :
                CheckDlgButton(hDlg,RPERCB[n].nId,0);
                break;

            case 1 :
                CheckDlgButton(hDlg,RPERCB[n].nId,1);
                break;

            default :
                break;
            };

        break;

    case WM_COMMAND :
        switch (wParam)
        {
        case RPER_FSTA :
            SendMessage (hDlg, WM_INITDIALOG, wParam,
                lParam);
            break;

        case RPER_FPER :
            Standard = FALSE;
            CheckRadioButton (hDlg, RPER_FSTA, RPER_FPER,
                wParam);
            RPERCB2[0].Etat = 1;
            RPERCB2[1].Etat = 0;
        }
    }
}
```



```
for (n=2; n<NRPERCB; n++)
{
    CheckDlgButton(hDlg,RPERCB[n].nId,0);
    RPERCB2[n].Etat = 0;
};
break;

case RPER_ER :
case RPER_EL :
case RPER_ED :
case RPER_RV :
    CheckDlgButton (hDlg, wParam,
        ! IsDlgButtonChecked (hDlg,wParam));

    RPERCB2[wParam-RPER_ER+2].Etat =
    RPERCB2[wParam-RPER_ER+2].Etat == 0 ? 1 : 0;

    Standard = TRUE;
    for (n=2; n<NRPERCB; n++)
        if (RPERCB2[n].Etat != RPERCB[n].Etat)
            {
                Standard = FALSE;
                break;
            };
    if (Standard)
        CheckRadioButton (hDlg, RPER_FSTA,
            RPER_FPER, RPER_FSTA);
    else
        CheckRadioButton (hDlg, RPER_FSTA,
            RPER_FPER, RPER_FPER);

    break;

case IDOK :
    EndDialog (hDlg, TRUE);
    break;

case IDCANCEL :
    EndDialog (hDlg, FALSE);
    break;

default :
    return FALSE;
}
break;

default :
    return FALSE;
}
return TRUE;
}
```

```

BOOL FAR PASCAL RCHRDlgProc (hDlg, iMessage, wParam, lParam)
    HWND          hDlg;
    unsigned      iMessage;
    WORD          wParam;
    LONG          lParam;
    {
    static BOOL    Standard;
    short         n;
    static struct InitCB RCHRCB2[NRCHRCB];

    switch (iMessage)
    {
    case WM_INITDIALOG :
        Standard = TRUE;

        for (n=0; n<NRCHRCB; n++)
            {
            RCHRCB2[n].nId = RCHRCB[n].nId;
            RCHRCB2[n].Etat = RCHRCB[n].Etat;
            };

        for (n=0; n<2; n++)
            SendMessage (GetDlgItem(hDlg,RCHRCB[n].nId),
                BM_SETCHECK, RCHRCB[n].Etat, 0L);

        for (n=2; n<NRCHRCB; n++)
            switch (RCHRCB[n].Etat)
            {
            case 0 :
                CheckDlgButton(hDlg,RCHRCB[n].nId,0);
                break;

            case 1 :
                CheckDlgButton(hDlg,RCHRCB[n].nId,1);
                break;

            default :
                break;
            };

        break;

    case WM_COMMAND :
        switch (wParam)
        {
        case RCHR_FSTA :
            SendMessage (hDlg, WM_INITDIALOG, wParam,
                lParam);
            break;

        case RCHR_FPER :
            Standard = FALSE;
            CheckRadioButton (hDlg, RCHR_FSTA, RCHR_FPER,
                wParam);
            RCHRCB2[0].Etat = 1;
            RCHRCB2[1].Etat = 0;
        }
    }
}

```

```
for (n=2; n<NRCHRCB; n++)
{
    CheckDlgButton(hDlg,RCHRCB[n].nId,0);
    RCHRCB2[n].Etat = 0;
};
break;

case RCHR_ER :
case RCHR_EL :
case RCHR_ED :
case RCHR_UC :
case RCHR_UQ :
case RCHR_UY :
case RCHR_UX :
case RCHR_UF :
case RCHR_RC :
case RCHR_RU :
case RCHR_RT :
    CheckDlgButton (hDlg, wParam,
                    ! IsDlgButtonChecked (hDlg,wParam));

    RCHRCB2[wParam-RCHR_ER+2].Etat =
    RCHRCB2[wParam-RCHR_ER+2].Etat == 0 ? 1 : 0;

    Standard = TRUE;
    for (n=2; n<NRCHRCB; n++)
        if (RCHRCB2[n].Etat != RCHRCB[n].Etat)
            {
                Standard = FALSE;
                break;
            };
    if (Standard)
        CheckRadioButton (hDlg,
            RCHR_FSTA, RCHR_FPER, RCHR_FSTA);
    else
        CheckRadioButton (hDlg,
            RCHR_FSTA, RCHR_FPER, RCHR_FPER);

    break;

case IDOK :
    EndDialog (hDlg, TRUE);
    break;

case IDCANCEL :
    EndDialog (hDlg, FALSE);
    break;

default :
    return FALSE;
}
break;

default :
    return FALSE;
}
return TRUE;
```

}

```

BOOL FAR PASCAL RCOMDlgProc (hDlg, iMessage, wParam, lParam)
    HWND  hDlg;
    unsigned  iMessage;
    WORD  wParam;
    LONG  lParam;
    {
    short  n;

    switch (iMessage)
    {
        case WM_INITDIALOG :
            for (n=0; n<NRCOMCB; n++)
                switch (RCOMCB[n].Etat)
                {
                    case 0 :
                        CheckDlgButton(hDlg,RCOMCB[n].nId,0);
                        break;

                    case 1 :
                        CheckDlgButton(hDlg,RCOMCB[n].nId,1);
                        break;

                    default :
                        break;
                };
            break;

        case WM_COMMAND :
            switch (wParam)
            {
                case RCOM_TITRE :
                case RCOM_DATE :
                case RCOM_TYPE :
                case RCOM_PERIODE :
                case RCOM_NOM :
                case RCOM_CALENDRIER :
                    CheckDlgButton (hDlg, wParam,
                                     ! IsDlgButtonChecked (hDlg,wParam));
                    break;

                case IDOK :
                    EndDialog (hDlg, TRUE);
                    break;

                case IDCANCEL :
                    EndDialog (hDlg, FALSE);
                    break;

                default :
                    return FALSE;
            }
            break;
    }
}

```

```

        default :
            return FALSE;
    }
    return TRUE;
}

```

## 5.6 SYNCHR.H

### Liste des procédures

SCOMDlgProc .....	86
SGLEVDlgProc .....	81
SGLOBDlgProc .....	83
SGLPTDlgProc .....	79

### Contenu du fichier

```

#define NSGLOBCB (sizeof SGLOBCB / sizeof SGLOBCB[0])
#define NSGLPTCB (sizeof SGLPTCB / sizeof SGLPTCB[0])
#define NSGLEVCB (sizeof SGLEVCB / sizeof SGLEVCB[0])
#define NSCOMCB (sizeof SCOMCB / sizeof SCOMCB[0])

```

```

struct InitCB SGLOBCB[] =
{
    SGLOB_FSTA, 1,
    SGLOB_FPER, 0,
    SGLOB_CR, 1,
    SGLOB_CM, 0,
    SGLOB_RD, 1,
    SGLOB_RF, 1,
},
SGLPTCB[] =
{
    SGLPT_R, 1,
    SGLPT_T, 1,
    SGLPT_TN, 0,
    SGLPT_TY, 1,
    SGLPT_TX, 0,
    SGLPT_TE, 0
},
SGLEVCB[] =
{
    SGLEV_C, 1,
    SGLEV_P, 1,
    SGLEV_T, 1,
    SGLEV_TN, 0,
    SGLEV_TY, 1,
    SGLEV_TX, 0,
    SGLEV_TE, 0
},
SCOMCB[] =
{
    SCOM_TITRE, 1,
    SCOM_DATE, 1,
    SCOM_TYPE, 1,
}

```

```

SCOM_PERIODE, 1,
SCOM_NOM,      1
};

```

```

struct InitCB SGLPTCB1[NSGLPTCB];
struct InitCB SGLEVCB1[NSGLEVCB];

```

**BOOL FAR PASCAL SGLPTDlgProc (hDlg, iMessage, wParam, lParam)**

```

HWND          hDlg;
unsigned      iMessage;
WORD          wParam;
LONG          lParam;
{
  BOOL          Temps;
  short        n;
  static struct InitCB SGLPTCB2[NSGLPTCB];

  switch (iMessage)
  {
    case WM_INITDIALOG :
      for (n=0; n<NSGLPTCB; n++)
      {
        SGLPTCB2[n].nId = SGLPTCB1[n].nId;
        SGLPTCB2[n].Etat = SGLPTCB1[n].Etat;
      };

      for (n=0; n<NSGLPTCB; n++)
        switch (SGLPTCB2[n].Etat)
        {
          case 0 :
            CheckDlgButton(hDlg,SGLPTCB2[n].nId,0);
            break;

          case 1 :
            CheckDlgButton(hDlg,SGLPTCB2[n].nId,1);
            break;

          default :
            break;
        };
      break;

    case WM_COMMAND :
      switch (wParam)
      {
        case SGLPT_R :
          CheckDlgButton (hDlg, wParam,
            ! IsDlgButtonChecked (hDlg,wParam));

          SGLPTCB2[0].Etat = SGLPTCB2[0].Etat == 0 ? 1 : 0;

          break;
      }
  }
}

```

```
case SGLPT_T :
    CheckDlgButton (hDlg, wParam,
                    ! IsDlgButtonChecked (hDlg,wParam));

    SGLPTCB2[1].Etat = SGLPTCB2[1].Etat == 0 ? 1 : 0;

    for (n=2; n<NSGLPTCB; n++)
        {
            CheckDlgButton (hDlg, SGLPTCB2[n].nId,
                            IsDlgButtonChecked
                            (hDlg,wParam));
            SGLPTCB2[n].Etat = SGLPTCB2[1].Etat;
        };
    break;

case SGLPT_TN :
case SGLPT_TY :
case SGLPT_TX :
case SGLPT_TE :
    CheckDlgButton (hDlg, wParam,
                    ! IsDlgButtonChecked (hDlg,wParam));

    SGLPTCB2[wParam-SGLPT_R].Etat =
    SGLPTCB2[wParam-SGLPT_R].Etat == 0 ? 1 : 0;

    Temps = FALSE;
    for (n=2; n<NSGLPTCB; n++)
        if (SGLPTCB2[n].Etat == 1)
            {
                Temps = TRUE;
                break;
            };
    if (Temps)
        {
            CheckDlgButton (hDlg, SGLPT_T,1);
            SGLPTCB2[1].Etat = 1;
        }
    else
        {
            CheckDlgButton (hDlg, SGLPT_T,0);
            SGLPTCB2[1].Etat = 0;
        };
    break;

case IDOK :
    for (n=0; n<NSGLPTCB; n++)
        SGLPTCB1[n].Etat = SGLPTCB2[n].Etat;

    EndDialog (hDlg, TRUE);
    break;

case IDCANCEL :
    EndDialog (hDlg, FALSE);
    break;

default :
    return FALSE;
```

```

        break;
    }
    default :
        return FALSE;
    }
    return TRUE;
}

```

**BOOL FAR PASCAL SGLEVDlgProc (hDlg, iMessage, wParam, lParam)**

```

HWND          hDlg;
unsigned      iMessage;
WORD         wParam;
LONG         lParam;
{
    BOOL          Temps;
    short         n;
    static struct InitCB SGLEVCB2[NSGLEVCB];

    switch (iMessage)
    {
        case WM_INITDIALOG :
            for (n=0; n<NSGLEVCB; n++)
            {
                SGLEVCB2[n].nId = SGLEVCB1[n].nId;
                SGLEVCB2[n].Etat = SGLEVCB1[n].Etat;
            };

            for (n=0; n<NSGLEVCB; n++)
                switch (SGLEVCB2[n].Etat)
                {
                    case 0 :
                        CheckDlgButton(hDlg,SGLEVCB2[n].nId,0);
                        break;

                    case 1 :
                        CheckDlgButton(hDlg,SGLEVCB2[n].nId,1);
                        break;

                    default :
                        break;
                };
            break;

        case WM_COMMAND :
            switch (wParam)
            {
                case SGLEV_C :
                case SGLEV_P :
                    CheckDlgButton (hDlg, wParam,
                                     ! IsDlgButtonChecked (hDlg,wParam));

                    SGLEVCB2[wParam-SGLEV_C].Etat =
                    SGLEVCB2[wParam-SGLEV_C].Etat == 0 ? 1 : 0;
                    break;
            }
    }
}

```



```
case SGLEV_T :
    CheckDlgButton (hDlg, wParam,
                    ! IsDlgButtonChecked (hDlg,wParam));

    SGLEVCB2[2].Etat = SGLEVCB2[2].Etat == 0 ? 1 : 0;
    for (n=3; n<NSGLEVCB; n++)
    {
        CheckDlgButton (hDlg, SGLEVCB2[n].nId,
                        IsDlgButtonChecked
                        (hDlg,wParam));
        SGLEVCB2[n].Etat = SGLEVCB2[2].Etat;
    };
    break;

case SGLEV_TN :
case SGLEV_TY :
case SGLEV_TX :
case SGLEV_TE :
    CheckDlgButton (hDlg, wParam,
                    ! IsDlgButtonChecked (hDlg,wParam));

    SGLEVCB2[wParam-SGLEV_C].Etat =
    SGLEVCB2[wParam-SGLEV_C].Etat == 0 ? 1 : 0;
    Temps = FALSE;
    for (n=3; n<NSGLEVCB; n++)
        if (SGLEVCB2[n].Etat == 1)
        {
            Temps = TRUE;
            break;
        };
    if (Temps)
    {
        CheckDlgButton (hDlg, SGLEV_T,1);
        SGLEVCB2[2].Etat = 1;
    }
    else
    {
        CheckDlgButton (hDlg, SGLEV_T,0);
        SGLEVCB2[2].Etat = 0;
    };
    break;

case IDOK :
    for (n=0; n<NSGLEVCB; n++)
        SGLEVCB1[n].Etat = SGLEVCB2[n].Etat;

    EndDialog (hDlg, TRUE);
    break;

case IDCANCEL :
    EndDialog (hDlg, FALSE);
    break;

default :
    return FALSE;
}
break;
```

```

        default :
            return FALSE;
    }
    return TRUE;
}

```

**BOOL FAR PASCAL SGLOBDlgProc (hDlg, iMessage, wParam, lParam)**

```

HWND          hDlg;
unsigned      iMessage;
WORD          wParam;
LONG          lParam;
{
    static BOOL      Standard;
    static BOOL      STSGLPT;
    static BOOL      STSGLEV;
    short          n;
    static struct InitCB  SGLOBCB2[NSGLOBCB];
    static FARPROC  lpfnSGLPTDlgProc;
    static FARPROC  lpfnSGLEVDlgProc;

    switch (iMessage)
    {
        case WM_INITDIALOG :
            Standard = TRUE;
            STSGLPT = TRUE;
            STSGLEV = TRUE;

            for (n=0; n<NSGLOBCB; n++)
            {
                SGLOBCB2[n].nId = SGLOBCB[n].nId;
                SGLOBCB2[n].Etat = SGLOBCB[n].Etat;
            };

            for (n=0; n<NSGLPTCB; n++)
            {
                SGLPTCB1[n].nId = SGLPTCB[n].nId;
                SGLPTCB1[n].Etat = SGLPTCB[n].Etat;
            };

            for (n=0; n<NSGLEVCB; n++)
            {
                SGLEVCB1[n].nId = SGLEVCB[n].nId;
                SGLEVCB1[n].Etat = SGLEVCB[n].Etat;
            };

            for (n=0; n<2; n++)
                SendMessage (GetDlgItem(hDlg,SGLOBCB[n].nId),
                    BM_SETCHECK, SGLOBCB[n].Etat, 0L);

            for (n=2; n<NSGLOBCB; n++)
                switch (SGLOBCB[n].Etat)
                {
                    case 0 :
                        CheckDlgButton(hDlg,SGLOBCB[n].nId,0);

```

```
        break;

    case 1 :
        CheckDlgButton(hDlg,SGLOB_CB[n].nId,1);
        break;

    default :
        break;
};

lpfnSGLPTDlgProc = MakeProcInstance (SGLPTDlgProc, hInst);
lpfnSGLEVDlgProc = MakeProcInstance (SGLEVDlgProc, hInst);

break;

case WM_COMMAND :
    switch (wParam)
    {
    case SGLOB_FSTA :
        SendMessage (hDlg, WM_INITDIALOG, wParam,
                    lParam);
        break;

    case SGLOB_FPER :
        Standard = FALSE;
        CheckRadioButton (hDlg, SGLOB_FSTA,
                        SGLOB_FPER, wParam);
        SGLOB_CB2[0].Etat = 0;
        SGLOB_CB2[1].Etat = 1;
        for (n=2; n<NSGLOB_CB; n++)
            {
            CheckDlgButton(hDlg,SGLOB_CB[n].nId,0);
            SGLOB_CB2[n].Etat = 0;
            };

        for (n=0; n<NSGLPT_CB; n++)
            SGLPT_CB1[n].Etat = 0;

        for (n=0; n<NSGLEV_CB; n++)
            SGLEV_CB1[n].Etat = 0;

        break;

    case SGLOB_CR :
    case SGLOB_CM :
    case SGLOB_RD :
    case SGLOB_RF :
        CheckDlgButton (hDlg, wParam,
                        ! IsDlgButtonChecked (hDlg,wParam));

        SGLOB_CB2[wParam-SGLOB_CR+2].Etat =
        SGLOB_CB2[wParam-SGLOB_CR+2].Etat == 0 ? 1 : 0;

        Standard = TRUE;
        for (n=2; n<NSGLOB_CB; n++)
```

```
        if (SGLOB_CB2[n].Etat != SGLOB_CB[n].Etat)
        {
            Standard = FALSE;
            break;
        };
    if (Standard && STSGLPT && STSGLEV)
        CheckRadioButton (hDlg, SGLOB_FSTA,
            SGLOB_FPER,
            SGLOB_FSTA);
    else
        CheckRadioButton (hDlg, SGLOB_FSTA,
            SGLOB_FPER,
            SGLOB_FPER);

    break;

case SGLOB_PT :
    DialogBox (hInst, "SGLPT", hDlg, lpfnSGLPTDlgProc);

    STSGLPT = TRUE;
    for (n=0; n<NSGLPTCB; n++)
        if (SGLPTCB1[n].Etat != SGLPTCB[n].Etat)
        {
            STSGLPT = FALSE;
            break;
        };
    if (Standard && STSGLPT && STSGLEV)
        CheckRadioButton (hDlg, SGLOB_FSTA,
            SGLOB_FPER,
            SGLOB_FSTA);
    else
        CheckRadioButton (hDlg, SGLOB_FSTA,
            SGLOB_FPER,
            SGLOB_FPER);

    break;

case SGLOB_EV :
    DialogBox (hInst, "SGLEV", hDlg, lpfnSGLEVDlgProc);

    STSGLEV = TRUE;
    for (n=0; n<NSGLEVCB; n++)
        if (SGLEVCB1[n].Etat != SGLEVCB[n].Etat)
        {
            STSGLEV = FALSE;
            break;
        };
    if (Standard && STSGLPT && STSGLEV)
        CheckRadioButton (hDlg, SGLOB_FSTA,
            SGLOB_FPER,
            SGLOB_FSTA);
    else
        CheckRadioButton (hDlg, SGLOB_FSTA,
            SGLOB_FPER,
            SGLOB_FPER);

    break;
```

```
        case IDOK :
            EndDialog (hDlg, TRUE);
            break;

        case IDCANCEL :
            EndDialog (hDlg, FALSE);
            break;

        default :
            return FALSE;
    }
    break;

    default :
        return FALSE;
}
return TRUE;
}
```

**BOOL FAR PASCAL SCOMDlgProc (hDlg, iMessage, wParam, lParam)**

```
    HWND    hDlg;
    unsigned iMessage;
    WORD    wParam;
    LONG    lParam;
    {
    short    n;

    switch (iMessage)
    {
        case WM_INITDIALOG :
            for (n=0; n<NSCOMCB; n++)
                switch (SCOMCB[n].Etat)
                {
                    case 0 :
                        CheckDlgButton(hDlg,SCOMCB[n].nId,0);
                        break;

                    case 1 :
                        CheckDlgButton(hDlg,SCOMCB[n].nId,1);
                        break;

                    default :
                        break;
                }
            break;

        case WM_COMMAND :
            switch (wParam)
            {
                case SCOM_TITRE :
                case SCOM_DATE :
                case SCOM_TYPE :
                case SCOM_PERIODE :
```

```
case SCOM_NOM :
    CheckDlgButton (hDlg, wParam,
                    ! IsDlgButtonChecked (hDlg,wParam));
    break;

case IDOK :
    EndDialog (hDlg, TRUE);
    break;

case IDCANCEL :
    EndDialog (hDlg, FALSE);
    break;

default :
    return FALSE;
}
break;

default :
    return FALSE;
}
return TRUE;
}
```

## 5.7 MEANDRES.H

### Contenu du fichier

#define IDOK	1	#define PCOM_NOM	108
#define IDCANCEL	2	#define PCOM_CALENDRIER	109
#define IDM_NOUVEAU	1	#define PGLOBAL_FSTA	121
#define IDM_OUVRIER	2	#define PGLOBAL_FPER	122
#define IDM_SELECT	3	#define PGLOBAL_DEC	125
#define IDM_ENREG	4	#define PGLOBAL_ACT	126
#define IDM_ENREGSOUS	5	#define PGLOBAL_INT	127
#define IDM_FERMER	6	#define PGLOBAL_REA	128
#define IDM_APERCU	7	#define PGLOBAL_TER	129
#define IDM_MISEPG	8	#define PGLOBAL_VIE	130
#define IDM_IMPRIMER	9	#define PGLOBAL_PRO	131
#define IDM_QUITTER	10	#define PGLOBAL_DUREE	132
		#define PGLOBAL_ETAT	133
#define IDM_ANNULER	11	#define PGLDR_CN	152
#define IDM_COUPER	12	#define PGLDR_CY	153
#define IDM_COPIER	13	#define PGLDR_CX	154
#define IDM_COLLER	14	#define PGLDR_CE	155
#define IDM_EFFACER	15	#define PGLDR_TN	156
#define IDM_TSELECT	16	#define PGLDR_TY	157
#define IDM_PGLOBAL	17	#define PGLDR_TX	158
#define IDM_PPER	18	#define PGLDR_TE	159
#define IDM_PCHR	19	#define PGLDR_IN	160
#define IDM_PCOM	20	#define PGLDR_IY	161
#define IDM_PCAL	21	#define PGLDR_IX	162
#define IDM_PRES	22	#define PGLDR_IE	163
#define IDM_PDEF	23	#define PGLDR_PY	164
		#define PGLDR_RN	165
#define IDM_RGLOB	24	#define PGLDR_RY	166
#define IDM_RPER	25	#define PGLDR_RX	167
#define IDM_RCHR	26	#define PGLDR_RE	168
#define IDM_RCOM	27	#define PGLDR_SY	169
#define IDM_RPROC	28	#define PGLDR_VY	170
#define IDM_RDEF	29		
		#define PGLET_TN	188
#define IDM_SGLOB	30	#define PGLET_TY	189
#define IDM_SCOM	31	#define PGLET_TX	190
#define IDM_SDEF	32	#define PGLET_TE	191
		#define PGLET_IN	192
#define IDM_AIDE	33	#define PGLET_IY	193
#define IDM_INFO	34	#define PGLET_IX	194
#define IDM_REORG	35	#define PGLET_IE	195
#define IDM_MASQUER	36	#define PGLET_CN	196
#define IDM_AFFICHER	37	#define PGLET_CY	197
		#define PGLET_CX	198
#define PCOM_TITRE	102	#define PGLET_CE	199
#define PCOM_DATE	103		
#define PCOM_TYPE	105	#define PPER_FSTA	201
#define PCOM_PERIODE	106	#define PPER_FPER	202

#define PPER_DEC	206	#define RGLUT_PY	289
#define PPER_ACT	207	#define RGLUT_PX	290
#define PPER_INT	208	#define RGLUT_PE	291
#define PPER_REA	209		
#define PPER_TER	210	#define RPER_FSTA	301
#define PPER_DRT	211	#define RPER_FPER	302
#define PPER_DRI	212	#define RPER_ER	305
#define PPER_DRR	213	#define RPER_EL	306
#define PPER_RPV	214	#define RPER_ED	307
#define PPER_RPS	215	#define RPER_RV	308
#define PCHR_FSTA	221	#define RCHR_FSTA	316
#define PCHR_FPER	222	#define RCHR_FPER	317
#define PCHR_DEC	225	#define RCHR_ER	321
#define PCHR_ACT	226	#define RCHR_EL	322
#define PCHR_INT	227	#define RCHR_ED	323
#define PCHR_REA	228	#define RCHR_UC	324
#define PCHR_TER	229	#define RCHR_UQ	325
#define PCHR_RPV	230	#define RCHR_UY	326
#define PCHR_RPT	231	#define RCHR_UX	327
#define PCHR_RPP	232	#define RCHR_UF	328
#define PCHR_DUREE	233	#define RCHR_RC	330
#define PCHR_ETAT	234	#define RCHR_RU	331
		#define RCHR_RT	332
#define PCHDR_T	237		
#define PCHDR_R	238	#define RCOM_TITRE	344
#define PCHDR_I	239	#define RCOM_DATE	345
		#define RCOM_TYPE	346
#define PCHET_T	241	#define RCOM_PERIODE	347
#define PCHET_N	242	#define RCOM_NOM	348
#define PCHET_F	243	#define RCOM_CALENDRIER	349
#define PCHET_I	244		
#define PCHET_C	245	#define SGLOB_FSTA	351
		#define SGLOB_FPER	352
#define PCAL_P	248	#define SGLOB_CR	355
#define PCAL_R	249	#define SGLOB_CM	356
		#define SGLOB_RD	357
#define RGLOB_FSTA	251	#define SGLOB_RF	358
#define RGLOB_FPER	252	#define SGLOB_PT	359
#define RGLOB_CC	256	#define SGLOB_EV	360
#define RGLOB_CE	257		
#define RGLOB_ER	258	#define SGLPT_R	366
#define RGLOB_EL	259	#define SGLPT_T	367
#define RGLOB_ED	260	#define SGLPT_TN	368
#define RGLOB_RC	261	#define SGLPT_TY	369
#define RGLOB_RE	262	#define SGLPT_TX	370
#define RGLOB_UTIL	263	#define SGLPT_TE	371
#define RGLUT_CN	280	#define SGLEV_C	376
#define RGLUT_CY	281	#define SGLEV_P	377
#define RGLUT_CX	282	#define SGLEV_T	378
#define RGLUT_CE	283	#define SGLEV_TN	379
#define RGLUT_RN	284	#define SGLEV_TY	380
#define RGLUT_RY	285	#define SGLEV_TX	381
#define RGLUT_RX	286	#define SGLEV_TE	382
#define RGLUT_RE	287		
#define RGLUT_PN	288	#define SCOM_TITRE	394



#define SCOM_DATE	395	#define IMPR_C	405
#define SCOM_TYPE	396	#define IMPR_T	406
#define SCOM_PERIODE	397	#define IMPR_S	407
#define SCOM_NOM	398	#define IMPR_D	408
		#define IMPR_A	409

## 5.8 MEANDRES.RC

### Contenu du fichier

```
#include "style.h"
#include "meandres.h"
```

```
meandres ICON MEANDRES.ICO
```

```
meandres MENU
```

```
BEGIN
  POPUP "&Fichier"
  BEGIN
    MENUITEM "&Nouveau",          IDM_NOUVEAU
    MENUITEM "&Ouvrir...",        IDM_OUVRIR
    MENUITEM "&SÉlectionner un objet...", IDM_SELECT
    MENUITEM "&Enregistrer",      IDM_ENREG
    MENUITEM "En&registrer sous...", IDM_ENREGSOUS
    MENUITEM "&Fermer",           IDM_FERMER
    MENUITEM SEPARATOR
    MENUITEM "&Aperu avant impression...", IDM_APERCU
    MENUITEM "Effectuer la &mise en page...", IDM_MISEPG
    MENUITEM "&Imprimer...",      IDM_IMPRIMER
    MENUITEM SEPARATOR
    MENUITEM "&Quitter",          IDM_QUITTER
  END

  POPUP "&Edition"
  BEGIN
    MENUITEM "&AnnulerAlt+Ret arr", IDM_ANNULER
    MENUITEM SEPARATOR
    MENUITEM "&CouperMaj+Supp",    IDM_COUPER
    MENUITEM "Co&pierCtrl+Ins",    IDM_COPIER
    MENUITEM "C&ollerMaj+Ins",     IDM_COLLER
    MENUITEM "&EffacerSupp",      IDM_EFFACER
    MENUITEM SEPARATOR
    MENUITEM "&Tout sÉlectionner",  IDM_TSELECT
  END

  POPUP "&Processus"
  BEGIN
    MENUITEM "St. &globales...",    IDM_PGLOB
    MENUITEM "St. &pÉriodiques...",  IDM_PPER
    MENUITEM "St. &chronologiques...", IDM_PCHR
    MENUITEM SEPARATOR
    MENUITEM "Co&mmentaires...",    IDM_PCOM
    MENUITEM "Ca&lendrier...",      IDM_PCAL
    MENUITEM "Ressources &requis",  IDM_PRES
    MENUITEM SEPARATOR
  END
END
```

```

    MENUITEM "&DÉfinir fiche standard", IDM_PDEF
END

POPUP "&Ressources"
BEGIN
    MENUITEM "St. &globales...",          IDM_RGLOB
    MENUITEM "St. &pÉriodiques...",       IDM_RPER
    MENUITEM "St. &chronologiques...",    IDM_RCHR
    MENUITEM SEPARATOR
    MENUITEM "Co&mmentaires...",         IDM_RCOM
    MENUITEM "Processus &requÉrants",     IDM_RPROC
    MENUITEM SEPARATOR
    MENUITEM "&DÉfinir fiche standard",   IDM_RDEF
END

POPUP "Pt. de &synchr."
BEGIN
    MENUITEM "St. &globales...",          IDM_SGLOB
    MENUITEM SEPARATOR
    MENUITEM "Co&mmentaires...",         IDM_SCOM
    MENUITEM SEPARATOR
    MENUITEM "&DÉfinir fiche standard",   IDM_SDEF
END

POPUP "E&cran"
BEGIN
    MENUITEM "&Aide...\tF1",              IDM_AIDE
    MENUITEM "&Informations",            IDM_INFO
    MENUITEM SEPARATOR
    MENUITEM "&RÉorganiser",             IDM_REORG
    MENUITEM "Masquer la &fiche",         IDM_MASQUER
    MENUITEM "Afficher les &rÉgles\tF2",  IDM_AFFICHER
END
END

meandres ACCELERATORS
BEGIN
    VK_DELETE, IDM_COUPER,    VIRTKEY, SHIFT
    VK_INSERT, IDM_COPIER,    VIRTKEY, CONTROL
    VK_INSERT, IDM_COLLER,    VIRTKEY, SHIFT
    VK_DELETE, IDM_EFFACER,   VIRTKEY
    VK_F1,      IDM_AIDE,      VIRTKEY
    VK_F2,      IDM_AFFICHER,  VIRTKEY
END

rcinclude meandres.dlg

```

## 5.9 MEANDRES.DLG

### Liste des boîtes de dialogue

IMPR.....	111
PCAL .....	110
PCHDR.....	97
PCHET.....	98
PCHR.....	104
PCOM.....	96
PGLDR.....	93
PGLET.....	92
PGLOB.....	106
PPER.....	95
RCHR.....	100
RCOM.....	109
RGLOB.....	107
RGLUT.....	99
RPER.....	101
SCOM.....	104
SGLEV.....	102
SGLOB.....	108
SGLPT.....	103

### Contenu du fichier

PGLET DIALOG LOADONCALL MOVEABLE DISCARDABLE 16, 13, 245, 92

STYLE WS\_DLGFRAME | WS\_POPUP

BEGIN

```

CONTROL "Etat&s", 180, "static", SS_LEFT | WS_GROUP | WS_CHILD, 10,
    10, 24, 10
CONTROL "Mi&n.", 181, "static", SS_LEFT | WS_GROUP | WS_CHILD, 150,
    30, 16, 8
CONTROL "Mo&y.", 182, "static", SS_LEFT | WS_GROUP | WS_CHILD, 170,
    30, 16, 8
CONTROL "Ma&x.", 184, "static", SS_LEFT | WS_GROUP | WS_CHILD, 190,
    30, 16, 8
CONTROL "&Ecart", 104, "static", SS_LEFT | WS_GROUP | WS_CHILD, 208,
    30, 22, 8
CONTROL "Nombre de proc. en a&ttente d'act.", 185, "static", SS_LEFT |
    WS_GROUP | WS_CHILD, 14, 40, 136, 8
CONTROL "Nombre de proc. &interrompus", 186, "static", SS_LEFT |
    WS_GROUP | WS_CHILD, 14, 54, 116, 10
CONTROL "Nombre de proc. en a&ctivitÈ", 187, "static", SS_LEFT |
    WS_GROUP | WS_CHILD, 14, 68, 116, 8
CONTROL "", PGLET_TN, "button", BS_CHECKBOX | WS_GROUP |
    WS_TABSTOP | WS_CHILD, 154, 40, 8, 12
CONTROL "", PGLET_TY, "button", BS_CHECKBOX | WS_TABSTOP |
    WS_CHILD, 174, 40, 8, 12
CONTROL "", PGLET_TX, "button", BS_CHECKBOX | WS_TABSTOP |
    WS_CHILD, 194, 40, 8, 12
CONTROL "", PGLET_TE, "button", BS_CHECKBOX | WS_TABSTOP |
    WS_CHILD, 214, 40, 8, 12
CONTROL "", PGLET_IN, "button", BS_CHECKBOX | WS_TABSTOP |

```

```

WS_CHILD, 154, 54, 8, 12
CONTROL "", PGLET_IY, "button", BS_CHECKBOX | WS_TABSTOP |
WS_CHILD, 174, 54, 8, 12
CONTROL "", PGLET_IX, "button", BS_CHECKBOX | WS_TABSTOP |
WS_CHILD, 194, 54, 8, 12
CONTROL "", PGLET_IE, "button", BS_CHECKBOX | WS_TABSTOP |
WS_CHILD, 214, 54, 8, 12
CONTROL "", PGLET_CN, "button", BS_CHECKBOX | WS_TABSTOP |
WS_CHILD, 154, 68, 8, 12
CONTROL "", PGLET_CY, "button", BS_CHECKBOX | WS_TABSTOP |
WS_CHILD, 174, 68, 8, 12
CONTROL "", PGLET_CX, "button", BS_CHECKBOX | WS_TABSTOP |
WS_CHILD, 194, 68, 8, 12
CONTROL "", PGLET_CE, "button", BS_CHECKBOX | WS_TABSTOP |
WS_CHILD, 214, 68, 8, 12
CONTROL "&Ok", IDOK, "button", BS_DEFPUSHBUTTON | WS_GROUP |
WS_TABSTOP | WS_CHILD, 146, 10, 36, 14
CONTROL "&Annuler", IDCANCEL, "button", BS_PUSHBUTTON |
WS_GROUP | WS_TABSTOP | WS_CHILD, 190, 10, 36, 14
END

```

Etats	Ok		Annuler	
	Min.	Moy.	Max.	Ecart
Nombre de proc. en attente d'act.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Nombre de proc. interrompus	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Nombre de proc. en activité	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Illustration 13 : Boîte de dialogue PGLET

```

PGLDR DIALOG LOADONCALL MOVEABLE DISCARDABLE 16, 17, 221, 153
STYLE WS_DLGFRAME | WS_POPUP
BEGIN

```

```

CONTROL "&Durées", 140, "static", SS_LEFT | WS_GROUP | WS_CHILD,
10, 10, 28, 10
CONTROL "Durée d'a&ctivité", 145, "static", SS_LEFT | WS_GROUP |
WS_CHILD, 14, 40, 68, 8
CONTROL "Mi&n.", 141, "static", SS_LEFT | WS_GROUP | WS_CHILD, 116,
30, 16, 8
CONTROL "Mo&y.", 142, "static", SS_LEFT | WS_GROUP | WS_CHILD, 136,
30, 16, 8
CONTROL "Ma&x.", 143, "static", SS_LEFT | WS_GROUP | WS_CHILD, 156,
30, 16, 8
CONTROL "&Ecart", 144, "static", SS_LEFT | WS_GROUP | WS_CHILD, 174,
30, 25, 9
CONTROL "Durée d'a&ttente d'act.", 146, "static", SS_LEFT | WS_GROUP |
WS_CHILD, 14, 54, 90, 8
CONTROL "Durée d'&interruption", 147, "static", SS_LEFT | WS_GROUP |
WS_CHILD, 14, 68, 86, 8

```

CONTROL "Durée moy. d'inter./&proc.", 148, "static", SS\_LEFT |  
 WS\_GROUP | WS\_CHILD, 14, 82, 108, 8  
 CONTROL "Durée de &repos", 149, "static", SS\_LEFT | WS\_GROUP |  
 WS\_CHILD, 14, 96, 62, 8  
 CONTROL "Durée moy. de repo&s/proc.", 150, "static", SS\_LEFT |  
 WS\_GROUP | WS\_CHILD, 14, 110, 104, 8  
 CONTROL "Durée moy. de &vie", 151, "static", SS\_LEFT | WS\_GROUP |  
 WS\_CHILD, 14, 124, 72, 8  
 CONTROL "", PGLDR\_CN, "button", BS\_CHECKBOX | WS\_GROUP |  
 WS\_TABSTOP | WS\_CHILD, 120, 40, 10, 12  
 CONTROL "", PGLDR\_CY, "button", BS\_CHECKBOX | WS\_TABSTOP |  
 WS\_CHILD, 140, 40, 8, 12  
 CONTROL "", PGLDR\_CX, "button", BS\_CHECKBOX | WS\_TABSTOP |  
 WS\_CHILD, 160, 40, 8, 12  
 CONTROL "", PGLDR\_CE, "button", BS\_CHECKBOX | WS\_TABSTOP |  
 WS\_CHILD, 180, 40, 8, 12  
 CONTROL "", PGLDR\_TN, "button", BS\_CHECKBOX | WS\_TABSTOP |  
 WS\_CHILD, 120, 54, 8, 12  
 CONTROL "", PGLDR\_TY, "button", BS\_CHECKBOX | WS\_TABSTOP |  
 WS\_CHILD, 140, 54, 8, 12  
 CONTROL "", PGLDR\_TX, "button", BS\_CHECKBOX | WS\_TABSTOP |  
 WS\_CHILD, 160, 54, 8, 12  
 CONTROL "", PGLDR\_TE, "button", BS\_CHECKBOX | WS\_TABSTOP |  
 WS\_CHILD, 180, 54, 8, 12  
 CONTROL "", PGLDR\_IN, "button", BS\_CHECKBOX | WS\_TABSTOP |  
 WS\_CHILD, 120, 68, 8, 12  
 CONTROL "", PGLDR\_IY, "button", BS\_CHECKBOX | WS\_TABSTOP |  
 WS\_CHILD, 140, 68, 8, 12  
 CONTROL "", PGLDR\_IX, "button", BS\_CHECKBOX | WS\_TABSTOP |  
 WS\_CHILD, 160, 68, 8, 12  
 CONTROL "", PGLDR\_IE, "button", BS\_CHECKBOX | WS\_TABSTOP |  
 WS\_CHILD, 180, 68, 10, 12  
 CONTROL "", PGLDR\_PY, "button", BS\_CHECKBOX | WS\_TABSTOP |  
 WS\_CHILD, 140, 82, 8, 12  
 CONTROL "", PGLDR\_RN, "button", BS\_CHECKBOX | WS\_TABSTOP |  
 WS\_CHILD, 120, 96, 8, 12  
 CONTROL "", PGLDR\_RY, "button", BS\_CHECKBOX | WS\_TABSTOP |  
 WS\_CHILD, 140, 96, 8, 12  
 CONTROL "", PGLDR\_RX, "button", BS\_CHECKBOX | WS\_TABSTOP |  
 WS\_CHILD, 160, 96, 8, 12  
 CONTROL "", PGLDR\_RE, "button", BS\_CHECKBOX | WS\_TABSTOP |  
 WS\_CHILD, 180, 96, 9, 12  
 CONTROL "", PGLDR\_SY, "button", BS\_CHECKBOX | WS\_TABSTOP |  
 WS\_CHILD, 140, 110, 8, 12  
 CONTROL "", PGLDR\_VY, "button", BS\_CHECKBOX | WS\_TABSTOP |  
 WS\_CHILD, 140, 124, 10, 12  
 CONTROL "&Ok", IDOK, "button", BS\_DEFPUSHBUTTON | WS\_GROUP |  
 WS\_TABSTOP | WS\_CHILD, 114, 10, 36, 14  
 CONTROL "&Annuler", IDCANCEL, "button", BS\_PUSHBUTTON |  
 WS\_GROUP | WS\_TABSTOP | WS\_CHILD, 164, 10, 36, 14

END

Durées	Ok		Annuler	
	Min.	Moy.	Max.	Ecart
Durée d'activité	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Durée d'attente d'act.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Durée d'interruption	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Durée moy. d'inter./proc.		<input type="checkbox"/>		
Durée de repos	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Durée moy. de repos/proc.		<input type="checkbox"/>		
Durée moy. de vie		<input type="checkbox"/>		

Illustration 14 : Boîte de dialogue PGLDR

```

PPER DIALOG LOADONCALL MOVEABLE DISCARDABLE 15, 12, 183, 198
STYLE WS_DLGFRAME | WS_POPUP
BEGIN
CONTROL "Statistiques pÉriodiques", 200, "static", SS_LEFT | WS_GROUP |
WS_CHILD, 10, 10, 102, 8
CONTROL "Fiche &standard", PPER_FSTA, "button", BS_RADIOBUTTON |
WS_GROUP | WS_TABSTOP | WS_CHILD, 14, 24, 66, 12
CONTROL "Fiche &personnalisÉe", PPER_FPER, "button",
BS_RADIOBUTTON | WS_CHILD, 14, 40, 90, 12
CONTROL "EvÈnements", 203, "button", BS_GROUPBOX | WS_GROUP |
WS_CHILD, 24, 50, 144, 44
CONTROL "DÉc&lancements", PPER_DEC, "button", BS_CHECKBOX |
WS_GROUP | WS_TABSTOP | WS_CHILD, 28, 60, 68, 12
CONTROL "A&ctivations", PPER_ACT, "button", BS_CHECKBOX |
WS_CHILD, 100, 60, 58, 12
CONTROL "&Interruptions", PPER_INT, "button", BS_CHECKBOX |
WS_CHILD, 28, 70, 66, 12
CONTROL "&RÉactivations", PPER_REA, "button", BS_CHECKBOX |
WS_CHILD, 100, 70, 66, 12
CONTROL "&Terminaisons", PPER_TER, "button", BS_CHECKBOX |
WS_CHILD, 28, 80, 62, 12
CONTROL "DurÉes", 204, "button", BS_GROUPBOX | WS_GROUP |
WS_CHILD, 24, 98, 144, 44
CONTROL "DurÉE moy. d'a&ttente d'act.", PPER_DRT, "button",
BS_CHECKBOX | WS_GROUP | WS_TABSTOP | WS_CHILD, 28,
108, 122, 12
CONTROL "DurÉE moy. d'&interruption", PPER_DRI, "button",
BS_CHECKBOX | WS_CHILD, 28, 118, 112, 12
CONTROL "DurÉE moy. de &repos", PPER_DRR, "button", BS_CHECKBOX |
WS_CHILD, 28, 128, 92, 12
CONTROL "RÉpartitions", 205, "button", BS_GROUPBOX | WS_GROUP |
WS_CHILD, 24, 148, 144, 34

```

```

CONTROL "Nombre d'ÉvÈnements", PPER_RPV, "button", BS_CHECKBOX
| WS_GROUP | WS_TABSTOP | WS_CHILD, 28, 158, 90, 12
CONTROL "DurÈe moy. d'attente et de repo&s", PPER_RPS, "button",
BS_CHECKBOX | WS_CHILD, 28, 168, 138, 12
CONTROL "&Ok", IDOK, "button", BS_DEF PUSHBUTTON | WS_GROUP |
WS_TABSTOP | WS_CHILD, 132, 10, 36, 14
CONTROL "&Annuler", IDCANCEL, "button", BS_PUSHBUTTON |
WS_GROUP | WS_TABSTOP | WS_CHILD, 132, 30, 34, 14

```

END

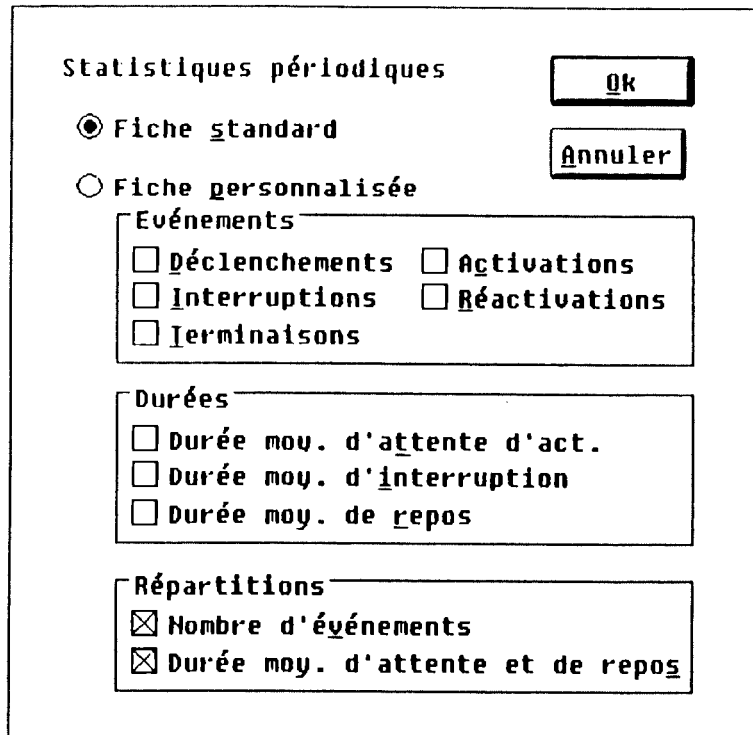


Illustration 15 : Boîte de dialogue PPER

```

PCOM DIALOG LOADONCALL MOVEABLE DISCARDABLE 15, 15, 171, 158
STYLE WS_DLGFRAME | WS_POPUP
BEGIN
CONTROL "Commentaires", 100, "static", SS_LEFT | WS_CHILD, 10, 10, 50,
10
CONTROL "Simulation", 101, "button", BS_GROUPBOX | WS_GROUP |
WS_CHILD, 20, 24, 86, 34
CONTROL "&Titre", PCOM_TITRE, "button", BS_CHECKBOX | WS_GROUP
| WS_TABSTOP | WS_CHILD, 24, 33, 34, 12
CONTROL "&Date d'exÉcution", PCOM_DATE, "button", BS_CHECKBOX |
WS_CHILD, 24, 44, 76, 12
CONTROL "Mesures statistiques", 104, "button", BS_GROUPBOX |
WS_GROUP | WS_CHILD, 20, 68, 126, 34
CONTROL "T&ype", PCOM_TYPE, "button", BS_CHECKBOX | WS_GROUP
| WS_TABSTOP | WS_CHILD, 24, 78, 30, 12
CONTROL "&PÉriode de prise en compte", PCOM_PÉRIODE, "button",
BS_CHECKBOX | WS_CHILD, 24, 88, 116, 12
CONTROL "Processus", 107, "button", BS_GROUPBOX | WS_GROUP |

```

```

WS_CHILD, 20, 112, 58, 34
CONTROL "&Nom", PCOM_NOM, "button", BS_CHECKBOX | WS_GROUP
| WS_TABSTOP | WS_CHILD, 24, 122, 28, 12
CONTROL "&Calendrier", PCOM_CALENDRIER, "button", BS_CHECKBOX |
WS_CHILD, 24, 132, 52, 12
CONTROL "&Ok", IDOK, "button", BS_DEFPUSHBUTTON | WS_GROUP |
WS_TABSTOP | WS_CHILD, 120, 6, 36, 14
CONTROL "&Annuler", IDCANCEL, "button", BS_PUSHBUTTON |
WS_GROUP | WS_TABSTOP | WS_CHILD, 120, 26, 36, 14
END

```

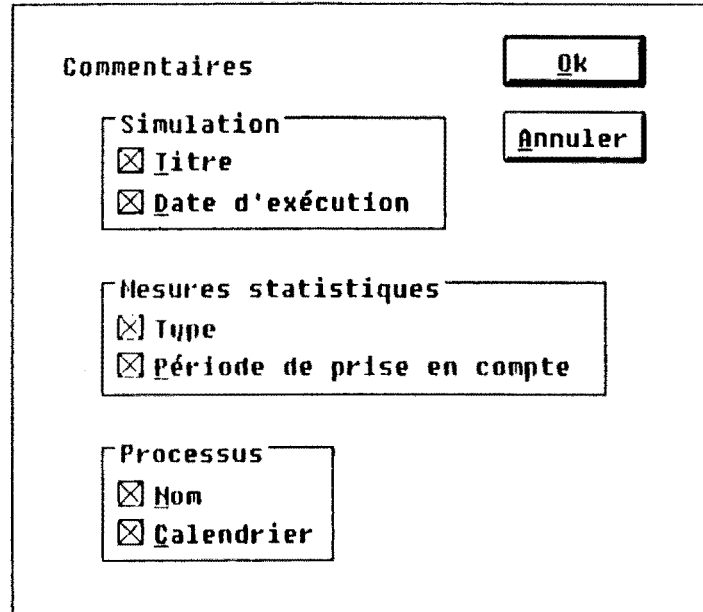


Illustration 16 : Boîte de dialogue PCOM

```

PCHDR DIALOG LOADONCALL MOVEABLE DISCARDABLE 16, 12, 166, 81
STYLE WS_DLGFRAME | WS_POPUP
BEGIN
CONTROL "Durées", 236, "static", SS_LEFT | WS_GROUP | WS_CHILD,
10, 10, 26, 8
CONTROL "Durée moy. d'attente d'act.", PCHDR_T, "button",
BS_CHECKBOX | WS_GROUP | WS_TABSTOP | WS_CHILD, 14,
30, 122, 12
CONTROL "Durée moy. de repos", PCHDR_R, "button", BS_CHECKBOX |
WS_TABSTOP | WS_CHILD, 14, 44, 90, 12
CONTROL "Durée moy. d'interruption", PCHDR_I, "button",
BS_CHECKBOX | WS_TABSTOP | WS_CHILD, 14, 58, 114, 12
CONTROL "&Ok", IDOK, "button", BS_DEFPUSHBUTTON | WS_GROUP |
WS_TABSTOP | WS_CHILD, 68, 8, 36, 14
CONTROL "&Annuler", IDCANCEL, "button", BS_PUSHBUTTON |
WS_GROUP | WS_TABSTOP | WS_CHILD, 112, 8, 36, 14
END

```



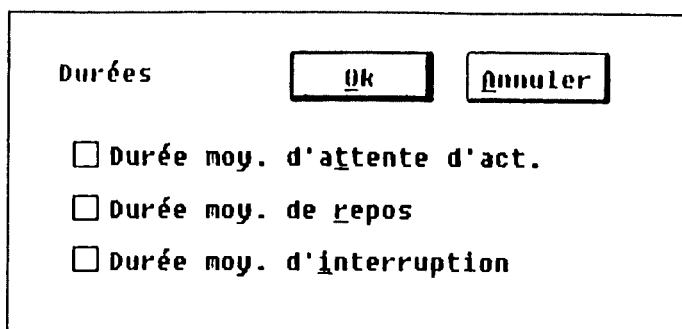


Illustration 17 : Boîte de dialogue PCHDR

```

PCHET DIALOG LOADONCALL MOVEABLE DISCARDABLE 16, 12, 189, 120
STYLE WS_DLGFRAME | WS_POPUP
BEGIN
CONTROL "Etats", 240, "static", SS_LEFT | WS_GROUP | WS_CHILD, 10,
    10, 22, 8
CONTROL "Nombre de proc. en a&ttente d'act.", PCHET_T, "button",
    BS_CHECKBOX | WS_GROUP | WS_TABSTOP | WS_CHILD, 14,
    30, 144, 12
CONTROL "Nombre mi&n. de proc en attente d'act.", PCHET_N, "button",
    BS_CHECKBOX | WS_TABSTOP | WS_CHILD, 14, 44, 162, 12
CONTROL "Nombre de proc. en attente d'act.", PCHET_F, "button",
    BS_CHECKBOX | WS_TABSTOP | WS_CHILD, 14, 58, 146, 12
CONTROL "Nombre de proc. &interrompus", PCHET_I, "button",
    BS_CHECKBOX | WS_TABSTOP | WS_CHILD, 14, 84, 122, 12
CONTROL "Nombre de proc. en a&ctivité", PCHET_C, "button",
    BS_CHECKBOX | WS_TABSTOP | WS_CHILD, 14, 98, 125, 12
CONTROL "‡ la &fin de la cellule", 246, "static", SS_LEFT | WS_GROUP |
    WS_CHILD, 24, 70, 92, 8
CONTROL "&Ok", IDOK, "button", BS_DEFPUSHBUTTON | WS_GROUP |
    WS_TABSTOP | WS_CHILD, 88, 8, 36, 14
CONTROL "&Annuler", IDCANCEL, "button", BS_PUSHBUTTON |
    WS_GROUP | WS_TABSTOP | WS_CHILD, 136, 8, 36, 14
END

```

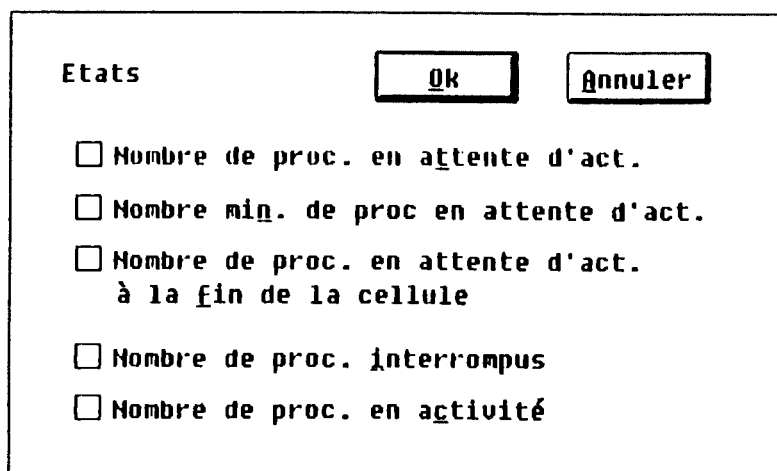


Illustration 18 : Boîte de dialogue PCHET

```

RGLUT DIALOG LOADONCALL MOVEABLE DISCARDABLE 15, 14, 219, 107
STYLE WS_DLGFRAME | WS_POPUP
BEGIN
    CONTROL "&Utilisation", 270, "static", SS_LEFT | WS_GROUP | WS_CHILD,
        10, 10, 48, 8
    CONTROL "Mi&n.", 271, "static", SS_LEFT | WS_GROUP | WS_CHILD, 124,
        30, 16, 8
    CONTROL "Mo&y.", 272, "static", SS_LEFT | WS_GROUP | WS_CHILD,
        144, 30, 16, 8
    CONTROL "Ma&x.", 273, "static", SS_LEFT | WS_GROUP | WS_CHILD,
        164, 30, 16, 8
    CONTROL "&Ecart", 274, "static", SS_LEFT | WS_GROUP | WS_CHILD, 182,
        30, 24, 8
    CONTROL "&CapacitÈ utilisÈe", 275, "static", SS_LEFT | WS_GROUP |
        WS_CHILD, 14, 40, 74, 8
    CONTROL "Nombre de &requêtes", 276, "static", SS_LEFT | WS_GROUP |
        WS_CHILD, 14, 60, 76, 8
    CONTROL "satisfaites simultanÈment", 277, "static", SS_LEFT | WS_CHILD,
        22, 68, 100, 8
    CONTROL "Nombre de &proc. en attente", 278, "static", SS_LEFT |
        WS_GROUP | WS_CHILD, 14, 80, 106, 8
    CONTROL "devant la ressource", 279, "static", SS_LEFT | WS_CHILD, 22, 88,
        80, 8
    CONTROL "", RGLUT_CN, "button", BS_CHECKBOX | WS_GROUP |
        WS_TABSTOP | WS_CHILD, 128, 40, 10, 14
    CONTROL "", RGLUT_CY, "button", BS_CHECKBOX | WS_TABSTOP |
        WS_CHILD, 148, 40, 8, 12
    CONTROL "", RGLUT_CX, "button", BS_CHECKBOX | WS_TABSTOP |
        WS_CHILD, 168, 40, 8, 12
    CONTROL "", RGLUT_CE, "button", BS_CHECKBOX | WS_TABSTOP |
        WS_CHILD, 188, 40, 8, 12
    CONTROL "", RGLUT_RN, "button", BS_CHECKBOX | WS_TABSTOP |
        WS_CHILD, 128, 60, 8, 12
    CONTROL "", RGLUT_RY, "button", BS_CHECKBOX | WS_TABSTOP |
        WS_CHILD, 148, 60, 8, 12
    CONTROL "", RGLUT_RX, "button", BS_CHECKBOX | WS_TABSTOP |
        WS_CHILD, 168, 60, 10, 12
    CONTROL "", RGLUT_RE, "button", BS_CHECKBOX | WS_TABSTOP |
        WS_CHILD, 188, 60, 8, 12
    CONTROL "", RGLUT_PN, "button", BS_CHECKBOX | WS_TABSTOP |
        WS_CHILD, 128, 80, 8, 12
    CONTROL "", RGLUT_PY, "button", BS_CHECKBOX | WS_TABSTOP |
        WS_CHILD, 148, 80, 8, 12
    CONTROL "", RGLUT_PX, "button", BS_CHECKBOX | WS_TABSTOP |
        WS_CHILD, 168, 80, 8, 12
    CONTROL "", RGLUT_PE, "button", BS_CHECKBOX | WS_TABSTOP |
        WS_CHILD, 188, 80, 8, 12
    CONTROL "&Ok", IDOK, "button", BS_DEFPUSHBUTTON | WS_GROUP |
        WS_TABSTOP | WS_CHILD, 122, 8, 36, 14
    CONTROL "&Annuler", IDCANCEL, "button", BS_PUSHBUTTON |
        WS_GROUP | WS_TABSTOP | WS_CHILD, 168, 8, 34, 14
END

```

Utilisation	Ok		Annuler	
	Min.	Moy.	Max.	Ecart
Capacité utilisée	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Nombre de requêtes satisfaites simultanément	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Nombre de proc. en attente devant la ressource	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Illustration 19 : Boîte de dialogue RGLUT

RCHR DIALOG LOADONCALL MOVEABLE DISCARDABLE 15, 13, 266, 181  
 STYLE WS\_DLGFRAME | WS\_POPUP  
 BEGIN

```

CONTROL "Statistiques chronologiques", 315, "static", SS_LEFT |
  WS_GROUP | WS_CHILD, 10, 10, 114, 8
CONTROL "Fiche &standard", RCHR_FSTA, "button", BS_RADIOBUTTON |
  WS_GROUP | WS_TABSTOP | WS_CHILD, 14, 24, 68, 12
CONTROL "Fiche &personnalisÉe", RCHR_FPER, "button",
  BS_RADIOBUTTON | WS_CHILD, 14, 40, 90, 12
CONTROL "EvÈnements", 318, "button", BS_GROUPBOX | WS_GROUP |
  WS_CHILD, 24, 50, 76, 44
CONTROL "&Requêtes", RCHR_ER, "button", BS_CHECKBOX |
  WS_GROUP | WS_TABSTOP | WS_CHILD, 28, 59, 46, 14
CONTROL "A&llocations", RCHR_EL, "button", BS_CHECKBOX |
  WS_CHILD, 28, 70, 56, 12
CONTROL "&DÈsallocations", RCHR_ED, "button", BS_CHECKBOX |
  WS_CHILD, 28, 80, 70, 12
CONTROL "Utilisation", 319, "button", BS_GROUPBOX | WS_GROUP |
  WS_CHILD, 24, 100, 230, 70
CONTROL "&CapacitÈ moyenne", RCHR_UC, "button", BS_CHECKBOX |
  WS_GROUP | WS_TABSTOP | WS_CHILD, 28, 110, 76, 12
CONTROL "Nombre moy. de re&quêtes satisfaites simultanÈment", RCHR_UQ,
  "button", BS_CHECKBOX | WS_CHILD, 28, 120, 210, 12
CONTROL "Nombre mo&y. de proc. en attente devant la ressource",
  RCHR_UY, "button", BS_CHECKBOX | WS_CHILD, 28, 130,
  220, 12
CONTROL "Nombre ma&x. de proc. en attente devant la ressource",
  RCHR_UX, "button", BS_CHECKBOX | WS_CHILD, 28, 140,
  218, 12
CONTROL "Nombre de proc. en attente devant la ressource", RCHR_UF,
  "button", BS_CHECKBOX | WS_CHILD, 28, 150, 204, 12
CONTROL "‡ la &fin de la cellule d'observation", 329, "static", SS_LEFT |
  WS_GROUP | WS_CHILD, 66, 160, 154, 8
CONTROL "RÈpartitions", 320, "button", BS_GROUPBOX | WS_GROUP |
  WS_CHILD, 110, 50, 144, 44
CONTROL "Nombre d'o&cc. des EvÈnements", RCHR_RC, "button",
  BS_CHECKBOX | WS_GROUP | WS_TABSTOP | WS_CHILD,
  
```

```

114, 60, 120, 12
CONTROL "Taux moyen d'utilisation", RCHR_RU, "button",
BS_CHECKBOX | WS_CHILD, 114, 70, 112, 12
CONTROL "Nombre moy. de proc. en attente", RCHR_RT, "button",
BS_CHECKBOX | WS_CHILD, 114, 80, 138, 12
CONTROL "&Ok", IDOK, "button", BS_DEFPUSHBUTTON | WS_GROUP |
WS_TABSTOP | WS_CHILD, 218, 8, 36, 14
CONTROL "&Annuler", IDCANCEL, "button", BS_PUSHBUTTON |
WS_GROUP | WS_TABSTOP | WS_CHILD, 218, 28, 36, 14
END

```

Illustration 20 : Boite de dialogue RCHR

```

RPER DIALOG LOADONCALL MOVEABLE DISCARDABLE 15, 14, 184, 136
STYLE WS_DLDFRAME | WS_POPUP
BEGIN
CONTROL "Statistiques pÉriodiques", 300, "static", SS_LEFT | WS_GROUP |
WS_CHILD, 10, 10, 102, 8
CONTROL "Fiche &standard", RPER_FSTA, "button", BS_RADIOBUTTON |
WS_GROUP | WS_TABSTOP | WS_CHILD, 14, 24, 68, 12
CONTROL "Fiche &personnalisÉe", RPER_FPER, "button",
BS_RADIOBUTTON | WS_CHILD, 14, 40, 94, 12
CONTROL "EvÉnements", 303, "button", BS_GROUPBOX | WS_GROUP |
WS_CHILD, 24, 50, 104, 44
CONTROL "&RequÉtes", RPER_ER, "button", BS_CHECKBOX |
WS_GROUP | WS_TABSTOP | WS_CHILD, 28, 60, 46, 12
CONTROL "A&llocations", RPER_EL, "button", BS_CHECKBOX |
WS_CHILD, 28, 70, 56, 12
CONTROL "&DÉsallocations", RPER_ED, "button", BS_CHECKBOX |
WS_CHILD, 28, 80, 70, 12
CONTROL "RÉpartitions", 304, "button", BS_GROUPBOX | WS_GROUP |
WS_CHILD, 24, 100, 104, 24

```

```

CONTROL "Nombre d'ÉvÈnements", RPER_RV, "button",
      BS_CHECKBOX | WS_GROUP | WS_TABSTOP | WS_CHILD,
      28, 110, 88, 12
CONTROL "&Ok", IDOK, "button", BS_DEFPUSHBUTTON | WS_GROUP |
      WS_TABSTOP | WS_CHILD, 132, 8, 36, 14
CONTROL "&Annuler", IDCANCEL, "button", BS_PUSHBUTTON |
      WS_GROUP | WS_TABSTOP | WS_CHILD, 132, 28, 36, 14
END

```

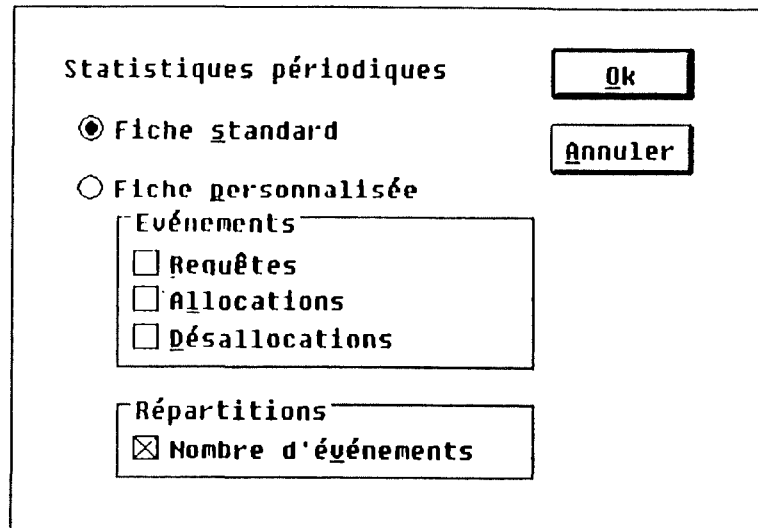


Illustration 21 : Boîte de dialogue RPER

```

SGLEV DIALOG LOADONCALL MOVEABLE DISCARDABLE 15, 14, 181, 112
STYLE WS_DLGFRAME | WS_POPUP
BEGIN
CONTROL "EvÈnements contributeurs", 375, "static", SS_LEFT | WS_GROUP |
      WS_CHILD, 10, 10, 96, 8
CONTROL "Nombre de &contributions", SGLEV_C, "button",
      BS_CHECKBOX | WS_GROUP | WS_TABSTOP | WS_CHILD,
      14, 24, 106, 12
CONTROL "Nombre de &participations", SGLEV_P, "button",
      BS_CHECKBOX | WS_GROUP | WS_TABSTOP | WS_CHILD,
      14, 40, 110, 12
CONTROL "&Temps de participation", SGLEV_T, "button", BS_CHECKBOX |
      WS_GROUP | WS_TABSTOP | WS_CHILD, 14, 56, 106, 12
CONTROL "Mi&nimum", SGLEV_TN, "button", BS_CHECKBOX |
      WS_GROUP | WS_CHILD, 40, 66, 46, 12
CONTROL "Mo&yenne", SGLEV_TY, "button", BS_CHECKBOX |
      WS_CHILD, 40, 76, 48, 12
CONTROL "Ma&ximum", SGLEV_TX, "button", BS_CHECKBOX |
      WS_CHILD, 40, 86, 42, 12
CONTROL "&Écart type", SGLEV_TE, "button", BS_CHECKBOX |
      WS_CHILD, 40, 96, 56, 12
CONTROL "&Ok", IDOK, "button", BS_DEFPUSHBUTTON | WS_GROUP |
      WS_TABSTOP | WS_CHILD, 136, 8, 36, 14
CONTROL "&Annuler", IDCANCEL, "button", BS_PUSHBUTTON |
      WS_GROUP | WS_TABSTOP | WS_CHILD, 136, 28, 36, 14
END

```

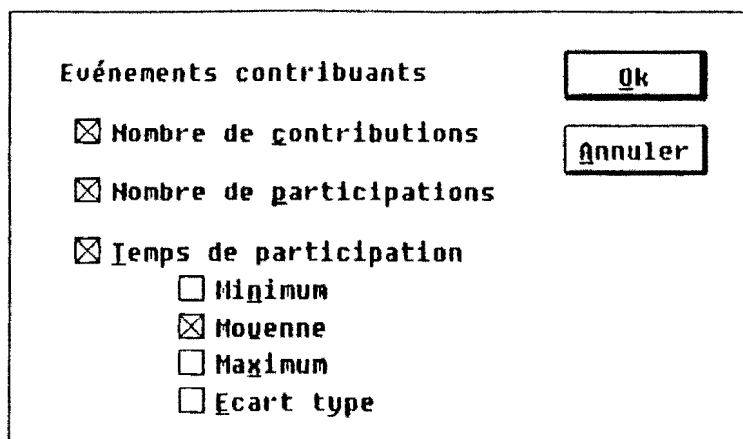


Illustration 22 : Boîte de dialogue SGLEV

```
SGLPT DIALOG LOADONCALL MOVEABLE DISCARDABLE 16, 13, 178, 100
STYLE WS_DLGFRAME | WS_POPUP
```

```
BEGIN
```

```
CONTROL "Point de synchronisation", 365, "static", SS_LEFT | WS_GROUP |
WS_CHILD, 10, 10, 100, 8
CONTROL "Nombre de &rÉalisations", SGLPT_R, "button", BS_CHECKBOX |
WS_GROUP | WS_TABSTOP | WS_CHILD, 14, 24, 100, 12
CONTROL "&Temps de rÉalisation", SGLPT_T, "button", BS_CHECKBOX |
WS_GROUP | WS_TABSTOP | WS_CHILD, 14, 40, 94, 12
CONTROL "Mi&nimum", SGLPT_TN, "button", BS_CHECKBOX |
WS_GROUP | WS_CHILD, 34, 50, 46, 12
CONTROL "Mo&yenne", SGLPT_TY, "button", BS_CHECKBOX |
WS_CHILD, 34, 60, 40, 12
CONTROL "Ma&ximum", SGLPT_TX, "button", BS_CHECKBOX |
WS_CHILD, 34, 70, 42, 12
CONTROL "&Ecart type", SGLPT_TE, "button", BS_CHECKBOX |
WS_CHILD, 34, 80, 56, 12
CONTROL "&Ok", IDOK, "button", BS_DEFPUSHBUTTON | WS_GROUP |
WS_TABSTOP | WS_CHILD, 128, 8, 36, 14
CONTROL "&Annuler", IDCANCEL, "button", BS_PUSHBUTTON |
WS_GROUP | WS_TABSTOP | WS_CHILD, 128, 28, 36, 14
```

```
END
```

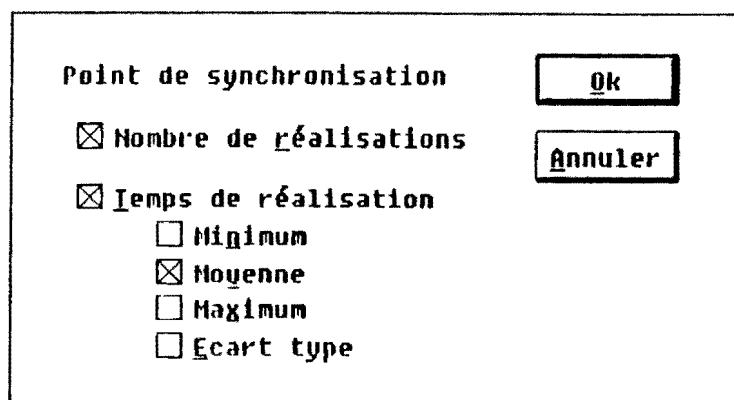


Illustration 23 : Boîte de dialogue SGLPT

```
SCOM DIALOG LOADONCALL MOVEABLE DISCARDABLE 17, 14, 174, 148
STYLE WS_DLGFRAME | WS_POPUP
BEGIN
  CONTROL "Commentaires", 390, "static", SS_LEFT | WS_GROUP |
    WS_CHILD, 10, 10, 50, 8
  CONTROL "Simulation", 391, "button", BS_GROUPBOX | WS_GROUP |
    WS_CHILD, 20, 24, 86, 34
  CONTROL "&Titre", SCOM_TITRE, "button", BS_CHECKBOX |
    WS_GROUP | WS_TABSTOP | WS_CHILD, 24, 33, 32, 12
  CONTROL "&Date d'exÉcution", SCOM_DATE, "button", BS_CHECKBOX |
    WS_CHILD, 24, 44, 78, 12
  CONTROL "Mesures statistiques", 392, "button", BS_GROUPBOX |
    WS_GROUP | WS_CHILD, 20, 68, 126, 34
  CONTROL "T&ype", SCOM_TYPE, "button", BS_CHECKBOX |
    WS_GROUP | WS_TABSTOP | WS_CHILD, 24, 78, 28, 12
  CONTROL "&PÉriode de prise en compte", SCOM_PERIODE, "button",
    BS_CHECKBOX | WS_CHILD, 24, 88, 116, 12
  CONTROL "Pt. de synchr.", 393, "button", BS_GROUPBOX | WS_GROUP |
    WS_CHILD, 20, 112, 68, 24
  CONTROL "&Nom", SCOM_NOM, "button", BS_CHECKBOX |
    WS_GROUP | WS_TABSTOP | WS_CHILD, 24, 122, 24, 12
  CONTROL "&Ok", IDOK, "button", BS_DEFPUSHBUTTON | WS_GROUP |
    WS_TABSTOP | WS_CHILD, 124, 8, 36, 14
  CONTROL "&Annuler", IDCANCEL, "button", BS_PUSHBUTTON |
    WS_GROUP | WS_TABSTOP | WS_CHILD, 124, 28, 36, 14
END
```

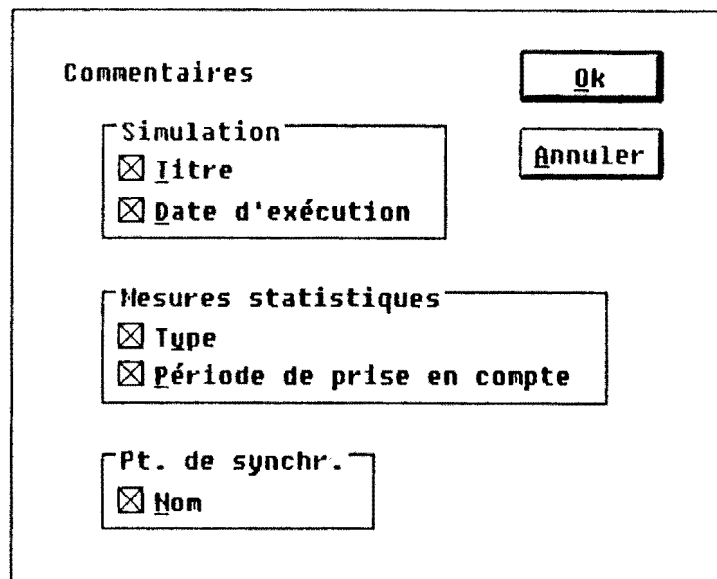


Illustration 24 : Boîte de dialogue SCOM

```
PCIIR DIALOG LOADONCALL MOVEABLE DISCARDABLE 15, 15, 193, 173
STYLE WS_DLGFRAME | WS_POPUP
BEGIN
  CONTROL "Statistiques chronologiques", 220, "static", SS_LEFT |
    WS_GROUP | WS_CHILD, 10, 10, 112, 8
  CONTROL "Fiche &standard", PCHR_FSTA, "button", BS_RADIOBUTTON |
```

```

WS_GROUP | WS_TABSTOP | WS_CHILD, 14, 24, 70, 12
CONTROL "Fiche &personnalisÉe", PCHR_FPER, "button",
BS_RADIOBUTTON | WS_CHILD, 14, 40, 88, 12
CONTROL "EvÈnements", 223, "button", BS_GROUPBOX | WS_GROUP |
WS_CHILD, 24, 50, 154, 44
CONTROL "DÉc&lenchements", PCHR_DEC, "button", BS_CHECKBOX |
WS_GROUP | WS_TABSTOP | WS_CHILD, 28, 60, 70, 12
CONTROL "A&ctivations", PCHR_ACT, "button", BS_CHECKBOX |
WS_CHILD, 100, 60, 56, 12
CONTROL "&I&nterruptions", PCHR_INT, "button", BS_CHECKBOX |
WS_CHILD, 28, 70, 64, 12
CONTROL "&RÉ&activations", PCHR_REA, "button", BS_CHECKBOX |
WS_CHILD, 100, 70, 64, 12
CONTROL "&T&erminaisons", PCHR_TER, "button", BS_CHECKBOX |
WS_CHILD, 28, 80, 60, 12
CONTROL "RÉ&partitions", 224, "button", BS_GROUPBOX | WS_GROUP |
WS_CHILD, 24, 98, 154, 44
CONTROL "Nombre d'É&vÈnements", PCHR_RPV, "button",
BS_CHECKBOX | WS_GROUP | WS_TABSTOP | WS_CHILD, 28,
108, 90, 12
CONTROL "DurÉE d'a&t&tente", PCHR_RPT, "button", BS_CHECKBOX |
WS_CHILD, 28, 118, 74, 12
CONTROL "Nombre de &proc. en att. ou en act.", PCHR_RPP, "button",
BS_CHECKBOX | WS_CHILD, 28, 128, 146, 12
CONTROL "&Durées...", PCHR_DUREE, "button", BS_PUSHBUTTON |
WS_GROUP | WS_TABSTOP | WS_CHILD, 44, 148, 42, 14
CONTROL "&Etats...", PCHR_ETAT, "button", BS_PUSHBUTTON |
WS_GROUP | WS_TABSTOP | WS_CHILD, 110, 148, 42, 14
CONTROL "&Ok", IDOK, "button", BS_DEFPUSHBUTTON | WS_GROUP |
WS_TABSTOP | WS_CHILD, 142, 10, 36, 14
CONTROL "&Annuler", IDCANCEL, "button", BS_PUSHBUTTON |
WS_GROUP | WS_TABSTOP | WS_CHILD, 142, 30, 36, 14

```

END

**Statistiques chronologiques** Ok

Fiche standard Annuler

Fiche personnalisée

**Evénements**

Déclenchements     Activations

Interruptions     Réactivations

Terminaisons

**Répartitions**

Nombre d'événements

Durée d'attente

Nombre de proc. en att. ou en act.

Durées...    Etats...

Illustration 25 : Boîte de dialogue PCHR



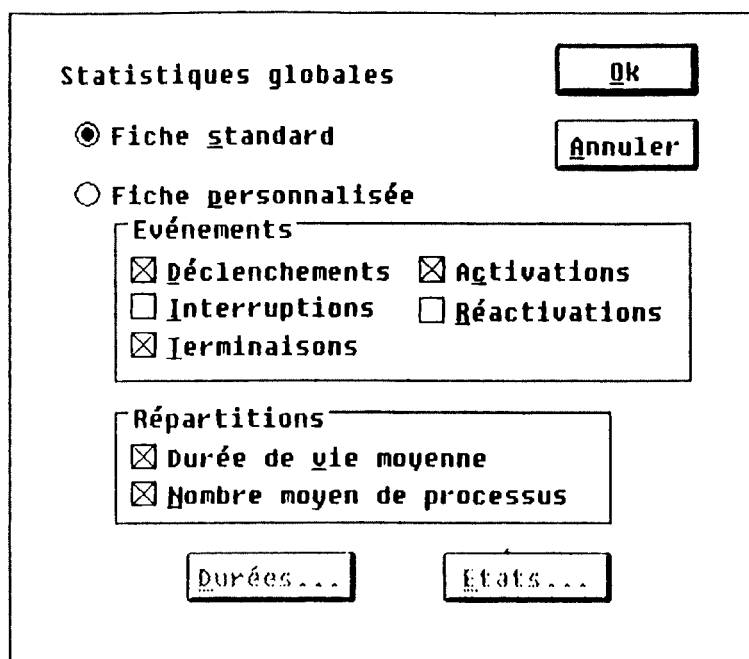


Illustration 26 : Boîte de dialogue PGLOB

PGLOB DIALOG LOADONCALL MOVEABLE DISCARDABLE 15, 15, 183, 170  
 STYLE WS\_DLGFRAME | WS\_POPUP  
 BEGIN

```

CONTROL "Statistiques globales", 120, "static", SS_LEFT | WS_CHILD, 10,
    10, 88, 8
CONTROL "Fiche &standard", PGLOB_FSTA, "button",
    BS_RADIOBUTTON | WS_GROUP | WS_TABSTOP |
    WS_CHILD, 14, 24, 68, 12
CONTROL "Fiche &personnalis e", PGLOB_FPER, "button",
    BS_RADIOBUTTON | WS_CHILD, 14, 40, 90, 12
CONTROL "Ev nements", 123, "button", BS_GROUPBOX | WS_GROUP |
    WS_CHILD, 24, 50, 144, 46
CONTROL "D c&lancements", PGLOB_DEC, "button", BS_CHECKBOX |
    WS_GROUP | WS_TABSTOP | WS_CHILD, 28, 60, 70, 12
CONTROL "A&ctivations", PGLOB_ACT, "button", BS_CHECKBOX |
    WS_CHILD, 100, 60, 58, 12
CONTROL "&Interruptions", PGLOB_INT, "button", BS_CHECKBOX |
    WS_CHILD, 28, 70, 66, 12
CONTROL "&R activations", PGLOB_REA, "button", BS_CHECKBOX |
    WS_CHILD, 100, 71, 66, 12
CONTROL "&Terminaisons", PGLOB_TER, "button", BS_CHECKBOX |
    WS_CHILD, 28, 80, 60, 12
CONTROL "R partitions", 124, "button", BS_GROUPBOX | WS_GROUP |
    WS_CHILD, 24, 100, 124, 34
CONTROL "Dur e de &vie moyenne", PGLOB_VIE, "button",
    BS_CHECKBOX | WS_GROUP | WS_TABSTOP | WS_CHILD, 28,
    110, 92, 12
CONTROL "&Nombre moyen de processus", PGLOB_PRO, "button",
    BS_CHECKBOX | WS_CHILD, 28, 120, 112, 12
CONTROL "&Dur es...", PGLOB_DUREE, "button", BS_PUSHBUTTON |
    WS_GROUP | WS_TABSTOP | WS_CHILD, 42, 142, 42, 14
    
```

```
CONTROL "&Etats...", PGLOB_ETAT, "button", BS_PUSHBUTTON |
    WS_GROUP | WS_TABSTOP | WS_CHILD, 106, 142, 42, 14
CONTROL "&Ok", IDOK, "button", BS_DEFPUSHBUTTON | WS_GROUP |
    WS_TABSTOP | WS_CHILD, 134, 6, 36, 14
CONTROL "&Annuler", IDCANCEL, "button", BS_PUSHBUTTON |
    WS_GROUP | WS_TABSTOP | WS_CHILD, 134, 26, 36, 14
```

END

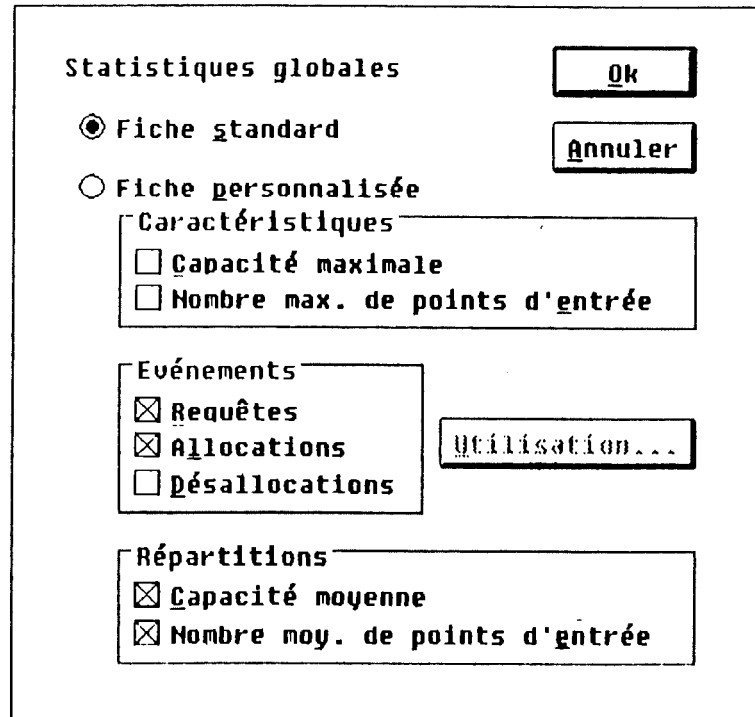


Illustration 27 : Boîte de dialogue RGLOB

```
RGLOB DIALOG LOADONCALL MOVEABLE DISCARDABLE 15, 15, 183, 187
STYLE WS_DLGFRAME | WS_POPUP
BEGIN
```

```
CONTROL "Statistiques globales", 250, "static", SS_LEFT | WS_GROUP |
    WS_CHILD, 10, 10, 86, 8
CONTROL "Fiche &standard", RGLOB_FSTA, "button", BS_RADIOBUTTON |
    WS_GROUP | WS_TABSTOP | WS_CHILD, 14, 24, 72, 12
CONTROL "Fiche &personnalis&e", RGLOB_FPER, "button",
    BS_RADIOBUTTON | WS_CHILD, 14, 40, 90, 12
CONTROL "Caract&eristiques", 253, "button", BS_GROUPBOX | WS_GROUP |
    WS_CHILD, 24, 50, 144, 34
CONTROL "&Capacit&e maximale", RGLOB_CC, "button", BS_CHECKBOX |
    WS_GROUP | WS_TABSTOP | WS_CHILD, 28, 60, 82, 12
CONTROL "Nombre max. de points d'&entr&e", RGLOB_CE, "button",
    BS_CHECKBOX | WS_CHILD, 28, 70, 134, 12
CONTROL "Ev&enements", 254, "button", BS_GROUPBOX | WS_GROUP |
    WS_CHILD, 24, 90, 76, 44
CONTROL "&Requ&tes", RGLOB_ER, "button", BS_CHECKBOX |
    WS_GROUP | WS_TABSTOP | WS_CHILD, 28, 100, 44, 12
CONTROL "A&llocations", RGLOB_EL, "button", BS_CHECKBOX |
```

```

        WS_CHILD, 28, 110, 56, 12
CONTROL "&DÉsallocations", RGLOB_ED, "button", BS_CHECKBOX |
        WS_CHILD, 28, 120, 70, 12
CONTROL "RÉpartitions", 255, "button", BS_GROUPBOX | WS_GROUP |
        WS_CHILD, 24, 140, 144, 34
CONTROL "&CapacitÉ moyenne", RGLOB_RC, "button", BS_CHECKBOX |
        WS_GROUP | WS_TABSTOP | WS_CHILD, 28, 150, 78, 12
CONTROL "Nombre moy. de points d'entrÉe", RGLOB_RE, "button",
        BS_CHECKBOX | WS_CHILD, 28, 160, 134, 12
CONTROL "&Utilisation...", RGLOB_UTIL, "button", BS_PUSHBUTTON |
        WS_GROUP | WS_TABSTOP | WS_CHILD, 104, 108, 64, 14
CONTROL "&Ok", IDOK, "button", BS_DEFPUSHBUTTON | WS_GROUP |
        WS_TABSTOP | WS_CHILD, 132, 8, 36, 14
CONTROL "&Annuler", IDCANCEL, "button", BS_PUSHBUTTON |
        WS_GROUP | WS_TABSTOP | WS_CHILD, 132, 28, 36, 14
END

```

```

SGLOB DIALOG LOADONCALL MOVEABLE DISCARDABLE 15, 15, 209, 161
STYLE WS_DLGFRAME | WS_POPUP
BEGIN

```

```

CONTROL "Statistiques globales", 350, "static", SS_LEFT | WS_GROUP |
        WS_CHILD, 10, 10, 92, 8
CONTROL "Fiche &standard", SGLOB_FSTA, "button", BS_RADIOBUTTON |
        WS_GROUP | WS_TABSTOP | WS_CHILD, 14, 24, 70, 12
CONTROL "Fiche &personnalisÉe", SGLOB_FPER, "button",
        BS_RADIOBUTTON | WS_CHILD, 14, 40, 88, 12
CONTROL "Condition de synchronisation", 353, "button", BS_GROUPBOX |
        WS_GROUP | WS_CHILD, 24, 50, 164, 34
CONTROL "P&rÉdicats", SGLOB_CR, "button", BS_CHECKBOX |
        WS_GROUP | WS_TABSTOP | WS_CHILD, 28, 60, 48, 12
CONTROL "&Mode de contribution", SGLOB_CM, "button", BS_CHECKBOX |
        WS_CHILD, 28, 70, 96, 12
CONTROL "RÉpartitions", 354, "button", BS_GROUPBOX | WS_GROUP |
        WS_CHILD, 24, 90, 164, 34
CONTROL "&DurÉe des Évts. contraignants", SGLOB_RD, "button",
        BS_CHECKBOX | WS_GROUP | WS_TABSTOP | WS_CHILD, 28,
        99, 128, 12
CONTROL "&FrÉquence des Évts. contraignants", SGLOB_RF, "button",
        BS_CHECKBOX | WS_CHILD, 28, 110, 144, 12
CONTROL "Pt. de s&ychr...", SGLOB_PT, "button", BS_PUSHBUTTON |
        WS_GROUP | WS_TABSTOP | WS_CHILD, 12, 132, 90, 14
CONTROL "E&vts. contribuants...", SGLOB_EV, "button",
        BS_PUSHBUTTON | WS_GROUP | WS_TABSTOP | WS_CHILD,
        110, 132, 90, 14
CONTROL "&Ok", IDOK, "button", BS_DEFPUSHBUTTON | WS_GROUP |
        WS_TABSTOP | WS_CHILD, 152, 8, 36, 14
CONTROL "&Annuler", IDCANCEL, "button", BS_PUSHBUTTON |
        WS_GROUP | WS_TABSTOP | WS_CHILD, 152, 28, 36, 14

```

END

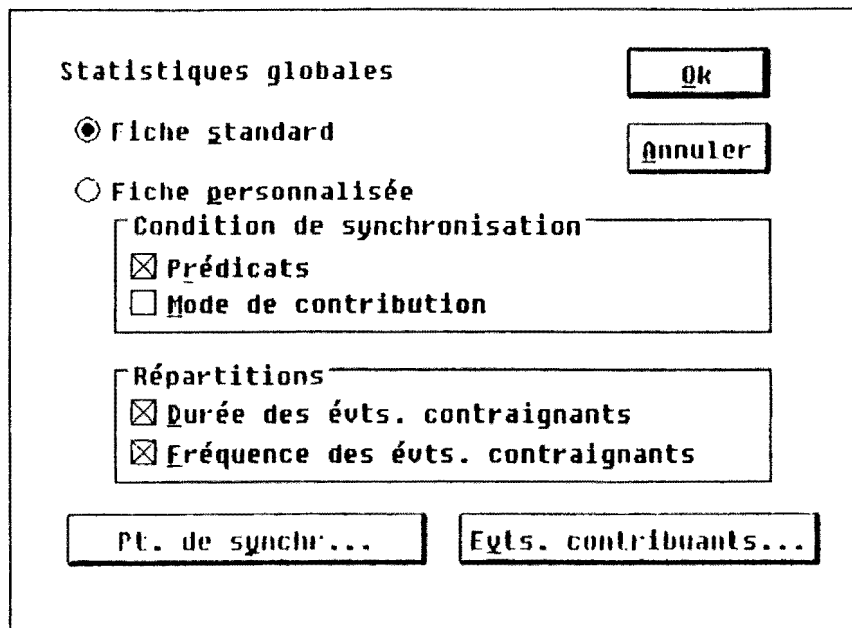


Illustration 28 : Boîte de dialogue SGLOB

```
RCOM DIALOG LOADONCALL MOVEABLE DISCARDABLE 17, 14, 176, 161
STYLE WS_DLGFRAME | WS_POPUP
BEGIN
  CONTROL "Commentaires", 340, "static", SS_LEFT | WS_GROUP |
    WS_CHILD, 10, 10, 54, 8
  CONTROL "Simulation", 341, "button", BS_GROUPBOX | WS_GROUP |
    WS_CHILD, 20, 24, 84, 34
  CONTROL "&Titre", RCOM_TITRE, "button", BS_CHECKBOX |
    WS_GROUP | WS_TABSTOP | WS_CHILD, 24, 33, 30, 12
  CONTROL "&Date d'exÉcution", RCOM_DATE, "button", BS_CHECKBOX |
    WS_CHILD, 24, 44, 76, 12
  CONTROL "Mesures statistiques", 342, "button", BS_GROUPBOX |
    WS_GROUP | WS_CHILD, 20, 68, 122, 34
  CONTROL "T&ype", RCOM_TYPE, "button", BS_CHECKBOX |
    WS_GROUP | WS_TABSTOP | WS_CHILD, 24, 78, 28, 12
  CONTROL "&PÉriode de prise en compte", RCOM_PÉRIODE, "button",
    BS_CHECKBOX | WS_CHILD, 24, 88, 116, 12
  CONTROL "Ressources", 343, "button", BS_GROUPBOX | WS_GROUP |
    WS_CHILD, 20, 112, 62, 34
  CONTROL "&Nom", RCOM_NOM, "button", BS_CHECKBOX | WS_GROUP |
    WS_TABSTOP | WS_CHILD, 24, 122, 28, 12
  CONTROL "&Calendrier", RCOM_CALENDRIER, "button", BS_CHECKBOX |
    WS_CHILD, 24, 132, 54, 12
  CONTROL "&Ok", IDOK, "button", BS_DEFPUSHBUTTON | WS_GROUP |
    WS_TABSTOP | WS_CHILD, 120, 8, 36, 14
  CONTROL "&Annuler", IDCANCEL, "button", BS_PUSHBUTTON |
    WS_GROUP | WS_TABSTOP | WS_CHILD, 120, 28, 36, 14
END
```

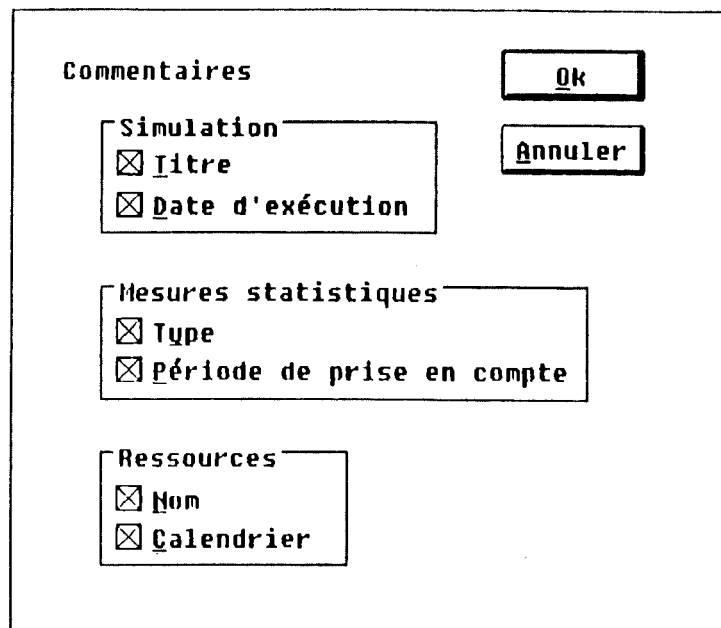


Illustration 29 : Boîte de dialogue ROOM

```
PCAL DIALOG LOADONCALL MOVEABLE DISCARDABLE 16, 12, 131, 60
STYLE WS_DLGFRAME | WS_POPUP
BEGIN
    CONTROL "Calendrier", 247, "static", SS_LEFT | WS_GROUP | WS_CHILD,
        10, 10, 46, 8
    CONTROL "&Processus", PCAL_P, "button", BS_CHECKBOX | WS_GROUP |
        WS_TABSTOP | WS_CHILD, 14, 26, 50, 12
    CONTROL "&Ressources", PCAL_R, "button", BS_CHECKBOX |
        WS_TABSTOP | WS_CHILD, 14, 40, 54, 12
    CONTROL "&Ok", IDOK, "button", BS_DEFPUSHBUTTON | WS_GROUP |
        WS_TABSTOP | WS_CHILD, 80, 8, 36, 14
    CONTROL "&Annuler", IDCANCEL, "button", BS_PUSHBUTTON |
        WS_GROUP | WS_TABSTOP | WS_CHILD, 80, 26, 36, 14
END
```

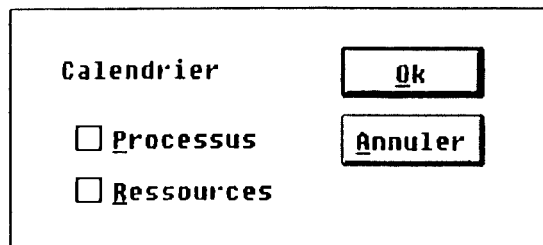


Illustration 30 : Boîte de dialogue PCAL

```
IMPR DIALOG LOADONCALL MOVEABLE DISCARDABLE 15, 15, 147, 99
STYLE WS_DLGFRAME | WS_POPUP
BEGIN
CONTROL "Imprimer", 400, "static", SS_LEFT | WS_CHILD, 10, 10, 43, 8
CONTROL "&Copies:", 401, "static", SS_LEFT | WS_CHILD, 24, 30, 31, 8
CONTROL "Nombre de pages:", 402, "static", SS_LEFT | WS_CHILD, 24, 50,
70, 8
CONTROL "&De:", 403, "static", SS_LEFT | WS_CHILD, 44, 82, 13, 8
CONTROL "&A:", 404, "static", SS_LEFT | WS_CHILD, 88, 82, 10, 8
CONTROL "", IMPR_C, "edit", ES_LEFT | WS_BORDER | WS_GROUP |
WS_TABSTOP | WS_CHILD, 56, 27, 15, 12
CONTROL "&Toutes", IMPR_T, "button", BS_RADIOBUTTON | WS_GROUP |
WS_TABSTOP | WS_CHILD, 34, 65, 34, 12
CONTROL "", IMPR_S, "button", BS_RADIOBUTTON | WS_TABSTOP |
WS_CHILD, 34, 80, 9, 12
CONTROL "", IMPR_D, "edit", ES_LEFT | WS_BORDER | WS_GROUP |
WS_TABSTOP | WS_CHILD, 60, 80, 20, 12
CONTROL "", IMPR_A, "edit", ES_LEFT | WS_BORDER | WS_GROUP |
WS_TABSTOP | WS_CHILD, 100, 80, 20, 12
CONTROL "&Ok", IDOK, "button", BS_DEFPUSHBUTTON | WS_GROUP |
WS_TABSTOP | WS_CHILD, 100, 7, 36, 14
CONTROL "&Annuler", IDCANCEL, "button", BS_PUSHBUTTON |
WS_GROUP | WS_TABSTOP | WS_CHILD, 100, 28, 36, 14
END
```

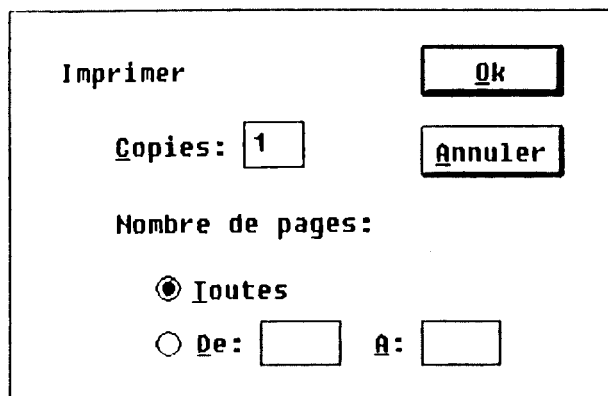


Illustration 31 : Boîte de dialogue IMPR

6

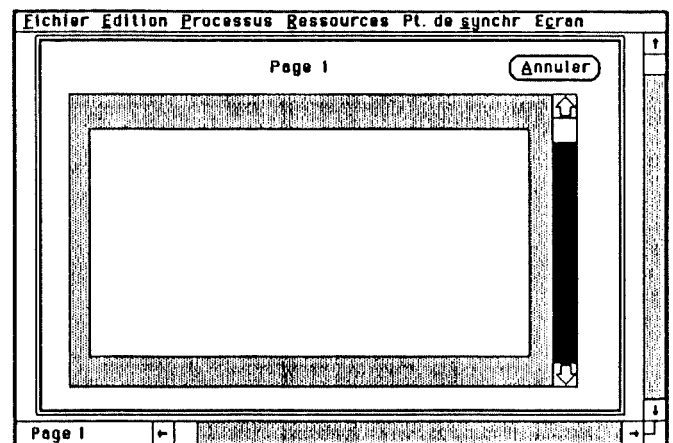
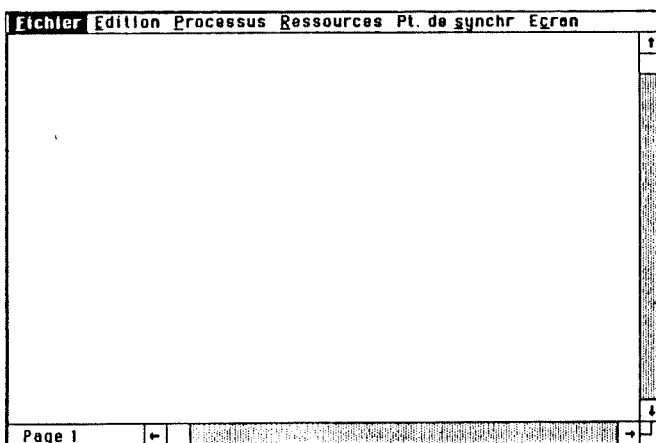
---

## MAQUETTE : PARTIE HYPERCARD

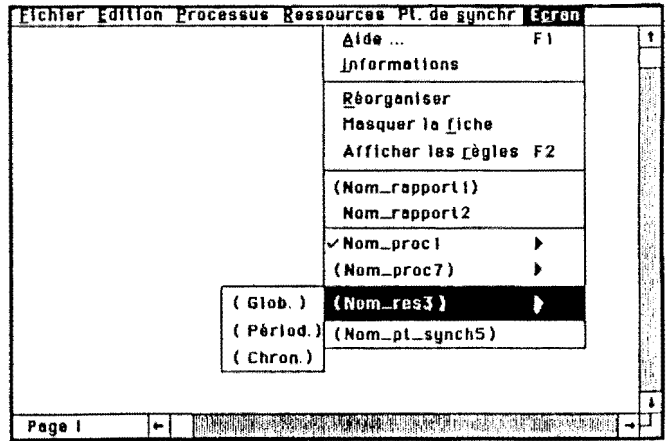
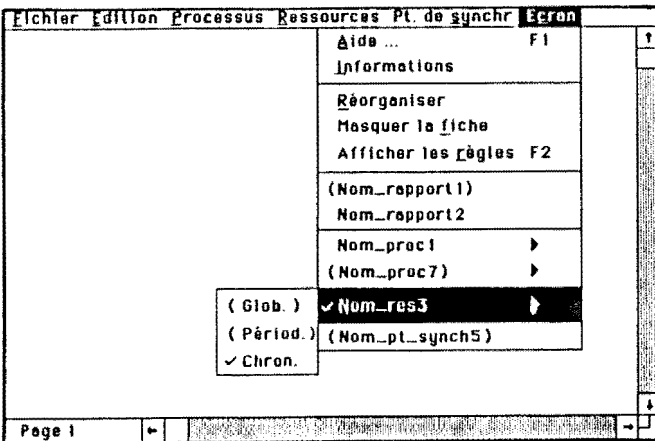
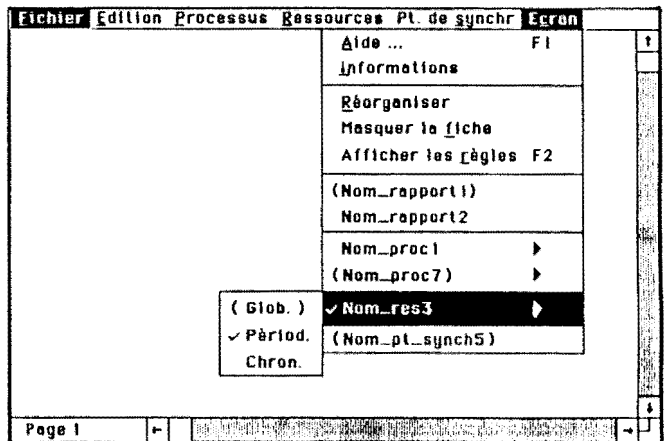
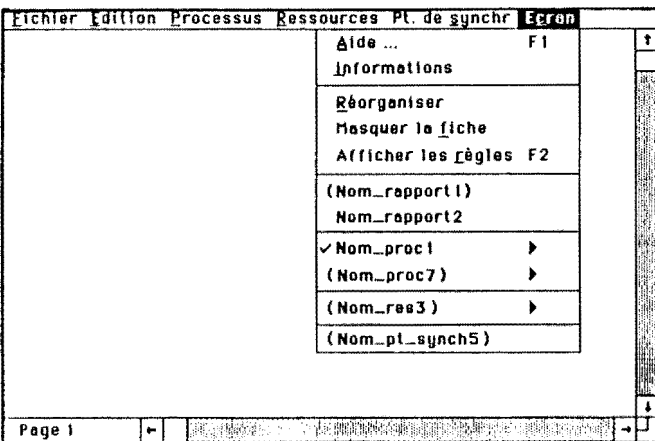
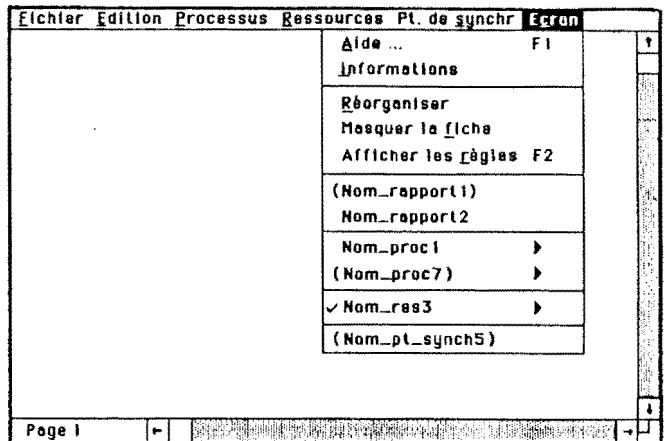
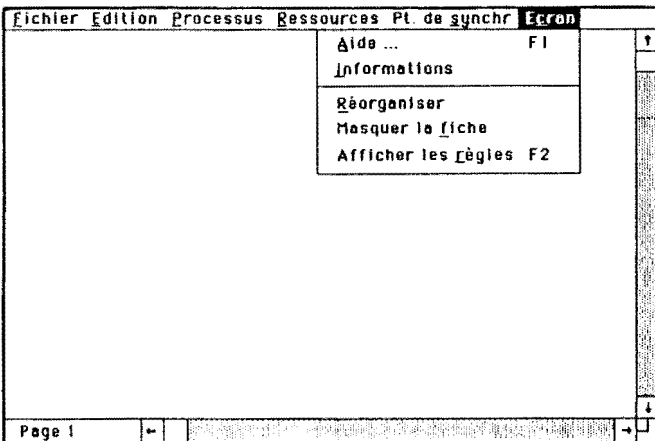
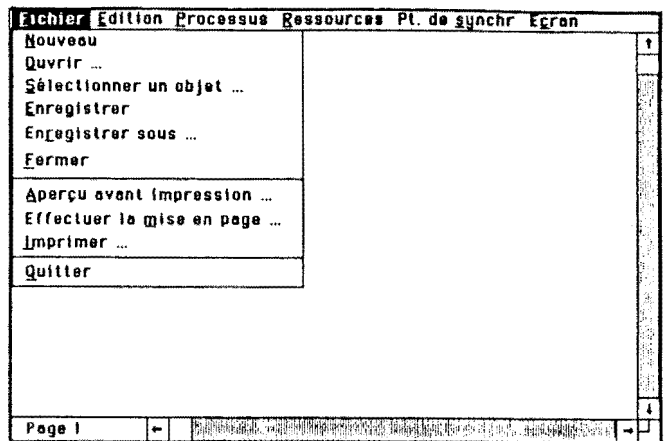
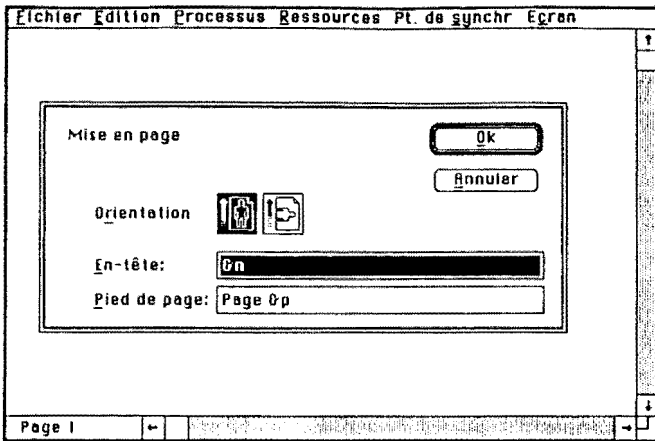
Nous vous livrons ci-après le contenu des piles *Simulation* et *Sélection* qui représentent la partie de la maquette réalisée avec HyperCard. Plus précisément, nous ne vous soumettons que le dessin des cartes, tant les scripts sous-jacents sont évidents : il s'agit de simples go to.

Le choix de scinder la maquette en deux piles résulte d'une raison historique. En effet, nous avons été amenés très tôt à imaginer un mécanisme de sélection dont nous avons voulu nous rendre compte du comportement grâce à HyperCard. La maquette de MEANDRES nous a été demandée bien plus tard. Nous aurions très bien pu tout regrouper au sein d'une même pile, ou au contraire éclater la maquette en un nombre plus élevé de piles.

### 6.1 PILE SIMULATION

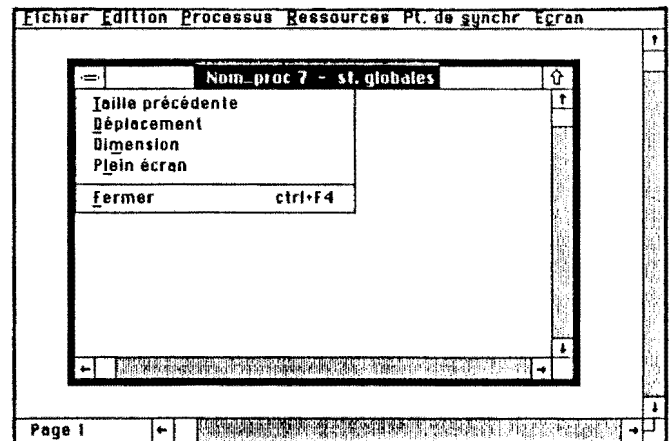
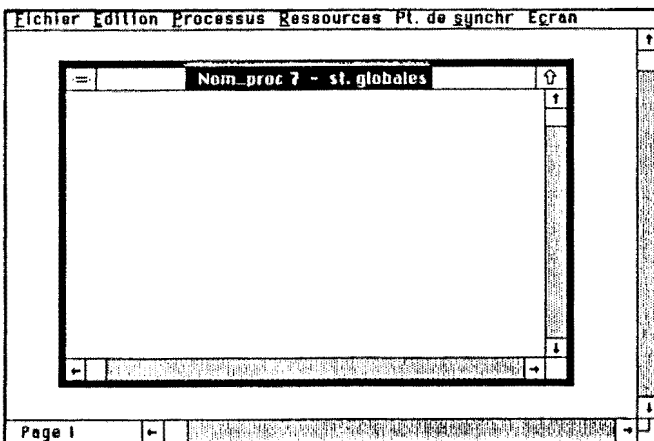
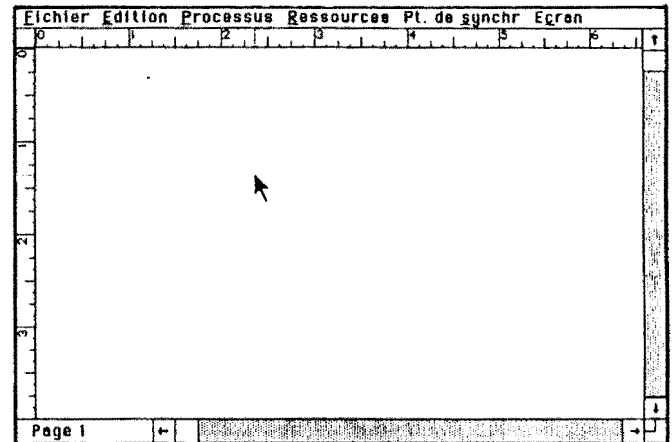
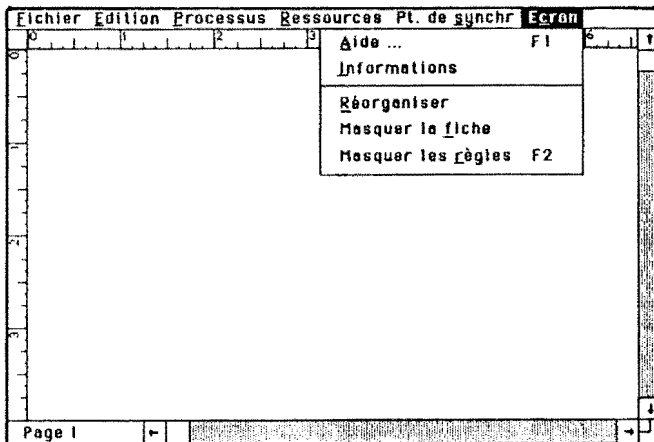
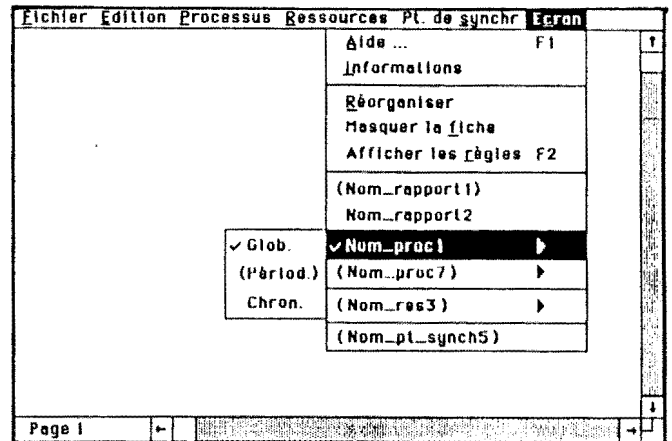
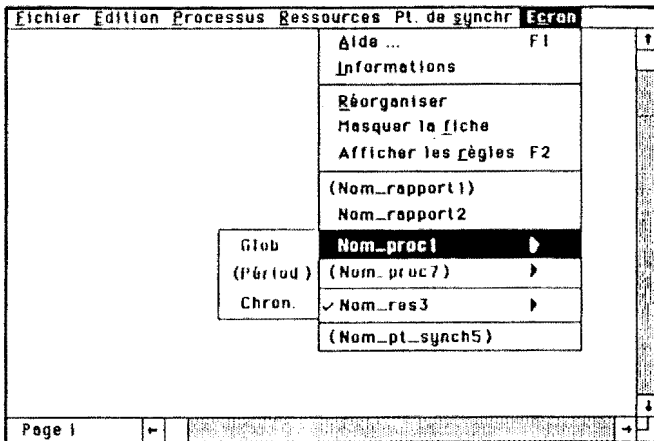
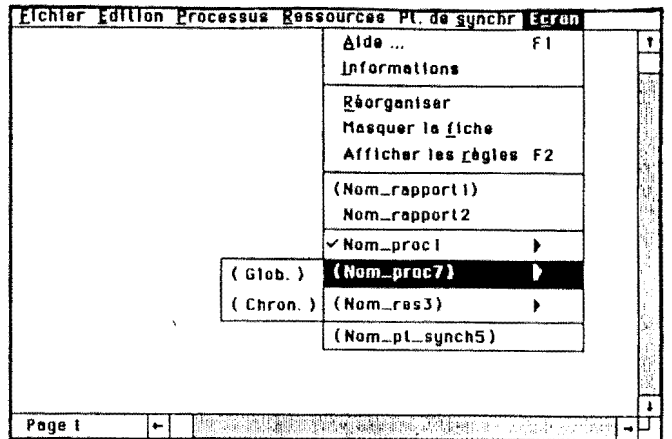
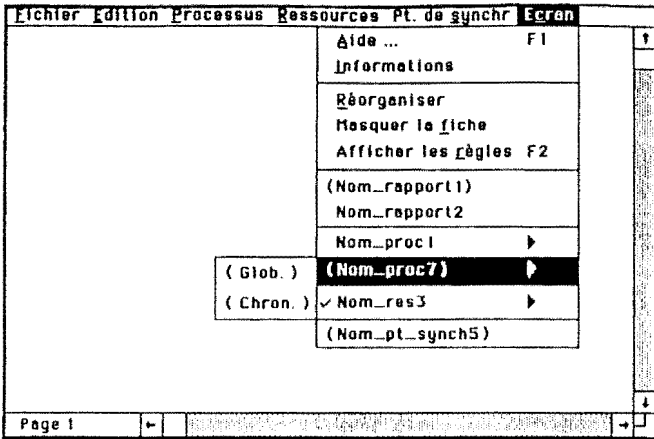


Maquette : partie HyperCard





Maquette : partie HyperCard

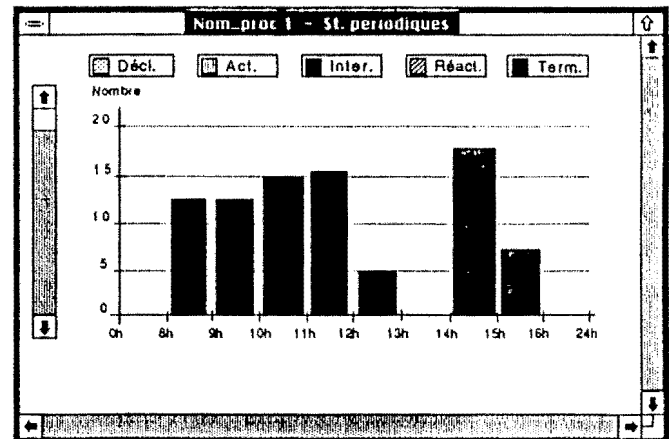
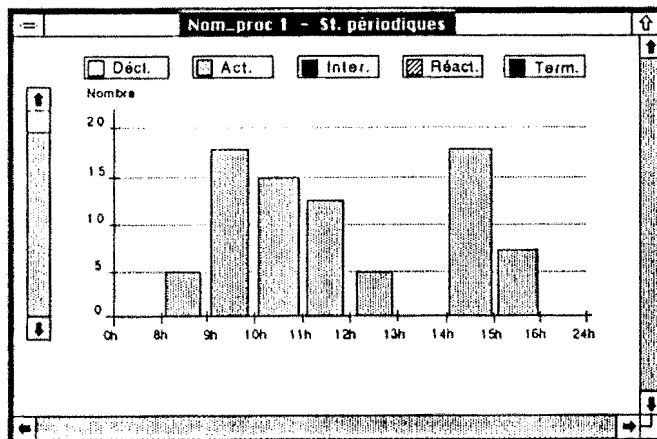
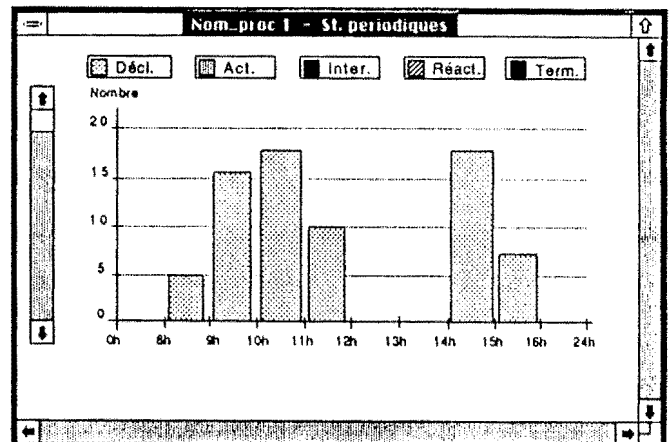
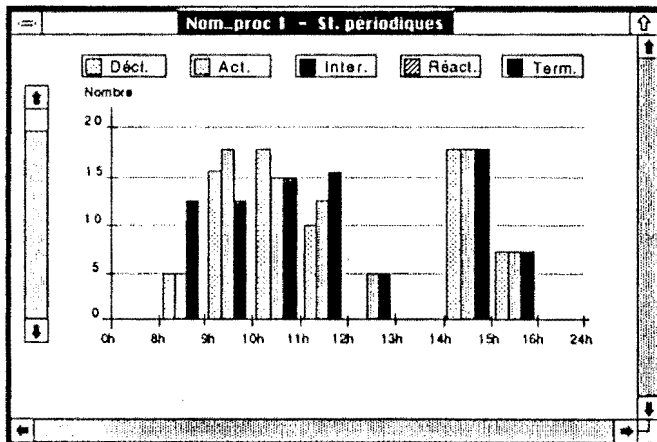
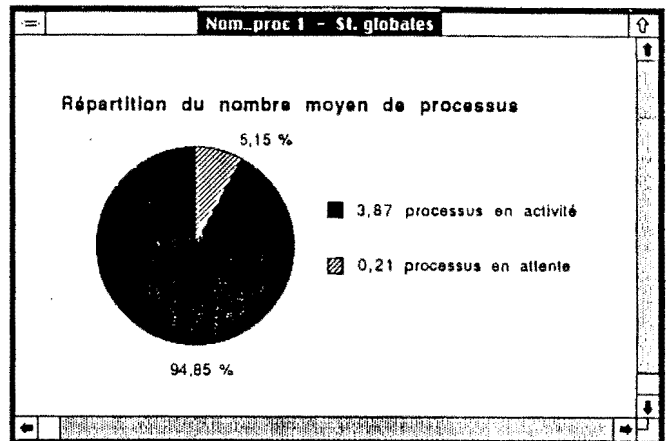
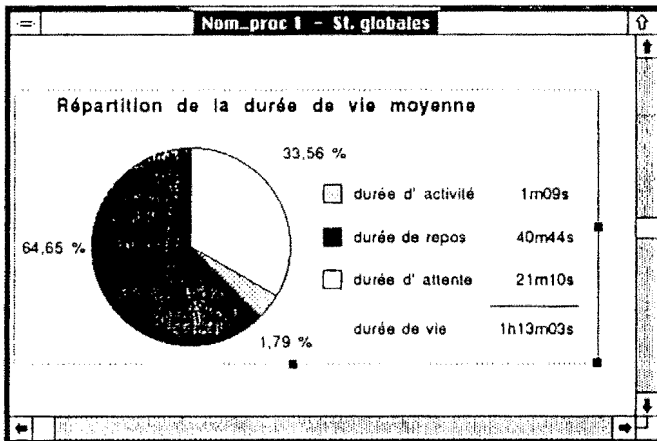
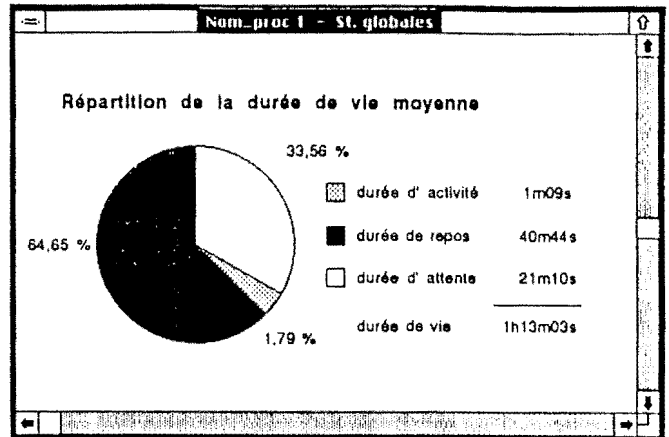


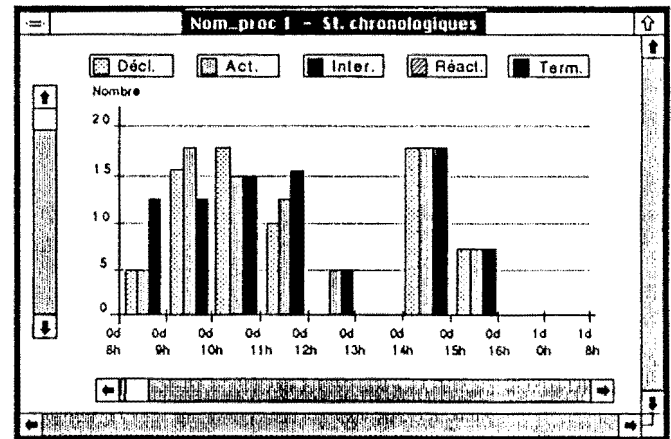
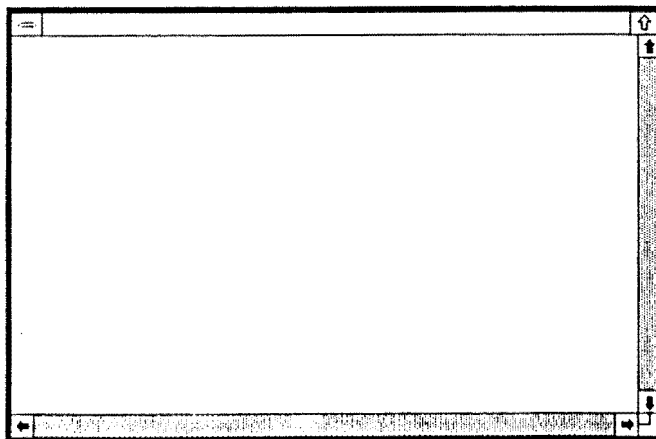
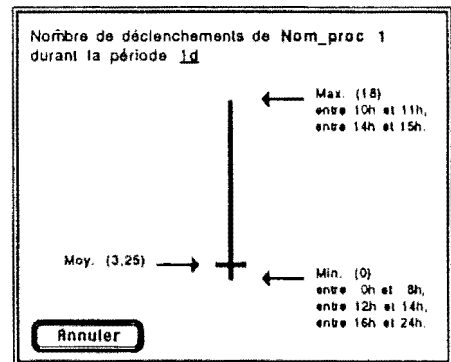
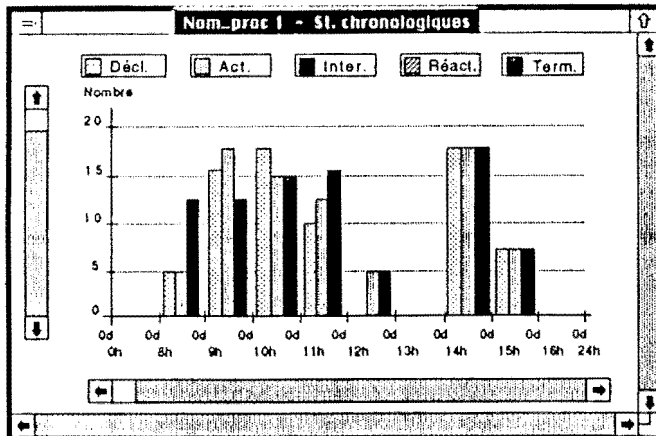
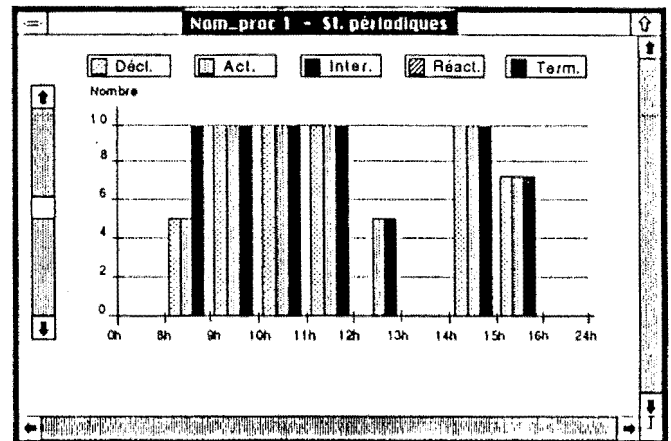
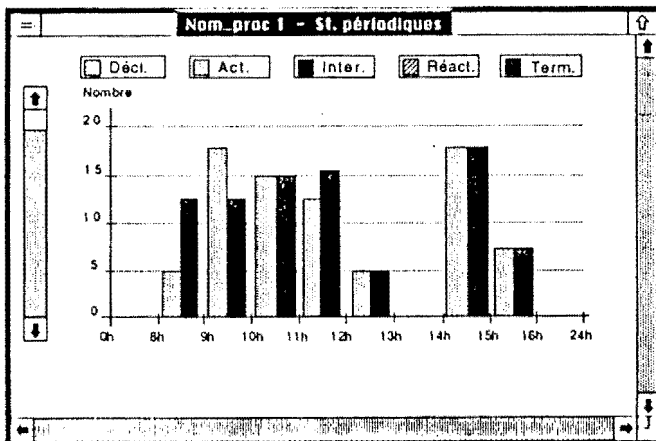
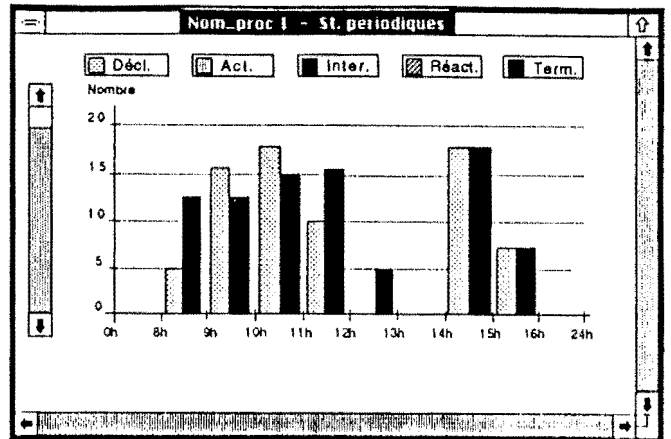
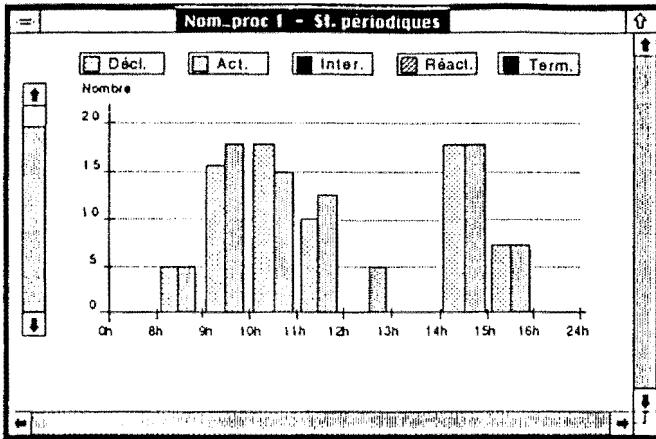
**Nom\_proc 1 - St. globales**

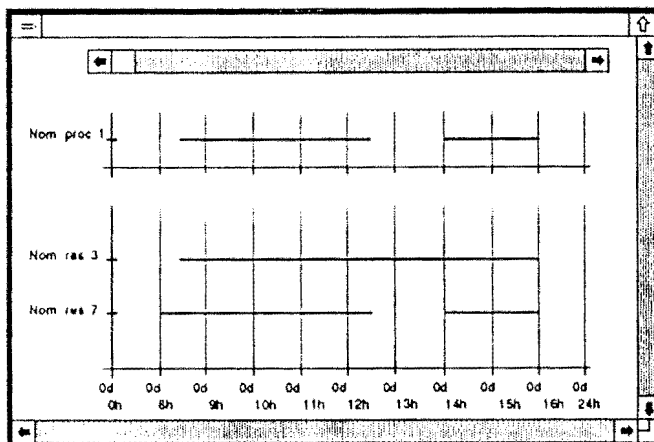
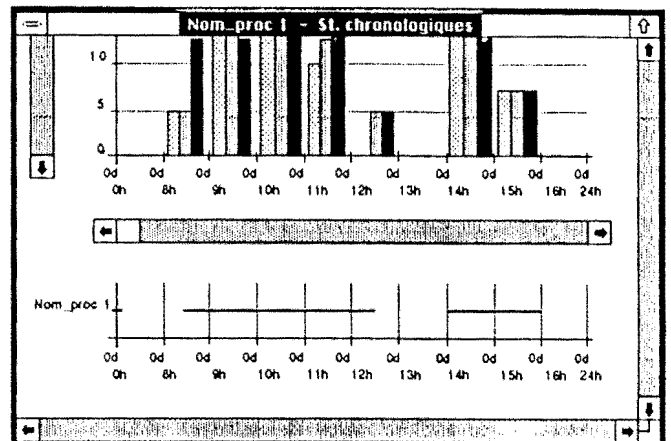
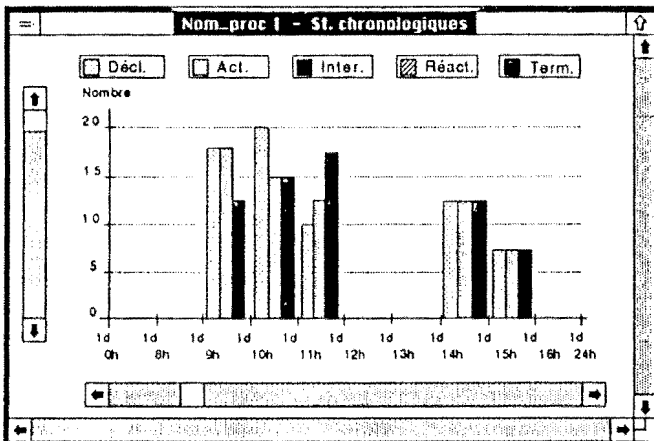
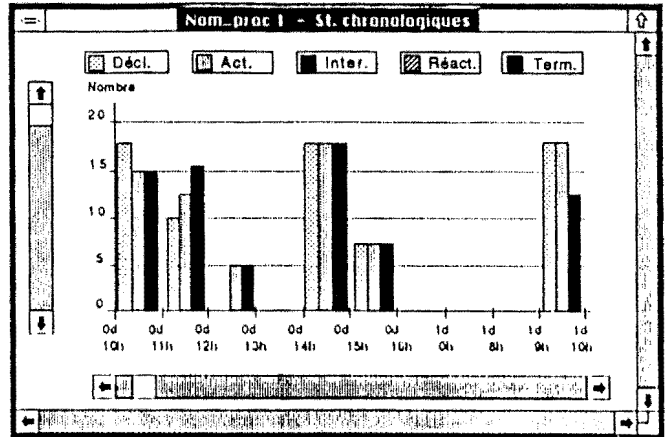
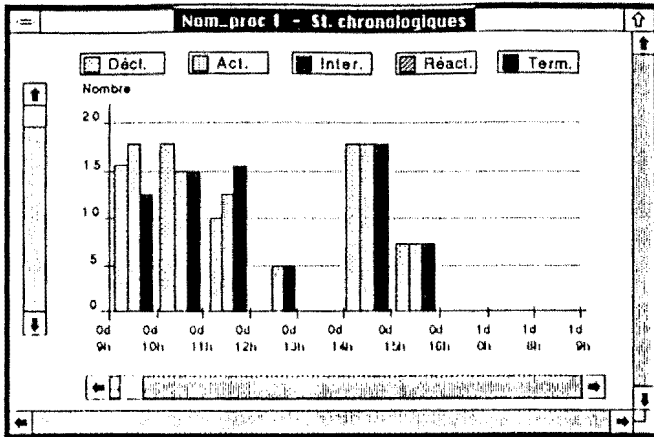
- Simulation Gestion Prêt exécutée le 12 avril 1991 à 15h57m59s
- Statistiques globales concernant les processus Nom\_proc 1 établies durant la période 0d à 5d
- Actifs de 8h30m à 12h30m, de 13h30m à 17h30m du 0 au 5<sup>ème</sup> jour de la semaine

**Evénements**

- 156 déclenchements
- 156 activations
- 160 terminaisons







Nom\_pt\_synch 7 - St. globales

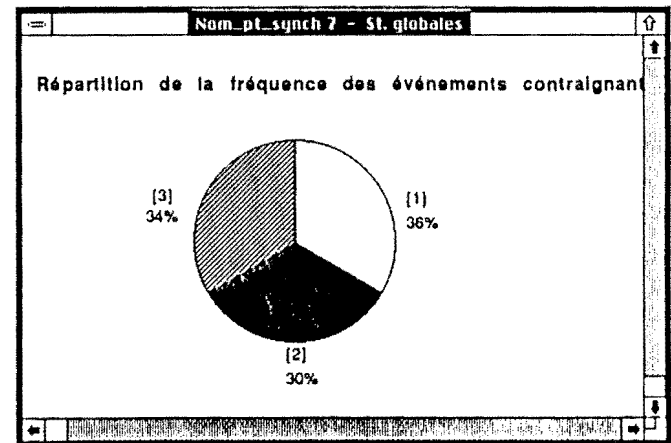
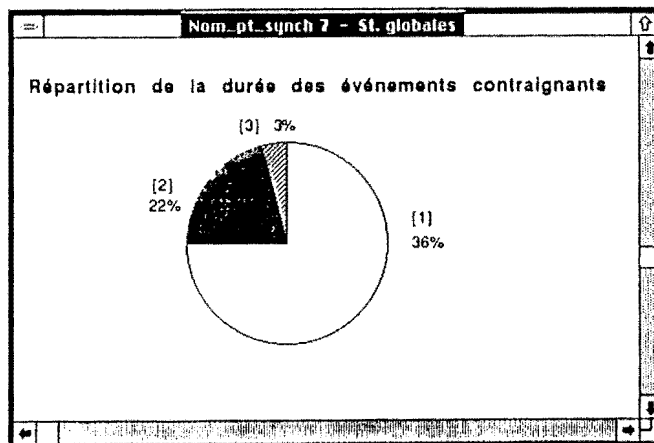
Condition de synchronisation

Réalisé quand

et

- [1] terminalson de Nom\_proc 7 (simple)
- et
- [2] réalisation de Nom\_pt\_synch 1 (simple)
- et
- [3] génération de Nom\_mes 13 (simple)

Nombre de réal. / contr.	Temps moyen de réal. / part.
[0] 7	[0] 1h18m30s
[1] 7	[1] 30m10s
[2] 7	[2] 66m45s
[3] 7	[3] 1h15m30s



## 6.2 PILE SELECTION

Processus	Ressources	Pt. de synchr.
Globales	Chronologiques	Annuler
Périodiques	Schéma	

Processus	Ressources	Pt. de synchr.
Nom-proc-1 Nom-proc-2 Nom-proc-3 Nom-proc-4 Nom-proc-5 Nom-proc-6 Nom-proc-7 Nom-proc-8 Nom-proc-9 Nom-proc-10 Nom-proc-11		
Globales	Chronologiques	Annuler
Périodiques	Schéma	

Processus	Ressources	Pt. de synchr.
Nom-proc-1 Nom-proc-2 Nom-proc-3 Nom-proc-4 Nom-proc-5 Nom-proc-6 <b>Nom-proc-7</b> Nom-proc-8 Nom-proc-9 Nom-proc-10 Nom-proc-11	Nom-res-3 Nom-res-9 Nom-res-11 Nom-res-17 Nom-res-19	
Globales	Chronologiques	Annuler
Périodiques	Schéma	

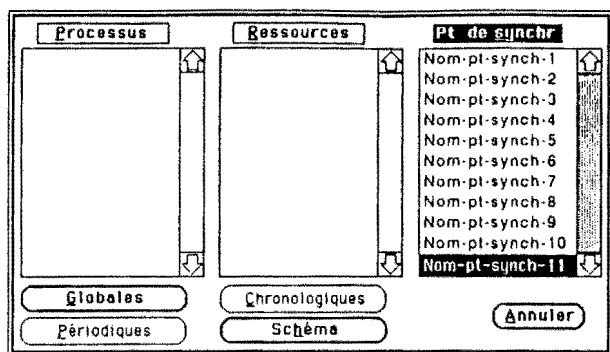
Processus	Ressources	Pt. de synchr.
Nom-proc-1 Nom-proc-7 Nom-proc-31	<b>Nom-res-11</b> Nom-res-12 Nom-res-13 Nom-res-14 Nom-res-15 Nom-res-16 Nom-res-17 Nom-res-18 Nom-res-19 Nom-res-20 Nom-res-21	
Globales	Chronologiques	Annuler
Périodiques	Schéma	

Processus	Ressources	Pt. de synchr.
	Nom-res-1 Nom-res-2 Nom-res-3 Nom-res-4 Nom-res-5 Nom-res-6 Nom-res-7 Nom-res-8 Nom-res-9 Nom-res-10 Nom-res-11	
Globales	Chronologiques	Annuler
Périodiques	Schéma	

Processus	Ressources	Pt. de synchr.
Nom-proc-1 Nom-proc-7 Nom-proc-31	Nom-res-1 Nom-res-2 Nom-res-3 Nom-res-4 Nom-res-5 Nom-res-6 Nom-res-7 Nom-res-8 Nom-res-9 Nom-res-10 <b>Nom-res-11</b>	
Globales	Chronologiques	Annuler
Périodiques	Schéma	

Processus	Ressources	Pt. de synchr.
<b>Nom-proc-7</b> Nom-proc-8 Nom-proc-9 Nom-proc-10 Nom-proc-11 Nom-proc-12 Nom-proc-13 Nom-proc-14 Nom-proc-15 Nom-proc-16 Nom-proc-17	Nom-res-3 Nom-res-9 Nom-res-11 Nom-res-17 Nom-res-19	
Globales	Chronologiques	Annuler
Périodiques	Schéma	

Processus	Ressources	Pt. de synchr.
		Nom-pt-synch-1 Nom-pt-synch-2 Nom-pt-synch-3 Nom-pt-synch-4 Nom-pt-synch-5 Nom-pt-synch-6 Nom-pt-synch-7 Nom-pt-synch-8 Nom-pt-synch-9 Nom-pt-synch-10 Nom-pt-synch-11
Globales	Chronologiques	Annuler
Périodiques	Schéma	



---

## Première maquette

### 7.1 OBJECTIFS

Cette maquette nous a servi à concrétiser les idées émises dans le mémoire de B. Sacré et de J. Lefèvre. Son but n'était rien d'autre que de susciter des réactions de tiers, de connaître leur avis sur l'outil de consultation de statistiques à réaliser. Elle s'est révélée un outil de communication très utile.

Elle a été réalisée peu de temps après que nous ayons appris le langage HyperTalk. Cela explique en partie pourquoi cette maquette reflète une philosophie très "hypercardienne". Bien que très imparfaite en regard du résultat final, cette maquette n'illustre pas moins un fait maintes fois constaté : des fonctionnalités identiques peuvent faire l'objet d'interfaces conceptuellement très différentes. Nous n'avons pas poursuivi la voie HyperCard, mais nous sommes persuadés qu'il aurait été possible de proposer un outil de consultation de statistiques et d'élaboration de rapports conforme aux concepts et normes d'HyperCard.

### 7.2 CONCEPTS

L'idée de base de cette maquette est d'associer une *fiche* à chacun des outils (c'est-à-dire ensembles de mesures statistiques) dégagés par B. Sacré et J. Lefèvre (cfr premier chapitre "Contexte et tâche") Le concept de fiche est, ici, très proche de celui de carte en HyperCard : toutes les informations sont visibles sur une fenêtre de taille fixe. A tout moment, l'utilisateur ne peut visualiser qu'une fiche.

D'autre part, nous étions soucieux de ne pas "encombrer" les fiches avec des objets interactifs tels que les barres de défilement afin de conserver un isomorphisme

parfait entre la fiche affichée à l'écran et celle imprimée. Cette volonté a de nombreuses conséquences sur l'interface.

### 7.3 PRESENTATION SOMMAIRE DE L'INTERFACE

L'écran de base se compose d'un menu icônographique permettant six actions :

- sélectionner une fiche;
- ranger une fiche,
- passer à une autre fiche présente sur le bureau,
- visualiser une autre partie d'un graphique,
- effectuer un changement d'échelle,
- imprimer.

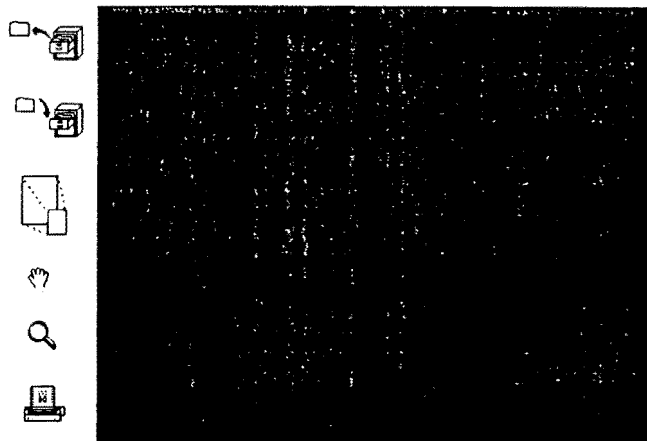


Illustration 32 : Ecran de base

#### 7.3.1 Sélectionner une fiche



Illustration 33 : Bouton de sélection de fiches

Comme l'illustre l'icône, ce bouton permet de choisir l'une des fiches présentes dans un tiroir. Ces fiches sont le reflet fidèle des suggestions de B. Sacré de et J. Lefèvre.

Après avoir cliqué sur le bouton, l'utilisateur doit choisir l'objet et les mesures qu'il souhaite consulter. Ces deux renseignements déterminent une fiche. Le mécanisme de sélection est analogue à celui présenté dans MEANDRES, si ce n'est que les boîtes de dialogue présentant les types de mesures statistiques sont moins riches. D'autre part, elles ont plus pour objet d'afficher une fiche particulière que de sélectionner des informations à afficher au sein d'une fiche globale. Autrement dit, contrairement à MEANDRES, à un objet et à un type de mesures statistiques (globale, périodique ou chronologique) peuvent correspondre plusieurs fiches.

Remarquez la présence du bouton *Calendrier* et l'absence du bouton *Schéma*.



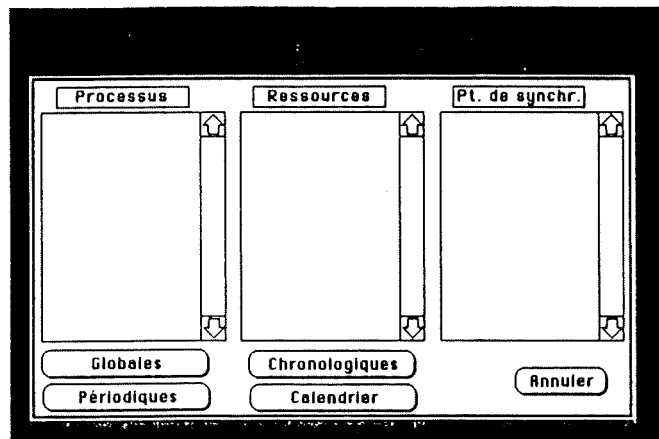


Illustration 34 : Boîte de dialogue de sélection d'un objet

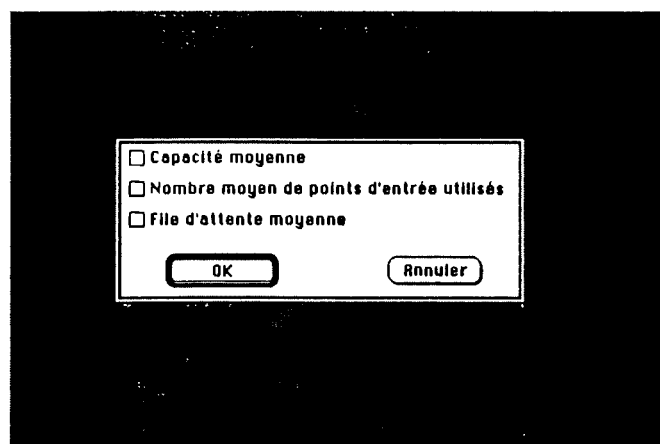
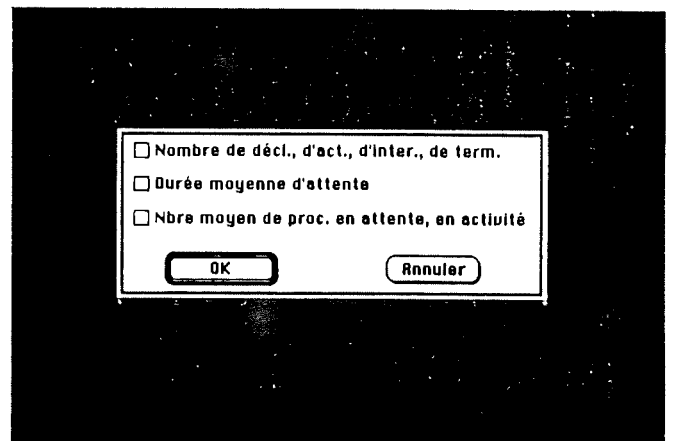


Illustration 35 : Boîtes de dialogue de sélection des mesures statistiques

Par défaut, chaque fiche est munie d'une en-tête reprenant un certain nombre de renseignements généraux (cfr commande **Commentaires** de MEANDRES). L'utilisateur peut faire disparaître et apparaître ces informations à son gré en cliquant

une seule fois n'importe où sur la fiche. Normalement, la disparition de l'en-tête provoque un réajustement spatial des données de la fiche.

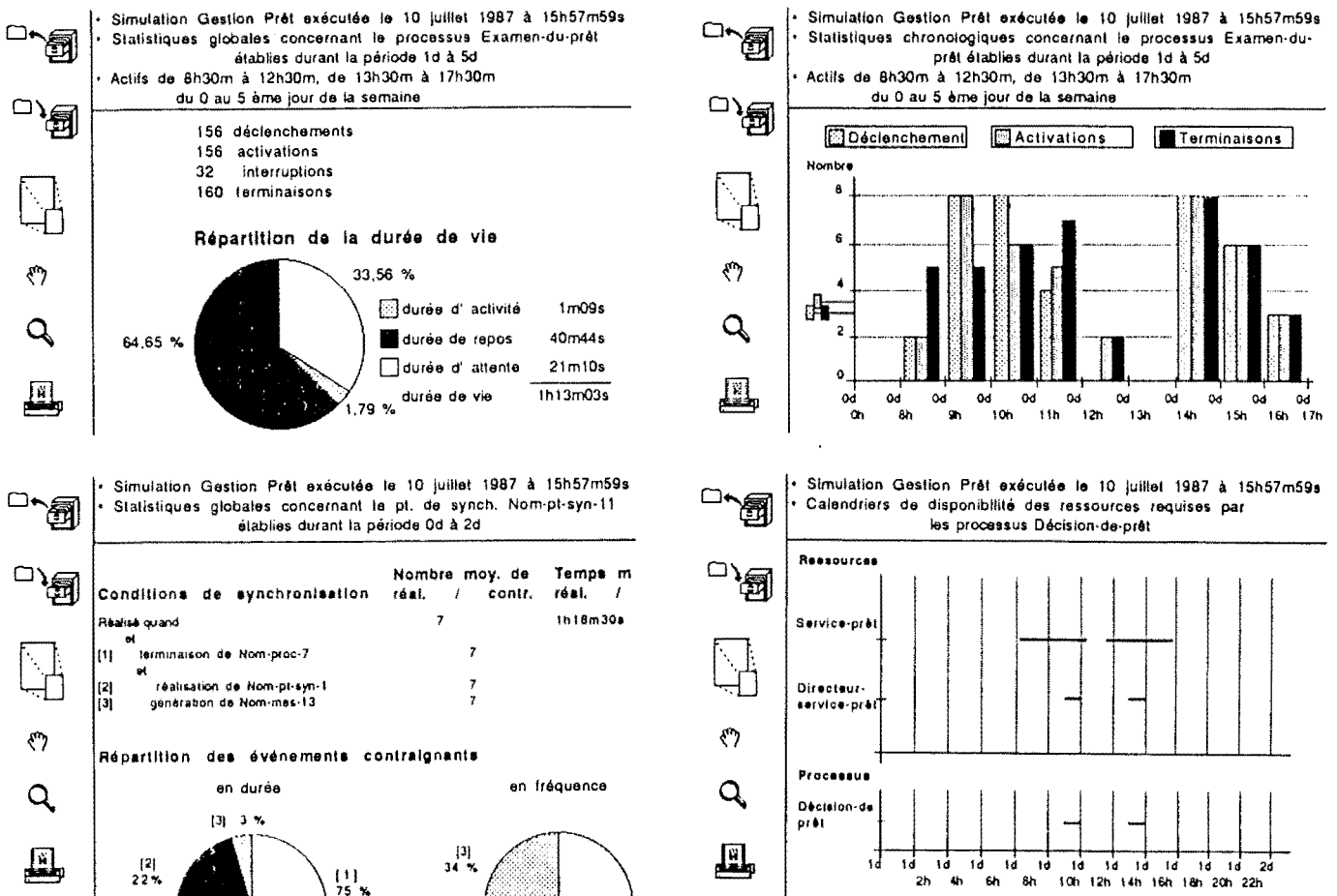


Illustration 36 : Exemples de fiches

### 7.3.2 Ranger une fiche

Dès que l'utilisateur a terminé de consulter une fiche, il peut la ranger dans le tiroir en cliquant sur le bouton illustré ci-dessous.



Illustration 37 : Bouton de rangement

S'il souhaite de nouveau la consulter, il devra repasser par le mécanisme de sélection expliqué précédemment.

### 7.3.3 Passer à une autre fiche présente sur le bureau

Tant que l'utilisateur n'a pas rangé la fiche, celle-ci reste présente sur le bureau, bien que cachée par la fiche courante. Autrement dit, il peut très rapidement la consulter grâce au bouton illustré ci-dessous :



Illustration 38 : Bouton affichant la liste des fiches présentes sur le bureau

Cliquer sur ce bouton en gardant enfoncé le bouton de la souris déroule un menu reprenant la liste des fiches présentes sur le bureau. Les fiches sont regroupées selon le type d'objets qu'elles concernent.

### 7.3.4 Visualiser une autre partie d'un graphique

L'utilisateur peut souhaiter visualiser une partie d'un histogramme non visible sur la fiche. L'outil représentant une main le lui permet.



Illustration 39 : Bouton de déplacement d'un graphique

Cliquer sur ce bouton transforme le curseur en une main. Pour visualiser une autre partie d'un histogramme, l'utilisateur doit "prendre" le graphique (c'est-à-dire cliquer dessus en gardant enfoncé le bouton de la souris) et le tirer dans la direction souhaitée. Par exemple, pour visualiser la période suivante sur un histogramme représentant des statistiques chronologiques, l'utilisateur prend l'histogramme, de préférence à droite, et bouge la souris vers la gauche. Le graphique suit le mouvement de la souris<sup>1</sup> : la cellule au dessus de laquelle l'utilisateur a cliqué se trouve maintenant à l'endroit où l'utilisateur a lâché le bouton de la souris<sup>2</sup>.

L'utilisateur peut également se servir de ce mécanisme pour déplacer le graphique dans le sens de l'axe des ordonnées. En fait, il peut déplacer le graphique dans tous les sens. C'est ce qui fait l'attrait de ce mécanisme par rapport à deux barres de défilement. De plus, il permet de préserver l'intégrité de l'isomorphisme des fiches en n'ajoutant pas d'objets interactifs visibles.

Ce mécanisme présente également l'avantage d'être, à notre avis, beaucoup plus intuitif et cohérent que les barres de défilement. En effet, malgré notre expérience, il nous arrive encore fréquemment de nous tromper en utilisant des barres de défilement : pour visualiser la suite d'un texte, par exemple, il faut le faire monter; par conséquent, nous avons tendance à cliquer sur la flèche vers le haut; pas de chance, c'est la flèche vers le bas qu'il fallait utiliser ! Il faut comprendre les flèches de défilement comme signifiant "c'est cette partie du document que vous verrez si vous cliquez sur moi" et non pas "c'est dans ce sens que le document défile quand vous cliquez sur moi". Ainsi, à cet objet interactif, on peut légitimement associer deux significations opposées ! Dans la mesure où notre mécanisme respecte nettement plus la métaphore du mini-monde [Bod90], il résout ce problème.

Pour quitter le mode de déplacement, il suffit à l'utilisateur de cliquer en dehors du convexe formé par le graphique.

Cet outil n'est disponible que pour des histogrammes présentant des mesures statistiques chronologiques.

<sup>1</sup> En fait, les axes ne bougent pas; seuls les bâtons se déplacent.

<sup>2</sup> Comme il est difficile de décrire ce mécanisme, nous renvoyons le lecteur perplexe à HyperCard. En effet, ce logiciel propose, en mode loupe, un mécanisme de déplacement analogue.

### 7.3.5 Effectuer un changement d'échelle

Le bouton illustré ci-dessus affiche une boîte de dialogue qui permet de modifier l'échelle en choisissant la valeur maximale de l'axe des ordonnées. Dans le cas d'un histogramme présentant des mesures chronologiques, la boîte de dialogue propose également un "zooming" de l'axe des abscisses.



Illustration 40 : Bouton de "zooming"

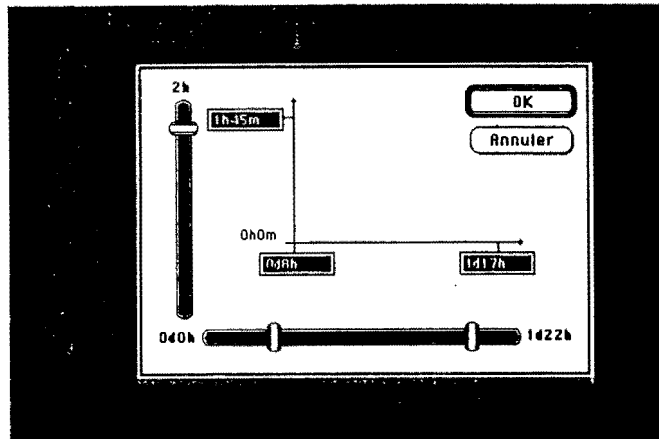


Illustration 41 : Boîte de dialogue de "zooming"

### 7.3.6 Imprimer

Enfin, l'utilisateur a la possibilité d'imprimer intégralement une fiche grâce au bouton suivant :



Illustration 42 : Bouton d'impression

Ce bouton affiche la boîte de dialogue qui gère les différents paramètres d'impression.

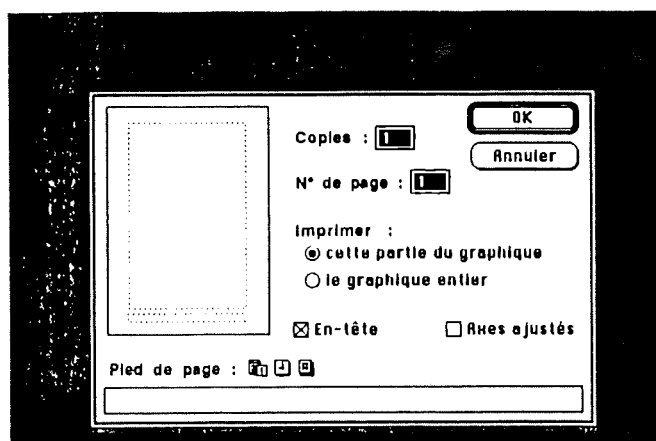


Illustration 43 : Boîte de dialogue gérant l'impression

Ces paramètres sont :

- Le nombre de copies.
- Le numéro de la page.
- Dans le cas de mesures statistiques chronologiques, l'utilisateur a la possibilité d'obtenir automatiquement une décomposition de l'histogramme par période. Ce mécanisme évite à l'utilisateur d'avoir à bouger lui-même l'histogramme et à l'imprimer période par période.
- La présence, ou non, de l'en-tête.
- L'ajustement, ou non, des axes. Le dessin ci-dessous illustre l'effet de cette option.

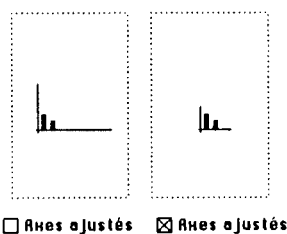


Illustration 44 : Ajustement des axes

- Le pied de page : date, heure, numéro de page, et texte libre.