Institutional Repository - Research Portal
Dépôt Institutionnel - Portail de la Recherche
researchportal.unamur.be
University of Namur

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Mise en oeuvre d'une méthode d'analyse fonctionnelle à l'aide du logiciel Isdos

Durigneux, F.

Award date:
1978

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.

Download date: 02. May. 2026

FACULTÉS UNIVERSITAIRES NOTRE-DAME DE LA PAIX - NAMUR

INSTITUT D'INFORMATIQUE

ANNEE ACADEMIQUE 1977-1978

Mise en œuvre d'une méthode d'analyse fonctionnelle à l'aide du Logiciel Isdos

Mémoire présenté en vue de l'obtention

du grade de

Licencié et Maître en Informatique

F. DURIGNEUX

FACULTES
UNIVERSITAIRES
N.-D. DE LA PAIX
NAMUR

Bibliothèque
FMB16
1978/8/1

FMB16/1978/8/1

FACULTÉS UNIVERSITAIRES NOTRE-DAME DE LA PAIX - NAMUR

INSTITUT D'INFORMATIQUE

ANNEE ACADEMIQUE 1977-1978

Mise en œuvre d'une méthode d'analyse fonctionnelle à l'aide du Logiciel Isdos

Mémoire présenté en vue de l'obtention

du grade de

Licencié et Maître en Informatique

F. DURIGNEUX

LBS 3213385

CS20-27025

TABLE DES MATIERES

	Table des matières	1
	Remerciements	4
	INTRODUCTION	5
1	Présentation du contenu du mémoire	5
2	Introduction aux logiciels d'aide à l'analyse	7
1°	Objectifs des logiciels d'aide à l'analyse	7
2°	Fonctions des logiciels d'aide à l'analyse	9
3°	Composants des logiciels d'aide à l'analyse et organisation générale	11
1.	METHODE D'ANALYSE FONCTIONNELLE	14
1. 1	Intégration de la méthode dans le processus de conception d'un système informatique	14
11. 1	Amont	14
11. 2	Aval	15
1. 2	Méthode proprement dite	15
12. 1	Concepts méthodologiques	15
121. 1	Cellule d'activité	15
121. 2	Sous-structure des données	16
121. 3	Sous-structure des traitements	18
121. 4	Intégration des deux nomenclatures en une métastructure du système d'information	21
12. 2	Structure de la méthode	22
122. 1	Données de base	22
122. 2	Corps de la méthode	22
122. 3	Outputs de la méthode	22
2.	INTRODUCTION AU LOGICIEL ISDOS	27
2. 1	Caractéristiques générales	27
21. 1	Caractéristiques du modèle	27
21. 2	Caractéristiques du langage	28
212. 1	PSL	28
212. 2	PSA ou Analyser	29
2. 2	Description des éléments du modéle d'ISDOS	29
22. 1	Objets et attributs	29
22. 2	Associations ("Relationships")	32
22. 3	Schéma synthétique du PSL	32

2. 3	Fonctions assurées par ISDOS	34
23. 1	Fonction de gestion de base de données	34
23. 2	Fonction de contrôle	41
232. 1	Contrôle du PSL Input	41
232. 2	Contrôle de cohérence	42
232. 3	Contrôle de complétude	45
23. 3	Fonction de documentation	47
233. 1	Types de rapports	47
233. 2	Présentation des rapports	49
23. 4	Fonction de dictionnaire	50
234. 1	Généralités	50
234. 2	Fonctions d'un dictionnaire	52
234. 3	Classification des dictionnaires de données	55
234. 4	Utilisation d'un dictionnaire de données	56
234. 5	Dictionnaire de la version 4.2 d'ISDOS	57
3.	MISE EN OEUVRE DE LA METHODE D'ANALYSE A L'AIDE D'ISDOS	60
3. 1	Méthode d'utilisation d'ISDOS	61
31. 1	Considérations générales	61
31. 2	Métastructure représentée à l'aide d'ISDOS	65
31. 3	Données initiales de la méthode	72
313. 1	Utilisation des paragraphes de commentaires	72
313. 2	Cahier des charges	74
313. 3	Cellule d'activité	74
313. 4	Etape 1. Description de la métastructure du système	77
313. 5	Etape 2. Description des spécifications fonctionnelles proprement dites	80
3135. 1	Description des données	80
3135. 2	Description des traitements	92
31. 4	Corps de la méthode	98
314. 1	Contrôles	98
314. 2	Production des OUTPUTS	98
3142. 1	Rapports existants	98
3142. 2	Rapports modifiés	107

4.	EVALUATION DE LA MISE EN OEUVRE DE LA METHODE D'ANALYSE A L'AIDE D'ISDOS	112
4. 1	Limites	112
41. 1	Données initiales de la méthode	112
41. 2	Fonctions d'ISDOS	115
412. 1	Fonction de contrôle	115
412. 2	Fonction de documentation	115
41. 3	Outputs de la méthode	115
5.	CONCLUSIONS	116
5. 1	Apport de l'utilisation d'un logiciel d'aide à l'analyse	116
5. 2	Apport d'ISDOS	117
5. 3	Extensions	118
53. 1	Extensions de la version 4.2	118
531. 1	Modifications du PSL	118
531. 2	Extensions du PSA	121
5. 4	Métagénérateur	122
	Bibliographie	124

REMERCIEMENTS

Nous remercions toutes les personnes qui nous ont aidé dans l'élaboration du présent travail et notamment:

J. FICHEFET, Directeur de l'Institut d'Informatique, FNDP
F. BODART, Professeur, FNDP
Y. PIGNEUR, Assistant, FNDP

Enfin nous remercions tous les membres du centre de calcul ainsi que les opérateurs du centre de la RTT pour l'aide qu'il nous ont apportée pendant la phase pratique de ce travail.

INTRODUCTION.

1. Présentation du contenu du mémoire.

L'objectif de ce travail est d'étudier les modalités de mise en oeuvre de la méthode d'analyse fonctionnelle en vigueur à l'institut d'informatique à l'aide du logiciel d'aide à l'analyse ISDOS.

Dans l'introduction, nous avons voulu introduire le concept de logiciel d'aide à la conception.
Nous allons continuer dans les deux chapitres suivants à délimiter le cadre dans lequel se situe ce travail.

Dans le chapitre 1, nous allons décrire la méthode d'analyse fonctionnelle en nous attachant à décrire quels en sont:

- les Inputs
- le corps: c'est à dire le travail d'analyse proprement dit
- les Outputs

Dans le chapitre 2, nous exposerons brièvement en quoi consiste le logiciel ISDOS, en accordant une attention particulière aux objectifs d'ISDOS et au modèle d'ISDOS ainsi qu'à la fonction de dictionnaire de données.

La mise en oeuvre de la méthode d'analyse à l'aide d'ISDOS vous sera présentée dans le chapitre 3.
On étudiera:

- les spécifications d'entrée
- les fonctions d'analyse
- les rapports obtenus en sortie

sous la forme d'une méthodologie d'utilisation d'ISDOS.

Le chapitre 4 sera consacré à l'évaluation de la mise en oeuvre de la méthode d'analyse à l'aide d'ISDOS.
Nous traiterons des limites d'ISDOS en ce qui concerne:

- la spécification des Inputs de la méthode d'analyse
- les fonctions de contrôle et de documentation du logiciel ISDOS
- les rapports produits par ISDOS.

Dans le dernier chapitre de ce travail, nous dégagerons l'apport de l'utilisation d'un logiciel d'aide à l'analyse.
Nous ferons de même en ce qui concerne ISDOS.
Enfin , nous envisagerons quels pourraient être des prolongements possibles à ce travail.

2. Introduction aux logiciels d'aide à l'analyse.

Dans cette deuxième partie de l'introduction, nous allons vous présenter très brièvement les outils d'aide à la conception.(1)
Nous vous parlerons des objectifs de ces logiciels, de leurs fonctions principales et enfin de leurs composants.
Enfin, nous verrons qu'ils sont un exemple de ce que l'on appelle la Conception Assistée par Ordinateur.

Logiciel d' aide à l'analyse.

Un logiciel d'aide à l'analyse est un ensemble de programmes qui sert d'instrument d'aide à la mise en oeuvre du processus de conception d'un système d'information.

1° Objectifs d'un logiciel d'aide à l'analyse.

* L'objectif principal est l'amélioration du processus de développement d'un système d'information.

  Cette amélioration touche surtout à l'amélioration de la productivité de tout le processus chargé de la conception du système d'information.

  Les moyens employés pour améliorer la productivité sont:

  - la mise à disposition des analystes de spécifications du système d'information qui soient: . plus fiables
    . plus complètes et plus précises
    . de meilleure qualité
    Ceci est réalisé en donnant aux utilisateurs du logiciel des dispositifs facilitant:
    . l'introduction de ces informations dans les bases de données du logiciel

---

(1) Dans la suite, nous utiliserons communément les termes logiciel d'aide à l'analyse et logiciel d'aide à la conception.

. le contrôle de ces informations
. la mise à jour de ces informations

- Les possibilités données aux analystes de disposer rapidement de toutes les informations nécessaires à l'accomplissement de leurs tâches.
  Ceci est réalisé par les fonctions de documentation de ces logiciels et les possibilités d'interrogation des bases de données gérées par ces logiciels.

  Tous ces moyens concourrent à réduire:

  . le temps consacré aux analyses et à la production des dossiers d'analyse;
  . le temps consacré à la programmation étant donné la qualité des dossiers d'analyse;
  . et enfin les coûts de plus en plus importants du personnel et du traitement par ordinateur.

  Dans le même ordre d'idée, l'amélioration de la maintenance est possible grâce à:

  . la centralisation des informations concernant le système dans une ou plusieurs banques de données;
  . à l'évolution rapide de l'impact du changement de tout élément du système sur les autres éléments du système;
  . l'évaluation des coûts de maintenance;
  . au déclenchement rapide du processus de mise à jour des données, de la documentation et des programmes.

* Enfin, ces logiciels poursuivent certains objectifs d'extensibilité.

  Un effort est fait pour qu'ils puissent:

  - accepter plusieurs formes d'entrées et proposer différents modes de représentation des résultats sous forme de rapports;

- offrir une pluralité d'utilisation. C'est à dire qu'ils devraient pouvoir être utilisés pour décrire un système d'information durant les différentes phases de son élaboration et ceci à différents niveaux de détail.
- être évolutifs: c'est à dire qu'ils devraient:
  . être capables, dans la mesure du possible, de prendre en considération les particularités de tout autre système d'information;
  . pouvoir s'adapter aux évolutions des idées en matière de système d'information.

2° Fonctions des logiciels d'aide à l'analyse.

a) Tout logicield'aide à la conception dispose d'une base de données et d'un système de gestion de base de données lui permettant de tenir à jour et d'interroger cette masse d'information concernant le système d'information

b) Au point de vue des contrôles de données; ils disposent de programmes qui:

   * contrôlent la syntaxe du langage décrivant les spécifications d'entrées;
   * construisent un dictionnaire des données introduites dans la base de données et de leurs utilisations;
   * contrôlent la cohérence et la complétude du système d'information pour découvrir les risques de redondance, les risques d'erreurs et les erreurs.

c) En ce qui concerne les rapports, ils offrent généralement un nombre fixé de types de rapports. Cependant il est possible de paramétrer chaque type de rapport, ce qui ouvre d'énormes possibilités aux utilisateurs.

En voici quelques exemples:

- Dictionnaire de données.
- Edition de matrices représentant l'utilisation des données par les process...
- Diagnostic d'erreurs.
- Edition de structures créées par les relations reliant les informations dans la base de données.
- Edition de résultats sous forme de diagrammes.
- Production de statistiques.

Cette liste est loin d'être exhaustive en ce qui concerne les possibilités des logiciels d'aide à la conception.

d) Les possibilités d'extensibilité varient d'un logiciel à l'autre. Cependant en règle générale, deux cas se présentent très souvent.

* Il est possible de créer assez facilement de nouveaux rapports.
* Il est beaucoup plus difficile de modifier le modèle de représentation du système d'information et le langage de spécification d'entrée.

e) Pour ce qui est de la standardisation, un effort semble être fait pour rendre les logiciels portables.
En effet, les programmes sont généralement écrits dans un langage de haut niveau plus ou moins standardisé.
Par exemple - BYBLOS développé par la SLIGOS est écrit en COBOL standard.
- ISDOS est écrit en FORTRAN 4 standard.

Cependant, le même effort n'est pas toujours fait en ce qui concerne le système de gestion de base de données.
C'est pourtant le cas pour ISDOS qui utilise ADBMS dérivé de CODASYL et qui est écrit en FORTRAN 4.

3° Composants des logiciels d'aide à la conception et organisation générale.

a) Modèle.

Par modèle on entendra la représentation mémorisée dans l'ordinateur du système en cours de conception et de production.

La description du modèle comprendra trois parties principales:

* La structuration et la nomenclature:
  - des données
  - des traitements

  Ceci est plus ou moins détaillé et varie d'un logiciel à l'autre.

* La représentation:

  Elle consiste surtout à donner des informations sur les formats ou sur les valeurs des données qui représentent les éléments du système d'information.

* Le fonctionnement:
  - Statique: description des traitements, de leurs entrées et de leurs sorties.
  - Dynamique: définition des fréquences des traitements, des fréquences et des volumes des entrées et des sorties et enfin des successions d'exécutions des traitements avec les évènements déclencheurs.

b) Méthode.

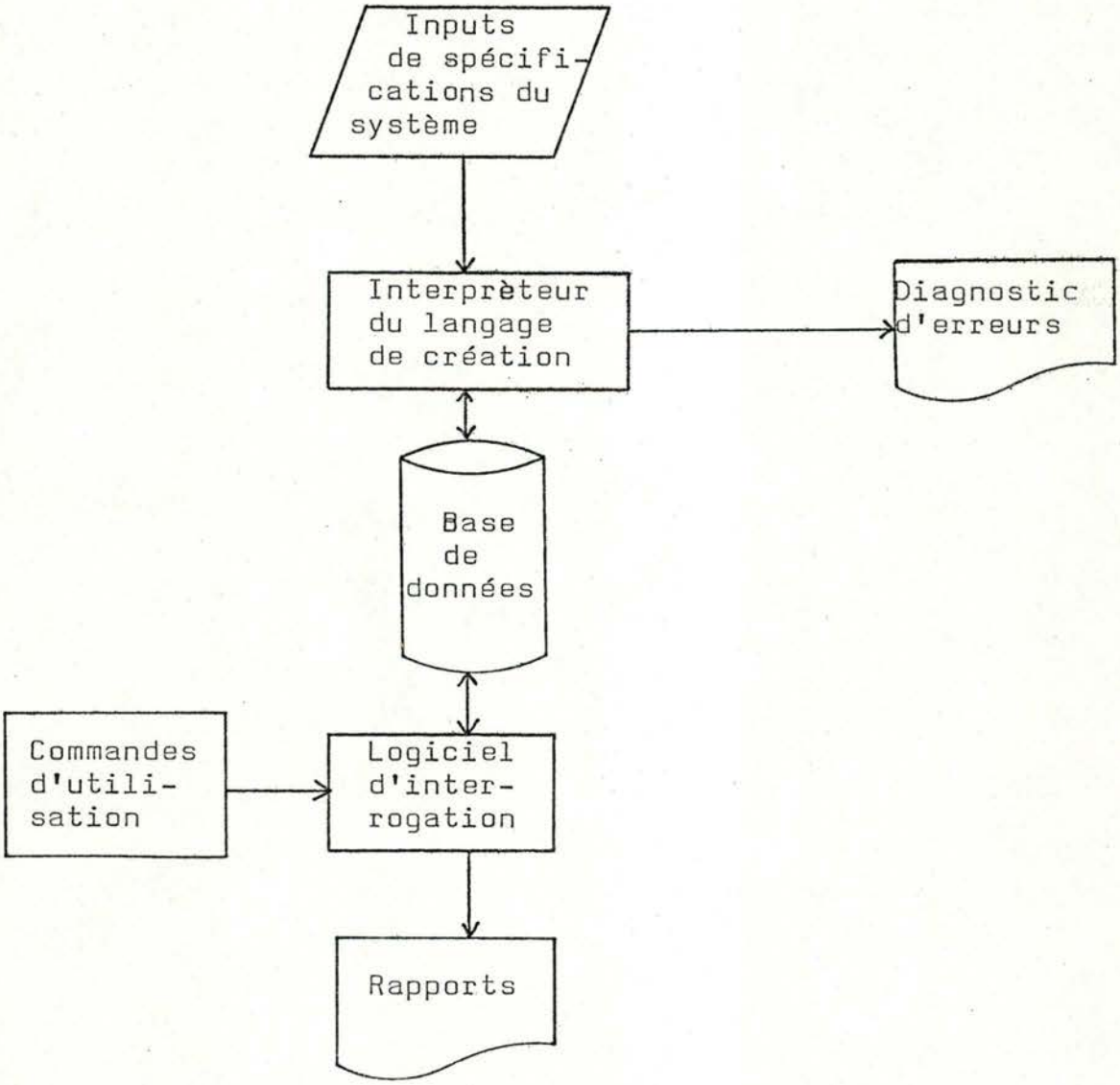
Cette méthode serait surtout un type d'approche du problème de l'analyse: (TOP-DOWN, par les INPUTS, par les OUTPUTS...).
Mais en pratique, il s'est avéré que les outils d'aide à la conception se voulaient le plus indépendant possible de toute méthode particulière.

c) Langage.

Le langage est nécessaire pour:

* d'une part exprimer les exigences de l'utilisateur sous la forme de spécifications.
* et d'autre part pour permettre à l'utilisateur de demander les rapports désirés avec les paramètres adéquats.

d) Organisation générale d'un logiciel d'aide à la conception.



Ces logiciels d'aide à la conception font partie d'un domaine d'étude plus vaste. Il s'agit de la Conception Assistée par Ordinateur. Les recherches dans ce domaine ont pour principale origine un besoin nouveau rencontré surtout chez les ingénieurs.

L'ingénieur souhaite disposer directement de l'ensemble des algorithmes et des données dont il a besoin pour définir son projet sans transcriptions ou transformations manuelles de données, génératrices d'erreurs et dispendieuses de temps.

Dans ce type de recherche, les problèmes posés sont de trois ordres.

* Il faut définir pour chaque domaine d'application la méthodologie de la conception ainsi que l'aide que peut lui apporter l'ordinateur.

* Il faut disposer d'outils informatiques pour:
  - recueillir, structurer et accéder aux données.
  - analyser les langages de communication.
  - définir et traiter les interactions.

* La complexité et le coût des systèmes réalisés impliquent des efforts particuliers quant à leur portabilité, leur modularité et leur adaptation rapide à d'autres problèmes similaires.

CHAPITRE 1. METHODE D'ANALYSE FONCTIONNELLE.

Ce chapitre va tout d'abord tenter de replacer l'analyse fonctionnelle dans le processus d'élaboration d'un système automatisé.

Nous dégagerons ensuite les concepts sur lesquels repose cette méthode. Ils sont d'ordre organisationnel et informationnel.

Enfin, nous détaillerons cette méthode pour en analyser:

- les INPUTS;
- le CORPS : c'est à dire les tâches à effectuer pour analyser le système d'information ou la partie du système d'information en cours d'élaboration;
- les OUTPUTS.

Analyse fonctionnelle:

> L'analyse fonctionnelle a pour but de préciser avec rigueur la description fonctionnelle des applications à automatiser. On peut la considérer comme un affinement de l'analyse de conception.
>
> BODART, 1977 - 1978.

1. 1 Intégration de la méthode dans le processus de conception d'un système informatique.

11. 1 Que trouvons-nous en amont?

En amont, nous trouvons l'analyse de conception qui fournit le dossier de conception.

Ce dossier se compose principalement:

* d'un Projet Cadre lui-même constitué de

  - une critique de l'existant;
  - la description des applications pour un domaine et un horizon donné;
  - contraintes de faisabilité et d'objectifs...

* de propositions de solutions et des critères qui ont conduit au choix de cette ou ces solutions dans l'ensemble proposé.

* du cahier des charges qui est en fait un affinement du projet cadre suite au choix précédent.

* de la définition d'un planning d'exécution des phases ultérieures et d'un budget de réalisation.

11. 2 Que trouvons-nous en aval?

En aval, nous trouvons l'analyse de programmation qui produit le dossier de programmation.

Elle a pour but de préparer l'implémentation, en vue de l'exploitation des solutions logiques dégagées au cours de l'analyse fonctionnelle.

Ce dossier comprendra notamment:

* le choix d'une implémentation qui optimise l'utilisation des ressources hardwares et softwares.

* la description rigoureuse et détaillée des programmes à réaliser sur base du dossier d'analyse fonctionnelle.

* les jeux de tests utilisés au cours de la mise au point des programmes.

1. 2 Méthode proprement dite.

12. 1 Concepts méthodologique.

Avant d'exposer ces concepts, nous engageons le lecteur à consulter la documentation qui concerne tous ces concepts. Il trouvera les renseignements voulu à la référence (1).

121. 1 Cellule d'activité.

Il s'agit par définition d'un centre d'activité homogène dans le temps et dans l'espace, doté de ressources et pourvu de règles de comportement nécessaires à son fonctionnement.

Elle sera caractérisée par le fait qu'elle est à la fois un élément organisationnel et informationnel du système.
Elle définit 3 types d'éléments du système:

* elle identifie un ensemble d'unités d'information qu'elle:
  - reçoit en Entrée;
  - produit en Sortie;
  - garde de façon interne.

* elle identifie les ressources humaines et matérielles nécessaires à l'accomplissement de son activité.

* elle identifie les règles de comportement appliquées à l'emploi des ressources et à l'exécution des traitements.

121. 2 Sous-structure des données.

La structure des données utilise les concepts d'entité, d'association et de propriété-valeur.
Voici un bref rappel de ces trois concepts:

* Entité.

" Une entité est ce qu'un individu ou un groupe voit comme un tout, ayant une existence propre.
Une entité est caractérisée par un ensemble de propriétés quantitatives et qualitatives et un comportement permanent; le nombre de propriétés et la permanence sont fonction du point de vue choisi."

Exemple: un client, un produit, une usine...

* Association.

" Une association est un ensemble de deux ou plusieurs entités où chacune assume un rôle donné. Une association peut posséder plusieurs propriétés."

L'existence d'une association est contingente à l'existence des entités qu'elle relate.

Exemple: ligne de commande est une association entre"commande"et"produit".

* Propriété-valeur.

" Une propriété appartenant à une entité ou une association est une qualité que les individus attribuent à cette entité ou cette association."

L'existence des propriétés attribuées est contingente à l'existence des entités ou de l'association concernée.

Plus formellement, une propriété est une relation (fonction) entre:

- une entité ou une association;
- et une ou plusieurs valeurs (objet) qui caractérisent l'entité ou l'association.
  ex.: le numéro, le nom, l'adresse, sont des propriétés attribuées à un client.

En fonction du modèle de structuration des données on établit la nomenclature que voici:

* Donnée élémentaire: Elément atomique correspondant au constituant le plus élémentaire de la nomenclature des données.

  - Elle n'a pas d'existence intrinsèque, indépendante des usages qui en sont faits au sein de l'organisation.

  - Une donnée élémentaire n'a de signification que dans une information élémentaire.

* Information élémentaire:

  Elle définit la configuration minimale de données élémentaires qui désigne une propriété d'usage associée à un objet au cours d'une période de temps ou à une date donnée.

  - Par objet on entend une Entité ou une Association.

* Unité d'information:

  Groupement des propriétés d'usage d'un même objet utilisées par un traitement logique homogène.

  - Cette notion d'unité d'information fournit à l'utilisateur et à l'analyste une synthèse des propriétés d'usage associées à un même objet dans un contexte homogène(celui de la cellule d'activité).

* Structure logique:

Ensemble structuré des unités d'information dont on a besoin pour la réalisation d'un traitement homogène.

- Elle comprend des entités, des associations et des propriétés d'usage de ces objets ainsi que des contraintes de cardinalité.

- La structure logique définit la configuration d'information nécessaire à l'exécution du traitement homogène d'une"phase"dans le cadre d'une cellule d'activité. (1)

* Structure conceptuelle:

Intégration des structures logiques associées aux traitements logiques qui réalisent les applications définies dans un projet cadre. (1)

- Elle est en fait un modèle général fournissant aux utilisateurs une représentation non redondante et exhaustive des informations élémentaires ainsi qu'une description de l'ensemble des contraintes d'intégrité.

121. 3 Sous-structure des traitements.

1. Modèle des traitements.

Le concept de traitement est lié à une activité qui consiste à transformer des Inputs en Outputs.

- Ces INPUTS et ces OUTPUTS seront des données qui font partie de la nomenclature des données décrites précédemment.

- Les traitements sont déclenchés par des évènements dont la survenance est régie par des conditions.
  ex.: Si le numéro de client n'existe pas, la phase de mise à jour du fichier client est déclenchée.

---

(1) Les notions de phase et d'application seront définies dans la nomenclature des traitements.

- Les traitements sont décomposables en d'autres traitements.

Au vu de ce concept on peut dire que la distinction entre traitement et donnée n'est pas intrinsèque mais d'ordre méthodologique.

En effet l'existence d'un traitement est contingente à l'existence des données inputs et des données outputs, et à la transformation de ces Inputs en Outputs.

A la limite on pourrait dire que les données font partie intégrante du traitement.

2. En fonction de ce modèle des traitements, on établit la nomenclature des traitements de la façon suivante:

* Fonction:
Elle correspond au niveau d'individualisation élémentaire qui reflète de façon arbitraire le degré de modularité choisi pour la programmation.

  - Le point terminal d'une fonction sera déterminé par un point de décision donnant lieu à la création ou à la modification d'une réalisation d'une unité d'information.

  - A la limite une fonction peut ne posséder qu'un seul point d'entrée et un seul point de sortie.

* Phase:
Traitement possédant une unité spatiotemporelle d'exécution

  - Une phase s'accomplit dans le contexte d'une cellule d'activité en se basant sur les informations définies dans une structure logique.

  - Elle est décomposable en fonctions.

* Application:
Elle décrit l'enchaînement des phases relatives à un flux d'informations, flux homogène et permanent dans le temps.

  - Elle se base sur la configuration des informations définissant une structure conceptuelle de données.

- Le flux d'information identifie pour l'application des Inputs bien définis et des Outputs bien définis.

* Sous-système:
Il regroupe des applications non indépendantes dans le cadre des interactions" exécution-gestion ".

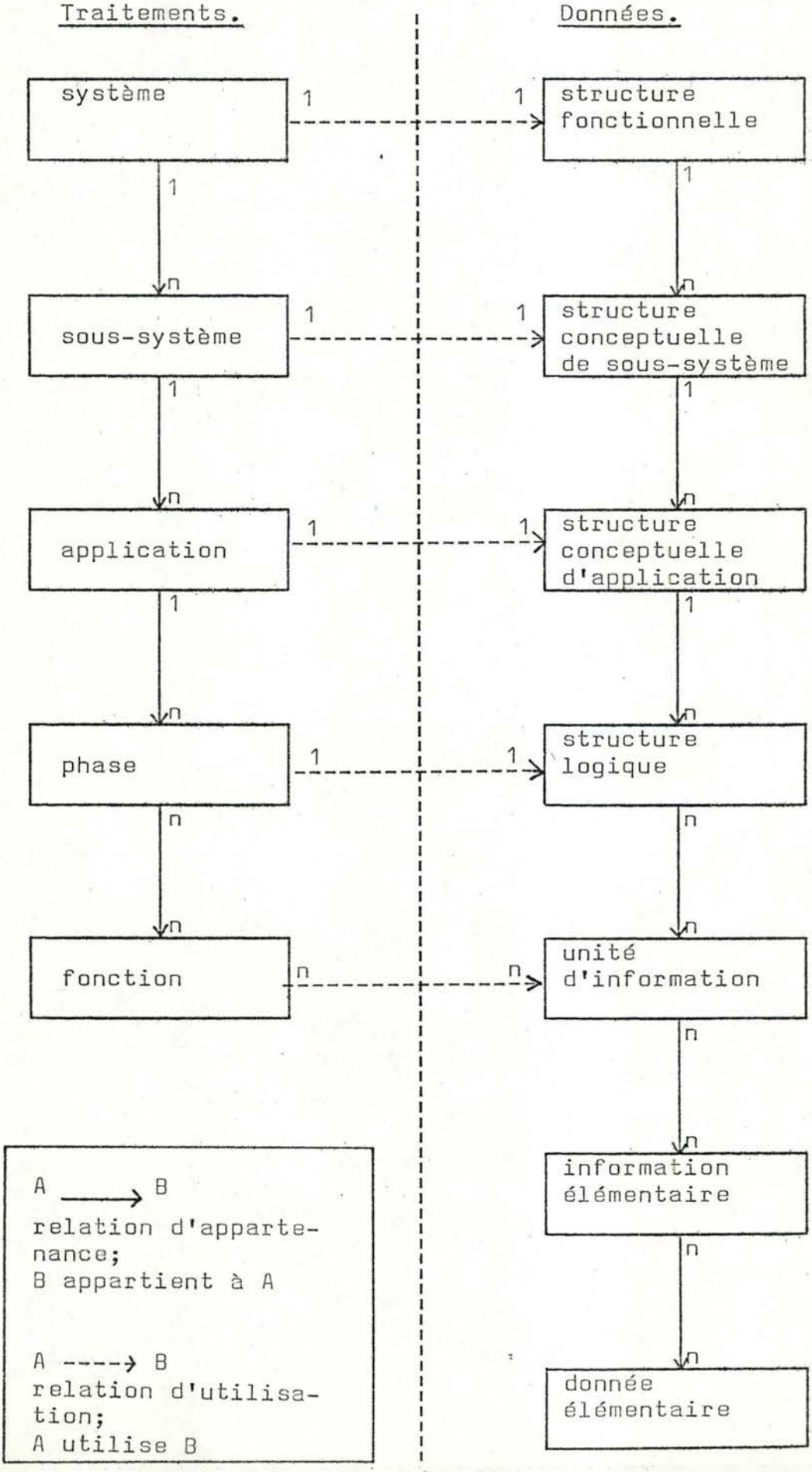
- Il se base sur l'intégration des structures conceptuelles de données de ses applications. Le résultat de cette intégration pourrait être appelé structure conceptuelle de sous-système.

* Sytème informatique:
Il est constitué de l'ensemble des flux, des fichiers d'information et des traitements qui décrivent l'état et le fonctionnement de l'organisme réel ainsi que le processus de décision.

- Toutes les données de ce système informatique seront regroupées dans ce que nous appellerons la structure fonctionnelle des données.

Afin de résumer tout ce qui a été dit dans ces deux nomenclatures, nous proposons au lecteur de synthétiser tous ces éléments et les relations qui existent entre eux par un schéma que vous trouverez à la page suivante.

121. 4 Intégration des deux nomenclatures en une métastructure du système d'information.



12. 2 Structure de la méthode.

122. 1 Données de base.

Les données de base seront contenues dans le dossier de conception.

Ou plus exactement, ce qui nous intéresse, ce sont toutes les informations concernant la ou les solutions informatiques retenues et décrites dans le cahier des charges.

Il y aura des informations concernant:

* l'environnement de la solution: organisation,planning..

* la solution retenue:

  - liste des données et des cellules d'activités concernées par ces données;
  - liste des traitements;
  - liste des contraintes de mise-en-oeuvre.

122. 2 Corps de la méthode.

L'analyse va consister en la production du dossier de développement qui contiendra:

* la description des données suivant la nomenclature des données;

* la description des traitements suivant la nomenclature des traitements.

Ces descriptions doivent être claires et détaillées. Ceci est possible grâce à l'utilisation de formulaires dont nous vous parlerons au paragraphe suivant.

122. 3 Outputs de la méthode.

Nous proposons ces Outputs dans un certain ordre, mais nous laissons toute liberté à l'utilisateur pour constituer le dossier de développement dans un ordre qu'il agrée.

Nous allons examiner les Outputs concernant les données puis ceux qui concernent les traitements.

A. Données.

1. Dictionnaire des informations.

Au cours de tout le processus d'élaboration

d'un système d'information, on créera ce dictionnaire et on le mettra régulièrement à jour.

Pour faire cela, l'analyste pourra utiliser un document du type DD/7.

Tous les documents dont nous parlerons maintenant se trouvent en annexe.

Ce dictionnaire des unités d'informations offre plusieurs avantages:

* Il permet de tenir à jour une documentation sous forme standard.

* Il est aisément communicable aux utilisateurs des services du centre informatique.

* Il servira d'outil d'analyse au cours de tout le processus d'analyse.

Malgré ce que nous avons dit au début de ce paragraphe 122. 3, nous estimons que ce dictionnaire doit se trouver au début du dossier de développement.

2. Tables de codification.

Il nous semble que ces tables de codification sont fortement liées au dictionnaire de données.

De toute façon il existe dans le document DD/7 une zone permettant de faire référence à la table de codification décrite à l'aide du document DD/8.

3. Répertoire d'Inputs.

Le document DC/3 permet de faire la liste des Inputs avec leurs cellules d'acquisition, leurs fréquences, leurs volumes.....

4. Répertoire d'Outputs.

Le document DC/4 donnera la liste des Outputs avec les cellules d'origines et les destinataires.

5. Description des unités d'information Inputs et Outputs.

L'analyste utilise le document DD/9 qui reprend quelque peu le formalisme de déclaration d'une structure en COBOL.

6. Le Document DD/10

présente une analyse des volumes des unités d'information par périodes.

7. Structure logique et structure conceptuelle.

L'analyste décrira ces deux structures dans le document DD/12 sous la forme d'un diagramme.

* La structure logique est la configuration d'information sur laquelle travaille chaque phase.

* En intégrant les structures logiques de toutes les phases d'une application, on obtient la structure conceptuelle d'une application.

* On peut remonter ainsi jusqu'au niveau du système.

B. Traitements.

1. Diagramme des flux d'information.

Le document de type DC/2 ou DD/5 permet de décrire les flux d'information. Nous l'utiliserons pour les applications, les phases et les fonctions.

* Application:
Pour chaque application on reprendra:

- les unités d'information entrantes.
- les unités d'information sortantes
- le ou les événements déclencheurs et la fréquence d'exécution.

* Phase:
Pour toutes les phases d'une application, on va représenter l'enchaînement des phases, avec pour chaque phase le même type information que pour l'application avec en plus les délais de transmission des unités d'informations ou des documents Inputs dans la cellule d'activité.

* Fonction:
Pour toutes les fonctions d'une phase, on représentera l'enchaînement de ces fonctions, ave pour chaque fonction le même type information que pour l'application et la phase.
L'analyste utilisera le document DC/2 ou DD/5 en prenant bien soin de mentionner à quel élément de la nomenclature des traitements se rapporte ce document .

2. Description des traitements.

Le document DD/1 permet de décrire en détail le traitement effectué dans un langage naturel, que ce soit une application, une phase ou une fonction.

3. Règles de traitement.

Le document DD/2 sert à décrire de façon formelle les règles de traitement d'une fonction.
Le formalisme employé pourrait être un pseudo-langage ou la technique des tables de décision.

4. Règles de contrôle.

Le document DD/3 permet de décrire les fonctions de contrôle avec pour chaque condition de validité:
le message d'erreur correspondant en cas de contrôle positif et un code de sévérité.

5. Documents de synthèse.

* Le document DD/11 donne une synthèse des modes d'utilisation des unités d'information par les phases d'une application.

* D'autres documents déjà décrits nous semblent des documents de synthèse; c'est le cas des documents:

  - donnant la liste des Inputs et Outputs d'une application;
  - représentant les enchaînements de phases et de fonctions;
  - décrivant la structure logique de chaque phase.

Le lecteur trouvera en annexe un exemple d'utilisation de tous ces documents.

C. Informations complémentaires.

Une troisième partie non négligeable reprendra:

* les objectifs et les contraintes fonctionnelles des applications analysées;

* la description des ressources hardware et software qui permettent de mettre en oeuvre la solution du cahier des charges.

- une analyse des coûts fonctionnels et de la faisabilité fonctionnelle de la solution retenue. Cette liste est loin d'être exhaustive, mais nous semble représentative des informations dites " complémentaires ".

CHAPITRE 2. INTRODUCTION AU LOGICIEL ISDOS.

Nous essayerons dans ce chapitre de familiariser le lecteur avec le logiciel d'aide à l'analyse ISDOS. (1)

Nous commencerons par présenter les objectifs d'ISDOS. Ces objectifs concerneront le modèle et les langages PSL (Problem Statement Language) et PSA (Problem Statement Analyser).

Ensuite, nous ferons une rapide description des éléments d'ISDOS. Cette partie concernera les objets et les attributs ainsi que des relations entre les objets et les attributs.

Et pour terminer cette deuxième partie, nous vous présenterons un schéma de synthèse du PSL.

Dans la troisième partie, nous exposerons quelles sont les fonctions assurées par ce logiciel d'aide à l'analyse. On distingue quatre grandes fonctions:

- la fonction de gestion de la base de données;
- la fonction de contrôle;
- la fonction de documentation;
- la fonction de dictionnaire qui sera développée plus particulièrement.

2. 1 Caractéristiques générales.

21. 1 Caractéristiques du modèle.

Le modèle utilisé dans ISDOS se veut assez complet pour offrir au niveau conceptuel un reflet acceptable d'un système d'information réel.

C'est à dire qu'il prend en charge plusieurs aspects caractérisant un système d'information:

* le fonctionnement du système

  - fonctionnement statique
  - fonctionnement dynamique

* la structuration des éléments du système

  - traitements
  - données

---

(1) La version utilisée pour ce travail est la version 4.2. Le lecteur est invité à consulter la littérature traitant d'ISDOS (10- 19).

* les renseignements nécessaires pour que les résultats obtenus à partir d'ISDOS offrent à l'utilisateur une bonne représentation du système étudié.

* les informations nécessaires pour localiser le plus exactement possible les éléments du système

  - localisation dans l'espace
  - localisation dans le temps

21. 2 Caractéristiques du langage.

ISDOS se compose de deux parties que nous ne qualifierons pas de langage : le PSL et le PSA.

Le PSL peut être considéré comme un langage, alors que la terminologie ISDOS préfère le terme analyser au terme PSA. Cette distinction provient certainement du fait que le PSA est composé d'un Analyseur du PSL

et d'un langage de spécification des types de fonctions et de rapports demandés par l'utilisateur

et enfin d'un Analyseur de ce langage de spécification de fonctions

212. 1 PSL.

* Afin d'assurer la facilité d'utilisation du PSL, les concepteurs du logiciel ISDOS ont fait du PSL un langage non-procédural. Il est non- procédural en ce sens qu'il n'existe pas de variables, d'instructions de branchement et que les instructions peuvent être mises dans n'importe quel ordre. Ceci est vrai au niveau des sections, mais également au niveau des statements qui composent ces sections.

  De ce fait, le PSL a pour mission de laisser tout loisir à l'utilisateur de décrire ses problèmes avec le niveau de détail désiré.

* D'après ce qui précède, on peut dire que le PSL se veut un langage de "transmission d'idées".

  Il doit donc remplir certaines conditions:

  1. Condition de lisibilité.

     Ce langage doit être le plus naturel possible et avoir un formalisme facilitant son interprétation par l'Analyseur.

2. Facilité de formation des utilisateurs.

Cette condition est une conséquence directe du point précédent.

3. Indépendance par rapport à toute méthode d'analyse particulière.

Cette condition n'est certainement pas remplie. On pourra s'en rendre compte dans la suite de ce travail.

4. Possibilités d'extensions.

Il existe certaines possibilités d'extensions qui cependant sont limitées.

212. 2 PSA ou Analyser.

Il poursuit deux objectifs principaux:

1. Offrir une présentation la plus proche possible du réel perçu.
Ceci est réalisé en assurant une bonne gestion de la base de données qui représente le modèle ISDOS.

2. Offrir les meilleurs services aux utilisateurs en ce qui concerne l'aide à l'analyse.
Cet objectif est réalisé par les fonctions que nous développerons dans le paragraphe 2.3 .

2. 2 Description des éléments du modèle d'ISDOS.

Le modèle d'ISDOS se compose de trois types d'éléments:

* les objets caractérisés par leur type
* les attributs relatifs à ces objets
* les "relationships" entre tous les éléments du modèle.

22. 1 Objets et attributs.

A. Types d'objets.

Nous les avons subdivisés en 5 classes.

1° Environnement du système à étudier.

INTERFACE ou ORGANISATIONAL UNIT : il s'agit d'un organe de communication du système avec son environnement.

Il génère des INPUTS à destination du système et il reçoit des OUTPUTS générés par le système.

2° Activités du système.

PROCESS: il s'agit d'un traitement qui contribue à transformer les INPUTS en OUTPUTS.

3° Données du système.

* INPUT: Entrées du système venant des interfaces.

* OUTPUT: Sorties du système à destination des interfaces.

* ENTITY: Il s'agit des entités sur lesquelles le système travaille. Une entité peut se décomposer en groupes et en éléments de même que les INPUTS et les OUTPUTS.

  - GROUP: Equivalent de la notion de groupe en COBOL. Un groupe est composé d'éléments.

  - ELEMENT: Il correspond à la notion de donnée élémentaire.

* RELATION: cet objet permet de décrire les relations sémantiques liant deux objets ENTITIES.

* SET: la notion de set correspond à celle de collection de données, en ce sens qu'un SET peut être décomposé en:
  - INPUTS
  - OUTPUTS
  - ENTITIES

4° Dynamique du système.

* EVENT: cet objet peut déclencher l'exécution d'un PROCESS et peut être causé par l'initialisation ou la fin d'un PROCESS, ou encore par le changement de l'état d'une condition.

* CONDITION: Elle peut suivant son état TRUE ou bien FALSE causer un événement.

* INTERVAL: cet objet permet de définir un intervalle que l'on utilisera dans la clause HAPPENS pour donner des renseignements sur la localisation temporelle de certains éléments d'ISDOS.

5° Gestion du système.

* PROBLEM-DEFINER: permet de définir le responsable de l'objet pour lequel on mentionne le RESPONSIBLE-PROBLEM-DEFINER.

* MAIL-BOX: cet objet est utilisé pour décrire l'endroit où les questions ou les informations au sujet de la description en PSL d'un problème peuvent être envoyées.

* SOURCE: sert à décrire un objet externe au problème décrit en PSL, mais concernant des éléments du système que l'on décrit.

* SECURITY: tout objet peut avoir un nom ISDOS servant de sécurité.

B. Types d'attributs d'objets.

1° Noms considérés comme objets.

* ATTRIBUTE: cet objet permet de définir toute sorte de noms d'attributs concernant les objets d'ISDOS.

* KEYWORD: cet objet permet de doter les objets d'ISDOS de mots-clés. Ce qui différencie les KEYWORDS des ATTRIBUTES, c'est que seuls ces derniers permettent de donner des valeurs aux noms d'attributs. Ce sont les ATTRIBUTES-VALUES.

* SYSTEM-PARAMETER: il s'agit comme son nom l'indique d'un objet décrivant un paramètre du système à analyser. Tout comme l'attribut, le paramètre système peut recevoir une valeur. Celle-ci peut lui être donnée sous la forme (MINI , MAXI).

2° Noms auxiliaires.

* Commentaires.

Pour tout objet, il est possible de créer des commentaires sous la forme de texte libre.
En voici quelques exemples:

- DESCRIPTION; valable pour tous les objets décrits par une section.

-PROCEDURE; réservé à l'objet PROCESS.

- ETC...

* Quantificateurs.

- CARDINALITY: permet de chiffrer le nombre d'occurences d'ENTITIES, de SETS,...

- CONNECTIVITY: sert à quantifier les relations sémantiques entre les ENTITIES.

- Les valeurs chiffrées ne peuvent être que des entiers.

22. 2 Associations ("relationships").

Ces associations permettent de décrire les relations liant les types d'objets d'ISDOS.
En voici deux exemples:

1. PROCESS CALCUL-FACTURE;
   RECEIVES BON-DE-CDE;

   L'association RECEIVES indique que le PROCESS CALCUL-FACTURE reçoit l'INPUT BON-DE-CDE venant de l'environnement du système.

2. INPUT BON-DE-CDE;
   RECEIVED BY CALCUL-FACTURE;

   Il s'agit de la même relation qu'à l'exemple précédent, mais cette fois sous la forme passive. Cette dualité des "relationships" permet d'effectuer des contrôles de cohérence entre la même relation écrite sous les deux formes.

22. 3 Schéma synthétique du PSL.

Vous trouverez à la page suivante un schéma présentant les principaux objets d'ISDOS, ainsi que les principales "relationships".

_____ Relation de décomposition des objets d'ISDOS.

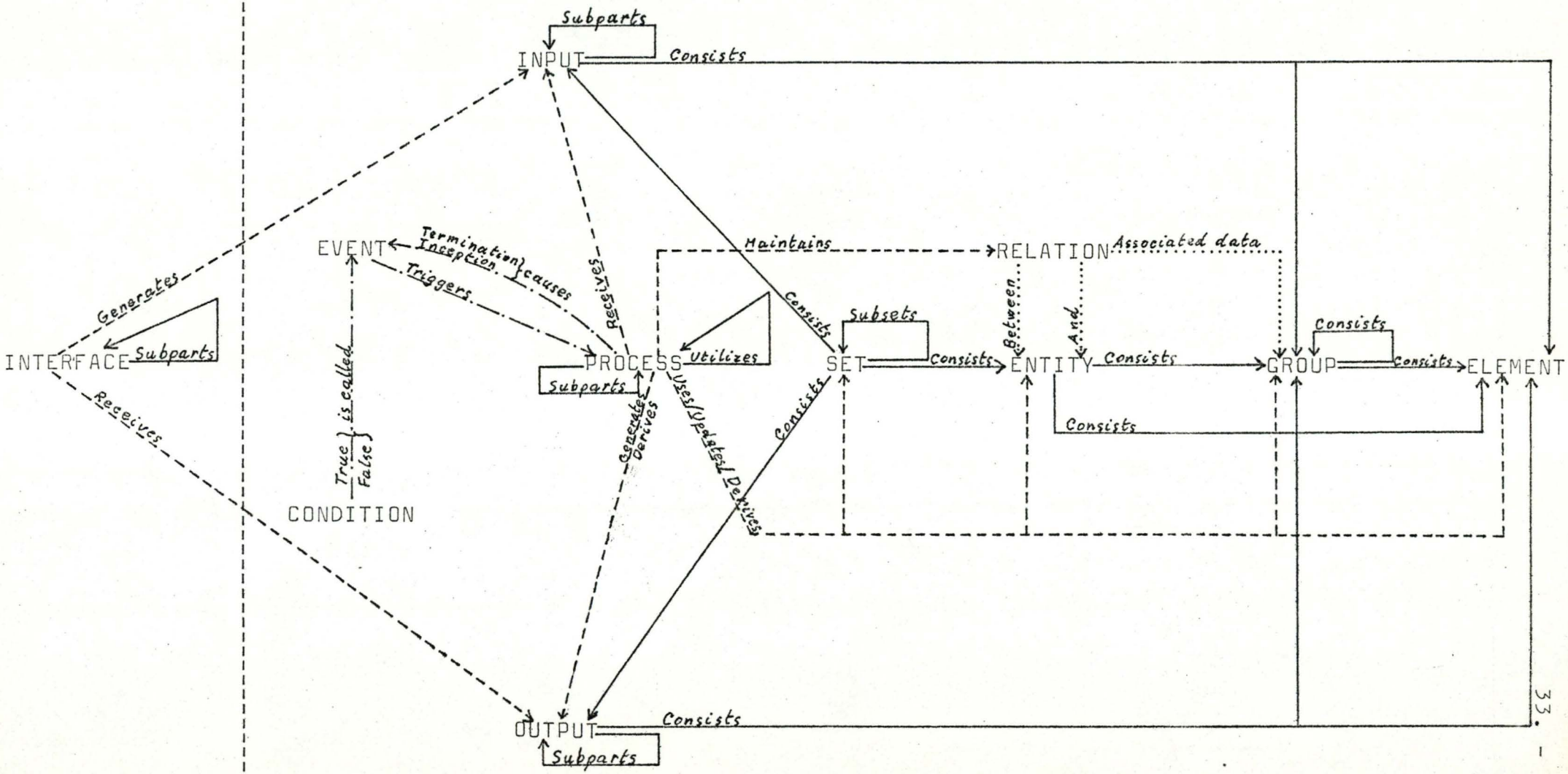
-------- Représente les relations des données avec les traitements ou les cellules d'organisation.

—·—·—·— Représente les relations décrivant le fonctionnement dynamique du système.

..... Représente les relations sémantiques que l'utilisateur désirera décrire entre les entités.

Environnement

Système à décrire



2. 3 Fonctions assurées par ISDOS.

Toutes ces fonctions sont assurées par l'Analyser qui, tout en assurant l'interprétation du PSL, constitue un langage composé de trois types d'instructions:

* Instructions de contrôle du PSA.

  exemple: SET
  Cette instruction permet entre autre de dire sur quelle base de données, on désire travailler. Elle indique également à l'Analyser le type d'outputs désirés: listing sur le terminal, à l'imprimante ou mémorisation dans un fichier catalogué.

* Instructions de gestion de la base de données.

  exemple:
  - IPSL: employée avec le paramètre UPDATE, cette instruction permet de créer la base de données àpartir des spécifications écrites en PSL.

  - DPSL: permet d'effacer des spécifications du PSL.

* Instructions de demande de rapports.

  Ces instructions sont paramétrables et permettent moyennant un rapport de sélection préalable, de varier les rapports obtenus à l'aide d'une même instruction PSA.

Les fonctions de l'Analyser, en dehors de l'interprétation, sont à classer en 3 catégories:

* fonction de gestion de base de données
* fonction de contrôle
* fonction de documentation

23. 1 Fonction de gestion de base de données.

La base de données d'ISDOS est de type CODASYL avec certaines restrictions et certaines extensions dont nous ne parlerons pas dans cette présentation sommaire de ADBMS. (1)

---

(1) Le lecteur est invité à consulter la documentation aux références (7 - 9)

A. Concepts de ADBMS.

1. Objets.

Il y a quatre sortes d'objets différents:
- les "items"
- les "records"
- les "areas"
- les "sets"

Item: C'est la plus petite unité de données qui puisse être définie dans ADBMS.
Un type d'item est identifié par son nom qui doit être unique dans le record concerné.
Il y a trois types d'items:

* les items simples: présents une fois dans le record.
* les items fixes répétés: répétés un nombre fixe de fois dans le record.
* les items variables répétés: répétés un nombre variable de fois dans le record.

Record: Un record dans la base de données est conceptuellement identique à un record dans un fichier conventionnel, en ce sens qu'il s'agit d'une collection d'items de données.
Comme principales caractéristiques d'un record on relèvera: son nom, son type, ses items, ...

Area: il s'agit d'un objet qui permet à l'utilisateur de sélectionner les pages dans lesquelles sont mémorisés différents types de records.

Set: le set est l'unité de base de la structure de la base de données.
Un type de set est défini dans le DDL, comme ayant un ou plusieurs types de records maîtres légaux et un ou plusieurs types de records membres légaux.

* Un set qui ne possède pas de membres est appelé "empty" set.

* Les critères d'ordre dans un set sont par exemple en cas de création les suivants:

  - FIFO: le record est ajouté au premier membre.
  - LIFO: le record est ajouté au dernier membre.
  - NEXT: le record est ajouté au membre suivant.
  - PRIOR: le record est ajouté au membre précédent.

  - SORTED: le record sera placé dans l'ordre ascendant de la clé de tri.
  - IMMATERIAL: il n'y a pas de différence entre l'ordre où les membres sont mémorisés et accédés.

2. Indicateurs de records courants.

Il y en a trois quien fait sont des pointeurs vers une occurence d'un record particulier:

* indicateur du type de record courant
* indicateur du maître courant d'un type de set
* indicateur du membre courant d'un type de set

3. Identification d'un record dans le programme UTILISATEUR.

Voici résumés dans un tableau les différentes méthodes d'identification et les arguments d'identification:

Méthode d'identification	Argument d'identification
par la clé de la base de données	clé de la base de données
comme maître courant	type de set
comme membre courant	type de set
comme record courant	type de record

4. Identification d'un set dans le programme utilisateur

On peut identifier un set en connaissant son type et un record maître particulier.

B. Structure de ADBMS concernant ISDOS.

Le DDL de la base de données se trouve à la figure1.

1. Records.

NAMREC: définit un nom défini par l'utilisateur.
L'item NAMET contient un nombre entier qui représente le type du nom.
L'item NAMEC contient le type du nom sous forme de caractère.

SYNREC: définit un synonyme pour un nom utilisateur. Il contient les deux items NAMET et NAMEC qui ont la même fonction que dans le record NAMREC.

NUBA, NUBB, NUBC, COM: Ces records sont utilisés pour relier les occurrences des records NAMREC en fonction des "relationships".
L'item RELTYP contient un code qui identifie le type de "relationship".
* NUBA est utilisé pour une connexion simple.
* NUBB et NUBC sont utilisés pour définir une connexion complexe.

* COM est utilisé pour spécifier la relation entre un record NAMREC et un record de commentaire COMLIN.
  L'item RELTYP contient un code identifiant le type de "relationship" commentaire.

NUMREC: permet de mémoriser un nombre entier. Il contient les items NAMET et NUMREC qui contiennent le code du type de ce nombre et sa forme en caractères.
L'item NUMVAL contient la valeur du nombre sous forme binaire.

COMLIN: permet de mémoriser une entrée commentaire de 72 caractères dans l'item LINEC.

AREC: Ces records sont utilisés comme mémoire tampon par l'Analyser pour sauver des pointeurs.
Les items sont:
* un item entier NMITEM
* deux items de clés de base de données DBKEYA et DBKEYB.
* un item de caractères CHITEM.

2. Sets.

ALLNAM: Il spécifie un ordre alphabétique entre tous les noms de base et les synonymes.

RELA: Il spécifie les connexions à main gauche des "relatioships" simples et complexes.

RELB: Il spécifie les connexions à main droite des "relationships" simples et complexes.

RELC: Il spécifie des connexions complexes impliquant plus de deux noms utilisateurs.

ALINE: Il spécifie la relation entre un record COM et ses records associés COMLIN.

SYNFOR: Il définit la relation entre un nom de base et ses synonymes.

ALPHA, BYVAL: Ces deux sets sont utilisés en conjonction avec le record AREC par l'"Analyser".

3. Types de connexions.

Vous trouverez à la figure 2 les principaux types de connexions de base.

C. Possibilités de ISDOS grâce à DBMS.

Au vu de la structure de la banque de données d'ISDOS, on comprendra aisément les restrictions que nous rencontrerons dans la variété des contrôles et des rapports obtenus à l'aide des commandes du PSA.

RECORDS:

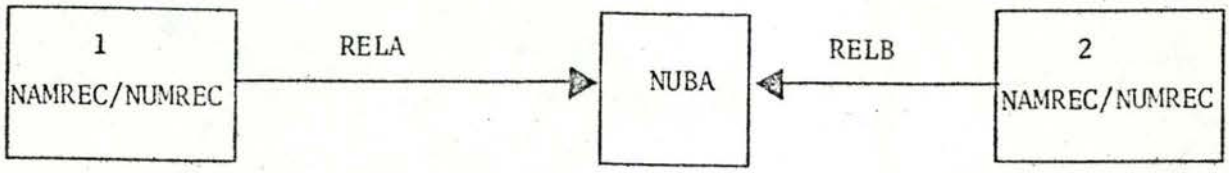
```
RECORD NAMREC
ITEM   NAMET   INTEG   16
ITEM   NAMEC   CHAR    30

RECORD SYNREC
ITEM   NAMET   INTEG   16
ITEM   NAMEC   CHAR    30

RECORD NUBA
ITEM   RELTYP  INTEG   16

RECORD NUBB
ITEM   RELTYP  INTEG   16

RECORD NUBC
ITEM   RELTYP  INTEG   16

RECORD COM
ITEM   RELTYP  INTEG   16

RECORD NUMREC
ITEM   NAMET   INTEG   16
ITEM   NAMEC   CHAR    30
ITEM   NUMVAL  INTEG   16

RECORD COMLIN
ITEM   LINEC   CHAR    72

RECORD AREC
ITEM   NMITEM  INTEG   16
ITEM   DBKEYA  DBKEY
ITEM   DBKEYB  DBKEY
ITEM   CHITEM  CHAR    30
```

SETS:

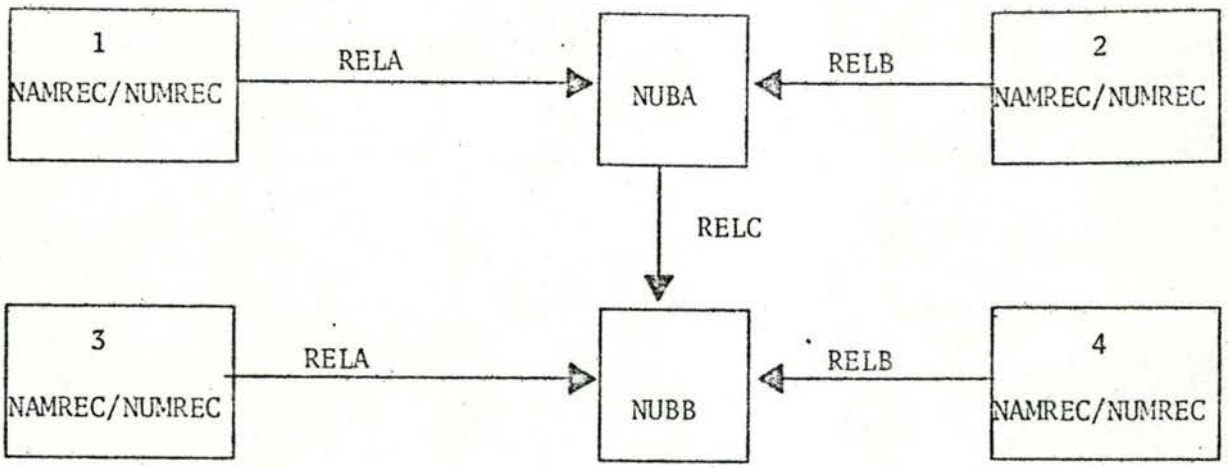
```
SET    ALLNAM SORTED   NAMEC
OWNER  SYSTEM
MEMBER NAMREC
MEMBER SYNREC

SET    RELA   SORTED   RELTYP
OWNER  NAMREC
OWNER  NUMREC
MEMBER NUBA
MEMBER NUBB
MEMBER NUBC
MEMBER COM

SET    RELB   SORTED   RELTYP
OWNER  NAMREC
OWNER  NUMREC
MEMBER NUBA
MEMBER NUBB
MEMBER NUBC

SET    RELC   NEXT
OWNER  NUBB
MEMBER NUBC
SET    ALINE  FIFO
OWNER  COM
MEMBER COMLIN

SET    SYNFOR SORTED   NAMEC
OWNER  NAMREC
MEMBER SYNREC

SET    ALPHA  SORTED   CHITEM
OWNER  SYSTEM
MEMBER AREC

SET    BYVAL  SORTED   NMITEM
OWNER  SYSTEM
MEMBER AREC
```

Figure 2: DDL for an Analyser Data Base

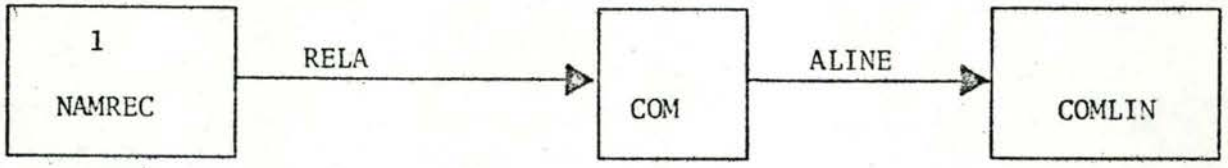
A. SIMPLE CONNECTION:



B. COMPLEX CONNECTION:



C. COMMENT ENTRY:



D. SYNONYMS:

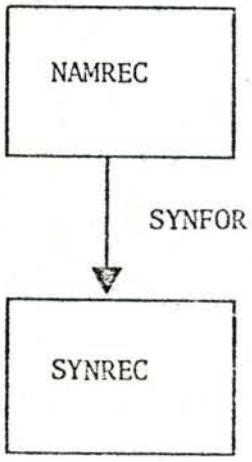


FIGURE 2. BASIC FORMS OF CONNECTION TYPES

23. 2 Fonction de contrôle.

Ces fonctions sont de trois types:

* le contrôle du PSL Input
* les contrôles de cohérence
* les contrôles de complétude

232. 1 Contrôle du PSL Input.

Le PSL étant un langage de spécification, l'Analyser va vérifier la syntaxe des instructions et produire des messages d'erreur s'il y a lieu. Le contrôle de syntaxe peut être fait dans plusieurs environnements possibles:

a. Interprétation sans références à la base de données

b. Interprétation avec références à la base de données.
Dans ce cas-ci, l'Analyser tient compte de ce qui a déjà été créé. Il pourra donc signaler que tel objet est déjà créé ou que tel type d'objet est incompatible avec la relation PSL qui l'emploie.

c. Interprétation avec mise à jour de la base de données.
En cas d'erreur, les instructions PSL erronées ayant un code d'erreur grave, ne donnent pas lieu à la mise à jour de la base de données. Ces instructions sont sautées jusqu'à l'instruction suivante ou la section suivante.

Une des grandes facilités du diagnostic d'erreurs, c'est le chaînage des erreurs. Ceci permet de voir ce qui est ou n'est pas pris en compte dans la gestion de la base de données. Et d'autre part cela permet de corriger immédiatement au terminal, sans sortir d'ISDOS, les erreurs commises au cours de l'IPSL qui vient d'être achevé.

Nous invitons le lecteur à consulter la documentation d'ISDOS, s'il désire plus de détails sur ce sujet.

232. 2 Contrôle de cohérence.

Nous allons examiner les rapports de l'Analyser qui permettent d'effectuer ce type de contrôle

1. ATTRIBUTES report.

Ce rapport permet non seulement d'avoir les objets avec leurs attributs, mais il permet également à l'analyste de voir:

* quels sont les objets qui normalement devraient posséder certains attributs.
* de même que ceux qui ne devraient pas posséder ces attributs.

2. CONTENTS-ANALYSIS report.

Utilisé avec l'option CONTAINED: ce rapport permet à l'analyste de voir quels objets, de niveau inférieur dans la structure des données,ne sont pas contenus dans les objets qui leur sont hiérarchiquement supérieurs.
Mais il faut faire attention, car un élément qui n'est contenu dans aucun groupe, entité, set, input ou output peut très bien être une donnée associée à une relation.

L'option CONSISTS permet de voir quels objets de niveau supérieur sont sans composants ou alors quels objets ont une structure similaire, ce qui peut être une cause de redondance que l'on désire éviter.

3. CONTENTS-COMPARISON report.

Ce rapport permet d'effectuer les mêmes contrôles de redondance que le rapport précédent.

4. DATA-PROCESS-INTERACTION report.

Ce rapport permet de connaître les relations d'utilisation des données par les traitements. Il permet d'effectuer plusieurs contrôles:

* utilisé avec le paramètre DATA, il permet de voir quelles données ne sont jamais utilisées par un PROCESS.
* utilisé avec le paramètre PROCESS, il permet de voir quels PROCESS n'utilisent aucune donnée.

Il permet également de vérifier la compatibilité entre les objets d'ISDOS et les "relationships".
exemple:
utilisation des instructions RECEIVES, GENERATES uniquement avec des INPUTS et des OUTPUTS.

5. ELEMENT-PROCESS-ANALYSIS report.

Ce rapport fournit des statistiques d'utilisation d'une entité pour une certaine période de temps (par exemple). Et il fait de même pour l'utilisation des autres données par les PROCESS.
Utilisé avec le PUNCHED-FILE de l'ELEMENT-PROCESS-USAGE report ,il permet de voir dans quelle mesure certains process pourraient être fusionnés s'ils utilisent les mêmes données, ou bien dans quelle mesure ils pourraient être éclatés en d'autres process.

6. EXTENDED-PICTURE report.

Ce rapport montre une vue générale de la structure des flux de données.
* Il peut aider à évaluer l'impact du changement d'un objet du système sur les autres objets.
* Il peut aussi aider à vérifier que des objets ne sont pas vides et à contrôler la compatibilité entre objets et relations PSL.

7. FREQUENCY report.

Ce rapport permet de tester si tous les objets logiquement reliés par leur fréquence sont groupés ensembles.
Il permet également d'avoir une idée ,par exemple pour une phase de la masse des outputs à produire ou des inputs à introduire dans le système. Ceci offre à l'utilisateur des informations qui peuvent servir de base à une certaine optimisation des ressources.

8. IDENTIFIER-ANALYSIS report.

Ce rapport présente toute information basée sur l'utilisation d'identificateurs pour les entités.

Il permet d'effectuer plusieurs contrôles concernant la cohérence et la complétude du système des données:

* Il permet de détecter les entités qui ont le même "IDENTIFIER".
* Si on utilise une codification bien définie, il permet de voir aisément qu'un "IDENTIFIER" est utilisé dans un mauvais contexte.

9. KEYWORD-IN-CONTEXT report.

Ce rapport produit une liste des objets ayant un ou plusieurs mots-clés. Ceci permet de créer des liens logiques qui regroupent des objets faisant partie d'un tout homogène. On peut donc à partir de ce rapport découvrir les objets intrus d'un concept fixé ou les objets nécessaires à la cohérence et à la complétude d'un "groupe logique".

10. PICTURE report.

Il fournit une représentation graphique du flux des données lié à un traitement. Par là même, il offre une vue partielle de la structure du système.
Il permet de contrôler l'utilisation adéquate des relations d'utilisation avec certains objets.

11. PROCESS-CHAIN report.

Il offre une représentation graphique de la dynamique de fonctionnement du système.
Il permet d'évaluer l'impact de changement d'un process ou d'un événement sur les objets restants du même type:

* quels process seront affectés par une modification d'un ou de plusieurs process.
* la même chose si on supprime un événement.

12. PROCESS-SUMMARY report.

Il permet de présenter tout ou partie de la description d'un PROCESS en langage PSL.
Il permet aussi de voir si les INPUTS et/ou les OUTPUTS sont compatibles avec leur utilisation par les PROCESS.

13. RELATION-STRUCTURE report.

Ce rapport donne des informations concernant les entités et les relations entre entités.
Il permet d'effectuer des contrôles de cohérence:

* une relation comporte-t-elle ou non deux entités reliées par l'instruction BETWEEN AND.
* cette relation a-t-elle une connectivity?
* cette entité est-elle reliée à une ou à plusieurs entités via une ou plusieurs relations.

14. STRUCTURE report.

Ce rapport permet de représenter toute structure implémentée dans ISDOS qu'elle soit de données ou de traitements.
L'analyste peut vérifier l'insertion adéquate de tout objet dans toute structure.

Nous avons vu au cours de cette énumération que les contrôles sont restreints:

* aux types des objets créés
* à l'existence ou non de certaines associations entre ces objets
* à la compatibilité des types d'objets avec la syntaxe des "relationships".

Toutes ces restrictions sont dues à la structure particulière de la base de données.

232. 3 Contrôles de complétude.

Dans ce paragraphe, nous allons seulement parler des rapports qui ont pour fonction principale le contrôle de complétude.
Car ,comme nous le verrons dans la suite de ce chapitre, il existe beaucoup de rapports qui assurent plusieurs fonctions suivant la manière dont on les utilise.

1. DATA-BASE-SUMMARY report.

Ce rapport, comme son nom l'indique, permet d'avoir un résumé de tous les objets créés dans la base de données.

2. DICTIONARY report.

Ce rapport permet aussi de présenter sous forme standard les objets ISDOS.
Mais contrairement au rapport précédent, il nécessite de lui spécifier les objets que l'on désire en Output.
Ce rapport peut recevoir en Input un nom d'objet tout comme un fichier de noms d'objets.

3. FORMATTED-PROBLEM-STATEMENT report.

Ce rapport permet de présenter tout ou partie de la description des objets du système d'information en langage PSL.
Il offre en outre la possibilité de présenter toutes les instructions PSL non employées dans chaque section effectivement employée.
Ceci est l'outil par excellence du travail de complétude.
Il permet de voir si on n'a rien oublié et il montre toutes les possibilités du PSL pour chaque section du langage.

4. LIST-CHANGE report.

Ce rapport fournit les informations ayant trait aux modifications apportées à la base de données:

* la commande de mise à jour
* le N° de séquence de celle-ci
* la date de la modification

Ceci est très utile en cas d'incident technique.

5. NAME-LIST report.

Ce rapport produit une liste de tous les éléments de la base de données avec en option:

* le type de l'objet
* les synonymes qui lui sont associés
* la date du dernier changement

6. NAME-SELECTION report.

Il s'agit sans aucun doute du rapport le plus utilisé. Il permet de sélectionner tous les noms d'objets qui satisfont à des critères de sélection paramétrables au gré de l'utilisateur.

Ce type de rapport sera utilisé généralement pour constituer un fichier (PUNCHED-FILE) qui sera utilisé comme Input d'autres rapports.
Ce fichier sera du type ISAM avec un nom d'objet par record de 80 caractères.

23. 3 Fonction de documentation.

233.1 Types de rapports.

Du fait de la structure de la base de données, le nombre de types de rapports est limité. Il n'existe donc pas à proprement parler de véritable langage d'interrogation qui permettrait d'obtenir n'importe quelle information concernant un objet d'ISDOS.

Cependant pour un rapport donné, ISDOS offre à l'utilisateur la possibilité de paramétrer la commande PSA. Ceci a pour résultat de donner un grand nombre de variantes possibles pour un type de rapport particulier.

Nous allons essayer de classer les rapports suivant le type d'objet qu'ils ont en Input:

* rapports concernant tous les types d'objets
* rapports concernant un type d'objet particulier
* rapports concernant un ensemble de types d'objets

La première classe comprend surtout en premier lieu des rapports de sélection tels que:

* l'ATTRIBUTES report
* le KEYWORD-IN-CONTEXT report
* le NAME-SELECTION report

tous ces rapports fournissent si on le désire leurs résultats sous la forme de listes.
De plus la plupart des rapports d'ISDOS permettent de créer un fichier "PUNCHED-FILE" qui contiendra les résultats du rapport. Ce fichier pourra ensuite être employé pour définir les Inputs d'un autre rapport.

Cette première classe comprend aussi:

* le DICTIONARY report et le NAME-LIST report qui offrent aussi des résultats sous la forme de liste.
* le PUNCH-COMMENT-ENTRY report qui imprime les paragraphes commentaires souhaités.

La deuxième et la troisième classe comprennent des rapports produisant:

* des listes d'objets
* des stuctures d'objets
* la représentation d'autres types de "relationships".

A. Rapports concernant un type d'objet.

*) Objet PROCESS

- PICTURE report
- PROCESS-SUMMARY report
- PROCESS-CHAIN report

*) Objet SET

- SUBSET-ANALYSIS report

B. Rapports concernant un ensemble de types d'objets.

1. Rapports de structure.

a) "Relationship": CONSISTS OF

- CONTENTS report
- CONTENTS-ANALYSIS report
- CONTENTS-COMPARISON report

b) Les structure de données ou de traitement.
- STRUCTURE report

2. Rapport de type d'utilisation des données par les traitements.

- DATA-PROCESS-INTERACTION report
- ELEMENT-PROCESS-ANALYSIS report
- ELEMENT-PROCESS-USAGE report

3. Rapport concernant toutes les relations entre traitements, données, événements.

- EXTENDED-PICTURE report

4. Rapport concernant les entités et les relations entre entités.

- RELATION-STRUCTURE report

233. 2 Présentation des rapports.

Certains rapports sont produits suivant trois modes de représentation:

* sous forme de structure
* sous forme de matrice
* sous forme de graphique

D'autres utilisent deux de ces modes et d'autres un seul.
Il existe aussi des rapports purement documentaires reprenant uniquement certains aspects de la description d'un objet comme les commentaires par exemple.
Enfin nous l'avons vu précédemment, les résultats sont aussi présentés sous la forme de listes d'objets.

Il est intéressant de posséder un tel éventail de présentation des résultats, car généralement les utilisateurs ont une préférence pour certains modes de représentation. Ces rapports peuvent être un bon outil de base de discussion avec les utilisateurs.

De plus les rapports ISDOS sont imprimés suivant le format FOLIO sur le listing. Cela permet par une découpe adéquate du listing de disposer de documentation moins volumineuse aux dimensions standards.

Enfin pour les graphiques très volumineux, ISDOS donne à l'utilisateur des renseignements lui permettant d'assembler facilement le document dans son entièreté.

23. 4 Fonction de dictionnaire.

234. 1 Généralités.

1. Notion de donnée: La donnée est une ressource de l'entreprise.
Elle a donc de la valeur et elle occupe une place très importante auprès des gestionnaires de l'entreprise

Modes de création d'une donnée dans l'entreprise:

* Elle existe dans les fichiers de l'entreprise
* Elle est transmise entre des individus et des départements
* elle provient ou est destinée à d'autres entreprises
* elle peut être le résultat de calculs et peut servir au calcul d'autres données

La gestion des données doit être optimisée le plus possible pour que le système d'information soit desservi de façon optimale, il faut:

* connaître les données qui existent et savoir comment elles sont utilisées et par quels traitements
* contrôler les modifications des données existantes et des traitements utilisant ces données
* contrôler les plans concernant les nouvelles utilisations des données et l'acquisition de nouveaux types de données

2. Notion de dictionnaire.(1)

Data Catalog: Outil software utilisé pour décrire chaque donnée élémentaire dans une base de données.

---

(1) Le lecteur trouvera, s'il le désire des compléments d'information aux références (2 - 4)

Data Dictionary ou Data Element Dictionary:
Outil software utilisé pour décrire chaque donnée élémentaire pour dire ce qu'elle est.

Data Element Directory: Outil software pour localiser chaque donnée élémentaire.

Data Element Dictionary / Directory: Outil software utilisé pour décrire, localiser et lister chaque donnée élémentaire d'une base de données. Il fournit un répertoire centralisé d'informations au sujet de chaque donnée élémentaire de façon à faciliter la gestion et le contrôle de la base de données et l'accès à cette base de données.

On parlera également de:

Data Dictionary System:
Par définition il s'agit d'un outil qui sert à enregistrer et à traiter l'information concernant la structure des données et leur utilisation

cfr (3)

Une autre définition nous parait plus générale:
Les dictionnaires de données sont des systèmes et des procédures manuelles ou automatisées servant à mémoriser et à manipuler les définitions des données concernant l'organisation.

cfr (2)

3. Evolution des techniques de mémorisation des données.

On peut s'interroger sur les circonstances qui ont contribué à la création de semblables outils.

Les données utilisées par les traitements automatisés sont souvent à la fois à vocation organisationnelle, fonctionnelle et opérationnelle.
Il en résulte des définitions multiples pour une même donnée et la création de nombreux fichiers indépendants. Cela entraîne beaucoup de redondance et de recouvrement dans les traitements des données et dans la gestion des fichiers contenant ces données.

Une première solution fut l'apparition des bases de données et des systèmes de gestion de ces bases de données.

Cependant, on a constaté que la masse des données avait augmenté en taille et en complexité. Ceci a eu pour conséquence de ne plus permettre la centralisation de toutes les données dans une seule base de données. Le travail de l'administrateur des données s'est alors fortement compliqué.

La solution fut l'utilisation de systèmes de dictionnaires de données.
Les améliorations apportées à la gestion des données furent les suivantes:

* Apport de contrôles simples et effectifs des données élémentaires
* Réduction de la redondance et du manque de pertinence des données
* Renforcement de l'usage de standards
* Renforcement des dispositifs de sécurité
* Détermination de l'impact du changement d'une donnée sur les autres
* La centralisation des données élémentaires comme moyen d'aide à la conception et au développement de nouveaux systèmes

234. 2 Fonctions d'un dictionnaire.

1. Dispositif d'acquisition des données.

Il existe plusieurs voies possibles:

* Acquisition directe par l'emploi d'un langage de spécification des Inputs
* Acquisition à partir des définitions des données dans les programmes écrits dans des langages de haut niveau comme COBOL
* Acquisition à partir d'instructions décrivant le traitement effectué dans un programme
* Acquisition à partir des définitions de sources dbms: CODASYL...

2. Fonction de test de cohérence.

Le dictionnaire de données doit assurer des tests de:

* formatage des informations mémorisées
* complétude des informations mémorisées
* cross-référence correctement constituée
* vérification de la bonne présentation du schéma conceptuel par rapport au système d'information réel

Tous ces tests sont effectués sur les Inputs ou à un autre moment lorsque les données disponibles sont suffisantes.

3. Dispositifs d'accès et de production des résultats.

Pour assurer la bonne qualité des services rendus à l'utilisateur, il faut doter celui-ci de moyens permettant la production de rapports sur demande qui offrent une grande variété de résultats:

* catalogues et description des éléments du système dans un certain formalisme
* Liste des éléments par catégories et suivant des mots-clés paramétrables
* listing représentant les structures de données ainsi que des cross-références
* définitions des données pour les programmes ou les bases de données en langage évolué

4. Génération de données pour les tests.

Cela est réalisé en incluant, au niveau de l'implémentation, des informations sur les valeurs d'intervalles pour les données. Ceci permettra d'effectuer certains tests qualitatifs sur les données.

5. Génération de descriptions des données.

Ces déclarations de données sont incorporées dans les programmes écrits en langages de haut niveau. Elles peuvent être placées dans ces programmes durant une précompilation.

Cela est possible, si la description atteint un niveau de détail suffisant.

6. Génération de code.

Cette génération concerne surtout des programmes standards tels que:

* des modules d'accès en Input ou en Output
* du code standard pour l'utilisation d'utilitaires
* si on a décrit les règles de validation de façon formelle, on pourra générer des programmes de contrôle de validité des données.

7. Contrôles d'accès.

Pour effectuer cette fonction, le dictionnaire doit:

* examiner les programmes d'application pour contrôler l'utilisation abusive de certaines données
* contrôler les liaisons entre utilisateurs et programmes à l'aide d'un interface
* placer des restrictions quant au nombre de personnes abilitées à modifier le dictionnaire, ainsi que des contraintes de "privacy"

8. Analyse de l'impact de changement.

Etant donné l'ampleur des systèmes d'information développés actuellement, l'utilisateur veut disposer de moyens qui lui permettent d'évaluer rapidement les conséquences des changements dans la structure d'information. Ceci afin de pouvoir:

* changer les planningsexistants ou tout simplement les modifier rapidement
* évaluer les effets de ces changements ainsi que leurs coûts

9. Maintien de plusieurs versions.

On peut disposer à un moment donné de plusieurs versions des programmes et des structures de données.
Le dictionnaire doit tenir compte:

* du profil de vie d'un projet
* des rapports sur les différences entre les versions d'un même projet

10. Interfaces d'utilitaires.

Le dictionnaire doit posséder des interfaces avec:

* des processeurs d'interrogation interactifs
* des systèmes interactifs d'édition et de mise à jour de bases de données
* des "reports writers"
* des programmes de restructuration de programmes
* d'autres dictionnaires

234. 3 Classification des dictionnaires de données.

- Dictionnaire
  - primaire
    - indépendant
    - dépendant
  - secondaire dépendant

Dictionnaire primaire.

Il s'agit d'un logiciel séparé et distinct de la base de données, dont les fonctions principales sont:

* l'identification
* la localisation
* le contrôle
* la manipulation

des informations ayant trait aux données élémentaires de la base de données.

Dictionnaire secondaire.

C'est un logiciel dans lequel existe la fonction de dictionnaire de données. Mais celle-ci n'est pas le but principal du logiciel.

Dictionnaire dépendant.

Il s'agit d'un logiciel conçu et implémenté pour un DBMS (Data Base Management System) bien spécifique. Il utilise des dispositifs de ce DBMS et par conséquent, il lui est fortement lié.

Dictionnaire indépendant.

Ce type de logiciel effectue les fonctions de contrôle et de gestion des données élémentaires d'une base de données indépendamment du DBMS utilisé pour gérer cette base de données.

234. 4 Utilisation d'un dictionnaire de données.

Tout dictionnaire de données qui se respecte peut être utilisé au cours du processus complet de Conception, d'Analyse et d'Implémentation d'un système d'information aux étapes suivantes:

* Analyse des données pour déterminer la structure fondamentale des données du système d'information.

* Analyse fonctionnelle.

* Conception de bases de données et de fichiers conventionnels.

* Déroulement opérationnel des applications.

* Collecte et évaluation des statistiques de performances des programmes d'application.

* Maintenance des applications et restructuration des bases de données.

234. 5 Dictionnaire de la version 4.2 d'ISDOS.

Nous allons reprendre les fonctions des dictionnaires développées au paragraphe 234. 2, et nous regarderons comment ISDOS réalise ces fonctions.

1. Acquisition des données.

Le PSL constitue un langage de spécification des Inputs qui permet une acquisition directe des données.

2. Fonction de test de cohérence.

Nous avons vu que ISDOS offre grâce à l'Analyser des moyens de contrôle de la syntaxe du PSL, de la cohérence et de la complétude des informations contenues dans la base de données.

3. Fonction de production de résultats.

Les rapports d'ISDOS comprennent des catalogues et des descriptions sous la forme de commentaires.
Il y a aussi des possibilités de production sélective de listes d'éléments par types d'objets ou suivant des valeurs de clés ou d'attributs.
Enfin il existe aussi des représentations de structures de données ainsi que des Cross-références.

4. Génération de données, de description de données, de code.

Dans l'état actuel d'ISDOS, cette fonction n'est pas réalisée.

Il faut bien se dire que : au niveau de l'analyse fonctionnelle, on ne parle pas de la programmation.
Cependant il est possible d'envisager une technique de remplacement des textes de commentaires comme celui du paragraphe PROCEDURE; par une véritable procédure écrite en langage de haut niveau.

5. Contrôle d'accès.

Le seul dispositif qu'offre ISDOS pour contrôler l'accès aux données est le verrouillage de la base de données à l'aide d'une clé.
Ce dispositif n'est d'ailleurs pas propre à ISDOS mais à la fonction du catalogue des fichiers assurée par le système d'exploitation.

6. Analyse de l'impact de changement.

ISDOS permet d'évaluer l'impact de changement au point de vue de la localisation dans la base de données des objets qui sont touchés par la modification d'autres objets.
Voici un petit exemple qui montre comment il faut procéder:

Supposons que le code produit est modifié en ce sens que sa longueur qui était de 7 caractères est portée à 8 caractères.
Le rapport CONTENTS-ANALYSIS permet de savoir dans quelles autres données est contenue cette donnée élémentaire.
Le rapport DATA-PROCESS-INTERACTION, utilisé avec les résultats du rapport précédent, va nous donner la liste de tous les PROCESS qui utilisent cette donnée.
Il suffira alors de procéder aux changements dans la base de données et, si les programmes sont déjà écrits, à la mise à jour de ces programmes.

7. Maintien de plusieurs versions.

Actuellement la version 4.2 ne sert que d'outil d'aide à l'analyse fonctionnelle. Dès lors un tel dispositif ne nous est d'aucune utilité.

8. Interfaces d'utilitaires.

* En dehors des rapports, ISDOS n'offre aucun outil d'interrogation.

* Les instructions de l'Analyser permettent de gérer la base de données.
  De plus certains rapports d'ISDOS permettent d'éditer tout ou partie de la base de données.

* Il est toujours possible de créer des interfaces avec d'autres dictionnaires ou avec des programmes de restructuration ou de simulation.
  Mais ces interfaces n'existent pas encore du moins à notre connaissance.

En résumé de ce qui précède, on peut dire qu'il ne manque pas grand chose à ISDOS pour être un véritable dictionnaire de données. Dans la suite de ce travail nous allons examiner les possibilités de combiner les "relationships" d'ISDOS de façon à avoir un dictionnaire de données applicable à la méthode d'analyse fonctionnelle développée dans le chapitre 1.

CHAPITRE 3. MISE EN OEUVRE DE LA METHODE D'ANALYSE A L'AIDE D'ISDOS.

Maintenant que nous disposons de la méthode d'analyse et du logiciel ISDOS, nous allons tenter de dégager une méthodologie de l'utilisation d'ISDOS.

Cette méthode comprendra trois étapes:

* La spécification des données initiales de la méthode d'analyse.

* Le corps de la méthode

* La production des outputs de la méthode

Cependant, avant d'entrer dans cette méthode, nous allons vous exposer la solution de structuration du système d'information à l'aide du PSL.

Les données initiales de la méthode comprennent surtout la partie du cahier des charges décrivant la ou les solutions informatiques choisies au cours de l'analyse de conception.
Nous proposerons une description en PSL:

* du cahier des charges et des cellules d'activités

* de la métastructure exposée au début de ce chapitre

* des spécifications détaillées des données et des traitements de la solution du cahier des charges.

Le corps de la méthode consiste en une phase de contrôle des données créées dans la base de données et à une phase de production des rapports de résultats.
Comme nous le verrons dans ce chapitre, il est difficile de dissocier la production des outputs du corps de la méthode.

Nous examinerons donc dans la dernière partie du chapitre trois: les rapports qui produisent ou aident à rédiger les documents composant le dossier de développement.

Nous parlerons tout d'abord des rapports existants dans ISDOS. Ces rapports peuvent être obtenus en utilisant les commandes de l'Analyser de la version 4.2 sans modifications.

Enfin nous vous exposerons les modifications que nous avons apportées à certains rapports afin de produire des résultats nécessaires au bon fonctionnement de la méthode d'analyse fonctionnelle.

Nous ne parlerons pas dans ce chapitre de modifications du PSL, parce que de telles modifications dépassent dans de grandes proportions le cadre de ce chapitre, dont l'objectif principal est d'exposer les tâches qui sont effectivement réalisées à l'heure actuelle.
Nous parlerons de ce sujet dans le chapitre suivant et dans les conclusions.

3. 1 Méthode d'utilisation d'ISDOS.

31. 1 Considérations générales.

A. Formation des noms d'objets.

Les noms d'objets sont tous composés au maximum de 30 caractères alphabétiques, numériques et du tiret.

Tout nom d'objet, à part certaines exceptions, comme les noms d'attributs et les valeurs, doit être univoque. De plus chaque objet est caractérisé par son type.

En pratique, nous avons observé qu'il est parfois difficile de distinguer le type de certains objets et ceci rien qu'à la lecture du source écrit en PSL.

C'est pourquoi nous avons décidé de préfixer les noms d'objets qui offraient cet inconvénient.

Nous proposons d'utiliser comme préfixe le synonyme abrégé de chaque type d'objet de la façon suivante: préfixe-nom d'objet.
Il faudra respecter la règle suivante sous peine d'erreur à l'interprétation du PSL:

(Nombre de caractères du préfixe + Nombre de caractères du nom de l'objet + 1) = 30

Type d'objet.	Préfixe.	
ATTRIBUTE	ATTR	
ATTRIBUTE-VALUE	ATTV	
CONDITION	COND	
ELEMENT	EL	différent du synonyme ELE
ENTITY	ENT	
EVENT	EV	
GROUP	GR	
INPUT	I	différent du synonyme INP
INTERFACE	INTF	
INTERVAL	INTV	
KEYWORD	KEY	
MAILBOX	BOX	
MEMO	MEM	il n'y a pas de synonyme pour MEMO
OUTPUT	O	différent du synonyme OUT
PROBLEM-DEFINER	PD	
PROCESS	PR	différent des synonymes: PROC ou PRC
RELATION	REL	différent du synonyme RLN
SECURITY	SEC	
SOURCE	SRC	
SET	SET	il n'y a pas de synonyme pour SET
SUBSETTING-CRITERION	SSCN	
SYSTEM-PARAMETER	SYSP	

Cette pratique pourra être appliquée par l'utilisateur comme il l'entend et elle n'est nullement indispensable au bon fonctionnement de la méthode

B. Aide-mémoire pour les noms d'objets.

Pour éviter certaines erreurs au moment de l'interprétation du PSL par l'Analyser, nous préconisons l'emploi d'un aide-mémoire des objets créés dans la base de données.

Celui-ci peut être constitué de deux façons:

* Par une liste des objets créés au fur et à mesure de la rédaction du PSL. Cette liste peut être tenue à jour à la main ou à l'aide d'outils informatiques d'édition comme l'EDT du système BS2000.

* Si la banque de données contient déjà certaines informations, on pourra demander un rapport préalable qui liste les objets créés jusqu'à présent.

Cette façon de faire évitera à l'utilisateur des créations redondantes d'objets existants, ainsi que la création de relations entre des noms d'objets différents des noms déjà créés dans la base de données.
Cependant le dernier inconvénient cité peut être résolu par l'utilisation de l'objet SYNONYM QUI permet de rattacher plusieurs versions de noms d'un même objet à la bonne forme de cet objet.

C. Conseils préliminaires à l'utilisateur du PSL.

Nous pensons que certaines mises en gardes ne sont pas inutiles avant d'utiliser le PSL.

* Nous conseillons aux utilisateurs d'ISDOS de créer des fichiers Input qui soient homogènes. Ils sont homogènes en ce sens qu'ils décrivent un même aspect fonctionnel du problème. Le lecteur pourra voir ce que nous entendons par là dans la suite de ce chapitre.

* Nous conseillons également à l'utilisateur de toujours garder toutes ses sources PSL. Ceci lui évitera beaucoup de désagréments s'il avait des problèmes avec le système d'exploitation ou avec la base de données. De plus lorsque des erreurs sont détectées dans le PSL, nous engageons l'utilisateur à

modifier également le source erroné, après avoir procédé aux modifications dans la base de données.
Toutes ces précautions lui permettront de rapidement remettre en état la base de données en cas d'incident technique ou autre.
L'utilisateur pourrait également faire des copies périodiques de ses banques de données en prenant bien soin de noter les modifications qui sont survenues entre deux copies successives. Il sera aidé en cela par le rapport ISDOS LIST-CHANGES dont nous avons déjà parlé dans le chapitre 2.
Ces précautions peuvent paraître superflues pour certains, mais en pratique on remarque que l'on ne prend jamais suffisamment de précautions pour garantir une base de données vis-à-vis de tout accident technique ou autre.

D. Démarche d'analyse antérieure à l'utilisation d'ISDOS.

A partir des données initiales de la méthode d'analyse, l'analyste va effectuer plusieurs tâches:

1° Il va devoir étudier la structure logique de chaque phase à partir de la description de la cellule d'activité associée à la phase.

2° Pour chaque application il va créer la structure conceptuelle intégrant les structures logiques des phases de l'application.
Il pourra faire de même jusqu'au niveau du schéma fonctionnel.

3° Enfin il va procéder à la décomposition des phases en fonctions, suivant les principes d'individualisation qu'il désire.

31. 2 Métastructure représentée à l'aide d'ISDOS.

A. Quels sont les problèmes à résoudre? (1)

1. On doit trouver un moyen de structurer les données d'une part et les traitements d'autre part.

   * En ce qui concerne les traitements; c'est possible, car tout PROCESS peut être décomposé en d'autres PROCESS à l'aide des "relationships": CONSISTS et UTILIZES.

   * Pour les données, nous rencontrons certaines difficultés. En effet un groupe ou une entité ne peuvent être constitués d'entités.
   Cette restriction du PSL peut être éludée en utilisant de façon particulière les propriétés de l'objet SET.
   Celui-ci peut:

     - être décomposé en SETS à l'aide de la relation SUBSETS.

     - être constitué d'ENTITIES à l'aide de la relation CONSISTS.

2. Partant d'un traitement, on veut connaître toutes les données utilisées par celui-ci et inversément.
   Ceci est possible grâce aux relations: USES, UPDATES, DERIVES et autres relations assimilées.

3. Partant d'une phase, on désire connaître sa structure logique.
   Et partant d'une entité ou d'une relation , on désire connaître les phases qui travaillent sur les structures logiques composées de cette entité ou de cette relation.
   Nous verrons dans les propositions de solutions, comment nous avons résolu ce problème.

---

(1) Nous invitons le lecteur à se référer au schéma de la métastructure du système d'information présenté dans le chapitre 1.

B. Propositions de solutions.

Une première solution qui nous est venue à l'esprit fut d'utiliser à fond la notion d'ATTRIBUTE.
Mais il nous est rapidement apparu que cela présentait deux grands inconvénients.

1. Pour distinguer les relations entre les entités ou les fonctions utilisées par une phase, on devait multiplier les noms d'attributs et les valeurs de ceux-ci.

   En voici un exemple:

```
RELATION            REL-CLI-CDE;
    BETWEEN         ENT-CLIENT
    AND             ENT-COMMANDE;
    ATTRIBUTES ARE  SYS ,SOUS-SYS1 ,APPL1 ,
                    PH1 ,FONC2;
```

   * Ce qui signifie que la relation REL-CLI-CDE entre les entités ENT-CLIENT et ENT-COMMANDE est utilisée dans la fonction FONC2 qui appartient à la phase PH1 qui fait partie de l'application APPL1 qui appartient au sous-système SOUS-SYS1 qui, lui enfin fait partie du système SYS.

   * Nous voyons grâce à cet exemple que l'énumération de tous ces attributs est fastidieuse et peut donc constituer une source d'erreurs.

   * De plus, si la même relation est utilisée par une autre fonction, les attributs mentionnés ne sont plus d'actualité pour cette fonction. Il faudra alors créer une autre relation semblable mais avec d'autres attributs. Cette redondance n'est vraiment pas acceptable.

2. L'utilisation des attributs permet de créer des relations implicites qui rendraient redondantes les "relationships" telles que: USES, SUBPARTS, UTILIZES...
   Nous estimons qu'une telle solution serait une solution de riches qui pourrait même à la limite remettre en question le langage PSL.

Proposition 2.

Elle consiste à utiliser les possibilités du PSL en ce qui concerne la structuration des Process.

Et d'autre part à utiliser les relations PSL: USES, MAINTAINS ainsi que UTILIZES pour relier les process respectivement aux Entités, aux Relations et aux Process.
On trouvera à la page suivante un exemple de cette solution.

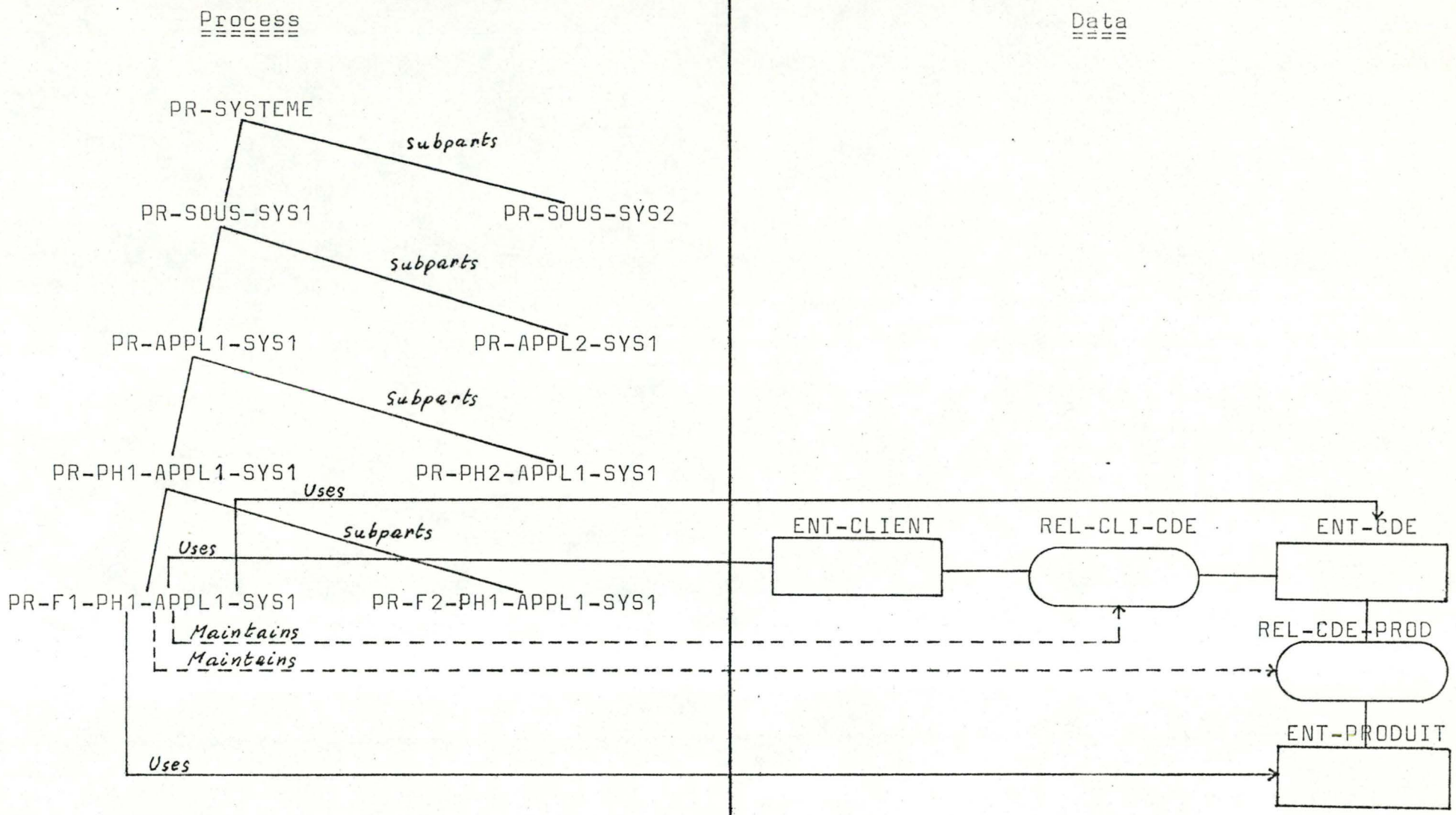
Caractéristiques de cette solution.

- Nous avons préfixé chaque nom d'objet pour avoir plus de clarté dans la dissociation de ces objets dès la lecture de PSL.

- Nous n'utilisons pas spécialement la notion d'attribut comme dans la solution précédente.

- Certaines recherches d'objets devront se faire en plusieurs étapes par une approche TOP-DOWN ou BOTTOM-UP.

- Cette solution ne représente pas de façon tangible la structure des données.
  C'est pourquoi nous lui préférons la solution suivante.



On peut aussi mettre les USES et les MAINTAINS dans la description de la phase.
Cela permet d'obtenir directement la structure logique sans devoir descendre au niveau des fonctions.

Proposition 3.

Cette solution crée une métastructure correspondant au schéma fonctionnel de la méthode d'analyse. Le lecteur trouvera dans l'exemple qui suit ce texte une application de cette solution qui combine l'utilisation des attributs de la première solution et la deuxième solution.

L'apport original de cette troisième solution est la structure des données parallèle à la structure des traitements.

Fonction des attributs.

* L'attribut TRAIT: permet par sa valeur d'identifier à quel élément de la nomenclature des traitements appartient le PROCESS ou le SET incriminé.
  Les valeurs de cet attribut sont;
  SYSTEME, SOUS-SYS, APPLICATION, PHASE, FONCTION pour les PROCESS et STR-SYS, CONC-SSYS, CONC-APPL, LOG-PH pour les SETS.

* L'attribut CLE: permet d'identifier à quel élément de nomenclature du niveau supérieur se rattache le process ou le set en question.

Cette solution diminue très fort le nombre d'étapes nécessaires pour accéder soit:

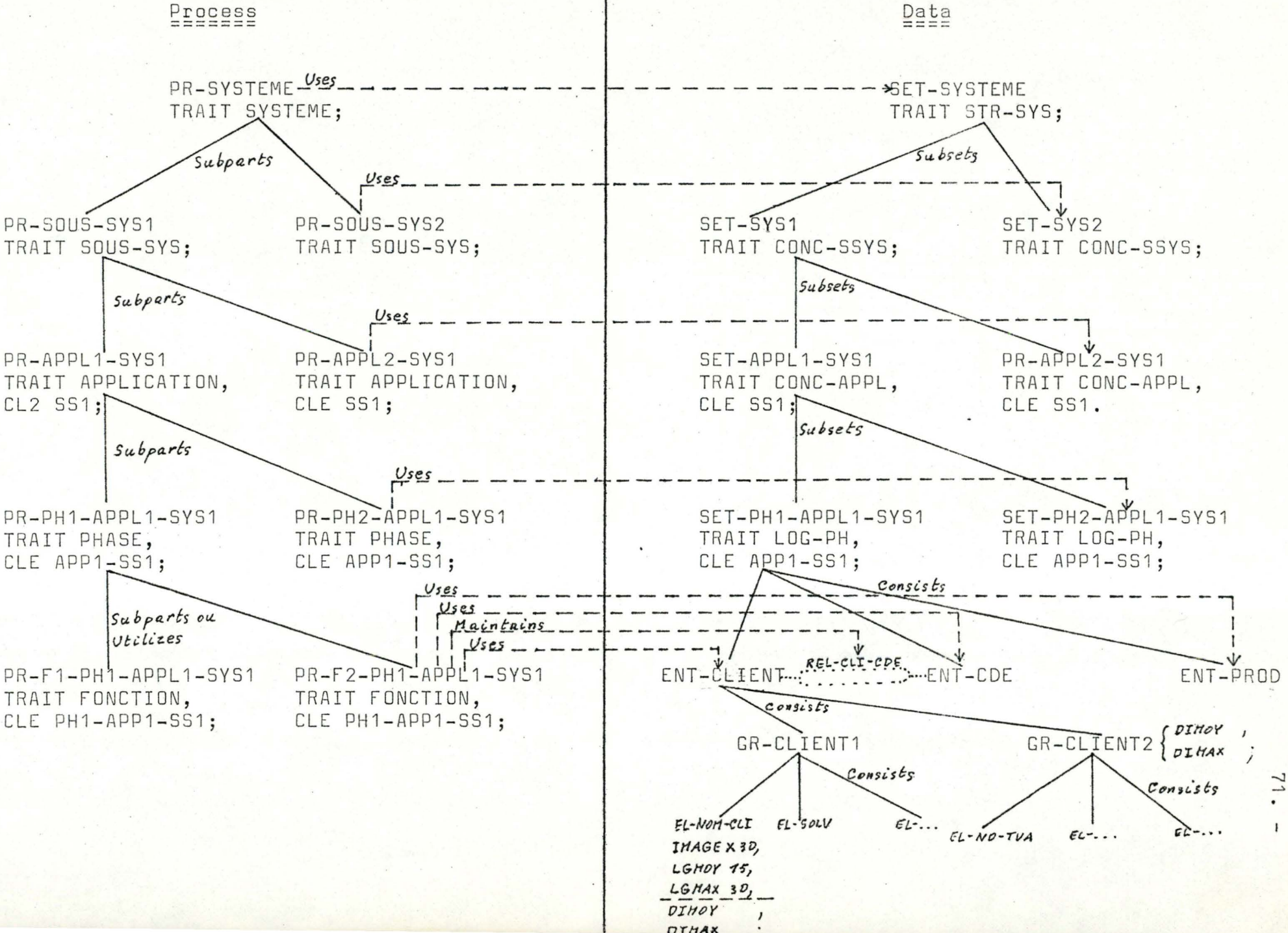
* à un type d'objet en faisant une sélection sur un type d'objet;

* à un type d'élément de la nomenclature en faisant une sélection sur le type d'objet et sur la valeur de l'attribut TRAIT correspondant à l'élément de la nomenclature désirée;

* à un ensemble d'éléments de la nomenclature qui composent un élément de niveau supérieur à celui des éléments incriminés.
  Pour faire cela, nous utiliserons le rapport de sélection en mentionnant conjointement:

  - le type d'objet : SET ou PROCESS;
  - l'attribut TRAIT avec la valeur correspondant au type d'élément de la nomenclature;
  - l'attribut CLE avec la valeur correspondant à une identification de l'élément de la nomenclature de niveau supérieur;

- à un objet particulier: en donnant le nom de cet objet particulier comme input du rapport désiré.


Process
Data
PR-SYSTEME
TRAIT SYSTEME;
Uses
SET-SYSTEME
TRAIT STR-SYS;
Subparts
Subsets
Uses
PR-SOUS-SYS1
TRAIT SOUS-SYS;
PR-SOUS-SYS2
TRAIT SOUS-SYS;
SET-SYS1
TRAIT CONC-SSYS;
SET-SYS2
TRAIT CONC-SSYS;
Subparts
Subsets
Uses
PR-APPL1-SYS1
TRAIT APPLICATION,
CL2 SS1;
PR-APPL2-SYS1
TRAIT APPLICATION,
CLE SS1;
SET-APPL1-SYS1
TRAIT CONC-APPL,
CLE SS1;
PR-APPL2-SYS1
TRAIT CONC-APPL,
CLE SS1.
Subparts
Subsets
Uses
PR-PH1-APPL1-SYS1
TRAIT PHASE,
CLE APP1-SS1;
PR-PH2-APPL1-SYS1
TRAIT PHASE,
CLE APP1-SS1;
SET-PH1-APPL1-SYS1
TRAIT LOG-PH,
CLE APP1-SS1;
SET-PH2-APPL1-SYS1
TRAIT LOG-PH,
CLE APP1-SS1;
Consists
Uses
Uses
Maintains
Uses
Subparts ou
Utilizes
PR-F1-PH1-APPL1-SYS1
TRAIT FONCTION,
CLE PH1-APP1-SS1;
PR-F2-PH1-APPL1-SYS1
TRAIT FONCTION,
CLE PH1-APP1-SS1;
ENT-CLIENT
REL-CLI-CDE
ENT-CDE
ENT-PROD
Consists
GR-CLIENT1
GR-CLIENT2
DIMOY ,
DIMAX ;
Consists
Consists
EL-NOM-CLI
IMAGE X 30,
LGMOY 15,
LGMAX 30,
DIMOY ,
DIMAX ;
EL-SOLV
EL-...
EL-NO-TVA
EL-...
EL-...

31. 3 Données initiales de la méthode.

Pour faciliter le travail de l'utilisateur, nous proposons l'utilisation de formulaires de spécifications du PSL.

Ces formulaires auront certaines particularités:

* Tout ce qui sera écrit en lettres minuscules correspondra à des directives d'utilisation de ces formulaires.

* Tout ce qui sera écrit en lettres majuscules correspondra à des instructions PSL.

313. 1 Utilisation des paragraphes de commentaire.

1. Pour tous les objets ISDOS décrits dans une section, l'analyste peut employer le paragraphe de commentaires suivant le mot ISDOS DESCRIPTION;

2. Les autres mots ISDOS permettant l'introduction de commentaires sont propres à certains objets d'ISDOS.
   L'analyste pourra se référer au tableau qui suit.

OBJET ISDOS	NOM DE PARAGRAPHE
ENTITY	LAYOUT; VOLATILITY;
INPUT	LAYOUT;
OUTPUT	LAYOUT;
PROCESS	PROCEDURE;
RELATION	DERIVATION;
SET	DERIVATION; VOLATILITY-MEMBER; VOLATILITY-SET;
CONDITION	FALSE-WHILE; TRUE-WHILE;

3. L'analyste utilisera le formulaire D/ISC en tête duquel il spécifiera le nom de paragraphe voulu. Il devra veiller à placer ce formulaire dans la suite de formulaires décrivant l'objet voulu et de terminer ses commentaires par;

Document de spécification des commentaires

```
1       *       *       *       *       *       *                72
Nom de paragraphe de commentaires
.....................;
```

313. 2 Cahier des charges.

Afin de décrire le cahier des charges ou plutôt la partie du cahier des charges qui concerne l'analyse fonctionnelle, nous proposons d'utiliser un PROCESS particulier.
Le Process aura pour valeur de l'attribut TRAIT: CHARGE.

* Il aura pour composants les objets du niveau supérieur de la nomenclature des traitements qui se trouvent dans la solution informatique choisie.

* Les paragraphes de commentaires serviront de supports à la description de tout ce qui concerne l'ensemble des solutions de ce cahier des charges.

313. 3 Cellules d'activité.

Un élément du système très important est très certainement la cellule d'activité.
Elle est l'organe de l'organisation dans le cadre de laquelle se déroule le traitement décrit par une phase.

Nous utiliserons l'objet PROCESS pour décrire une cellule d'activité avec le préfixe CELL-.

* Ce Process aura pour valeur de l'attribut TRAIT le mot CELLULE.

* Elle sera reliée à la phase qui lui correspond par la relation SUBPARTS.

* Elle sera reliée à d'autres cellules d'activité par la relation UTILIZES. Ces cellules d'activité lui fournissent des unités d'information ou reçoivent des Unités d'informations venant de la cellule d'activité.

* Il utilisera des ressources que nous décrirons à l'aide de l'objet ELEMENT, par la relation USES.
  Cet ELEMENT sera décrit comme suit:

  - Il aura pour attribut: UNITE et ce nom d'attribut aura un nom d'attribute-value définissant l'unité de la ressource.

- Il aura pour valeur: le nombre d'unités de la ressource consommées pour le traitement de la phase sous la forme(mini THRU MAXI).

- Pour la différencier des autres éléments, on lui donnera le préfixe RESS-.

On risque de rencontrer certains problèmes pour définir les unités de temps pendant lesquelles sont consommées les ressources.

* Il faudra définir toutes ces informations dans le paragraphe de commentaires DESCRIPTION de la cellule d'activité, car ces unités ne dépendent que de la cellule d'activité et non pas des ressources consommées.

* Il faudra également spécifier dans les commentaires quel est l'apport INPUT des autres cellules ou à quelles cellules sont destinés les OUTPUTS de la cellule.

Ce formalisme nous semble respectueux de la notion de cellule d'activité exposée au chapitre 1.
En effet, on peut définir:

- les ressources décrites à l'aide des objets ELEMENTS
- les unités d'information utilisées en input et en output.
  Car la cellule d'activité est reliée à la phase correspondante dans la structure des traitements. Elle est aussi en relation avec les origines et les destinations de ces unités d'informations.
- les règles de traitement sont décrites dans la phase, mais rien n'empêche l'utilisateur de les décrire dans la cellule, s'il le désire.

Pour décrire les cellules d'activité, l'analyste pourra utiliser le formulaire D/ISO que vous trouverez à la page suivante.

Informations à spécifier.

Toutes les indications ont été données dans ce paragraphe.
Cependant, il faudra veiller à ce que les ressources consommées ne soient pas encore décrites, auquel cas, on ommettra de remplir la partie du formulaire consacrée à la description des ELEMENTS sauf la relation USED BY.

Document de description d'une cellule d'activité

1 * * * * * * 72

Nom de PROCESS CELL-.....;
ATTRIBUTES ARE TRAIT CELLULE;
UTILIZES CELL-.....,
CELL-.....,
CELL-.....,
CELL-.....,
CELL-.....,
CELL-.....;
SUBPARTS ARE PR-.....; Nom de la phase correspondante

Nom d'une ressource
ELEMENT RESS-.....;
ATTRIBUTES ARE UNITE .....; Désignation de l'unité
VALUES ARE ..... THRU .....;
Nombre entier minimum — Nombre entier maximum
USED BY CELL-.....;
Nom de la cellule décrite dans ce document

Même chose que pour la ressource précédente
ELEMENT RESS-.....;
ATTRIBUTES ARE UNITE .....;
VALUES ARE ..... THRU .....;
USED BY PR-.....;

ELEMENT RESS-.....;
ATTRIBUTES ARE UNITE .....;
VALUES ARE ..... THRU .....;
USED BY PR-.....;

ELEMENT RESS-.....;
ATTRIBUTES ARE UNITE .....;
VALUES ARE ..... THRU .....;
USED BY PR-.....;

La rédaction des spécifications devrait se faire en deux étapes.
La première vise à décrire la structure du système à analyser et la seconde consiste à analyser plus en profondeur chaque niveau de la structure.

313. 4 Etape 1.

Description de la métastructure du système.

Comme le modèle exposé au chapitre 3: par. 334. 2 le montre, nous constatons le parallélisme entre la structure de PROCESS et la structure des SETS. Nous allons proposer à l'analyste un document qui va faciliter sa tâche:
il s'agit du formulaire D/IS1

Informations à spécifier pour le process.

1. Nom du PROCESS à définir.

2. SUBPARTS: liste de tous les PROCESS utilisés pour décomposer le process décrit dans ce document. S'il n'y a pas de composants, on laissera ces zones à blanc.

3. SYNONYMS: nom utilisateur utilisé comme synonyme pour le process spécifié. Cette spécification est facultative.

4. ATTRIBUTES.

   a) TRAIT. L'analyste devra cocher la case en regard de la valeur qu'il doit donner à l'attribut TRAIT.
   Il ne peut donner qu'une et une seule valeur à cet attribut.

   b) CLE. La valeur de cet attribut doit permettre de distinguer à quel process de niveau supérieur est rattaché le process spécifié. Cet attribut n'est pas à mentionner lorsque l'on décrit la racine de la structure des process.

5. USES.

   Le nom du SET correspondant dans la structure parallèle des données. Ce nom sera celui qui sera spécifié dans la seconde partie de ce document.

Du moins ce sera le cas pour les SYSTEMES,SOUS-SYSTEMES, APPLICATIONS, PHASES.

Informations à spécifier pour le set.

1. Nom du SET à définir.

2. SUBSETS: liste de tous les SETS correspondants au process spécifiés dans le statement SUBPARTS. Cependant, il faut noter qu'il n'y a pas de correspondants lorsque la valeur de l'Attribut TRAIT du process spécifié a les valeurs: PHASE, FONCTION, CONTROLE (1).
Ceci est dû au fait que le parallélisme entre process et set est rompu à partir des phases.
S'il n'y a pas de composants, ces zones sont laissées à blanc.

3. ATTRIBUTES.

   a) TRAIT: l'analyste devra cocher la valeur voulue tout comme pour le process spécifié dans ce document, avec les restrictions concernant les FONCTIONS et les CONTROLES.

   b) CLE: valeur identique à celle spécifiée pour l'attribut CLE du process décrit dans ce document.

---

(1) L'explication de l'utilisation de cette valeur de l'attribut TRAIT se trouve au numéro 3 du paragraphe 3132. 1.

Document de description de la métastructure

1 * * * * * * 72

Nom de PROCESS PR-.....;

Liste de process SUBPARTS PR-.....,
PR-.....,
PR-.....,
PR-.....,
PR-.....,
PR-.....,
PR-.....;

Nom de SYNONYMS PR-.....;

ATTRIBUTES

TRAIT
- ☐ SYSTEME ,
- ☐ SOUS-SYS ,
- ☐ APPLICATION ,
- ☐ PHASE ,
- ☐ FONCTION ,
- ☐ CONTROLE , } pas de uses vers un set

CLE valeur de l'attribut clé .....;

USES SET-.....; Nom du set correspondant dans la métastructure

==========

Nom de SET SET-.....;

Liste de sets SUBSETS SET-.....,
SET-.....,
SET-.....,
SET-.....,
SET-.....,
SET-.....;

ATTRIBUTES

TRAIT
- ☐ STR-SYS ,
- ☐ CONC-SSYS ,
- ☐ CONC-APPL ,
- ☐ LOG-PH ,

CLE .....;

Valeur identique à celle de l'attribut cle du process décrit dans ce document

313. 5 Etape 2.

Description des spécifications fonctionnelles proprement dites.

Maintenant que la métastructure est définie, nous allons décrire les données et les traitements.

3135.1 Desription des données.

Au cours de la phase précédente de structuration, l'analyste aura identifié les données dont il a besoin pour produire les résultats désirés.

A. Description de la structure logique des données.

* L'analyste a identifié les entités nécessaires à l'exécution de la phase qu'il doit décrire.
* Il a aussi relevé les relations sémantiques entre les entités.
* Et enfin il a déterminé les informations qui caractérisent ces entités et ces relations.

A partir de toutes ces informations, il va utiliser le formulaire D/IS2 pour décrire la structure logique de chaque phase d'une application.

Informations à spécifier dans ce formulaire.

a) Informations concernant les deux entités.

1. ENTITY: nom de l'entité décrite.

2. IDENTIFIED: nom de l'élément ou du groupe qui permet de distinguer cette entité parmi d'autres entités.
   Si la donnée associée est un groupe, l'analyste placera le préfixe GR dans les deux cases prévues à cet effet. On fera de même avec le préfixe EL, s'il s'agit d'un élément.

3. CONTAINED: nom du set correspondant à la phase que l'analyste décrit.
   S'il décrit la phase PR-PH1-APP1-SS1: le nom du set correspondant aurait pour nom: SET-PH1-APP1-SS1.

4. CARDINALITY: nom de system parameter spécifiant le nombre d'occurrences de cette entité.

5. CONSISTS: liste de system parameters suivis d'un nom de groupe ou d'élément.
Le system parameter spécifie le nombre de fois que le groupe ou l'élément est contenu dans l'entité spécifiée.
Si le composant est un groupe, l'analyste placera le préfixe GR dans les deux cases prévues à cet effet. Il fera la même chose avec le préfixe EL, si le composant est un élément.

b) Informations concernant la relation.

1. RELATION: nom de la relation à décrire.

2. BETWEEN: nom d'une des deux entités décrites dans le présent document.

* AND: nom de la seconde entité décrite dans ce document.

Remarques:

* Si les deux entités concernées par la relation ont déjà été définies, l'analyste n'utilisera que la partie "RELATION" de ce document.

* Cependant, l'analyste devra vérifier si ces deux entités déjà décrites font parties du set correspondant à la phase en cours d'analyse.

3. CARDINALITY: nom du système parameter spécifiant le nombre d'occurrences de la relation décrite.

4. CONNECTIVITY: nom de system parameter spécifiant les quantifications de la relation sémantique gauche-droite.

* TO: nom de system parameter spécifiant les quantifications de la relation sémantique droite-gauche.
N.B. ISDOS offre à l'analyste la possibilité de donner des valeurs (MIN - MAX) aux system parameters.

5. ASSOCIATED_DATA: liste de groupes et/ou d'éléments qui caractérisent la relation spécifiée.

Document de description de la structure logique

1 * * : * * * * 72

Nom d'ENTITY ENT-.....;
IDENTIFIED BY ☐☐-.....;
CONTAINED IN SET-.....;
CARDINALITY IS SYSP-.....;
CONSISTS OF Liste de composants
SYSP-..... ☐☐-.....,
SYSP-..... ☐☐-.....,
SYSP-..... ☐☐-.....,
SYSP-..... ☐☐-.....,
SYSP-..... ☐☐-.....,
SYSP-..... ☐☐-.....;

==========

ENTITY ENT-.....;
IDENTIFIED BY ☐☐-.....;
CONTAINED IN SET-.....;
CARDINALITY IS SYSP-.....;
CONSISTS OF
SYSP-..... ☐☐-.....,
SYSP-..... ☐☐-.....,
SYSP-..... ☐☐-.....,
SYSP-..... ☐☐-.....,
SYSP-..... ☐☐-.....,
SYSP-..... ☐☐-.....;

==========

Nom de la RELATION REL-.....;
BETWEEN ENT-.....
AND ENT-.....;
CARDINALITY IS SYSP-.....;
CONNECTIVITY IS
SYSP-..... TO SYSP-.....;
ASSOCIATED DATA ARE ☐☐-.....,
☐☐-.....,
☐☐-.....,
☐☐-.....,
☐☐-.....;

B. Description des INPUTS, OUTPUTS, SETS, GROUPS et ELEMENTS.

Nous allons faire ces descriptions, sans penser à toutes les relations qui vont relier ces objets aux process.
Pour accomplir cette tâche, nous proposons à l'analyste d'utiliser le document D/IS3.

1° Informations à spécifier pour un INPUT ou un OUTPUT.

1. Nom de l'INPUT à décrire. Mettre INPUT dans TYPE, et utiliser le préfixe I. S'il s'agit d'un OUTPUT, on utilisera OUTPUT pour le TYPE et O pour le préfixe.

2. SUBPARTS:
   * liste d'INPUTS qui font partie de l'INPUT spécifié: utilisation du préfixe I.

   * liste d'OUTPUTS qui font partie de l'OUTPUT spécifié: utilisation du préfixe O.

3. CONSISTS:
   liste de groupes d'éléments et/ou d'éléments qui sont des composants de l'INPUT spécifié.

   Si le composant est un groupe, l'analyste placera le préfixe GR dans les deux cases prévues à cet effet.

   Si le composant est un élément, il fera de même avec le préfixe EL.

   L'analyste peut utiliser un système parameter pour indiquer le nombre d'occurrences du composant dans l'INPUT.

4. SYNONYMS:
   * Nom d'un synonyme de l'INPUT. Ceci est facultatif, utilisation du préfixe I.

* Nom d'un synonyme: utilisation du préfixe O.

5. HAPPENS:
   nom de system parameter indiquant le nombre d'occurrences de l'INPUT ou de l'OUTPUT spécifié.

   * TIMES-PER: nom de l'intervalle pendant lequel les occurrences surviennent.

Document de description d'un INPUT ou d'un OUTPUT

1 * * * * * * 72

Type de l'objet [□□□□□] [□]-.....,

SUBPARTS ARE Liste des objets faisant partie de l'input ou de l'output

[□]-.....,
[□]-.....,
[□]-.....,
[□]-.....,
[□]-.....,
[□]-.....,
[□]-.....;

CONSISTS OF Liste de composants

SYSP-..... [□□]-.....,
SYSP-..... [□□]-.....,
SYSP-..... [□□]-.....,
SYSP-..... [□□]-.....,
SYSP-..... [□□]-.....,
SYSP-..... [□□]-.....,
SYSP-..... [□□]-.....,
SYSP-..... [□□]-.....,
SYSP-..... [□□]-.....,
SYSP-..... [□□]-.....,
SYSP-..... [□□]-.....,
SYSP-..... [□□]-.....,
SYSP-..... [□□]-.....,
SYSP-..... [□□]-.....,
SYSP-..... [□□]-.....;

SYNONYMS ARE [□]-.....;

HAPPENS SYSP-.....

TIMES-PER INTV-.....;

2° Informations à spécifier pour un SET.

Les sets que nous allons décrire ici sont distincts de ceux que l'on a décrit dans le document D/IS1.
Ceux-là permettaient de décrire des concepts parallèles aux concepts de système, sous-système, application et phase de la méthode d'analyse.
Tandis que ceux-ci doivent être considérés comme les SET au sens d'ISDOS.

L'analyste utilisera le document D/IS4 pour décrire un SET.

1. Nom du SET à décrire.

2. SYNONYMS:
   Nom d'un synonyme du SET. Cette clause est facultative.

3. SUBSETS:
   liste des Noms de SETS faisant partie du SET décrit.

4. CONSISTS:
   liste d'ENTITIES, d'INPUTS, d'OUTPUTS qui composent le SET spécifié.
   L'analyste peut indiquer le nombre d'occurrences de ces objets en spécifiant un nom de system parameter.

   * S'il s'agit d'une entité, l'analyste complètera le nom de l'entité à la suite du préfixe ENT-.

   * S'il s'agit d'un INPUT, l'analyste indiquera le préfixe I, dans la case prévue à cet effet, suivi du nom de l'INPUT.

   * S'il s'agit d'un OUTPUT, il fera la même chose, mais avec le préfixe O.

Document de description d'un SET

```
1        *        *        *        *        *        *        72
Nom de SET                    SET-.........................;

SYNONYMS ARE                  SET-.........................;

SUBSETS ARE  Liste des sets composants ce set
                    SET-.........................;
                    SET-.........................,
                    SET-.........................,
                    SET-.........................,
                    SET-.........................,
                    SET-.........................;

CONSISTS OF  Liste des composants de ce set

                    a. liste d'entités

SYSP-.......................  ENT-.........................,
SYSP-.......................  ENT-.........................,
SYSP-.......................  ENT-.........................,
SYSP-.......................  ENT-.........................,
SYSP-.......................  ENT-.........................,
SYSP-.......................  ENT-.........................,

                    b. liste d'inputs ou d'outputs

SYSP-.......................  □-.........................,
SYSP-.......................  □-.........................,
SYSP-.......................  □-.........................,
SYSP-.......................  □-.........................,
SYSP-.......................  □-.........................,
SYSP-.......................  □-.........................,
SYSP-.......................  □-.........................,
SYSP-.......................  □-.........................,
SYSP-.......................  □-.........................,
SYSP-.......................  □-.........................,
SYSP-.......................  □-.........................,
SYSP-.......................  □-.........................,
SYSP-.......................  □-.........................,
SYSP-.......................  □-.........................,
SYSP-.......................  □-.........................;
```

3° Infomations à spécifier pour un groupe ou un élément.

Pour accomplir ce travail, l'analyste utilisera le document D/IS5.

1. Suivant le type de l'objet décrit, l'analyste cochera la case en regard du type adéquat.

   * S'il s'agit d'un groupe, l'analyste mettra le préfixe GR dans les deux cases prévues à cet effet, suivi du nom de groupe.

   * S'il s'agit d'un élément, il agira de même avec le préfixe EL.

2. Spécifications propres à un groupe.

   * CONSISTS: liste de groupes ou d'éléments composant le groupe décrit. Le system parameter permet d'indiquer le nombre d'occurrences de l'objet composant du groupe. L'utilisation des préfixes est la même qu'au point 1.

   * ATTRIBUTES:

     - DIMOY: dimension moyenne du groupe spécifié.
     - DIMAX: dimension maximum du groupe spécifié.

3. Spécifications propres à un élément.

   ATTRIBUTES.

   - IMAGE: image de l'élément spécifié.
   - LGMOY: longueur moyenne de l'élément spécifié en caractères.
   - LGMAX: longueur maximum de l'élément. spécifié en caractères.
   - DIMOY: dimension moyenne de l'élément spécifié.
   - DIMAX: dimension maximum de l'élément spécifié.

Document de description d'1 élément ou d'1 groupe

1 * * * * * * 72

Nom de l'objet ☐ ELEMENT ☐ GROUP ☐-.....;

Groupe

CONSISTS OF Liste de composants

SYSP-.....
☐-.....,
SYSP-.....
☐-.....,
SYSP-.....
☐-.....,
SYSP-.....
☐-.....,
SYSP-.....
☐-.....,
SYSP-.....
☐-.....,
SYSP-.....
☐-.....,
SYSP-.....
☐-.....,
SYSP-.....
☐-.....,
SYSP-.....
☐-.....,
SYSP-.....
☐-.....,
SYSP-.....
☐-.....;

ATTRIBUTES ARE

DIMOY ..... ,
Dimension moyenne

DIMAX .....;
Dimension maximum

Elément

Codification de l'image:

cppppp-qq-uu

c définit la classe de la donnée:
****

X = alphanumérique
N = numérique non signé
S = numérique signé

ppppp = nombre de chiffres entiers ou de caractères

qq = nombre de chiffres décimaux

uu = codification de la clause usage de COBOL

C = computational
C1 = computational-1
C2 = computational-2
C3 = computational-3
I = indexed

ATTRIBUTES ARE

IMAGE .....-.... ,
cppppp-qquu

LGMOY ..... ,
Longueur moyenne

LGMAX ..... ,
Longueur maximum

DIMOY ..... ,
Dimension moyenne

DIMAX ..... ;
Dimension maximum

C. Description de l'environnement du système à analyser.

Pour accomplir cette tâche, l'analyste utilisera le document D/IS6.

Information à spécifier pour un interface.

1. Nom de l'interface à décrire.

2. SYNONYMS ARE.
   Nom du synonyme de l'interface spécifié.
   Cette clause est facultative.

3. SUBPARTS:
   liste des interfaces qui font partie de l'interface spécifié.

4. GENERATES:
   liste des INPUTS générés par l'interface spécifié à destination du système analysé.

5. RECEIVES:
   liste des OUTPUTS reçus par l'interface spécifié et provenant du système analysé.

Description d'un interface du système

```
1         *         *         *         *         *         *         72
Nom de l' INTERFACE              INTF-.........................;
SYNONYMS ARE                     INTF-.........................;
SUBPARTS ARE  Liste d'interfaces
                        INTF-.........................,
                        INTF-.........................,
                        INTF-.........................,
                        INTF-.........................,
                        INTF-.........................,
                        INTF-.........................;
GENERATES               I-............................,
                        I-............................,
                        I-............................,
                        I-............................,
                        I-............................,
                        I-............................,
                        I-............................,
                        I-............................,
                        I-............................,
                        I-............................;
RECEIVES                O-............................,
                        O-............................,
                        O-............................,
                        O-............................,
                        O-............................,
                        O-............................,
                        O-............................,
                        O-............................,
                        O-............................,
                        O-............................,
                        O-............................;
```

3135. 2 Description des traitements.

A. Description du fonctionnement statique d'un process.

Cette partie de l'analyse consiste en l'étude des relations entre les process et les données. Pour accomplir cette tâche, l'analyste dispose du document D/IS7a et de son prolongement par le document D/IS7b.

Informations à spécifier pour un process.

1. Nom du process à décrire.

2. GENERATES: liste d'OUTPUTS générés par le process à destination de l'environnement du système analysé.

3. RECEIVES: liste d'INPUTS reçus par le process en provenance de l'environnement du système analysé.

4. USES: liste de SETS, INPUTS, ELEMENTS, GROUPS, ENTITIE utilisées par le process pour:

   * DERIVE : dériver une liste de SETS, OUTPUTS, ELEMENTS, GROUPS, ENTITIES.

     Suivant le type d'objet mentionné dans ce document ainsi que dans son prolongement, l'analyste devra utiliser les préfixes correspondants:

     Pour le type SET: préfixe SET-
     Pour le type INPUT: préfixe I-
     Pour le type OUTPUT: préfixe O-
     Pour le type ELEMENT: préfixe EL-
     Pour le type GROUP: préfixe GR-
     Pour le type ENTITY: préfixe ENT-

5. USES: voir 4.

   * UPDATE : liste de SETS, ELEMENTS, GROUPS, ENTITIES mis à jour par le process spécifié.

Remarques:

1. Tout ce qui a été dit aux points 4 et 5 est valable pour l'extension D/IS7b du formulaire D/IS7a.

2. En ce qui concerne les PROCESS, l'analyste devra organiser d'une certaine façon les paragraphes de commentaires: DESCRIPTION ET PROCEDURE.

a) Dans tous les cas, la description claire et concise du traitement effectué par le PROCESS sera développée dans le paragraphe DESCRIPTION;

b) De plus s'il s'agit dûne fonction de contrôle: c'es à dire si la valeur de l'attribut TRAIT est CONTROLE, l'analyste devra spécifier les conditions, les messages d'erreur et un code sévérité ainsi que la légende de ces codes à la suite de la description du contrôle.

c) Dans tous les cas, l'analyste décrira la logique du traitement dans le paragraphe PROCEDURE;. Pour ce faire il utilisera un formalisme tel que celui des tables de décision ou bien un pseudo-langage ou bien même une description dans un langage de programmation.

Document de description d'1 traitement (a)

```
1        *        *        *        *        *        *        72
Nom du PROCESS                  PR-.............................;
GENERATES Liste d'outputs       O-.............................,
                                O-.............................,
                                O-.............................,
                                O-.............................,
                                O-.............................,
                                O-.............................,
                                O-.............................,
                                O-.............................,
                                O-.............................,
                                O-.............................;
RECEIVES  Liste d'inputs        I-.............................,
                                I-.............................,
                                I-.............................,
                                I-.............................,
                                I-.............................,
                                I-.............................,
                                I-.............................,
                                I-.............................,
                                I-.............................,
                                I-.............................;
USES ..............................., ...............................
TO DERIVE          ...............................,
                   ...............................;
USES ..............................., ...............................
TO UPDATE          ...............................,
                   ...............................;
USES               ................................
TO DERIVE          ................................;
USES               ................................
TO DERIVE          ................................;
USES               ................................
TO UPDATE          ................................;
USES               ................................
TO UPDATE          ................................;
```

Description d'1 traitement document (b)

1 * * * * * * 72

Continuation du formulaire D/IS7a

USES ....., ...,
..... , ...

TO UPDATE .....,
..... , ...;

USES ....., ...,
..... , ...

TO UPDATE .....,
..... , ...;

USES ....., ...,
..... , ...

TO DERIVE .....,
..... , ...;

USES ....., ...,
..... , ...

TO DERIVE .....;

USES .....

TO DERIVE .....;

USES .....

TO DERIVE .....;

USES .....

TO DERIVE .....;

USES .....

TO UPDATE .....;

USES .....

TO UPDATE .....;

USES .....

TO UPDATE .....;

B. Description du fonctionnement dynamique du système.

Cette phase d'analyse englobe les objets EVENT, PROCESS CONDITION. L'analyste utilisera le formulaire D/IS8.

1° Informations à spécifiér pour une condition.
****

1. Nom de CONDITION à décrire.

2. FALSE ou TRUE WHILE; paragraphe commentaire décrivant la condition FALSE OU TRUE.

3. BECOMING: liste de noms d'événements causés par le passage de la condition à l'état TRUE. La même chose concerne la passage de la condition à l'état FALSE.

2° Informations à spécifier pour un événement.
****

1. Nom d'événement à décrire.

2. * HAPPENS: system-parameter indiquant le nombre d'occurrences de cet événement.

   * TIMES-PER: Nom d'intervalle pendant lequel surviennent toutes ces occurrences.

3. TRIGGERS: liste de noms de process déclenchés par une occurrence de l'événement spécifié.

3° Informations à spécifier pour un process.
****

1. Nom de process à décrire dynamiquement.

2. HAPPENS: Même chose qu'au point 2 concernant un événement, mais pour un process.

3. * INCEPTION-CAUSES: liste d'événements du process causés par le début.

   * TERMINATION: listed'événements du process causés par la fin du process.

Description du fonctionnement dynamique du système

```
1       *       *       *       *       *       *       72
Nom de la CONDITION          .............................;

    □ FALSE
    □ TRUE    WHILE;  Lignes de commentaires de 72 caractères
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................;
BECOMING TRUE IS CALLED      EV-.........................,
EV-........................., EV-.............................;
BECOMING FALSE IS CALLED     EV-.........................,
EV-........................., EV-.............................;
=====================================================================
Nom d' EVENT                 EV-.........................;
HAPPENS              SYSP-.....................
   TIMES-PER         INTV-.....................;
TRIGGERS             PR-.........................,
PR-........................., PR-.............................,
PR-........................., PR-.............................;
=====================================================================
Nom de PROCESS       PR-.........................;
HAPPENS              SYSP-.....................
   TIMES-PER         INTV-.....................;
INCEPTION-CAUSES     EV-.........................,
EV-........................., EV-.............................,
EV-........................., EV-.............................;

TERMINATION-CAUSES EV-.........................,
EV-........................., EV-.............................,
EV-........................., EV-.............................;
```

31. 3 Corps de la méthode.

314. 1 Contrôles.

Il est difficile de dissocier certains contrôles de la production des résultats. Et cela, parce que les même rapports effectuent ces deux fonctions.

Sauf en ce qui concerne le contrôle du PSL, on peut procéder de deux façons:

- Soit qu'on effectue tous les contrôles et toutes les modifications nécessaires, avant de passer à la production des résultats.

- Soit qu'on ne dissocie pas les contrôles de la production des résultats. Dès lors on effectuera ces deux opérations parallèlement.

La première solution risque, à priori, d'amener une forte duplication de la production de rapports nécessaires aux contrôles et au dossier d'analyse. En revanche elle assure la production d'un dossier basé sur une base de données " cohérente " et " complète ".

La deuxième solution fait gagner des duplications de rapports. Par contre elle risque de demander des révisions du dossier déjà produit si des sources importantes d'erreurs étaient découvertes lors des contrôles.

314. 2 Production des OUTPUTS.

Pour présenter ces résultats, nous allons reprendre chaque document faisant partie du dossier de développement et nous étudierons les moyens qui sont mis à notre disposition par ISDOS pour obtenir les résultats souhaités.

3142. 1 Rapports existants.

En premier lieu, nous nous intéresserons aux possibilités de la version 4.2 sans modifications.

1. Fiche descriptive des traitements d'une phase.

   Le rapport doit reprendre pour chaque phase la description claire et précise du traitement accompli dans la phase.

a) Modalités d'obtention du rapport.
****

Ce rapport peut être obtenu en deux étapes successives:

* La première consiste à effectuer une recherche sélective de tous les objets de type PROCESS ayant PHASE pour valeur de l'attribut TRAIT.
L'instruction PSA correspondante est de la forme: NS S= 'PROCESS AND ATTR = TRAIT, PHASE'.
Le résultat de cette instruction sera un fichier tampon prit par défaut contenant toutes des phases décrites dans la base de données. Si l'utilisateur désire garder ce fichier, il ajoutera le paramètre PUNCH = nom de fichier utilisateur.

* La seconde étape consiste en l'utilisation du rapport PROCESS-SUMMARY de façon à imprimer, pour chaque phase, la section PROCESS correspondante.
Il faudra cependant indiquer à l'Analyser que l'on désire imprimer uniquement les commentaires du paragraphe DESCRIPTION.
Ceci pourra être réalisé grâce à l'instruction PSA: PSUM NPRCD NENT NLEV NOTH NPG

b) Autres possibilités.
****

En variant les critères de sélection avec les attributs TRAIT ainsi que CLE et leurs valeurs, mais en gardant le type PROCESS dans cette sélection: l'utilisateur peut obtenir les descriptions pour tous les types de traitements.

2. Règles de traitement d'une phase ou d'autres traitements.

Ce document doit reprendre pour chaque phase et pour chaque fonction un formalisme représentant les règles de traitement.

a) Modalités d'obtention du rapport.
****

Comme pour le résultat précédent , la procédure comprend deux étapes:

* Une sélection de la forme vue au rapport précédent.

* L'emploi du PROCESS-SUMMARY report de la façon suivante: PSUM PRCD NDESC NENT NLEV NOTH NPG.

Ceci suppose que les règles de traitement ont été correctement décrites sous la forme de commentaires dans le paragraphe PROCEDURE de chaque PROCESS.

b) Autres possibilités.
****

Nous renvoyons le lecteur au rapport précédent en ce qui concerne les extensions possibles.

3. Règles de contrôle des fonctions.

Ce document doit contenir une liste reprenant:

- les conditions de validité.

- les messages d'erreur en cas de contrôle positif.

- Le code sévérité correspondant au message d'erreur.

Pour obtenir ces données, nous avons décidé de regrouper les fonctions de contrôle dans un " groupe logique " caractérisé par la valeur CONTROLE de l'attribut TRAIT.
Pour éviter toute production redondante de rapports, on évitera d'utiliser cette valeur: CONTROLE dans les critères de sélection des deux rapports précédents.

* Modalités d'obtention.
****

Il suffit de procéder comme pour le rapport précédent, mais en modifiant la valeur d'attribut TRAIT dans le rapport NAME-SELECTION et en le remplaçant par CONTROLE.

4. Diagramme d'enchainement des fonctions.

Dans ce document, on désire représenter l'enchainement des fonctions d'une phase par un graphe tel que:

- les sommets représentent les unités d'information ainsi que les fonctions.

- et les arcs représentent les relations qui lient les unités d'information entrantes ou sortantes aux fonctions.
  Les arcs représentent aussi les relations de succession des fonctions.

ISDOS possède un formalisme qui est très proche de celui exposé ci-dessus.

Le lecteur trouvera un exemple comparatif de ces deux formalismes dans les annexes.

a) Modalités d'obtention.
   ****

   - Dans une première étape, nous allons faire une sélection de la ou des phases désirées.

   - Ensuite en utilisant le rapport STRUCTURE, nous allons créer un fichier que nous appelerons DIAGRAM qui contiendra les noms des phases et les noms des fonctions composants ces phases.
     Ceci peut être obtenu par l'instruction PSA: STR PN = DIAGRAM NP LEVELS = 1

   - Enfin à partir du fichier DIAGRAM, nous pourrons obtenir le graphique de chaque phase suivi du graphique de ses fonctions.
     Ceci est fait à l'aide du PICTURE report avec l'instruction PSA: PICT F = DIAGRAM NSTR.

b) Contraintes de manipulations.
   ****

   Le rapport final à inclure au dossier demandera un certain travail manuel, car les résultats du PICTURE report se trouvent dans le format suivant:

   - 1 process par page avec les INPUTS, OUTPUTS, SET, ENTITIES, GROUPS, ELEMENTS qu'il utilise.

   - le PROCESS en amont.

   - et si on emploie le paramètre STR: les PROCESS en aval.

De plus les rapports ISDOS sont présentés suivant le format standard. Il suffit donc de découper les listings à l'aide d'une rogneuse pour disposer de feuilles de format standard.

Enfin, si l'utilisateur désire avoir un diagramme d'une seule pièce, il pourra procéder à un assemblage des résultats. Ce travail sera facilité par les informations données dans ce rapport sur les process qui se trouvent en amont ou en aval.

c) Autres possibilités.
****

Ces même résultats peuvent être obtenus pour l'enchaînement des phases au niveau d'une application. Il suffira d'opérer une sélection de la ou des applications souhaitées et d'utiliser la même procédure.

5. Dictionnaire des informations.

Comme on a pu le voir dans un précédent chapitre sur les dictionnaires de données, il est possible de créer une sorte de métastructure des données parallèlement à la structure des traitements.

Ceci pourra permettre à l'utilisateur de se servir d'ISDOS comme d'un véritable dictionnaire de données. Pour cela il devra combiner de façon adéquate des rapports ISDOS.

a) Besoins de l'utilisateur.
****

* Il désire pouvoir disposer d'une fiche pour chaque information qui reprenne:

le nom de la donnée	les règles d'obtention ou de vérification
la définition	les synonymes
le format	les références à des tables de codification

* Il peut vouloir savoir où est contenue une donnée dans les autres.

* Il peut aussi désirer connaître toutes les données élémentaires qui composent une ENTITE, ...

b) Modalités d'obtention.
****

Les fiches d'information peuvent être obtenues grâce au rapport DICTIONARY.
Il permet d'avoir pour tout objet d'ISDOS:

* le nom des objets,

* la description des objets sous forme de commentaires,

* les synonymes associés au nom d'objet,

* le responsable de la donnée,

* les attributs avec leurs valeurs
* les mots-clés

} associés au nom d'objet

* il faudra utiliser le Name-Selection conjointement avec le DICTIONARY report. L'instruction PSA correspondante serait par exemple: DICT NKEY NEW-PAGE.

- Pour connaître où est contenue une donnée l'utilisateur dispose des CONTENTS reports. On peut utiliser le CONTENTS ANALYSIS reports qui permet de connaître tous les objets qui contiennent les données à analyser. Les résultats peuvent être obtenus sous forme de motrice:

  Si on l'utilise avec l'option contained:

  . les colonnes représentent les données à analyser.
  . les lignes représentent les données qui contiennent ces données à analyser.
  . la présence d'une * à l'intersection d'une ligne et d'une colonne signifie que la donnée à analyser est contenue dans la donnée résultat.

  De tels résultats peuvent être obtenus par l'instr. PSA: CA CNTD
  Auparavant on aura peut-être utilisé un Name-Sélection.

- Pour connaître les composants d'un objet donné , on utilisera le CONTENTS report qui permet de connaître à n'importe quel niveau les données faisant l'objet d'un consists.
  En effet on peut dire jusqu'à quel niveau on désire aller avec le paramètre LEVELS = integer/ALL/LOWEST.
  Ceci peut être obtenu à l'aide de l'instruction PSA: CONT.

c) Contraintes d'utilisation.
****

* La définition de l'objet, les règles d'obtention ou de vérification et les valeurs des codes avec leurs significations devront être placées comme commentaires dans le paragraphe DESCRIPTION.

* Tout ce qui concerne les attributs des ELEMENTS sera traité dans le paragraphe 313.2 de ce chapitre.

6. Description d'une unité d'information.

Ce document a pour but de représenter la structure d'une unité d'information suivant le mode de représentation COBOL avec:

- un nombre niveau.
- un nom de la donnée.
- un libellé.
- une picture.
- des longueurs (moy, max).
- des dimensions (moy, max).

Il existe le rapport STRUCTURE qui permet de produire ce document à l'exception des trois dernières informations de cette liste.
Ce rapport sera modifié et nous verrons comment dans le paragraphe 3132.2 de ce chapitre.

7. Analyse d'activité des Applications.

Ce document permet de savoir dans quelles phases d'une application sont utilisées les unités d'information et quel usage en est fait.

a) Modalité d'obtention.
****

Le DATA-PROCESS INTERACTION report permet d'avoir pour des process donnés toutes les entités, inputs... utilisés par ces process et le type d'utilisation.

Les résultats sont présentés sous forme de matrice:

* les colonnes sont les process
* les lignes figurent les données

A l'intersection des lignes et des colonnes on trouve un code qui signifie telle donnée est:

* créée
* mise à jour
* utilisée comme INPUT et/ou OUTPUT

} De tels PROCESS

Ceci peut être obtenu en utilisant l'instruction: DPI PROCESS.

b) Extensions possibles.
****

Cette analyse peut également être faite à partir des données. Il suffit alors de remplacer le paramètre PROCESS par DATA et bien sûr de fournir un fichier input contenant les noms de données que l'on désire traiter.

7.BIS. Liste des INPUTS ou des OUTPUTS d'une application.

L'utilisateur pourra utiliser la même procédure que pour le rapport précédent.

8. Diagramme des relations entre unités d'information.

Ce document comprend un graphe dont les sommets de type A représentent des entités et les sommets de type B des associations reliant une ou plusieurs entités par des arcs. On désire également connaître les propriétés des entités ainsi que les propriétés des relations liant 2 entités ou même plusieurs.

a) Modalités d'obtention.
****

Le PSA offre la possibilité de recevoir plusieurs types de résultats concernant la structure logique d'une phase.
Comme input on peut avoir soit des entités, soit des relations.
Nous allons prendre le cas des entités en INPUT.

* On peut disposer de plusieurs matrices de résultats. Mais ce qui nous intéresse, c'est surtout
* le RELATION SUMMARY.

Il présente chaque relation avec:

* les entités reliées
* les valeurs de CONNECTIVITY

Mais actuellement, il n'est pas possible d' avoir les données associées à ces relations. Ce type de résultat peut être obtenu grâce au rapport RELATION-STRUCTURE obtenu avec l'instruction PSA: RSTR NEMAT NESUM NSMAT.

b) Contraintes.
****

Il devra par contre représenter lui-même par un graphique la structure logique. Les données associées peuvent être obtenues par le rapport CONTENTS pour les entités et FPS pour les relations.

9. Analyse des volumes pour les U.I.

Ce document a pour but de présenter sous forme de tableau une analyse de volumes par périodes. Pour chaque période on désire avoir:

* le nombre moyen et maximum d'occurences;

* ainsi que le volume moyen et maximum calculé à partir d'une unité de volume pour laquelle on aura une longueur moyenne et maximum.

a) Modalités d'obtention.
****

Il existe dans ISDOS un rapport qui permet de regrouper tous les objets, contenant une clause HAPPENS et qui ont les mêmes fréquences. Il s'agit du rapport FREQUENCY. En entrée on peut avoir soit des noms d'intervalles avec le paramètre INTERVAL, soit des noms d'objets contenant une clause HAPPENS: c'est à dire des objets du type EVENT , INPUT , OUTPUT et/ou PROCESS. Ce rapport peut être obtenu par l'instruction PSA: FREQ.

On peut remarquer que nous sommes loin du document à produire. A notre connaissance, il n'existe aucun rapport dans ISDOS permettant de donner de tels résultats.

Une solution de rechange pourrait être d'utiliser un paragraphe de commentaires pour décrire les périodes et les informations nécessaires à la rédaction d'une analyse de volume à partir des résultats du FREQUENCY report.

10. Document décrivant le fonctionnement dynamique d'une phase.

Ce résultat n'est pas prévu dans le dossier d'analyse, mais il peut facilement être obtenu à l'aide PROCESS-CHAIN report.
Il suffit de lui fournir en Input les noms des phases et d'utiliser l'instruction PSA: PC.

* Extensions possibles.
  ****

  Ce rapport peut être obtenu pour tous les Process. Il faudra seulement varier les critères de sélection.

Le lecteur aura pu voir que l'Analyser offre de grandes possibilités de documentation.
Nous vous avons présenté celles qui intéressent la méthode d'analyse de l'institut.
Toute liberté vous est laissée pour produire d'autres documents à l'aide des rapports existants dans la version 4.2 d'ISDOS.

3142. 2 Rapports modifiés.

Dans l'étude des rapports existants, nous avons noté quelques faiblesses du PSA en ce qui concerne la méthode d'analyse.
Nous vous proposons donc ces modifications dans la suite de ce paragraphe.

1. Rapport CONTENTS.

La fiche de description des unités d'information ne peut pas être obtenue directement grâce aux rapports existants.
En effet: le CONTENTS report permet d'avoir une représentation de toute structure de données avec nombres niveaux et décalage de colonnes suivant les nombres niveaux. On peut également avoir pour chaque donnée la nature de l'objet ainsi que le nom de la donnée.

D'autre part: le rapport ATTRIBUTES permet de représenter:

- les données;
- les attributs de ces données: c'est à dire les noms d'ATTRIBUTS;
- et si on le désire les valeurs de ces attributs.

* Solution proposée.
****

Finalement nous avons décidé de combiner les avantages du rapport ATTRIBUTES avec ceux du CONTENTS.
Afin de signaler à l'Analyser que l'on désire les valeurs des attributs, l'utilisateur emploiera le paramètre DPIC dans la phrase PSA demandant le rapport CONTENTS.

* Caractéristiques de la solution proposée.
****

a) Nombre d'attributs.

Finalement nous avons décidé d'utiliser 5 noms d'attributs:

- IMAGE: représentation de la picture du COBOL.
- LGMOY: longueur moyenne en caractères.
- LGMAX: longueur maximum en caractères.
- DIMOY: dimension moyenne.
- DIMAX: dimension maximum.

b) Valeurs des attributs.

- LGMOY, LGMAX, DIMOY, DIMAX ont pour valeur un nombre entier de 11 chiffres maximum.

- IMAGE aura pour valeur CPPPPP¯QQUU

  C définit la classe de la donnée:

X= Alphanumérique.
N= Numérique non signé.
S= Numérique signé.

  PPPPP= nombre de caractères ou bien nombre de chiffres entiers.

  QQ= nombre de décimales.

  UU= ceci ne devrait pas faire partie de l'analyse fonctionnelle, mais nous pouvons déjà y penser pour des extensions futures.

UU serait une codification de la clause USAGE de COBOL.

C= computational, C1= computational-1,C2 = computational-2
C3= computational -3,I = Indesced.

Exemple:
N4-2C3: il s'agit d'un nombre signé packé qui aure 4 chiffres entiers et 2 décimales au maximum.

c) Le nombre de niveaux.

Si le nombre de niveau dépasse un certain nombre fixé, il n'est plus possible de mettre sur une même ligne tous les attributs. Il existe deux solutions possibles:

1. On garde les décalages suivant les niveaux et si le nombre de niveaux dépasse le contingent fixé, on imprime sur la page suivante les attributs à leur emplacement normal sur la ligne. L'analyste en découpant cette page peut l'ajuster à la suite de la donnée à laquelle se rapportent les attributs " décalés ".

2. On ne fait plus de décalage suivant les niveaux.
   Dès lors on aura par ligne:

   - un emplacement pour le type d'objet.
   - un emplacement pour le nom de la donnée: maximum 30 caractères.
   - un emplacement pour la valeur de l'attribut longueur moyenne.
   - un emplacement pour la valeur de l'attribut longueur maximum.
   - un emplacement pour la valeur de l'attribut dimension moyenne.
   - un emplacement pour la valeur de l'attribut dimension maximum.

Sur cette page, on trouvera un exemple plus explicite de la deuxième solution qui, finalement, a été retenue.

Remarque: toutes les valeurs d'attribut ne seront données que pour les ELEMENTS. Les GROUPS: ne pourront avoir comme attributs que DIMOY et DIMAX.

Exemple de présentation.
****

		IMAGE	LGMOY	LGMAX	DIMOY	DIMAX
1*(ENTITY)	ENT-CLIENT					
2(GROUP)	GR-DESIGN-CLIENT					
3(ELEMENT)	EL-NO-CLIENT	N6	3	6		
3(ELEMENT)	EL-NOM-CLIENT	X30	15	30		
2(ELEMENT)	EL-SOLVABILITE	X1	1	1		
-						
-						
-						
-						

2. Modification du rapport RELATION-STRUCTURE.

Au départ nous désirions ajouter quelque chose à une partie de ce rapport appelée RELATION-SUMMARY. Ce " sous-rapport " devait imprimer pour chaque relation:

- le nom de la relation.
- les deux entités reliées par la relation.
- les connectivities.

Mais en pratique, il s'est avéré que ce sous-rapport n'existait pas.
Nous l'avons écrit pour qu'il donne en plus des informations sur les données associées à la relation.

La présentation est la suivante:

(RELATION) nom de la relation.
BETWEEN nom de l'entité de gauche AND nom de l'entité de droite.
ASSOCIATED DATA
(type de la donnée) nom de la donnée.

Ceci terminait la présentation de la méthode d'utilisation d'ISDOS.
Nous rappelons au lecteur qu'il pourra trouver un exemple des résultats de cette méthode dans les annexes ainsi que les sources des modifications de rapports.

CHAPITRE 4. EVALUATION DE LA MISE EN OEUVRE DE LA METHODE D'ANALYSE A L'AIDE D'ISDOS.

Cette évaluation de la mise en oeuvre de la méthode d'analyse fonctionnelle va surtout s'attacher aux limites du logiciel dans ce contexte particulier.

En effet, il nous a semblé que les possibilités d'ISDOS avaient été suffisamment développées dans les chapitres 1 et 2.

Nous examinerons ces limites sous trois angles:

- le premier concernera les données initiales de la méthode.
- le second concernera les fonctions d'ISDOS qui regroupent les fonctions de contrôle et de documentation.
- le troisième concernera les résultats de la méthode produits par ISDOS.

4. 1 Limites.

Nous avons pu déceler certaines faiblesses d'ISDOS au cours des manipulations que nous avons faites avec ce logiciel.
En voici un exposé que nous espérons suffisamment critique.

41. 1 Données initiales de la méthode.

La critique des Inputs sous-entend la critique du modèle d'ISDOS et la critique du langage PSL.

* Désignation des objets.

ISDOS identifie chaque objet grâce à son type et à un nom de 30 caractères maximum.
De plus le paragraphe de commentaires DESCRIPTION permet de définir plus précisément un objet particulier ainsi qu'une foule de détails concernant cet objet.
C'est ici que se trouve, à notre avis, une petite imperfection.
En rendant ces commentaires variables (en longueur et en nature d'information), on a hypothéqué toute chance de pouvoir les utiliser comme support de description des objets dans tout rapport.

Certains nous rétorquerons que 30 caractères permettent d'avoir un identifieur complet. Peut être, mais lorsque nous avons retranché le préfixe et les tirets, il ne reste plus tellement de place pour définir un mot facile à décrypter. Cela sera à fortiori rendu plus malaisé à partir du moment ou chacun possède ses règles personnelles d'élision et d'abrégé.

Si on accepte ces difficultés et si on considère qu'il est très difficile et pratiquement impensable d'insérer la description de tous les objets dans tous les rapports, on pourrait envisager une solution qui sera proposée dans les extensions de ce travail.

* Structuration des objets.

Nous n'avons rien à redire en ce qui concerne les PROCESS.
Cependant, nous avons recontré quelques problèmes avec la structuration des données.
Voici résumés dans un tableau les possibilités de décomposition des données à l'aide de la relation PSL CONSISTS.

SET	INPUT	OUTPUT	ENTITY	GROUP
INPUT	INPUT	OUTPUT	GROUP	GROUP
OUTPUT	GROUP	GROUP	ELEMENT	ELEMENT
ENTITY	ELEMENT	ELEMENT		

On voit bien qu'il existe plusieurs points de rupture dans la décomposition de objets.

- Le premier se trouve au niveau du SET.

Il est très ennuyeux que le SET ne puisse pas contenir de groupes et même d'éléments qui lui soit propre.
Lorsque l'on désire définir un fichier temporaire, on doit le décrire à partir d'INPUTS, OUTPUTS et ENTITIES existants ou que l'on définira par après. Si certaines informations ne se trouvent pas dans ces objets, il faudra créer un objet d'un de ces trois types qui contiendra ces informations et que l'on reliera au SET que l'on a défini.

Cette façon de faire nous semble lourde et fastidieuse.

- Le second se trouve au niveau des INPUTS et des OUTPUTS.

  Contrairement au SET, ces deux types d'objets ne peuvent pas contenir d'ENTITIES. Par contre ils peuvent contenir des GROUPS et des ELEMENTS.

- Le dernier se trouve au niveau des ENTITIES.

  Une entité ne peut pas être constituée d'entités. C'est ce qui nous a forcé à utiliser l'objet SET pour définir les notions de structure logique, structure conceptuelle.

* Paragraphes de commentaires.

Il semble que les paragraphes de commentaires, en plus des informations descriptives, contiennent tout ce qui ne trouve pas de solution formelle dans le modèle.
Un exemple sera plus explicite:

- l'expression d'une condition fait l'objet de commentaires, alors qu'il existe de nombreuses façons de formaliser les expressions conditionnelles.

* Spécifications concernant le cahier des charges.

Nous estimons que la solution proposée dans le chapitre 3 est acceptable pour tout ce qui concerne les informations textuelles du cahier des charges.
Cependant, le concepteur éprouvera certaines difficultés s'il désire décrire la configuration hardware nécessaire à l'implémentation décrite dans le cahier des charges. Il pourra le faire dans un paragraphe de commentaires, mais il risque de déboucher sur une solution très sophistiquée, s'il veut faire cela en utilisant les objets d'ISDOS.

* Description d'une cellule d'activité.

La solution proposée au chapitre 3 nous parait acceptable en attendant une modification du PSL que nous proposerons comme extension à ce travail.
Car, comme nous l'avons vu, il subsiste des problèmes d'identification:

- des cellules fournissant les informations Inputs et des cellules recevant les informations Outputs.
- des quantités de ressources consommées.

41. 2 Fonctions d'ISDOS.

Nous avons déjà parlé du nombre limité des rapports qui est dû à la structure de la base de données ADBMS.

412. 1 Fonction de contrôle.

Par le fait même du nombre limité de rapports, les fonctions de contrôle sont également limitées:

* à la syntaxe du PSL

* à la complétude de la base de données par rapport au PSL existant

* à certains contrôles de cohérence entre les types d'objets utilisés par les "RELATIONSHIPS"

412. 2 Fonction de documentation.

La limite principale est une conséquence de la structure de la base de données: c'est la non existence d'un véritable outil d'interrogation.

Une autre limite concerne certaines faiblesses rencontrées dans les résultats du RELATION-STRUCTURE report.

41. 3 Outputs de la méthode.

La plupart des outputs de la méthode d'analyse peuvent être produits par ISDOS.
Seuls les documents décrivant:

- les structures logiques
- les diagrammes d'enchaînement des fonctions

nécessitent un certain travail de mise en forme.

A part cela, nous n'avons pas d'autres critiques à formuler quant aux documents du dossier de développement produits par ISDOS.

CHAPITRE 5. CONCLUSIONS.

Nous estimons qu'il n'est pas nécessaire de reprendre en d'autres termes le contenu de la table des matières.

Par contre, nous allons tenter de dégager ce que devrait être l'apport de l'utilisation d'un logiciel d'aide à l'analyse et plus particulièrement, quel est l'apport d'ISDOS.

Ensuite nous proposerons des extensions possibles à ce travail.

* Une première extension visera à une modification de la version 4.2 qui portera surtout sur des propositions de modification du PSL.

* La seconde extension concernera l'utilisation du métagénérateur qui sera bientôt mis au point par les chercheurs chargés de l'étude du projet ISDOS.

5. 1 Apport de l'utilisation d'un logiciel d'aide à l'analyse.

L'utilisation d'un logiciel d'aide à l'analyse doit améliorer le processus de développement d'un système d'information.

1. Il doit améliorer la productivité des analystes, en leur donnant des outils qui visent à accroître la qualité des spécifications du système à analyser.

   Les spécifications seront:

   * plus fiables
   * plus précises
   * plus complètes
   * plus pertinentes

   La qualité accrue de ces informations va déboucher sur la production de dossiers de développement:

   * plus complets
   * mieux structurés
   * avec le niveau de détail désiré
   * respectant des standards de documentation
   * produits plus rapidement que par le passé.

   Les améliorations des dossiers conduiront à une amélioration de la programmation et de l'implémentation des applications.

2. Il doit faciliter la maintenance du système d'information.

En centralisant dans une banque de données toutes les informations concernant le système d'information, le logiciel d'aide à l'analyse permet:

* d'évaluer rapidement l'impact de tout changement dans le système d'information

* de calculer les coûts de ces changements

* de rapidement mettre en oeuvre les procédures de mise à jour de la base de données, des dossiers, et des programmes d'application.

3. Il doit ouvrir des perspectives futures d'amélioration du processus de conception des systèmes d'information:

* en facilitant la création d'outils permettant la simulation du comportement réel du système

* en étant ouvert à tout changement des structures d'information du système,

5. 2 Apport d'ISDOS.

Il est difficile de parler des apports du logiciel ISDOS dans l'amélioration du processus d'élaboration d'un système d'information, sans être redondant avec le paragraphe précédent. Nous allons essayer de résumer ces avantages le plus possible.

En ce qui concerne l'amélioration de la qualité de l'analyse fonctionnelle.
Il est inutile de revenir sur la qualité:

* des spécifications
* des possibilités de contrôle
* des possibilités de documentation
* des outputs obtenus.

Il est bien évident que cette qualité des dossiers de développement va se répercuter jusque l'exploitation des programmes d'application.

La qualité de la maintenance du système d'information sera également améliorée quant à:

* la rapidité de localisation des éléments du système à modifier

* la rapidité d'exécution des procédures de mise à jour de la documentation et des programmes d'application

Enfin les extensions futures ne sont pas un vain mot en ce qui concerne ISDOS. Nous en parlerons dans la suite de ce chapitre.

5. 3 Extensions.

53. 1 Extensions de la version 4.2.

Le système ISDOS est conçu de telle façon qu'il aurait été très difficile de mettre en oeuvre toute modification du PSL tout en ayant la possibilité de terminer ce travail dans les délais qui nous étaient impartis.
Cela explique pourquoi nous vous proposons ces modifications en tant qu'extensions à ce travail.

531. 1 Modifications du PSL.

1. Modification de la désignation des objets.

   Afin de pouvoir faire figurer, au gré de l'utilisateur, une description de chaque objet ISDOS qui soit de longueur standard, nous proposons la création d'un nouveau paragraphe de commentaires.

   * Il aurait pour nom: SIGNIFICATION;
   * Il aurait une longueur fixe et suffisante pour décrire tout nom d'objet ISDOS de façon non équivoque.
   * Si l'utilisateur le désire, il peut le faire figurer sur tout rapport. Le formatage des rapports devra être modifié en conséquence. Mais cela ne devrait pas poser trop de problèmes du fait que cette signification aura une longueur fixe.
   * Le paragraphe DESCRIPTION; servira à tout complément d'information concernant cet objet.

2. Spécification des informations concernant une cellule d'activité.

   En remplacement de la solution proposée au point 513. 3 du chapitre 3, on pourrait envisager la création de nouveaux types d'objets ISDOS:

   * CELLULE: représentera une cellule d'activité.

* RESSOURCE: représentera un type de ressource.
* UNITE-DE-MESURE: définira l'unité de consommation d'une ressource.

Nous devons créer également de nouvelles relations PSL entre ces objets.
Voici comment nous concevons chaque nouvelle section PSL.

CELLULE Section

PERFORMS Process-name(s)

HOLDING (Set- / Entity- / Group- / Element-) name(s)

GETTING (Set- / Entity- / Input- / Group- / Element-) name(s)

PUTTING (Set- / Entity- / Output- / Group- / Element-) name(s)

CONSUMING Ressource-name(s) [AT] RATE [OF] integer Unite-de-mesure-name
[,Ressource-name(s) [AT] RATE [OF] integer Unité-de-mesure-name ...;]

NEXT-CELL [ARE] Cellule-name(s);

FOR-CELL [ARE] Cellule-name(s);

RULE; comment-entry;

* Une cellule exécute (PERFORMS) un ou plusieurs traitements en gardant des sets, des entités, des groups ou des éléments
  - en accédant à des sets, des entités, des inputs, des groups ou des éléments
  - en produisant des sets, des entités, des outputs, des groups ou des éléments
  - en consommant une ou plusieurs ressources à un taux de autant d'unités de mesure.

* La cellule est reliée (NEXT-CELL) aux cellules à qui sont destinées les unités d'information produites par la cellule.

* La cellule est reliée (FOR-CELL) aux cellules d'où proviennent les unités d'information produites par la cellule.

* Les règles de traitement de la cellule sont décrites dans le paragrphe de commentaires RULES;

RESSOURCE Section

MESURED [BY] Unité-de mesure-name;
CONSUMED [BY] Cellule-name(s) [AT] RATE [OF]
integer [TO] PERFORM Process-name(s)
[,Cellule-name(s) [AT] RATE [OF]
integer [TO] PERFORM Process-name(s) ...;]
SUBPARTS ARE Ressource-name(s);

* Une ressource est mesurée par une unité de mesure.

* Une ressource est consommée par une ou des cellules d'activité à un taux d'un nombre entier d'unités de mesures pour exécuter un ou plusieurs process.

* Une ressource peut être décomposée en d'autres ressources.

UNITE-DE-MESURE Section.

MESURES Ressource-name(s);

* Une unité de mesure sert à mesurer la consommation de une ou plusieurs ressources.

Nous n'avons placé dans ces sections que le strict minimum. On peut y ajouter des statements PSL comme:

* ATTRIBUTES ARE liste d'attributs et de valeurs...;
* DESCRIPTION; commentaires;
* SYNONYMS ARE liste de synonymes...;
* ETC...

3. Une autre extension à ce travail pourrait être:
un prolongement direct à ce mémoire qui étudierait l'expression de la méthode d'analyse organique à l'aide du logiciel ISDOS ou de tout autre outil d'aide à l'analyse.

531.2 Extensions du PSA.

1. Extensions dues aux modifications du PSL.

a) Pour faire figurer le paragraphe de commentaires SIGNIFICATION; dans certains rapports, il faudrait procéder à des modifications qui porteraient sur:

- l'accession à la SIGNIFICATION dans la base des données.

- la présentation du rapport compte tenu de l'ajoute de ces commentaires.

b) Il faudrait envisager la production de rapports concernant les cellules d'activités et entre autres:

- un rapport représentant le diagramme des flux d'information entre les cellules d'activité;

- un rapport qui résume la cellule d'activité et qui reprendrait:

  * les unités d'information entrantes et leur origine, sortantes et leurs destinations, gardées dans la cellule d'activité.
  * le ou les process concernés par la cellule d'activité.
  * les ressources consommées par la cellule d'activité ainsi que les informations concernant la quantité de ressources consommées.
  * les règles de traitement concernant l'activité de la cellule d'activité.

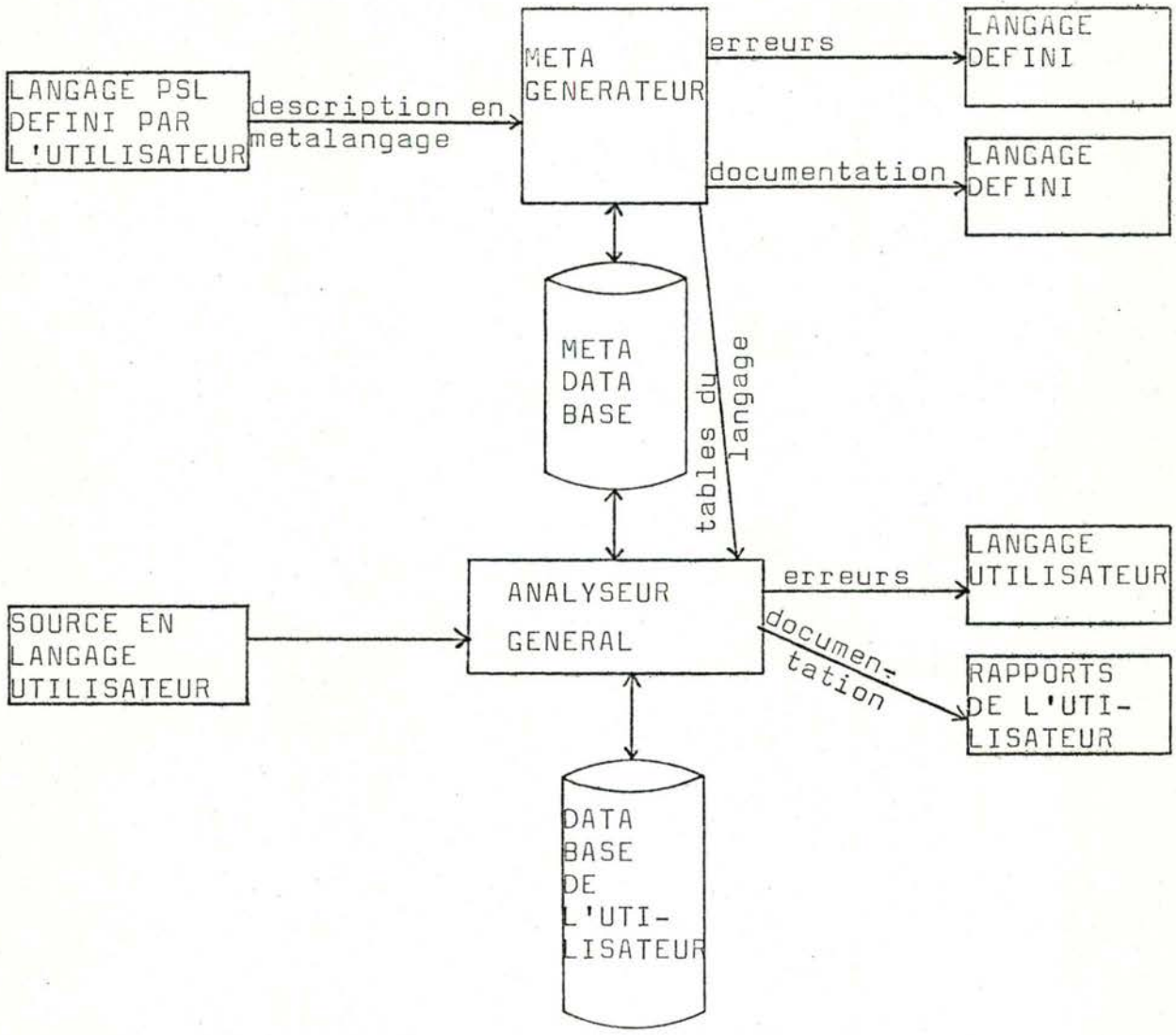
2. Extensions à la version 4.2 non modifiée.

On pourrait envisager la création d'un rapport produisant sous la forme d'un graphe les relations entre les entités créées par l'utilisateur.

5. 4 Métagénérateur.

Les chercheurs du projet ISDOS seront très bintôt à même de fournir un métasystème ISDOS.

Le schéma général de ce système est le suivant:



- A partir du méta-langage, l'utilisateur décrit son PSL.

- Le métagénérateur crée une Métadatabase définissant le nouveau langage avec ses objets et ses relations ainsi que les tables du nouveau langage.

- La source écrite en nouveau PSL est compilée par l'analyseur général qui crée la base de données de l'utilisateur qui peut ensuite demander les rapports désirés.

Ce métasystème ouvre réellement des horizons nouveaux à la conception assistée par ordinateur à l'aide d'ISDOS.

Des études devront être faites pour connaître les possibilités de ce nouveau système ainsi que ses limites.

Mais au premier abord, nous pouvons dire que ce métasystème nous entraîne loin des possibilités de la version 4.2 d'ISDOS que nous avons utilisée au cours de ce travail.

BIBLIOGRAPHIE.

1. Références à la méthode d'analyse fonctionnelle.

(1) F. BODART; Problèmes d'Organisation et Méthodes d'Analyse Fonctionnelle;

Cours FNDP, 1977 - 1978, Namur.

2. Références aux Dictionnaires de données et à la conception assistée par ordinateur.

(2) BELKIS LEONG-HONG and BEATRICE MARRON;

Technical Profile of Leven Data Element Dictionary Directory Systems;

Institute for Computer Sciences and Technology, National Bureau of Standards;
Washington, D.C.20234, February 1977.

(3) The British Computer Society Data Dictionary Systems Working Party Report;

SIGMOD Recors, VOL 9, NO 2, 1977.

(4) Installing a Data Dictionary;

in E.D.P. Analyser, vol. 16, N° 1, January 1978.

(5) Y. PIGNEUR; Synthèse et Propositions de Spécifications Théoriques d'un Système de Documentation;

Mémoire présenté lors de l'année académique 1976 - 1977 à l'Institut d'Informatique FNDP, Namur.

(6) J.P. CRESTIN; La C.A.O. Outil de l'Ingénieur;

dans le Bulletin de l'IRIA (Institut de Recherche d'Informatique et d'Automatique), N° 37, page 2, Avril 1977.

3. Références au système ISDOS.

(7) A. HERSHEY and M.G. GRAULICH;

The Structure and Contents of a PSA Data Base;

ISDOS WORKING PAPER, N° 166, August 1976.

(8) Ernest A. HERSEY and G. FLEMENG.

A Data Base Management System (ADBMS);

based on DETG 71, November 1977

(9) Y. YAMAMOTO; ADBMS Commands Summary;

ISDOS Ref. 7730-0192-0, October 1977.

(10) D. TEISCHROEW and Z. GACKWSKI;

Beginning Use of PSL/PSA;

ISDOS Working Paper, Nº 184, June 1977.

(11) Problem Statement Language (PSL)
Introduction and Users Manual,Version 4.2;

ISDOS Project, University of Michigan, July 1977.

(12) Problem Statement Language (PSL)
Language Reference Manual, Version 4.2;

ISDOS Project, May 1977.

(13) Problem Statement Language (PSL)
Language Reference Summary, Version 4.2;

ISDOS Project, November 1977

(14) Problem Statement Analyser Reports, Version 4.2

ISDOS Project, July 1977.

(15) Problem Statement Analyser, Version 4.2;
Command Description;

ISDOS Project, June 1977.

(16) Problem Statement Analyser, Version 4.2;
Users Manual;

ISDOS Project, May 1977.

(17) Problem Statement Analyser (PSA), Version 4.2;
Command Reference Summary;

ISDOS Project, November 1977

(18) Ernest A. HERSHEY, Paul W. MESSINK and Bruce D. FRANSON;

Procedure and Coventions for Developing User's Loftware for PSA, Version 4.1;

ISDOS Working Paper Nº 164, August 1976.

(19) R. van GHENT, F.A. DETTI and M. TULIC;

Structured Analysis Guidelines and Standards;

Chase Manhattan Bank, N.A.; May 1978.

BUMP
0 0 3 2 1 3 3 8 5
*FM B16/1978/08/1

FACULTÉS UNIVERSITAIRES NOTRE-DAME DE LA PAIX - NAMUR

INSTITUT D'INFORMATIQUE

ANNEE ACADEMIQUE 1977-1978

Mise en œuvre d'une méthode d'analyse fonctionnelle à l'aide du Logiciel Isdos

(Annexes)

Mémoire présenté en vue de l'obtention

du grade de

Licencié et Maître en Informatique

F. DURIGNEUX

8/

FACULTES
UNIVERSITAIRES
N.-D. DE LA PAIX
NAMUR

Bibliothèque
F-MB 16
1978/8/2

FM B 16 / 1978 / 8 / 2

FACULTÉS UNIVERSITAIRES NOTRE-DAME DE LA PAIX - NAMUR

INSTITUT D'INFORMATIQUE

ANNEE ACADEMIQUE 1977-1978

Mise en œuvre d'une méthode d'analyse fonctionnelle à l'aide du Logiciel Isdos

(Annexes)

Mémoire présenté en vue de l'obtention

du grade de

Licencié et Maître en Informatique

F. DURIGNEUX

LBS 321386

BIBLIOTHEQUE
NAMUR
F.N.D.P.

TABLE DES MATIERES.

1.	ENONCE DE L'EXEMPLE	3
2.	METHODE D'ANALYSE DE L'INSTITUT	8
	Dictionnaire des informations	9
	Table de codification	10
	Fiche descriptive du traitement	11
	Diagramme de flux d'information	12
	Diagramme d'enchaînement des fonctions	13
	Règles de traitement	15
	Règles de contrôle	16
	Description d'une unité d'information	17
	Analyse des volumes	18
	Analyse d'activité	19
	Diagramme des relations entre les unités d'information	20
3.	RESULTATS DE LA METHODE D'ANALYSE ASSISTEE PAR ISDOS	21
3. 1	Modifications de rapports	21
31. 1	CONTENTS report	21
	Routine INITIM	21
	Routine PRPIC	22
	Source compilée d'INITIM	24
	Source compilée de PRPIC	28
31. 2	RELATION-STRUCTURE report	34
	Source compilée de la routine SUMRUT	35
3. 2	Résultats obtenus à l'aide d'ISDOS	39
32. 1	Documents de spécification.	39
	Métastructure	40
	Structure logique	41
	Output	42
	Set	43
	Groupe	44
	Elément	45
	Interface	46
	Traitement statique	47
	Traitement dynamique	48
32. 2	Rapports d'ISDOS	49
	Complétude uniquement	50
	Dossier de développement	55
	Compléments	71

INTRODUCTION.

Dans ces annexes, nous allons vous présenter les résultats de la phase pratique de notre travail.

La première partie de ces annexes sera consacrée à l'énoncé d'un exemple que nous avons utilisé pour illustrer la méthode d'analyse et la méthode d'utilisation d'ISDOS.

La deuxième partie des annexes vous présentera les documents de la méthode d'analyse de l'institut appliquée à l'exemple exposé dans la première partie.

La dernière partie de ces annexes vous présentera les résultats de la méthode d'analyse assistée par le logiciel ISDOS.
Dans un premier temps nous vous exposerons de façon un peu plus détaillée les modifications apportées au rapports existants:

* CONTENTS-ANALYSIS report

* RELATION-STRUCTURE report

Et enfin, nous vous présenterons les résultats obtenus à partir des rapports d'ISDOS, ainsi que les spécifications. Nous prendrons soin de vous montrer les résultats correspondant aux documents présentés dans la deuxième partie de ces annexes. Cela permettra de comparer les deux versions du dossier de développement.

1. ENONCE DE L'EXEMPLE.

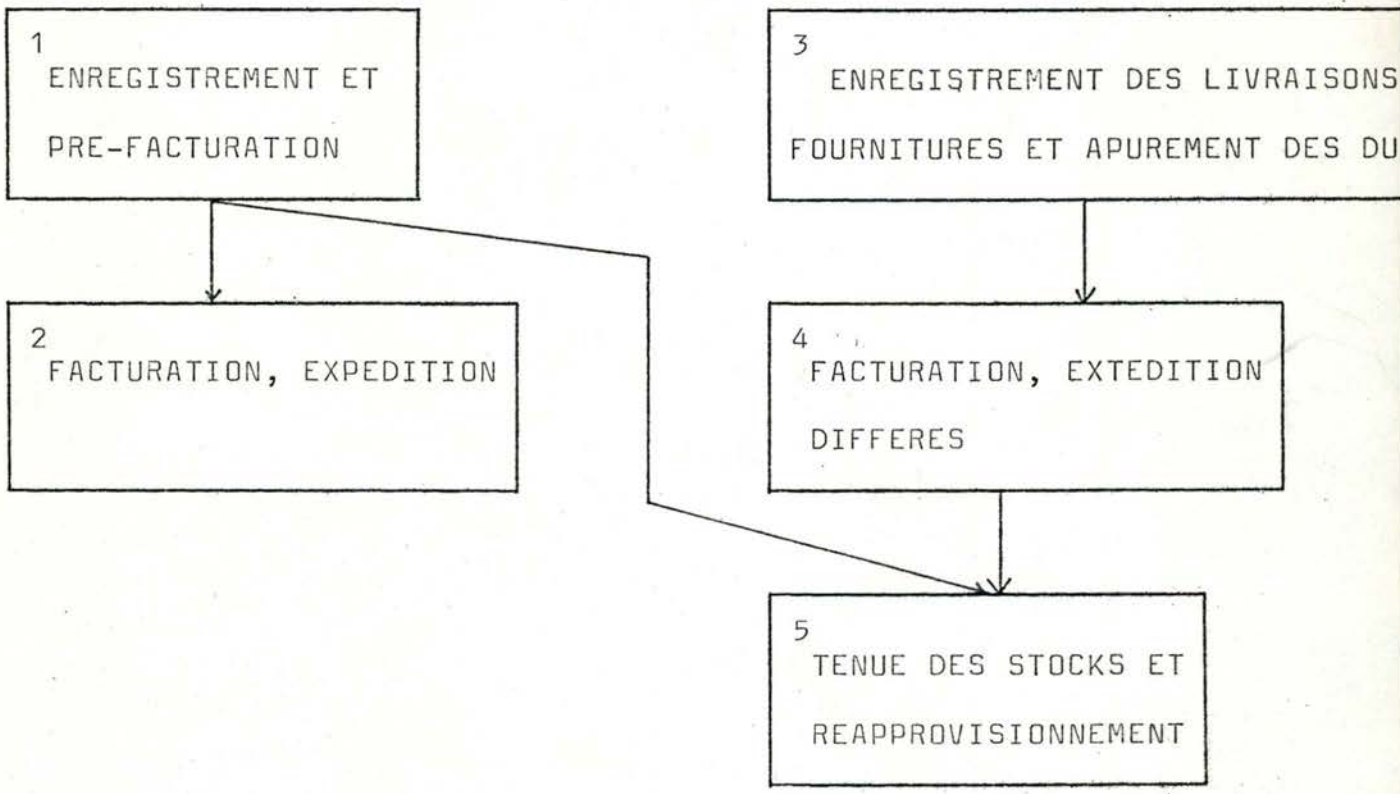
Afin d'illustrer la méthode d'analyse nous avons repris une partie d'un cas de traitement de l'information proposé par Monsieur F. BODART dans le cadre de son cours intitulé MIS (Management Information System).

Il existe une version BATCH et une version TELETRAITEMENT de cet exemple. Nous avons repris la version BATCH. Nous nous sommes limités à l'analyse d'une phase de l'application proposée, c'est pourquoi nous ne vous présenterons que la phase Enregistrement Commande Client et Préfacturation.

APPLICATION : FACTURATION ET REAPPROVISIONNEMENT.

On considère une application facturation et réapprovisionnement dans une entreprise de vente par correspondance.

Elle contient les 5 phases suivantes:



PHASE ENREGISTREMENT COMMANDE CLIENT ET PRE-FACTURATION

- *Evénement déclencheur :* Bon de commande (M1)
  Paramètres : cfr. Informations du Bon de commande

- *Périodicité :* quotidienne

- *Messages :*
  - Messages Refus d'une commande (M2), (M20)
  - Bon de réquisition (M3)
  - Liste des dus par produits (M4)
  - Message Refus d'une commande pour cause de non-solvabilité (M5)
  - Pre-facture (M6)
  - Liste des commandes différées (M40)

N.B. La pré-facture n'est pas imprimée, elle est stockée sur un support auxiliaire dans un fichier des pré-factures.

o

o o

DESCRIPTION DE LA PHASE

PAS 1 : Pour chaque commande

PAS 1.1. : Contrôle du client : N° Client, adresse

PAS 1.1.1. : Si client existe dans le fichier client et si adresse inchangée, aller au PAS 1.2.

PAS 1.1.2. : Sinon, déclenchement de phase mise-à-jour du fichier client

PAS 1.2. : Contrôle de solvabilité du client

PAS 1.2.1. : Si code solvabilité +, alors aller au PAS 1.3.

PAS 1.2.2. : Sinon, si code -, alors,
- déclencher refus de la commande, message (M5)
- aller au PAS 1.10.

PAS 1.3. : Contrôle du chèque (il existe signature, libellé correct du montant)

PAS 1.3.1. : Si chéque invalide, alors
- imprimer message de refus de la commande (M20)
- aller au PAS 1.10.

PAS 1.3.2. : Sinon, aller au PAS suivant

PAS 1.4. : Pour chaque ligne, contrôle de la commande

PAS 1.4.1. : Si REF PROD et $Q^{té}$ $C^{dée}$ existent, alors, aller au PAS 1.4.3.

PAS 1.4.2. : Sinon, enregistrer le refus de la ligne

PAS 1.4.3. : Fin de traitement d'une ligne

PAS 1.5. :

PAS 1.5.1. : Si pas de refus de ligne, alors aller au PAS 1.6.

PAS 1.5.2. : Sinon,
- imprimer message de refus de la commande (M2)
- aller au PAS 1.10.

HYPOTHESE : Une commande est acceptée ou rejetée *globalement* ; si la référence d'un produit est incorrect, on envoie au client un formulaire où tous les éléments de sa commande sont repris en lui demandant de préciser les données incorrectes. La commande n'est pas mise en attente. Lorsque le client renvoie le formulaire, ce dernier sera considéré comme une nouvelle commande. Cette hypothèse d'organisation est faite pour des raisons d'efficacité dans le cadre du traitement "Batch".

PAS 1.6. : Enregistrement du montant du chèque et mise-à-jour de l'avoir
N.AVOIR = AVOIR + MONT. chèque
N.B. : Une variable précédée de la lettre N indique une nouvelle valeur prise par la variable.

PAS 1.7. : Pour chaque article d'une commande, procéder au contrôle de disponibilité

PAS 1.7.1. : Si stock = 0, alors
- $Q^{té}$ due = $Q^{té}$ $C^{dée}$
- M.A.J. Fichier dus par produits
  N.TOT DUS = TOT DUS + $Q^{té}$ due
- M.A.J. Fichier dus par COMM. clients (COMM. DIFF.)
  DIFF = $Q^{té}$ due
- Aller au PAS 1.7.3.

N.B. : DIFF signifie différés.

PAS 1.7.2. : Sinon,
- $Q^{té}$ livrée = Min (Stock, $Q^{té}$ $C^{dée}$)
- N. Stock = Stock - $Q^{té}$ livrée
- N. TOT. LIV = TOT. LIV + $Q^{té}$ livrée
- Montant par PROD = $Q^{té}$ livrée x P.U.
- M.A.J. Fichier Stock PROD : Stock, TOT. LIV.
- Imprimer Bon de réquisition

N.B. : P.U. signifie Prix unitaire
TOT. LIV. signifie Total (cumul) des livraisons depuis le début de la période.

PAS 1.7.2.1. : Si $Q^{té}$ livrée = $Q^{té}$ $C^{dée}$, alors aller au PAS 1.7.3.

PAS 1.7.2.2. : Sinon,
- $Q^{té}$ due = $Q^{té}$ $C^{dée}$ - $Q^{té}$ livrée
- M.A.J. Fichier dus pas produits
  N.TOT DUS = TOT. DUS + $Q^{té}$ DUE
- M.A.J. Fichier dus par COMM. clients (COMM. DIFF.)
  DIFF = $Q^{té}$ due

PAS 1.7.3. : Fin traitement d'un article d'une commande

PAS 1.8. : Calculer montant total hors-taxe (Mont. H.T.)
Mont. H.T. = Σ Montant par PROD.

PAS 1.8.1. : Si Mont. H.T. = 0 (Stock = 0 pour $\forall$ PROD $\in$ COMM), alors aller au PAS 1.10.

PAS 1.8.2. : Sinon,
- Calcul Montant T.V.A. (Mont. T.V.A.)
  Mont. T.V.A. = Mont. H.T. x T.V.A.
- Calcul Montant Total (Mont. T.T.)
  Mont. T.T. = MONT H.T. + Mont. T.V.A.

HYPOTHÈSE : le taux T.V.A. est le même pour chaque produit.

PAS 1.9. : Etablissement de la pré-facture

PAS 1.10. : Fin traitement d'une commande

PAS 2. : Etablissement liste des dus par produits

PAS 3. : Etablissement liste des dus par commande-clients, c'est-à-dire la liste des COMM. DIFF.

o

o o

2. METHODE D'ANALYSE DE L'INSTITUT.

Dans le souci de rendre le volume de ces annexes minimum, nous nous sommes limités à vous présenter un exemple caractéristique de chaque document composant le dossier de développement.

Vous trouverez les résultats correspondants à ces documents dans la troisième partie de ces annexes.

DD/7 DICTIONNAIRE DES INFORMATIONS

NOM: EL-NO-CLIENT
Version: 1 Folio: 9. -

Système: Entreprise de vente par correspondance Date: 7-8-78
Sous-système: Service commercial
Application: Facturation et réapprovisionnement Auteur: Durigneux
Dossier: Dossier de développement

RUBRIQUE / UNITES (biffer)

LIBELLE

Il s'agit du numéro d'un client de l'entreprise par correspondance.

FORMAT / STRUCTURE

TYPE: numérique non signé

LONGUEUR: 6 chiffres entiers

STRUCTURE:
Premier chiffre: code pays
deuxième chiffre: code province, si le pays est la Belgique
quatre derniers chiffres: numéro de client interne à l'entreprise

REGLE D'OBTENTION OU DE VERIFICATION

Ces informations se trouvent dans la table de codification.

SYNONYMES	rfr. UNITES	rfr TABLES DE CODIFICATION
		Table DD8/ 1

RUBRIQUE : EL-NO-CLIENT
CLASSE : Numérique non sign

DD/8 | | TABLE DE CODIFICATION

Version : 1 Folio : 1

Système : Entreprise de vente par correspondance
Sous-système : Service commercial Date : 7-8-78
Application : Facturation et réapprovisionnement
Dossier : Dossier de développement Auteur : Durigneux

RUBRIQUE : Numéro de client : 6 caractères numériques
CLASSE DE SIGNIFICATION :

REPERTOIRE DES VALEURS	SIGNIFICATIONS
X X XXXX	
*	Code pays
1	Belgique
2	CEE
*	Code province si Belgique sinon 0
1	Brabant
2	Flandre occidentale
3	Flandre orientale
4	Luxembourg
5	Liège
6	Namur
7	Limbourg
8	Hainaut
9	Anvers
*	Numéro donné au client par ordre croissant d'introduction dans le signalétique

DD/1 | FICHE DESCRIPTIVE DU TRAITEMENT

PHASE: Enregistrement commande client et préfacturation
PROCESSUS:
Version: 1 Folio: 11. -

Sous-système: Service commercial
Application: Facturation et réapprovisionnement

Date: 7-8-78
Auteur: Durigneux

Cette phase est déclenchée par la réception d'un bon de commande client.

* Pour chaque commande, on va procéder au contrôle de validité au niveau du client et au niveau du produit.

* Pour les commandes acceptées, on effectue un contrôle de disponibilité pour chaque produit. Ceci permet de distinguer les commandes différées composées de lignes de commande différées des commandes exécutables composées de lignes de commande exécutables.
Les commandes différées sont imprimées.

* On traite ensuite les commandes exécutables en mettant à jour le fichier stock et en éditant la liste des dus par produit ainsi que les bons de réquisition à destination du service de gestion des stocks.

* On calculera enfin les préfactures pour chaque commande prise en considération et on les mémoriseras dans un fichier des préfactures.

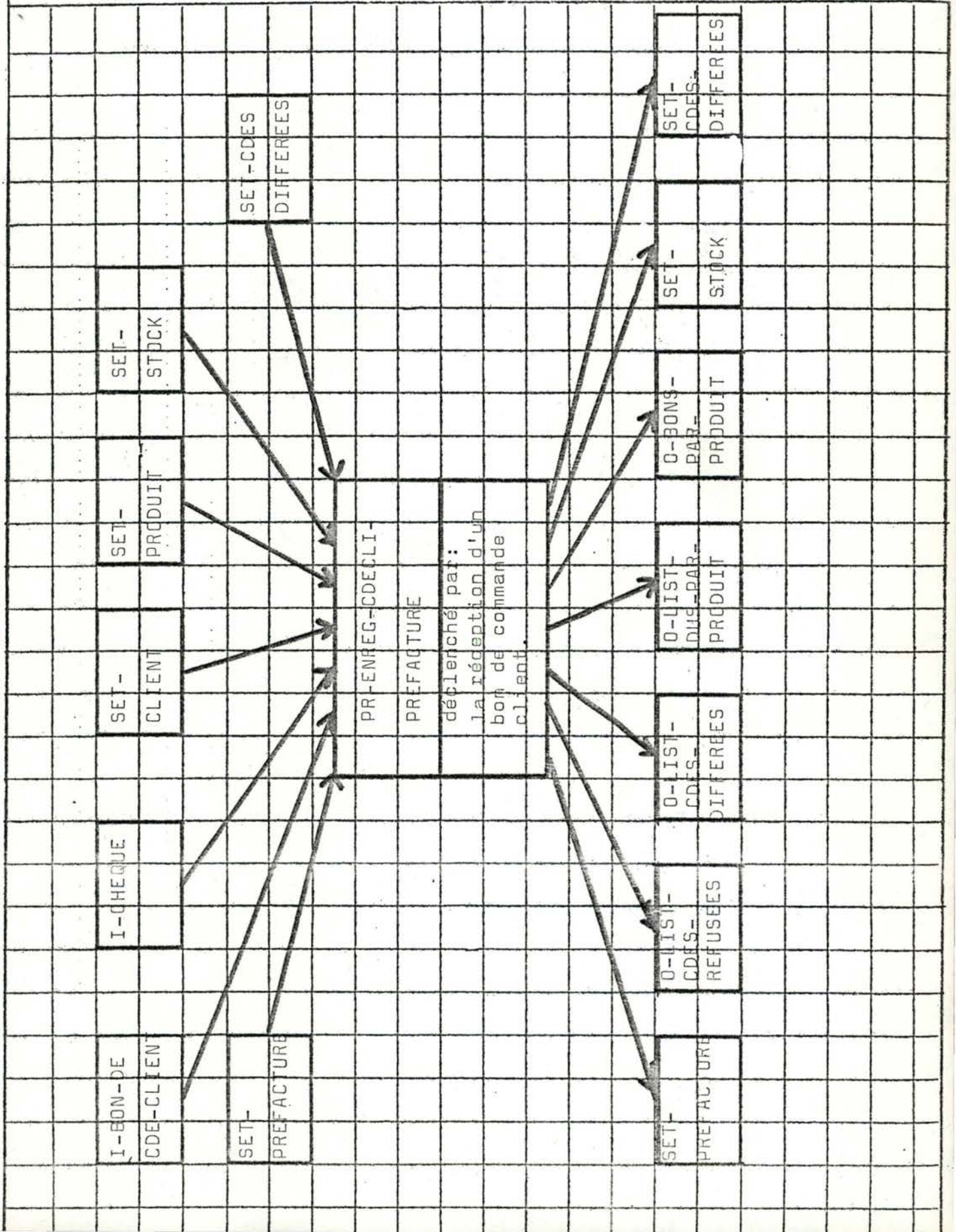
DIAGRAMME DE FLUX D'INFORMATION	DC/2 VERSION FOLIO

. IDENTIFICATION DU DIAGRAMME — Désignation du diagramme

Sous-système : Service commercial

Application : Facturation et réapprovisionnement — DATE: 7-8-78

Diagramme n° : 1: Phase Enreg. commande client et préfacturation



DD/5 | DIAGRAMME D'ENCHAINEMENT DES FONCTIONS

PHASE: Enregistremnent commande client et préfacturation

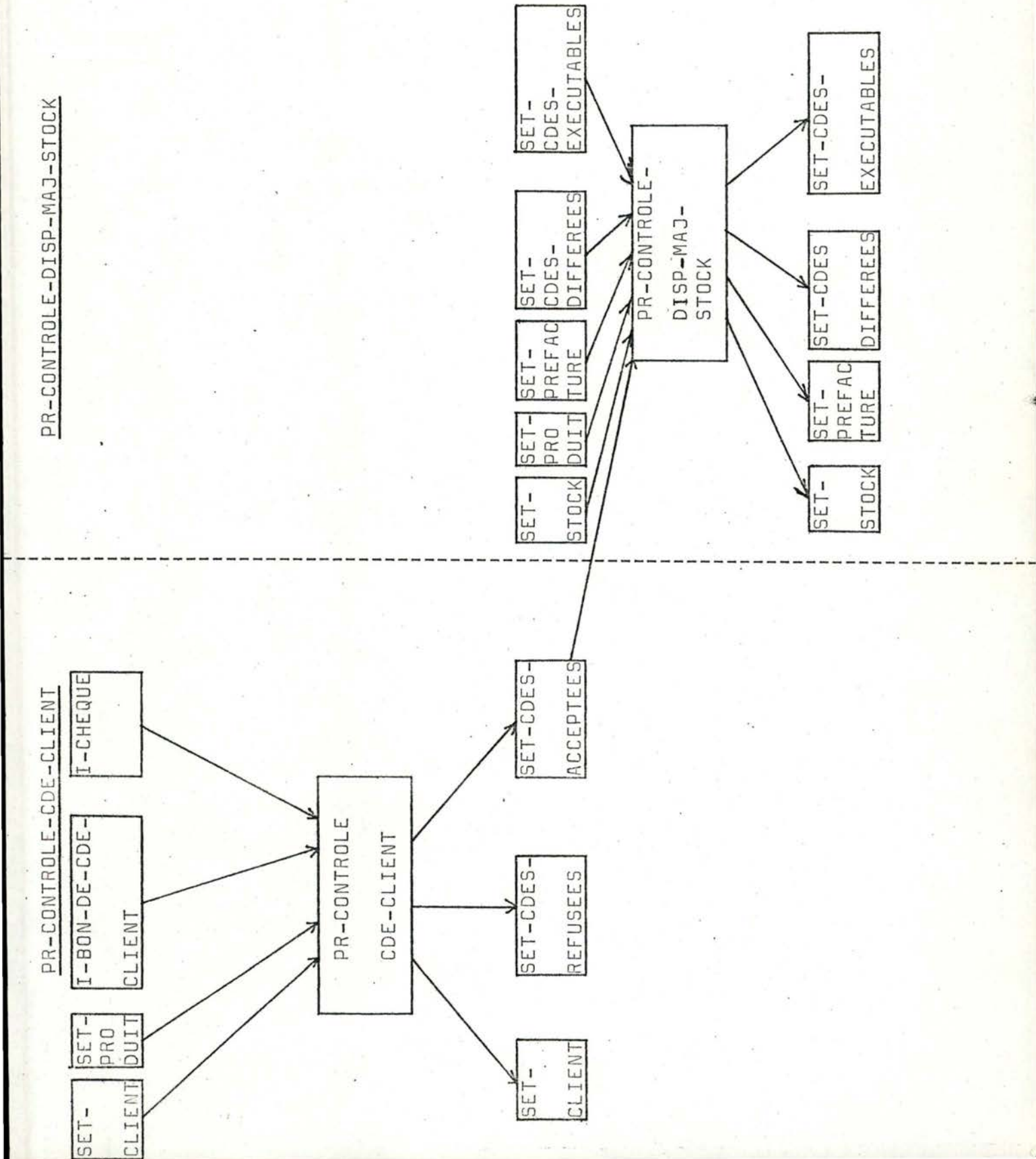
Version: 1 Folio: 1

Sous-système: Service commercial

Application: Facturation et réapprovisionnement

Date: 7-8-78

Auteur: Durigneux



DD/5 | DIAGRAMME D'ENCHAINEMENT DES FONCTIONS

PHASE: Enregistrement cde client et préfactuation

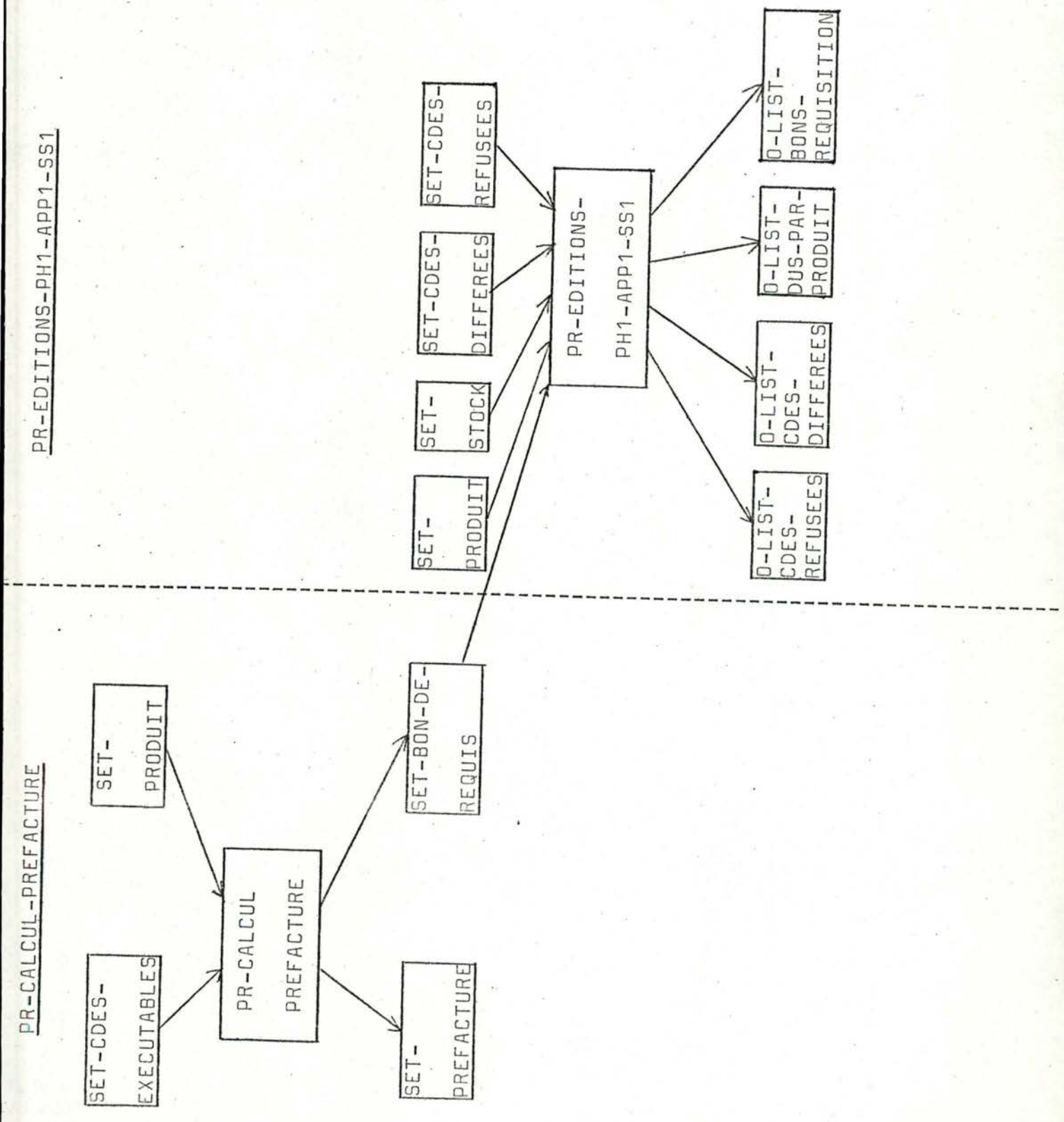
Version: 1 Folio: 2

Sous-système: Service commercial

Application: Facturation et réapprovisionnement

Date: 7-8-78

Auteur: Durigneux



PHASE

DD/2 DESCRIPTION DE PROCESSUS: REGLES DE TRAITEMENT

Phase Enreg. cdes clie
PROCESSUS préfacturatio

Version: 1 Folio:

Sous-système: Service commercial Date: 8-8-78

Application: Facturation et réapprovisionnemen Auteur: Durigneux

Fonction n°	CONDITIONS / ACTIONS
1.	Contrôle commande client: PR-CONTROLE-CDE-CLIENT.

Table N° 1

Le client existe dans le fichier client	Y				N
L'adresse du client est inchangée	Y			N	
Le code solvabilité est positif	Y		N		
Le chèque est valide	Y	N			
Aller à la table N° 2.	X				
Editer le message M20		X			
Editer le message M5			X		
Déclencher la phase de m-à-j client				X	X

Table N° 2

Dernière ligne d'une commande	Y			N		
La référence produit existe	Y		N	Y		N
La quantité commandée existe	Y	N		Y	N	
Imprimer la ligne incorrecte Positionner l'indicatif cde refusée		X	X		X	X
Aller à la table N° 3		X				
Aller à la table N° 2					X	

Table N° 3

Indicatif commande refusée positionné	Y	N
Imprimer le message M2 Positionner indicatif cde acceptée	X	
Former l'enregistrement cde refusée	X	
Former l'enregistremnet cde acceptée		X
Effectuer le contrôle de disponibilité à la fonction N° 2		X

PHASE

DD/3 | | DESCRIPTION DE PROCESSUS — Phase: Enreg. cdes client préfacturation

REGLES DE CONTROLE — PROCESSUS

Version: 1 Folio:

Sous-système: Service commercial Date: 8-8-78

Application: Facturation réapprovisionnement Auteur: Durigneux

	CONDITION DE VALIDITE	SEVER.	MESSAGE D'ERREUR
	Le client n'existe pas dans le fichier	2	Déclenchement de la phase de m-à-j du fichier client
	Adresse du client ≠te	2	Déclenchement de la phase de m-à-j du fichier client
	Code solvabilité du client -	4	Message M5
	Chèque invalide	4	Message M20
	Référence produit incorrecte	4	Message M2
	Quantité incorrecte	4	Message M2

SEVERITE: 1 = suspicion d'erreur, accepté
2 = erreur corrigée d'office, accepté
3 = erreur formelle, accepté
4 = erreur fatale, rejeté

DD/9 DESCRIPTION D'UNE UNITE D'INFORMATION

UNITE: CLIENT
Version: 1 Folio:

Système: Entreprise de vente par correspond. Date: 8-8-78
Sous-système: Service commercial
Application: Facturation et réapprovisionnement Auteur: Durigneux

LIBELLE: Client de l'entreprise par correspondance

niv	RUBRIQUE			CODIF.	FORMAT	LONGUEUR		DIMENSION		REM.
	mnémon.	*	libellé		type	moy.	max.	moy.	max.	
01	ENT-CLIENT				ENTITE					
02	GR-DESIGN-CLIENT				Groupe					
03	EL-NO-CLIENT			DD8/1	Numéric	3	6			
03	EL-NOM-CLIENT				Alphanum	20	40			
03	EL-ADRESSE				Alphanum	35	70			
02	EL-INDIC-SOLVABILITE				Numéric	1	1			
02	EL-ANC-AVOIR				Numéric	6	12			
02	EL-NOUV-AVOIR				Numéric	6	12			

UNITE:

DD/10

ANALYSE DES VOLUMES Version: Folio:

Sous-système: Date:
Application: Auteur:

RAPPEL: LONGUEUR UNITE:	Moy.	Max.	

PERIODE	OCCURRENCES	VOLUME	REM.
	Moy.		
	Max.		
	Moy.		
	Max		
	Moy.		
	Max.		
	Moy.		
	Max		
	Moy.		
	Max.		

ORIGINE:
PEREMPTION:
ARCHIVAGE:

Comme nous ne disposons pas des informations nécessaires à la rédaction de ce document, nous nous contentons de vous en donner un exemple.

/11 ANALYSE D'ACTIVITE

APPLICATION: Facturation et réapprovisionnement

Version: 1 Folio:

-système: Service commercial

Date: 8-8-78

Auteur: Durigneux

ITES Classe	PHASES				
CLIENT	Phase 1			Phase 4	
C	Contrôle code client				
A					
M	M-à-j de l'avoir			M-à-j de l'avoir	
S					
D					
C					
A					
M					
S					
D					
C					
A					
M					
S					
D					
C					
A					
M					
S					
D					
C					
A					
M					
S					
D					

nsultations A = adjonctions M = modifications

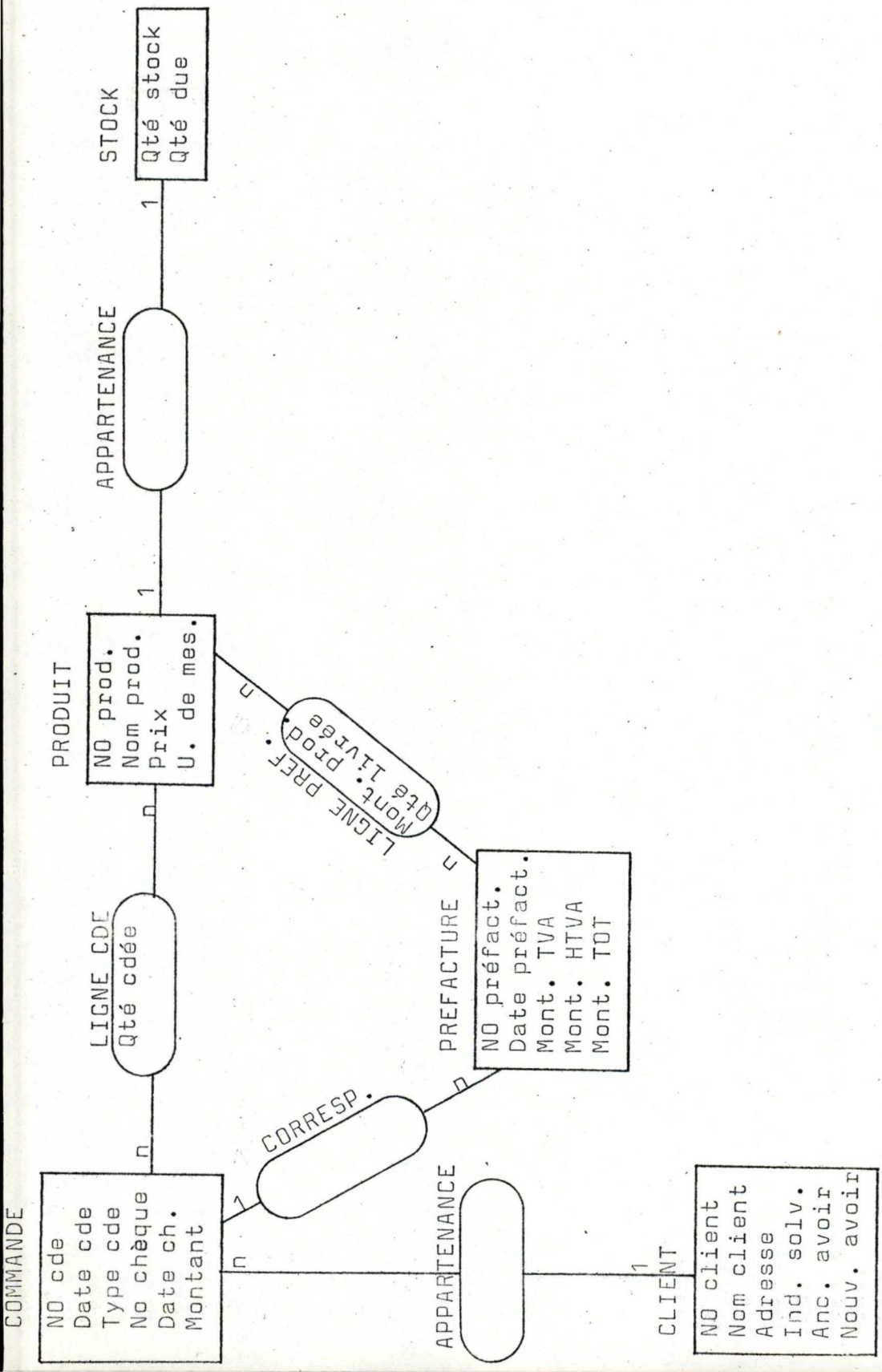
ppressions D = délai de réponse

12 DIAGRAMME DES RELATIONS ENTRE LES UNITES D'INFORMATION

APPLIC.: Facturation et réapprovisionnement
PHASE: Enreg. cdes client et préfacturation
Version: 1 Folio:

-système: Service commercial
cation: Facturation réapprovisionnement

Date: 8-8-78
Auteur: Durigneux



3. RESULTATS DE LA METHODE D'ANALYSE ASSISTEE PAR ISDOS.

3. 1 Modifications de rapports.

Dans cette partie des annexes, nous allons vous présenter les routines fortran que nous avons dû écrire pour modifier les rapports: CONTENTS et RELATION-STRUCTURE.

31. 1 CONTENTS-report.

Pour modifier ce rapport, nous avons créé deux routines:

* INITIM
* PRPIC

Afin de comprendre ce qui va suivre, il est utile de connaître le type de connection de la "relation" ATTRIBUTES ARE...
Vous trouverez ce schéma à la figure 1.

Routine INITIM.

Cette routine permet d'initialiser le rapport utilisé avec le paramètre DPIC. En clair, ce paramètre signifie "display picture".

Le lecteur pourra trouver le listing de compilation de la routine à la suite de ce texte.

Cette routine est en fait une fonction logique dont le résultat déterminera si le rapport se déroulera ou nom avec les modifications:

* si INITIM= .TRUE. : le rapport se déroulera avec la version normale de son code.

* si INITIM= .FALSE. : le rapport se déroulera selon la version modifiée du code.

L'initialisation du rapport consiste à placer la valeur de la clé de la base de données correspondant à chacun des Attribute-names:

LGMOY		KLMOY
LGMAX		KLMAX
DIMOY	dans les variables:	KDMOY
DIMAX		KDMAX
IMAGE		KIMAGE

Si les Attributes-names: DIMOY et/ou DIMAX n'existent pas dans la base de données, cela n'a aucune influence sur la valeur finale de la fonction.

Par contre si l'un des autres Attribute-name n'existe pas dans la base de données, la fonction INITIM prendra pour valeur: .TRUE. ,ce qui aura pour effet de remettre le switch PICSW,qui se trouve dans le common CONTIN, à la valeur .FALSE.

Dans tous les cas de non-existence d'un Attribute-name, l'utilisateur recevra un message d'erreur lui indiquant que l'attribut ne se trouve pas dans la base de données.

ROUTINE PRPIC.

Cette routine imprime les Attributes-names et leurs valeurs pour certains composants des objets ISDOS traités dans le CONTENTS report:

* GROUP:
  Ce type d'objet ne pourra avoir que les valeurs correspondant aux attributs DIMAX et DIMOY.

* ELEMENT:
  Ce type d'objet peut avoir des valeurs pour chaque attribut: IMAGE, LGMOY, LGMAX, DIMOY, DIMAX.

* UNDEFINED:
  Ce type d'objet pourra être facilement classifiable, si l'utilisateur a pris soin de préfixer tout objet de la base de données.

Nous limiterons cette description à ces quelques explications pour ne pas devoir entrer trop profondément dans la structure modulaire des rapports d'ISDOS.

Extrait de: ISDOS Working Paper NO. 166

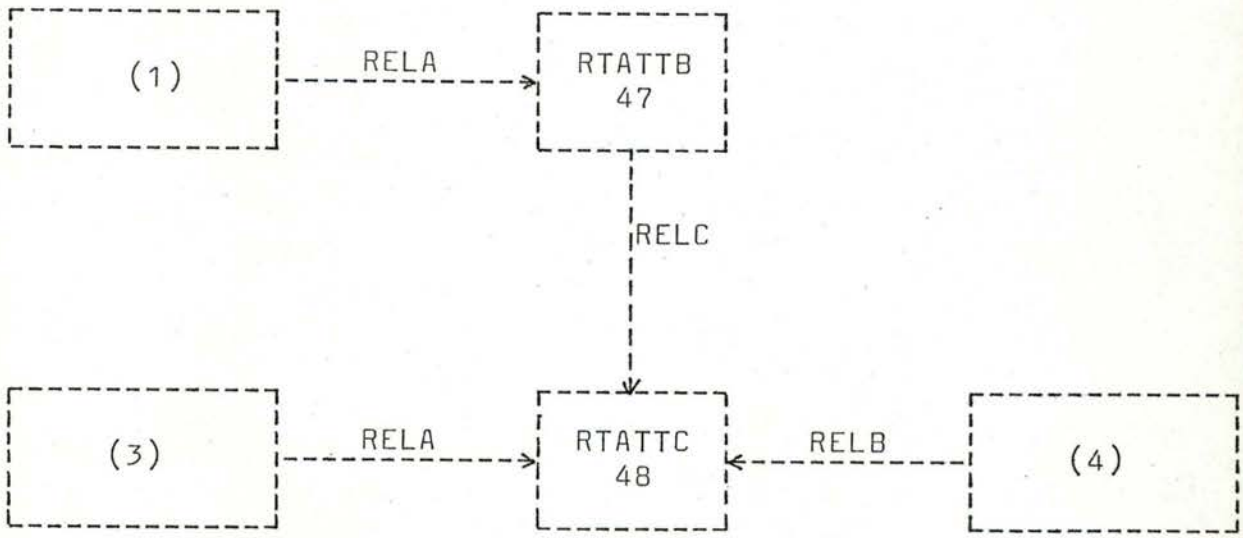
Connection type: T2S

Object Types

1	Any except (ATTRIBUTE, ATTRIBUTE-VALUE)
3	ATTRIBUTE
4	ATTRIBUTE-VALUE, NUMBER

Statements:

1 ATTRIBUTES ARE (3 4)...



- Figure 1 -

```
      LOGICAL FUNCTION INITIM(LOWEST)                                  24. -
C
C INITIALISATION FOR CONTENTS IF PICSW=.TRUE.
C
      COMMON/BLANKS/BLANK
      INTEGER BLANK(30)
C
C CONTAINS BLANKS FOR VARIOUS PURPOSES
C  INITIALIZED BY BLOCK DATA
C
      COMMON/CONTIN/NAMIND,MPLEV,NUMONE,LEVELS,T,NCFLAG,LOWEST,LINSW,
     &  LEVSW,NAMSW,PICSW
      INTEGER NAMIND,MPLEV,NUMONE,LEVELS,T
      LOGICAL NCFLAG,LOWEST,LINSW,LEVSW,NAMSW,PICSW
C
C FOR CONTENT REPORT
C
C NAMIND - NUMBER OF SPACES TO IDNDENT EACH NAME AS GO DOWN LEVELS
C MPLEV  - MAX NUMBER OF SPACES TO INDENT
C NCFLAG - WHETHER TO GIVE COMMENT IF UNDEF/GROUP HAS NO CONTENTS
C NUMONE - NUMBER OF LEVELS ONE THINGS HANDLED
C LEVELS - MAX NUMBER OF LEVELS TO GO DOWN (0=> ALL; -1=> LOWEST)
C LOWEST - WHETHER OR NOT TO GIVE ONLY LOWEST STUCTURE
C T      - HOLDS THE NAME TYPE OF THE CURRENT NAME
C LINSW  - WHETHER OR NOT TO PRINT LINE NUMBERS
C LEVSW  - WHETHER OR  NOT TO PRINT LEVEL NUMBERS
C NAMSW  - WHETHER OR NOT TO PRINT NAME TYPES
C PICSW  - WHETHER OR NOT TO PRINT SOME ATTIBUTE VALUES
C
      COMMON/NMCODE/ NMUNDF,NMATTR,NMATTV,NMCOND,NMELMT,NMENTY,
     & NMEVNT,NMGRUP,NMINPP,NMINTV,NMKWRD,NMMBOX,NMMEMO,NMOUTP,
     & NMPRDF,NMPROC,NMRWEY,NMRELN,NMSCTY,NMSETT,NMSRCE,NMSSCN,
     & NMSYSP,NMNUMB
      INTEGER      NMUNDF,NMATTR,NMATTV,NMCOND,NMELMT,NMENTY,
     & NMEVNT,NMGRUP,NMINPP,NMINTV,NMKWRD,NMMBOX,NMMEMO,NMOUTP,
     & NMPRDF,NMPROC,NMRWEY,NMRELN,NMSCTY,NMSETT,NMSRCE,NMSSCN,
     & NMSYSP,NMNUMB
C
C CONTAINS CODES FOR NAME TYPES - INITIALIZED BY BLOCK DATA
C
C       0 NMUNDF  UNDEFINED
C       1 NMATTR  ATTRIBUTE
C       2 NMATTV  ATTRIBUTE VALUE
C       3 NMCOND  CONDITION
C       4 NMELMT  ELEMENT
C       5 NMENTY  ENTITY
C       6 NMEVNT  EVENT
C       7 NMGRUP  GROUP
C       8 NMINPP  INPUT
C       9 NMINTV  INTERVAL
```

```
C       10 NMKWRD  KEYWORD
C       11 NMMBOX  MAILBOX
C       12 NMMEMO  MEMO
C       13 NMOUTP  OUTPUT
C       14 NMPRDF  PROBLEM DEFINER
C       15 NMPROC  PROCESS
C       16 NMRWEY  REAL WORLD ENTITY
C       17 NMRELN  RELATION
C       18 NMSCTY  SECURITY
C       19 NMSETT  SET
C       20 NMSRCE  SOURCE
C       21 NMSSCN  SUBSETTING CRITERION
C       22 NMSYSP  SYSTEM PARAMETER
C       23 NMNUMB  NUMBER
C
C       NOTE: ADD 100 IF SYNONYM FOR SOMETHING
C
        COMMON/PICTAB/DIMAX,DIMOY,IMAGE,LGMAX,LGMOY,
       & KDMAX,KDMOY,KIMAGE,KLMAX,KLMOY
        INTEGER DIMAX(2),DIMOY(2),IMAGE(2),LGMAX(2),LGMOY(2)
        INTEGER KDMAX,KDMOY,KIMAGE,KLMAX,KLMOY
C
C ATTRIBUTES NAMES FOR DESCRIPTION OF GROUP AND/OR ELEMENT
C
C DIMAX - DIMENSION MAXIMUM FOR AN ELEMENT ODER A GROUP
C DIMOY - MEAN DIMENSION FOR AN ELEMENT ODER A GROUP
C IMAGE - PICTURE FOR AN ELEMENT
C LGMAX - MEAN LENGTH FOR AN ELEMENT
C LGMOY - MAXIMUM LENGTH FOR AN ELEMENT
C
C KDMAX - VALUE OF D.B. KEY FOR ATTRIBUTE DIMAX
C KDMOY - VALUE OF D.B. KEY FOR ATTRIBUTE DIMOY
C KIMAGE - VALUE OF D.B. KEY FOR ATTRIBUTE IMAGE
C KLMAX - VALUE OF D.B. KEY FOR ATTRIBUTE DIMAX
C KLMOY - VALUE OF D.B. KEY FOR ATTRIBUTE DIMOY
C
        COMMON/REPORT/WIDTH,MLIN,HEADSW,PAGENO,LIN,HSKIP,CSKIP,
       & LPCOL,HEADP,VER,TIM,DAT,PNAM,
       & HEAD,COL,BUF
        INTEGER VER(2),TIM(2),DAT(3),PNAM(8)
        INTEGER HEAD(30),COL(30),BUF(30)
        INTEGER WIDTH,MLEN,PAGENO,LIN,HSKIP,CSKIP
        LOGICAL HEADSW,LPCOL,HEADP
C
C VARIABLES FOR REPORT SYSTEM
C
C WIDTH  - OUTPUT WIDTH IN CHARACTERS
C MLIN   - MAX NUMBER OF LINES ON A PAGE
C HEADSW - WHETHER TO PRINT HEADING AT ALL
C PAGENO - PAGE NUMBER
```

```
101 C LIN     - CURRENT LINE NUMBER
102 C HSKIP   - NUMBER OF LINES TO SKIP AFTER HEADING
103 C CSKIP   - NUMBER OF LINES TO SKIP AFTER COLUMN HEADING
104 C LPCOL   - WHETHER OR NOT TO PRINT COLUMN HEADING AT TOP OF PAGE
105 C HEADP   - WHETHER OR NOT USER HAS GIVEN HEADING
106 C VER     - 8 CHARACTER VERSION NUMBER
107 C TIM     - TIME AT START OF REPORT HH:MM.SS
108 C DAT     - TODAYS DATE - MMM DD, 19YY
109 C PNAM    - BUFFER FOR PROBLEM NAME
110 C HEAD    - BUFFER FOR PAGE HEADING
111 C COL     - COLUMN HEADING BUFFER
112 C BUF     - LINE BUFFER
113 C
114       INTEGER KEY,IPIC,SW
115       LOGICAL ERRSW
116       INTEGER NAM(3)
117 C^^^^^^^^^^^^^^^^
118       ERRSW = .FALSE.
119       IPIC = 1
120       CALL SMOVE(NAM,1,DIMAX,1,5)
121 C
122 C IS NAM IN D.B. AND IS NAM AN ATTRIBUTE?
123 C
124     5 CONTINUE
125       CALL SMOVE(NAM,6,BLANK,1,25)
126       CALL GKB(NAM,KEY,T,SW)
127       IF (SW .GT. 0) GOTO (150,150,100,100,100),IPIC
128    10 CONTINUE
129       IF (T .EQ. NMATTR) GOTO (210,220,230,240,250),IPIC
130       GOTO (200,200,180,180,180),IPIC
131 C
132 C NAM IS NOT IN D.B.
133 C
134   100 CONTINUE
135       ERRSW = .TRUE.
136   150 CONTINUE
137       CALL HYUSRN(426,30,NAM)
138       GOTO 10
139 C
140 C NAM NOT AN ATTRIBUTE
141 C
142   180 ERRSW = .TRUE.
143   200 CONTINUE
144       CALL HYUSRN(427,30,NAM)
145       GOTO (210,220,230,240,250),IPIC
146 C
147 C SAVE KEY IN ATTRIBUTE-KEY VARIABLE
148 C
149   210 CONTINUE
150       KDMAX = KEY
```

```
      IPIC = 2
      CALL SMOVE(NAM,1,DIMOY,1,5)
      GOTO 5
  220 CONTINUE
      KDMOY = KEY
      IPIC = 3
      CALL SMOVE(NAM,1,LGMAX,1,5)
      GOTO 5
  230 CONTINUE
      KLMAX = KEY
      IPIC = 4
      CALL SMOVE(NAM,1,LGMOY,1,5)
      GOTO 5
  240 CONTINUE
      KLMOY = KEY
      IPIC = 5
      CALL SMOVE(NAM,1,IMAGE,1,5)
      GOTO 5
  250 CONTINUE
      KIMAGE = KEY
C
C INITIALISATION OF COL HEADING IF ERRSW=.FALSE.
C
  500 CONTINUE
      IF (ERRSW) GOTO 850
      LOWEST = .FALSE.
      CALL COLBLD(54,1,5,'IMAGE')
      CALL COLBLD(67,1,5,'LGMOY')
      CALL COLBLD(80,1,5,'LGMAX')
      CALL COLBLD(93,1,5,'DIMOY')
      CALL COLBLD(106,1,5,'DIMAX')
C
C PRINT COL HEADING
C
      CALL PCOL(2,2)
      LPCOL = .TRUE.
  850 CONTINUE
      INITIM = ERRSW
C
C ALWAYS DONE
C
  900 RETURN
      END
```

```
      SUBROUTINE PRPIC(T,PICKEY,PICNUM)
C
C PRINT ATTRIBUTES AND VALUES FOR CONTENTS IF PICSW=.TRUE.
      INTEGER PICKEY,PICNUM
C
C PICKEY - HOLDS THE D.B. KEY OF THE CURRENT NAME
C PICNUM - LEVEL OF ITEM TO BE PRINTED
C
      COMMON/BLANKS/BLANK
      INTEGER BLANK(30)
C
C CONTAINS BLANKS FOR VARIOUS PURPOSES
C  INITIALIZED BY BLOCK DATA
C
      COMMON/CONTIN/NAMIND,MPLEV,NUMONE,LEVELS,T,NCFLAG,LOWEST,LINSW,
     &  LEVSW,NAMSW,PICSW
      INTEGER NAMIND,MPLEV,NUMONE,LEVELS,T
      LOGICAL NCFLAG,LOWEST,LINSW,LEVSW,NAMSW,PICSW
C
C FOR CONTENT REPORT
C
C NAMIND - NUMBER OF SPACES TO IDNDENT EACH NAME AS GO DOWN LEVELS
C MPLEV  - MAX NUMBER OF SPACES TO INDENT
C NCFLAG - WHETHER TO GIVE COMMENT IF UNDEF/GROUP HAS NO CONTENTS
C NUMONE - NUMBER OF LEVELS ONE THINGS HANDLED
C LEVELS - MAX NUMBER OF LEVELS TO GO DOWN (0=> ALL; -1=> LOWEST)
C LOWEST - WHETHER OR NOT TO GIVE ONLY LOWEST STUCTURE
C T      - HOLDS THE NAME TYPE OF THE CURRENT NAME
C LINSW  - WHETHER OR NOT TO PRINT LINE NUMBERS
C LEVSW  - WHETHER OR  NOT TO PRINT LEVEL NUMBERS
C NAMSW  - WHETHER OR NOT TO PRINT NAME TYPES
C PICSW  - WHETHER OR NOT TO PRINT SOME ATTIBUTE VALUES
C
      COMMON/NAME/NAMEL,MNAMEL,NAME,NAME1
      INTEGER NAME(8),NAME1(8)
      INTEGER NAMEL,MNAMEL
C
C HANDY PLACE FOR STORING A USER-DEFINED NAME
C   INITIALIZED BY BLOCK DATA
C
C NAMEL  - LENGTH OF NAME IN CHARACTERS
C MNAMEL - MAXIMUM NAME LENGTH
C NAME   - CHARATER FORM OF NAME, MUST BE DIMENSIONED
C NAME1  - CHARACTER FOR OF A NAME, MUST BE DIMENSIONED
C***** LARGE ENOUGH TO HOLD MAXIMUM SIZE NAME
C
      COMMON/NMCODE/ NMUNDF,NMATTR,NMATTV,NMCOND,NMELMT,NMENTY,
     & NMEVNT,NMGRUP,NMINPP,NMINTV,NMKWRD,NMMBOX,NMMEMO,NMOUTP,
     & NMPRDF,NMPROC,NMRWEY,NMRELN,NMSCTY,NMSETT,NMSRCE,NMSSCN,
     & NMSYSP,NMNUMB
```

```
      INTEGER       NMUNDF,NMATTR,NMATTV,NMCOND,NMELMT,NMENTY,
     & NMEVNT,NMGRUP,NMINPP,NMINTV,NMKWRD,NMMBOX,NMMEMO,NMOUTP,
     & NMPRDF,NMPROC,NMRWEY,NMRELN,NMSCTY,NMSETT,NMSRCE,NMSSCN,
     & NMSYSP,NMNUMB
C
C CONTAINS CODES FOR NAME TYPES - INITIALIZED BY BLOCK DATA
C
C       0 NMUNDF  UNDEFINED
C       1 NMATTR  ATTRIBUTE
C       2 NMATTV  ATTRIBUTE VALUE
C       3 NMCOND  CONDITION
C       4 NMELMT  ELEMENT
C       5 NMENTY  ENTITY
C       6 NMEVNT  EVENT
C       7 NMGRUP  GROUP
C       8 NMINPP  INPUT
C       9 NMINTV  INTERVAL
C      10 NMKWRD  KEYWORD
C      11 NMMBOX  MAILBOX
C      12 NMMEMO  MEMO
C      13 NMOUTP  OUTPUT
C      14 NMPRDF  PROBLEM DEFINER
C      15 NMPROC  PROCESS
C      16 NMRWEY  REAL WORLD ENTITY
C      17 NMRELN  RELATION
C      18 NMSCTY  SECURITY
C      19 NMSETT  SET
C      20 NMSRCE  SOURCE
C      21 NMSSCN  SUBSETTING CRITERION
C      22 NMSYSP  SYSTEM PARAMETER
C      23 NMNUMB  NUMBER
C
C      NOTE: ADD 100 IF SYNONYM FOR SOMETHING
C
       COMMON/PICTAB/DIMAX,DIMOY,IMAGE,LGMAX,LGMOY,
     & KDMAX,KDMOY,KIMAGE,KLMAX,KLMOY
       INTEGER DIMAX(2),DIMOY(2),IMAGE(2),LGMAX(2),LGMOY(2)
       INTEGER KDMAX,KDMOY,KIMAGE,KLMAX,KLMOY
C
C ATTRIBUTES NAMES FOR DESCRIPTION OF GROUP AND/OR ELEMENT
C
C DIMAX - DIMENSION MAXIMUM FOR AN ELEMENT ODER A GROUP
C DIMOY - MEAN DIMENSION FOR AN ELEMENT ODER A GROUP
C IMAGE - PICTURE FOR AN ELEMENT
C LGMAX - MEAN LENGTH FOR AN ELEMENT
C LGMOY - MAXIMUM LENGTH FOR AN ELEMENT
C
C KDMAX - VALUE OF D.B. KEY FOR ATTRIBUTE DIMAX
C KDMOY - VALUE OF D.B. KEY FOR ATTRIBUTE DIMOY
C KIMAGE - VALUE OF D.B. KEY FOR ATTRIBUTE IMAGE
```

```
C KLMAX - VALUE OF D.B. KEY FOR ATTRIBUTE DIMAX
C KLMOY - VALUE OF D.B. KEY FOR ATTRIBUTE DIMOY
C
      COMMON/RECRDS/NAMREC,SYNREC,NUBA,NUBB,NUBC,
     & COM,COMLIN,
     & NAMET,NAMEC,RELTYP,LINEC,NUMREC,NUMVAL,
     & AREC,NMITEM,DBKEYA,DBKEYB,CHITEM,NAMEDC,
     & UPDREC,UPDSEQ,UPDCMD,UPDDAT,UPDTIM
      INTEGER NAMREC(2),SYNREC(2),NUBA(2),NUBB(2),NUBC(2)
      INTEGER COM(2),COMLIN(2)
      INTEGER NAMET(2),NAMEC(2),RELTYP(2),LINEC(2),NUMREC(2),NUMVAL(2)
      INTEGER AREC(2),NMITEM(2),DBKEYA(2),DBKEYB(2),CHITEM(2),NAMEDC(2
      INTEGER UPDREC(2),UPDSEQ(2),UPDCMD(2),UPDDAT(2),UPDTIM(2)
C
C CHARACTER FORM OF RECORD NAMES
C  INITIALIZED BY BLOCK DATA
C
      COMMON/REPORT/WIDTH,MLIN,HEADSW,PAGENO,LIN,HSKIP,CSKIP,
     & LPCOL,HEADP,VER,TIM,DAT,PNAM,
     & HEAD,COL,BUF
      INTEGER VER(2),TIM(2),DAT(3),PNAM(8)
      INTEGER HEAD(30),COL(30),BUF(30)
      INTEGER WIDTH,MLEN,PAGENO,LIN,HSKIP,CSKIP
      LOGICAL HEADSW,LPCOL,HEADP
C
C VARIABLES FOR REPORT SYSTEM
C
C WIDTH  - OUTPUT WIDTH IN CHARACTERS
C MLIN   - MAX NUMBER OF LINES ON A PAGE
C HEADSW - WHETHER TO PRINT HEADING AT ALL
C PAGENO - PAGE NUMBER
C LIN    - CURRENT LINE NUMBER
C HSKIP  - NUMBER OF LINES TO SKIP AFTER HEADING
C CSKIP  - NUMBER OF LINES TO SKIP AFTER COLUMN HEADING
C LPCOL  - WHETHER OR NOT TO PRINT COLUMN HEADING AT TOP OF PAGE
C HEADP  - WHETHER OR NOT USER HAS GIVEN HEADING
C VER    - 8 CHARACTER VERSION NUMBER
C TIM    - TIME AT START OF REPORT HH:MM.SS
C DAT    - TODAYS DATE - MMM DD, 19YY
C PNAM   - BUFFER FOR PROBLEM NAME
C HEAD   - BUFFER FOR PAGE HEADING
C COL    - COLUMN HEADING BUFFER
C BUF    - LINE BUFFER
C
      COMMON/RTCODE/ RTASSD,RTBECT,RTBECF,RTRELB,RTRELC,RTCARD,RTCNNB,
     & RTCNNC,RTCONB,RTCONC,RTDRVB,RTDRVC,RTSUBS,RTGENS,RTHAPB,RTHAPC,
     & RTIRWE,RTIDNT,RTINCP,RTKWRD,RTMAIL,RTMAIN,RTSUBP,RTRCVS,RTRESF,
     & RTSCTY,RTMEMO,RTSRCE,RTSSCN,RTTERM,RTTRIG,RTUPDB,RTUPDC,RTUSES,
     & RTVALB,RTVALS,RTVALR,RTUTIL,RTATTB,RTATTC,RTSCNB,RTSCNC
      INTEGER        RTASSD,RTBECT,RTBECF,RTRELB,RTRELC,RTCARD,RTCNNB,
```

```
151       & RTCNNC,RTCONB,RTCONC,RTDRVB,RTDRVC,RTSUBS,RTGENS,RTHAPB,RTHAPC,
152       & RTIRWE,RTIDNT,RTINCP,RTKWRD,RTMAIL,RTMAIN,RTSUBP,RTRCVS,RTRESP,
153       & RTSCTY,RTMEMO,RTSRCE,RTSSCN,RTTERM,RTTRIG,RTUPDB,RTUPDC,RTUSES,
154       & RTVALB,RTVALS,RTVALR,RTUTIL,RTATTB,RTATTC,RTSCNB,RTSCNC
155 C
156 C RELATION TYPES FOR NON COMMENT ENTRIES
157 C
158 C   INITIALIZED BY BLOCK DATA
159 C
160       COMMON/SETS/ALLNAM,RELA,RELB,RELC,ALINE,
161       & SYNFOR,ALLUPD
162       INTEGER ALLNAM(2),RELA(2),RELB(2),RELC(2),ALINE(2)
163       INTEGER SYNFOR(2),ALLUPD(2)
164 C
165 C CHARACTER FORM OF SET NAMES
166 C  INITIALIZED BY BLOCK DATA
167 C
168       INTEGER IERR,KEY,IPIC,KV,BPTR,NML
169 C.....................................
170 C
171 C PUT LEVEL NUMBER IN BUF
172 C
173       IF (LEVSW) CALL NINBUF(PICNUM,17,5)
174 C
175 C GET NAME OF CURRENT OBJECT
176 C
177       CALL GFK(NAMEC,PICKEY,NAME,IERR)
178 C
179 C PUT OBJECT NAME IN BUF
180 C
181       CALL BUFBLD(23,1,30,NAME)
182 C
183 C IS NAME  NO GROUP AND NO ELEMENT?
184 C
185       IF (T .NE. NMGRUP .AND. T .NE. NMELMT .AND. T .NE. NMUNDF) GOTO 90
186 C
187 C DO WE HAVE THE CURRENT NAME AN ATTRIBUTE?
188 C
189       CALL SMOVE(NAME,1,BLANK,1,30)
190       CALL SKFM(RELA,PICKEY,RTATTB,IERR)
191       IF (IERR .LT. 0) GOTO 900
192 C
193 C FIND NUBC FOR CURRENT NAME
194 C
195    10 CONTINUE
196       CALL SOM(RELC,RELA,IERR)
197    20 CONTINUE
198       CALL SMM(RELA,RELC,IERR)
199       CALL GKO(RELA,KEY,IERR)
200       IF (T .EQ. NMGRUP) GOTO 60
```

```
C
C DO WE HAVE GOOD ATTRIBUTES FOR AN ELEMENT?
C
   30 CONTINUE
      IF (KEY .NE. KIMAGE) GOTO 40
      IPIC = 1
      GOTO 100
   40 CONTINUE
      IF (KEY .NE. KLMOY) GOTO 50
      IPIC = 2
      GOTO 100
   50 CONTINUE
      IF (KEY .NE.KLMAX) GOTO 60
      IPIC = 3
      GOTO 100
   60 CONTINUE
      IF (KEY .NE. DIMOY) GOTO 70
      IPIC = 4
      GOTO 100
   70 CONTINUE
      IF (KEY .NE. DIMAX) GOTO 80
      IPIC = 5
      GOTO 100
C
C SET RELA TO NEXT ATTRIBUTE
C
   80 CONTINUE
      CALL FNM(RELC,IERR)
      IF (IERR .LT. 0) GOTO 900
      GOTO 20
C
C CORRECT ATTRIBUTE FOUND... GET ATTR VALUE
C
  100 CONTINUE
      CALL SMM(RELB,RELC,IERR)
      CALL GFO(NAMEC,RELB,NAME,IERR)
      IF (IERR .LT.0) GOTO 80
      NML = NAMLEN(NAME)
      GOTO (210,220,230,240,250),IPIC
C
C PUT ATTRIBUTE VALUE IN BUF
C
  210 CONTINUE
      BPTR = 54
      GOTO 490
  220 CONTINUE
      BPTR = 67
      GOTO 490
  230 CONTINUE
      BPTR = 80
```

```
      GOTO 490
  240 CONTINUE
      BPTR = 93
      GOTO 490
  250 CONTINUE
      BPTR = 106
C
C CHECK IF ERROR LENGTHS IN VALUE
C
  490 CONTINUE
      IF (NML .LE. 11) GOTO 500
      CALL BUFBLD(BPTR,1,12,'ERROR LENGTH')
      GOTO 80
C
C PUT ATTRIBUTE VALUE IN BUF
C
  500 CALL BUFBLD(BPTR,1,NML,NAME)
      GOTO 80
C
C ALLWAYS DONE
C
  900 CONTINUE
      RETURN
      END
```

31. 2 RELATION-STRUCTURE report.

Nous avons plus particulièrement modifié une partie de ce rapport que l'on appelle RELATION-SUMMARY.

Nous avons remarqué en réalité que cette partie du rapport était mentionnées dans la documentation, mais n'était pas implémentée.

Nous avons donc écrit la routine SUMRUT qui en plus des résultats décrits dans la documentation offre également les données associées aux relations entre les entités.

Vous trouverez le schéma du type de connection de la "relation" ASSOCIATED-DATA à la figure 2.

La routine va pour chaque clé de la base de données correspondant à une relation aller chercher dans la matrice RMAT les valeurs des clés de la base de données qui identifient l'entité droite et l'entité gauche de la relation.
Et, s'il existe des données associées à la relation, on les imprimera à la suite des noms d'entités avec le type qui les caractérise.

Extrait de: ISDOS Working Paper NO. 166

Connection Type: S1

Object Types:

1 RELATION
2 GROUP, ELEMENT

Statements:

1 ASSOCIATED-DATA IS 2 name-list

2 ASSOCIATED WITH 1 name-list

- Figure 2 -

```
      SUBROUTINE SUMRUT
C
C PRINT THE RELATION SUMMARY
C
      COMMON/COLKEY/NCOL,MCOL,COL(400)
      INTEGER NCOL,MCOL,COL
C
C STORAGE FOR KEYS OF COLUMNS OF MATRICES
C
C NCOL   - CURRENT NUMBER OF COLUMNS
C MCOL   - MAX NUMBER OF COLUMNS
C COL    - VECTOR OF D.B. KEYS OF CORRESPONDING COL
C
      COMMON/ROWKEY/NROW,MROW,ROW(400)
      INTEGER NROW,MROW,ROW
C
C STORAGE FOR KEYS OF ROWS OF MATRICES
C
C NROW   - CURRENT NUMBER OF ROWS
C MROW   - MAX NUMBER OF ROWS
C ROW    - VECTOR OF D.B. KEYS OF CORRESPONDING ROW
C
      COMMON/RSTRCM/ RMAT,EMAT,SMAT,XMAT1,XMAT2,
     &  LH,SHIFT,TAB,MPOSN,
     &  USEDSW(400)
       INTEGER RMAT,EMAT,SMAT,XMAT1,XMAT2,LH,SHIFT,TAB,MPOSN
       LOGICAL USEDSW
C
C GLOBAL VARIABLES FOR THE RELATION STRUCTURE REPORT
C
C RMAT   - SPARSE MATRIX INDEX FOR THE RELATION - ENTITY MATRIX
C EMAT   - SPARSE MATRIX INDEX FOR THE ENTITY INTERACTION MATRIX
C SMAT   - SPARSE MATRIX INDEX FOR THE SHORTEST PATH MATRIX
C XMAT1  - SPARSE MATRIX INDEX FOR SCRATCH MATRIX NUMBER ONE
C XMAT2  - SPARSE MATRIX INDEX FOR SCRATCH MATRIX NUMBER TWO
C LH     - LEFT HAND MARGIN FOR MOST OUTPUT
C SHIFT  - SPACES BETWEEN THE NUMBERS AND NAMES IN THE ENTITY STRUCTUR
C TAB    - SPACES TO TAB FOR EACH SUCCESSIVE LEVEL IN THE ENTITY STRUC
C MPOSN  - MAXIMUM STARTING POINT FOR NAMES IN THE ENTITY STRUCTURE
C USEDSW - SWITCHES TO INDICATE WHETHER OR NOT A NAME HAS BEEN
C          INCLUDED IN AN ENTITY STRUCTURE
C
      COMMON/RTCODE/ RTASSD,RTBECT,RTBECF,RTRELB,RTRELC,RTCARD,RTCNNB,
     & RTCNNC,RTCONB,RTCONC,RTDRVB,RTDRVC,RTSUBS,RTGENS,RTHAPB,RTHAPC,
     & RTIRWE,RTIDNT,RTINCP,RTKWRD,RTMAIL,RTMAIN,RTSUBP,RTRCVS,RTRESP,
     & RTSCTY,RTMEMO,RTSRCE,RTSSCN,RTTERM,RTTRIG,RTUPDB,RTUPDC,RTUSES,
     & RTVALB,RTVALS,RTVALR,RTUTIL,RTATTB,RTATTC,RTSCNB,RTSCNC
       INTEGER        RTASSD,RTBECT,RTBECF,RTRELB,RTRELC,RTCARD,RTCNNB,
     & RTCNNC,RTCONB,RTCONC,RTDRVB,RTDRVC,RTSUBS,RTGENS,RTHAPB,RTHAPC,
     & RTIRWE,RTIDNT,RTINCP,RTKWRD,RTMAIL,RTMAIN,RTSUBP,RTRCVS,RTRESP,
```

```
     & RTSCTY,RTMEMO,RTSRCE,RTSSCN,RTTERM,RTTRIG,RTUPDB,RTUPDC,RTUSES,
     & RTVALB,RTVALS,RTVALR,RTUTIL,RTATTB,RTATTC,RTSCNB,RTSCNC
C
C RELATION TYPES FOR NON COMMENT ENTRIES
C
C   INITIALIZED BY BLOCK DATA
C
      COMMON/SETS/ALLNAM,RELA,RELB,RELC,ALINE,
     & SYNFOR,ALLUPD
      INTEGER ALLNAM(2),RELA(2),RELB(2),RELC(2),ALINE(2)
      INTEGER SYNFOR(2),ALLUPD(2)
C
C CHARACTER FORM OF SET NAMES.
C  INITIALIZED BY BLOCK DATA
C
      COMMON/TYPCOM/PLEN,PTYPE
      INTEGER PTYPE(11)
      INTEGER PLEN
C
C STORAGE OF NAME TYPES FOR CONVERSION ROUTINES
C
C PLEN   - LENGTH RETURNED FROM TYPCNV OR TCNV
C PTYPE  - CHAR FORM OF TYPE (42 CHARACTERS)
C
      COMMON/RECRDS/NAMREC,SYNREC,NUBA,NUBB,NUBC,
     & COM,COMLIN,
     & NAMET,NAMEC,RELTYP,LINEC,NUMREC,NUMVAL,
     & AREC,NMITEM,DBKEYA,DBKEYB,CHITEM,NAMEDC,
     & UPDREC,UPDSEQ,UPDCMD,UPDDAT,UPDTIM
      INTEGER NAMREC(2),SYNREC(2),NUBA(2),NUBB(2),NUBC(2)
      INTEGER COM(2),COMLIN(2)
      INTEGER NAMET(2),NAMEC(2),RELTYP(2),LINEC(2),NUMREC(2),NUMVAL(2)
      INTEGER AREC(2),NMITEM(2),DBKEYA(2),DBKEYB(2),CHITEM(2),NAMEDC(2
      INTEGER UPDREC(2),UPDSEQ(2),UPDCMD(2),UPDDAT(2),UPDTIM(2)
C
C CHARACTER FORM OF RECORD NAMES
C  INITIALIZED BY BLOCK DATA
C
      INTEGER D,I,J,KLEFT,KRIGHT,Y,NUM,IRC,IERR,T
      INTEGER NAM(8)
C.....
      CALL BPAGE
      CALL BLBUF
C
C INITIALISATION OF BUF
C
      CALL BUFBLD(36,1,7,'BETWEEN')
      CALL BUFBLD(78,1,3,'AND')
C
C LOOP FOR SEARCH AND PRINT OF RELATION,ENTITIES AND ASS. DATA
```

```
C
      DO      150 I=1, NROW
              D = -1
C
C LOOP FOR SEARCH OF LEFT AND RIGHT ENTITIES FOR A RELATION
C
              DO      100 J=1, 2
C
C GET THE NEXT ELEMENT IN ROW I OF ARRAY NO RMAT
C
                      CALL SMRNXT(RMAT,I,Y,NUM,D,IRC)
                      IF (NUM .EQ. 1) KLEFT = COL(Y)
                      IF (NUM .EQ. 2) KRIGHT = COL(Y)
  100                 CONTINUE
C
C GET RELATION AND ENTITIES NAMES
C
              CALL GFK(NAMEC,ROW(I),NAM,IERR)
              CALL BUFBLD(6,1,30,NAM)
              CALL GFK(NAMEC,KLEFT,NAM,IERR)
              CALL BUFBLD(46,1,30,NAM)
              CALL GFK(NAMEC,KRIGHT,NAM,IERR)
              CALL BUFBLD(83,1,30,NAM)
C
C PUT BUF  WITH OBJECT NAMES
C
              CALL PBUF(2,0,.FALSE.)
C
C DOES RELATION HAVE ASSOCIATED DATA
C
              CALL SKFM(RELA,ROW(I),RTASSD,IERR)
              IF (IERR .LT. 0) GOTO 150
C
C INITIALISATION OF BUFFER COL
C
              CALL BLCOL
              CALL COLBLD(6,1,19,'ASSOCIATED DATA ARE')
              CALL PCOL(1,0)
C
C GET ASSOCIATED DATA IN D.B.
C
  110         CALL BLCOL
              CALL SMM(RELB,RELA,IERR)
              CALL GKO(RELB,KLEFT,IERR)
C
C GET TYPE OF ASSOCIATED DATA
C
              CALL GFK(NAMET,KLEFT,T,IERR)
              CALL TYPC9(T,PLEN,PTYPE)
C
```

```
151 C PUT TYPE IN BUFFER COL
152 C
153                 CALL COLBLD(6,1,1,'(')
154                 CALL COLBLD(7,1,PLEN,PTYPE)
155                 CALL COLBLD(PLEN+7,1,1,')')
156 C
157 C GET ASSOC. DATA NAME
158 C
159                 CALL GFK(NAMEC,KLEFT,NAM,IERR)
160 C
161 C PUT IT IN BUFFER COL
162 C
163                 CALL COLBLD(PLEN+9,1,30,NAM)
164 C
165 C PRINT BUFFER COL WITH ASSOC. DATA
166 C
167                 CALL PCOL(0,0)
168 C
169 C FIND AN OTHER ASSOC. DATA
170 C
171         CALL FNSK(RELA,RTASSD,IERR)
172         IF (IERR .GE. 0) GOTO 110
173   150 CONTINUE
174   900 RETURN
175         END
```

3. 2 Résultats obtenus à l'aide d'ISDOS.

Dans cette dernière partie des annexes, nous allons vous présenter:

* Des exemples de formulaires de spécification du PSL

* Des extraits des rapports obtenus à l'aide d'ISDOS

Nous avons voulu limiter le plus possible ces exemples de façon à présenter un éventail complet de résultats tout en minimisant le volume des listings de ces rapports.

Afin d'éviter de vous présenter des pages presque blanches, nous effectuerons certains montages de plusieurs pages de listings et nous placerons également d'éventuels commentaires sur ces listings.

---

32. 1 Documents de spécification.

Vous trouverez dans les pages qui suivent des documents rédigés en PSL.

Document de description de la métastructure

1 * * * * * * 72

Nom de PROCESS PR-APPLICATION:FACTUR:REAPPROV;

Liste de process SUBPARTS PR-ENREG:CDE:CLI:PREFACTURE...,
PR-FACTUR:EXPEDITION.....,
PR-ENREG:LIVR:FOUR:APUREM:DUS.,
PR-FACTUR:EXPEDIT:CDES:DIFF...,
PR-.....,
PR-GESTION:STOCK:REAPPROV.....;

Nom de SYNONYMS PR-APP1:SS1.....;

ATTRIBUTES

TRAIT
- ☐ SYSTEME ,
- ☐ SOUS-SYS ,
- ☒ APPLICATION ,
- ☐ PHASE ,
- ☐ FONCTION ,
- ☐ CONTROLE , } pas de uses vers un set

CLE valeur de l'attribut clé SS1.....;

USES SET-APP1:SS1.....; Nom du set correspondant dans la métastructure

==========

Nom de SET SET-APP1:SS1.....;

Liste de sets SUBSETS SET-PH1:APP1:SS1.....,
SET-PH2:APP1:SS1.....,
SET-PH3:APP1:SS1.....,
SET-PH4:APP1:SS1.....,
SET-.....,
SET-PH5:APP1:SS1.....;

ATTRIBUTES

TRAIT
- ☐ STR-SYS ,
- ☐ CONC-SSYS ,
- ☒ CONC-APPL ,
- ☐ LOG-PH ,

CLE SS1.....;

Valeur identique à celle de l'attribut cle du process décrit dans ce document

Document de description de la structure logique

1 * * * * * * 72

Nom d'ENTITY ENT-PRODUIT.....;
IDENTIFIED BY [EL]-NO:PRODUIT.....;
CONTAINED IN SET-PH1:APP1:SS1.....;
CARDINALITY IS SYSP-NBRE:PRODUITS.....;
CONSISTS OF Liste de composants
SYSP-..... [GR]-DESIGN:PRODUIT.....,
SYSP-..... [EL]-PRIX:UNITAIRE.....,
SYSP-..... [ ]-.....,
SYSP-..... [ ]-.....,
SYSP-..... [ ]-.....,
SYSP-..... [EL]-UNITE:MESURE.....;

==========

ENTITY ENT-STOCK.....;
IDENTIFIED BY [EL]-NO:PRODUIT.....;
CONTAINED IN SET-PH1:APP1:SS1.....;
CARDINALITY IS SYSP-NBRE:PRODUITS.....;
CONSISTS OF
SYSP-..... [EL]-NO:PRODUIT.....,
SYSP-..... [EL]-QUANTITE:STOCK.....,
SYSP-..... [ ]-.....,
SYSP-..... [ ]-.....,
SYSP-..... [ ]-.....,
SYSP-..... [EL]-QUANTITE:DUE.....;

==========

Nom de la RELATION REL-PRODUIT:STOCK.....;
BETWEEN ENT-PRODUIT.....
AND ENT-STOCK.....;
CARDINALITY IS SYSP-NBRE:PRODUITS.....;
CONNECTIVITY IS
SYSP-1..... TO SYSP-1.....;
ASSOCIATED DATA ARE [ ]-.....,
[ ]-.....,
[ ]-.....,
[ ]-.....,
[ ]-.....;

Document de description d'un INPUT ou d'un OUTPUT

1 * * * * * * 72

Type de l'objet OUTPUT 0-LIST:CDES:REFUSEES.....,

SUBPARTS ARE Liste des objets faisant partie de l'input ou de l'output

0-MESS:REFUS:CHEQUE:INVALIDE..,
0-MESS:REFUS:CLI:INSOLVABLE...,
□-.....,
□-.....,
□-.....,
□-.....,
0-MESS:REFUS:LIGNE:INVALIDE..;

CONSISTS OF Liste de composants

SYSP-..... □□-.....,
SYSP-..... □□-.....,
SYSP-..... □□-.....,
SYSP-..... □□-.....,
SYSP-..... □□-.....,
SYSP-..... □□-.....,
SYSP-..... □□-.....,
SYSP-..... □□-.....,
SYSP-..... □□-.....,
SYSP-..... □□-.....,
SYSP-..... □□-.....,
SYSP-..... □□-.....,
SYSP-..... □□-.....,
SYSP-..... □□-.....,
SYSP-..... □□-.....;

SYNONYMS ARE 0-REFUS.....;

HAPPENS SYSP-NBRE:CDES:REFUSEES.....

TIMES-PER INTV-JOUR.....;

Document de description d'un SET

1 * * * * * * 72

Nom de SET SET-CDES:ACCEPTEES.....;

SYNONYMS ARE SET-.....;

SUBSETS ARE Liste des sets composants ce set

SET-.....,
SET-.....,
SET-.....,
SET-.....,
SET-.....,
SET-.....;

CONSISTS OF Liste des composants de ce set

a. liste d'entités

SYSP-..... ENT-.....,
SYSP-..... ENT-.....,
SYSP-..... ENT-.....,
SYSP-..... ENT-.....,
SYSP-..... ENT-.....,
SYSP-..... ENT-.....,

b. liste d'inputs ou d'outputs

SYSP-..... □-.....,
SYSP-..... □-.....,
SYSP-..... □-.....,
SYSP-..... □-.....,
SYSP-..... □-.....,
SYSP-..... □-.....,
SYSP-..... □-.....,
SYSP-..... □-.....,
SYSP-..... □-.....,
SYSP-..... □-.....,
SYSP-..... □-.....,
SYSP-..... □-.....,
SYSP-..... □-.....,
SYSP-..... □-.....,
SYSP-..... □-.....,
SYSP-NBRE:CDES:ACCEPTEES..... I-BON:DE:CDE:CLIENT.....;

Document de description d'1 élément ou d'1 groupe

1 * * * * * * 72

Nom de l'objet ☐ ELEMENT ☒ GROUP GR-LIGNE-CDE.....;

Groupe

CONSISTS OF Liste de composants

SYSP-.....
EL-NOM-PRODUIT.....,
SYSP-.....
□□-.....,
SYSP-.....
□□-.....,
SYSP-.....
□□-.....,
SYSP-.....
□□-.....,
SYSP-.....
□□-.....,
SYSP-.....
□□-.....,
SYSP-.....
□□-.....,
SYSP-.....
□□-.....,
SYSP-.....
□□-.....,
SYSP-.....
□□-.....,
SYSP-.....
EL-QUANTITE-CDEE.....;

ATTRIBUTES ARE

DIMOY ..... ,
Dimension moyenne

DIMAX .....;
Dimension maximum

Elément

Codification de l'image:

cppppp-qq-uu

c définit la classe de la donnée:
****

X = alphanumérique
N = numérique non signé
S = numérique signé

ppppp = nombre de chiffres entiers ou de caractères

qq = nombre de chiffres décimaux

uu = codification de la clause usage de COBOL

C = computational
C1 = computational-1
C2 = computational-2
C3 = computational-3
I = indexed

ATTRIBUTES ARE

IMAGE ....-.... ,
cppppp-qquu

LGMOY ..... ,
Longueur moyenne

LGMAX ..... ,
Longueur maximum

DIMOY ..... ,
Dimension moyenne

DIMAX ..... ;
Dimension maximum

Document de description d'1 élément ou d'1 groupe

1 * * * * * * 72

Nom de l'objet ☒ ELEMENT ☐ GROUP [EL]-MONTANT-TVA.....;

Groupe

CONSISTS OF Liste de composants

SYSP-.....
☐☐-.....,
SYSP-.....
☐☐-.....,
SYSP-.....
☐☐-.....,
SYSP-.....
☐☐-.....,
SYSP-.....
☐☐-.....,
SYSP-.....
☐☐-.....,
SYSP-.....
☐☐-.....,
SYSP-.....
☐☐-.....,
SYSP-.....
☐☐-.....,
SYSP-.....
☐☐-.....;

ATTRIBUTES ARE

DIMOY ..... ,
Dimension moyenne

DIMAX .....;
Dimension maximum

Elément

Codification de l'image:

cppppp-qq-uu

c définit la classe de la donnée:
****

X = alphanumérique
N = numérique non signé
S = numérique signé

ppppp = nombre de chiffres entiers ou de caractères

qq = nombre de chiffres décimaux

uu = codification de la clause usage de COBOL

C = computational
C1 = computational-1
C2 = computational-2
C3 = computational-3
I = indexed

ATTRIBUTES ARE

IMAGE N12.....,
cppppp-qquu

LGMOY 6.....,
Longueur moyenne

LGMAX 12.....,
Longueur maximum

DIMOY .....,
Dimension moyenne

DIMAX .....;
Dimension maximum

Description d'un interface du système

```
1        *        *        *        *        *        *        72
Nom de l' INTERFACE          INTF-SERVICE-CLIENT.....;
SYNONYMS ARE                 INTF-.....;
SUBPARTS ARE  Liste d'interfaces
                             INTF-.....,
                             INTF-.....,
                             INTF-.....,
                             INTF-.....,
                             INTF-.....,
                             INTF-.....;
GENERATES                    I-BON-DE-CDE-CLIENT.....,
                             I-.....,
                             I-.....,
                             I-.....,
                             I-.....,
                             I-.....,
                             I-.....,
                             I-.....,
                             I-.....,
                             I-CHEQUE.....;
RECEIVES                     O-LIST-CDES-REFUSEES.....,
                             O-.....,
                             O-.....,
                             O-.....,
                             O-.....,
                             O-.....,
                             O-.....,
                             O-.....,
                             O-.....,
                             O-LIST-CDES-DIFFEREES.....;
```

Document de description d'1 traitement (a)

```
1        *        *        *        *        *        *                72
Nom du PROCESS                  PR-CALCUL:PREFACTURE.....;
GENERATES Liste d'outputs       O-.............................,
                                O-.............................,
                                O-.............................,
                                O-.............................,
                                O-.............................,
                                O-.............................,
                                O-.............................,
                                O-.............................,
                                O-.............................,
                                O-.............................;
RECEIVES  Liste d'inputs        I-.............................,
                                I-.............................,
                                I-.............................,
                                I-.............................,
                                I-.............................,
                                I-.............................,
                                I-.............................,
                                I-.............................,
                                I-.............................,
                                I-.............................;
USES SET:CDES:EXECUTABLES....., SET:PRODUIT.....................
TO DERIVE          SET:PREFACTURE.....,
                   SET:BON:DE:REQUIS.....;
USES ..............................., ...............................
TO UPDATE          ...............................,
                   ...............................;
USES               ...............................
TO DERIVE          ...............................;
USES               ...............................
TO DERIVE          ...............................;
USES               ...............................
TO UPDATE          ...............................;
USES               ...............................
TO UPDATE          ...............................;
```

Description du fonctionnement dynamique du système

```
1         *         *         *         *         *         *         72
Nom de la CONDITION          COND:CONTR:CDE:CLIENT.....;

    ☐ FALSE
    ☒ TRUE    WHILE;  Lignes de commentaires de 72 caractères
.............EV:TERM:C:CDE:CLIENT.ET.TOUS.LES.CONTROLES.DE.....
.............SEVERITE.4.SONT.NEGATIFS.....................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................;
BECOMING TRUE IS CALLED      EV-.........................,
EV-........................., EV-INIT:CONTROLE:DISP.....;

BECOMING FALSE IS CALLED     EV-.........................,
EV-........................., EV-TERM:TRAIT:PH1.....;
=====================================================================
Nom d' EVENT                 EV-PH1:APP1:SS1.....;
HAPPENS            SYSP-NBRE:CDES.....
   TIMES-PER       INTV-JOUR.....;

TRIGGERS           PR-.........................,
PR-........................., PR-.............................,
PR-........................., PR-CONTROLE:CDE:CLIENT.....;
=====================================================================
Nom de PROCESS     PR-CONTROLE:CDE:CLIENT.....;

HAPPENS            SYSP-NBRE:CDES.....
   TIMES-PER       INTV-JOUR.....;

INCEPTION-CAUSES   EV-.........................,
EV-........................., EV-.........................,
EV-........................., EV-.........................;

TERMINATION-CAUSES EV-.........................,
EV-........................., EV-.........................,
EV-........................., EV-TERM:C:CDE:CLIENT.....;
```

32. 2 Rapports d'ISDOS.

INSTITUT D'INFORMATIQUE FNDP

INPUT SOURCE LISTING

Base de données

PARAMETERS: DB=BOD.PAS INPUT=BOD.IPSL SOURCE-LISTING NOCROSS-REFERENCE NOUPDATE
DATA-BASE-REFERENCE

```
LINE  S T M T                                                          ID FIELD

   1 >/*************************************************************/
   2 >     /*     DESCRIPTION DES INTERFACES DE LA PHASE PH1-APP1-SS1     */
   3 >     INTERFACE           INTER-SERVICE-CLIENT;
   4 >           GENERATES     I-BON-DE-CDE-CLIENT,
   5 >                         I-CHEQUE;
   6 >           RECEIVES      O-LIST-CDES-REFUSEES,
   7 >                         O-LIST-CDES-DIFFEREES;
   8 >           DESCRIPTION;
   9 >                         CE SERVICE S'OCCUPE DE TOUS LES CONTACTS AVEC
  10 >                         LA CLIENTELE:
  11 >                         - RECEPTION DES BONS DE COMMANDE ET DES CHEQUES
  12 >                         - TRANSMISSION DE CEUX-CI AU SERVICE
  13 >                           INFORMATIQUE
  14 >                         - ENVOIS AU CLIENT DE NOTE DE REFUS DE COMMANDE
  15 >                         POUR CAUSE DE  - NON-SOLVABILITE
  16 >                                        - CHEQUE INVALIDE
  17 >                                        - LIGNE DE COMMANDE INVALIDE;
  18 >EOF
```

Voici un exemple de listing d'interprétation d'une section PSL sans mise à jour de la base de données.

* La base de données s'appelle BOD.PAS

* Le source se trouve dans le fichier ISAM: BOD.IPSL dans des enregistrement de 80 caractères, et le dernier enregistrement du fichier contient: EOF

INSTITUT D'INFORMATIQUE FNDP

DATA BASE SUMMARY

PARAMETERS: DD=BOD.DURI.PAS PERCENT SYNONYM NODESCRIPTION NORESPONSIBLE-PROBLEM-DEFINER
NOSOURCE NOSECURITY NOKEYWORD NOATTRIBUTE

NAME TYPE	COUNT	NUMBER WITH SYNONYM	PERCENT WITH SYNONYM
ATTRIBUTE	5	0	
ATTRIBUTE-VALUE	28	0	
CONDITION	2	0	
ELEMENT	37	0	
ENTITY	5	0	
EVENT	9	0	
GROUP	11	0	
INPUT	5	0	
INTERFACE	2	0	
INTERVAL	1	0	
OUTPUT	17	0	
PROCESS	32	11	34.38
RELATION	5	0	
SET	30	0	
SYSTEM-PARAMETER	22	0	
** TOTAL **	211	11	5.21

Voici un résumé des types d'objets contenus dans la base de données.

INSTITUT D'INFORMATIQUE FNDP

FORMATTED PROBLEM STATEMENT

```
57 OUTPUT                                          O-LIST-CDES-REFUSEES;
58       /*   DATE OF LAST CHANGE - 07/12/78 193, 09.18.33 */
59   /* SYNONYMS ARE
60   /* DESCRIPTION
61   /* SEE-MEMO
62   /* KEYWORDS
63   /* ATTRIBUTES ARE
64      GENERATED BY:   PR-ENREG-CDECLI-PREFACTURE,
65                      PR-EDITIONS-PH1-APP1-SS1;
66      RECEIVED BY:    INTER-SERVICE-CLIENT;
67      SUBPARTS ARE:   O-MESS-REFUS-CHEQUE-INVALIDE,
68                      O-MESS-REFUS-CLI-INSOLVABLE,
69                      O-MESS-REFUS-LIGNE-INVALIDE;
70   /* PART OF
71   /* CONSISTS OF
72   /* CONTAINED IN
73   /* LAYOUT
74   /* DERIVED BY
75   /*    USING
76   /* HAPPENS                TIMES-PER
77   /* RESPONSIBLE-PROBLEM-DEFINER IS
78   /* SECURITY IS
79   /* SOURCE IS
80       /* !Z */
```

*/ Ce graphisme signifie que ce
*/ statement n'a pas été employé
*/ dans cette section OUTPUT.

INSTITUT D'INFORMATIQUE FNDP

STRUCTURE REPORT

PARAMETERS: DB=BOD.DURI.PAS NAME=PR-SYSTEME INDENT=3 NOINDEX NOPUNCHED-NAMES LEVELS=ALL
LINE-NUMBERS LEVEL-NUMBERS STATISTICS NONEW-PAGE PRINT

```
COUNT LEVEL NAME

    1     1 PR-SYSTEME
    2      2   PR-SOUS-SYSTEME-COMMERCIAL
    3       3     PR-APPLICATION-FACTUR-REAPPROV
    4        4       PR-ENREG-CDECLI-PREFACTURE
    5         5         PR-CONTROLE-CDE-CLIENT
    6         5         PR-CONTROLE-DISP-MAJ-STOCK
    7         5         PR-CALCUL-PREFACTURE
    8         5         PR-EDITIONS-PH1-APP1-SS1
    9        4       PR-FACTUR-EXPEDITION
   10        4       PR-ENREG-LIVR-FOUR-APUREM-DUS
   11        4       PR-FACTUR-EXPEDIT-CDES-DIFF
   12        4       PR-GESTION-STOCK-REAPPROV
   13       3     PR-APP2-SS1
   14        4       PR-PH1-APP2-SS1
   15         5        PR-F1-PH1-APP2-SS1
   16       3     PR-APP3-SS1
   17      2   PR-SS2
   18       3     PR-APP1-SS2
   19       3     PR-APP2-SS2
   20        4       PR-PH1-APP2-SS2
   21         5        PR-F1-PH1-APP2-SS2
   22         5        PR-FONC-TRI-FIXE
   23        4       PR-PH2-APP2-SS2
   24         5        PR-F1-PH2-APP2-SS2
   25         5        PR-F2-PH2-APP2-SS2
   26        4       PR-PH3-APP3-SS2
   27      2   PR-SS3
   28       3     PR-APP1-SS3
   29        4       PR-PH1-APP1-SS3
   30        4       PR-PH2-APP1-SS3
   31         5        PR-F1-PH2-APP1-SS3
```

PROCESS

Nomenclature des niveaux

Niveau	Elément de la nomenclature
1	SYSTEME
2	SOUS-SYSTEME
3	APPLICATION
4	PHASE
5	FONCTION

Il est bien entendu que cette nomenclature des niveaux n'a d'existence que par rapport à la nomenclature des traitements de la méthode d'analyse.

INSTITUT D'INFORMATIQUE FNDP

SUBSET ANALYSIS REPORT

PARAMETERS: DB=800.DURI.PAS NAME=SET-SYSTEME PRINT NOPUNCHED-NAMES STRUCTURE INDENT=3
LINE-NUMBERS NODE-NUMBERS LEVELS=ALL SUBSETTING-CRITERIA NOLEVEL-STATISTICS PRINT-MATRIX
PRINT-COUNT-TABLE NOINDEX

SUBSET STRUCTURE

COUNT	LEVEL	NAME	SUBSETTING-CRITERIA
1	1.0	SET-SYSTEME	
2	1.1	SET-SS1	
3	1.1.1	SET-APP1-SS1	
4	1.1.1.1	SET-PH1-APP1-SS1	
5	1.1.1.2	SET-PH2-APP1-SS1	
6	1.1.1.3	SET-PH3-APP1-SS1	
7	1.1.1.4	SET-PH4-APP1-SS1	
8	1.1.1.5	SET-PH5-APP1-SS1	
9	1.1.2	SET-APP2-SS1	
10	1.1.2.1	SET-PH1-APP2-SS1	
11	1.1.3	SET-APP3-SS1	
12	1.2	SET-SS2	
13	1.2.1	SET-APP1-SS2	
14	1.2.2	SET-APP2-SS2	
15	1.2.2.1	SET-PH1-APP2-SS2	
16	1.2.2.2	SET-PH2-APP2-SS2	
17	1.2.2.3	SET-PH3-APP2-SS2	
18	1.3	SET-SS3	
19	1.3.1	SET-APP1-SS3	
20	1.3.1.1	SET-PH1-APP1-SS3	
21	1.3.1.2	SET-PH2-APP1-SS3	

Nomenclature des niveaux

Niveau	Elément de la nomenclature
1.0	Structure conceptuelle de système
1.1	Structure conceptuelle de sous-système
1.1.1	Structure conceptuelle d'application
1.1.1.1	Structure logique d'une phase

DICTIONARY REPORT

PARAMETERS: DB=BOD.DURI.PAS NAME=EL-NO-CLIENT NOINDEX NOPUNCHED-NAMES DESCRIPTION
SYNONYMS KEYWORDS RESPONSIBLE-PD ATTRIBUTES SPACING=2 NEW-PAGE PRINT

1 EL-NO-CLIENT ELEMENT

DESCRIPTION:

LIBELLE DE L'INFORMATION

IL S'AGIT DU NUMERO D'UN CLIENT DE L'ENTREPRISE PAR CORRESPONDANCE.

FORMAT/STRUCTURE

- PREMIER CARACTERE: CODE PAYS
- DEUXIEME CARACTERE: CODE PROVINCE SI LE CLIENT EST BELGE
- CARACTERES SUIVANTS: NO DE CLIENT INTERNE @ L'NTREPRISE

TABLE DE CODIFICATION

REPERTOIRE DES VALEURS X X XXXX	SIGNIFICATION
@	CODE PAYS
1	BELGIQUE
2	CEE
@	CODE PROVINCE SI BELGIQUE SINON 0
1	BRABANT
2	FLANDRE OCCIDENTALE
3	FLANDRE ORIENTALE
4	LUXEMBOURG
5	LIEGE
6	NAMUR
7	LIMBOURG
8	HAINAUT

DICTIONARY REPORT

9 ANVERS

NO DONNE AU CLIENT PAR ORDRE CROISSANT
D'INTRODUCTION DANS LE SIGNALETIQUE

ATTRIBUTE:	VALUE:
IMAGE	N6
LGMOY	3
LGMAX	6

INSTITUT D'INFORMATIQUE FNDP

PROCESS SUMMARY

1* PR-ENREG-CDECLI-PREFACTURE

CETTE PHASE EST DECLENCHEE PAR LA RECEPTION D'UN BON DE COMMANDE CLIENT.

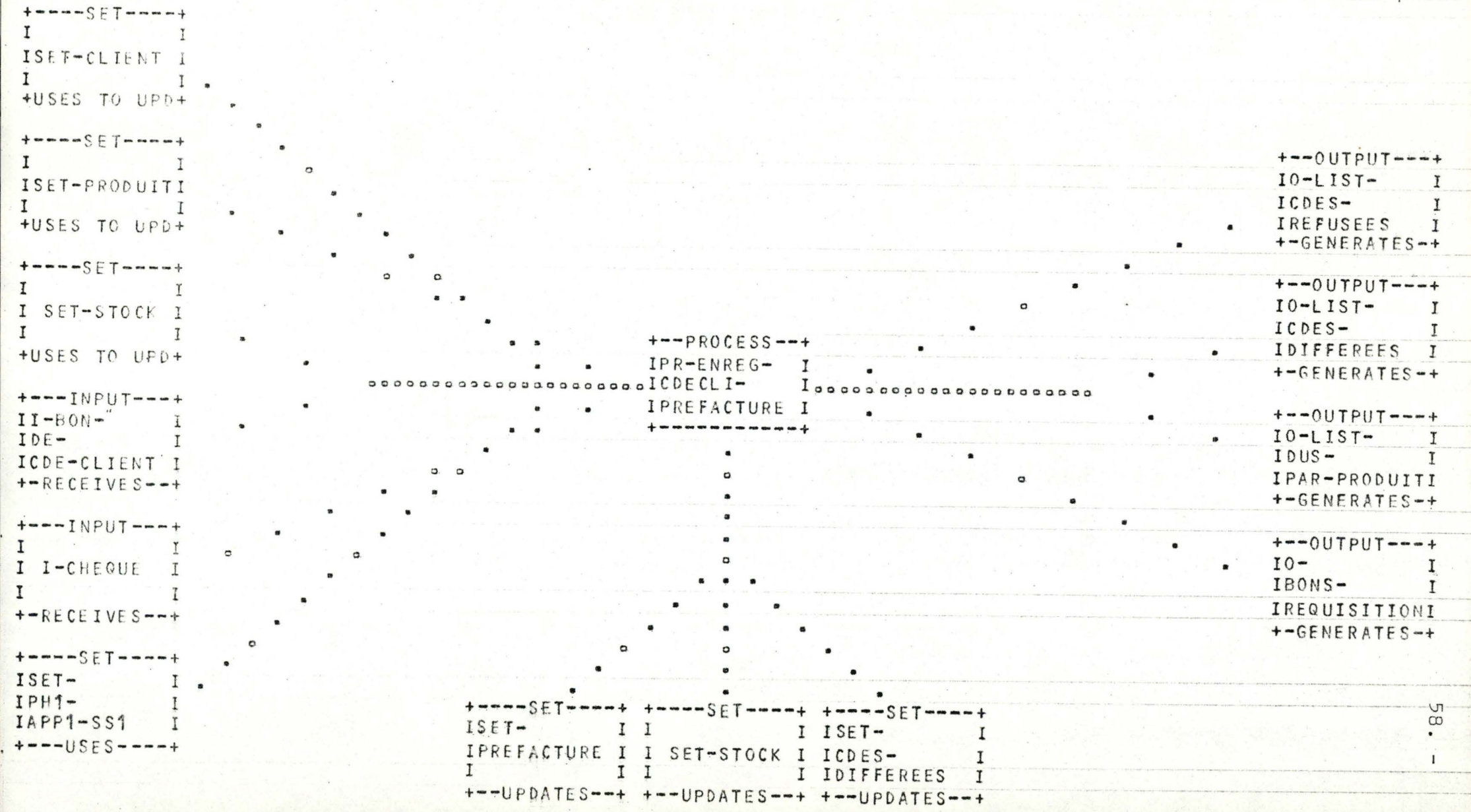
- POUR CHAQUE COMMANDE ON VA PROCEDER AU CONTROLE DE VALIDITE AU NIVEAU DU CLIENT ET AU NIVEAU DU PRODUIT.
- POUR LES COMMANDES ACCEPTEES, ON EFFECTUE UN CONTROLE DE DISPONIBILITE POUR CHAQUE PRODUIT. CECI PERMET DE DISTINGUER LES COMMANDES DIFFEREES COMPOSEES DE LIGNES DIFFERLES DES COMMANDES EXECUTABLES COMPOSEES DE LIGNES EXECUTABLES.
  LES COMMANDES DIFFEREES SONT IMPRIMEES.
- ON TRAITE ENSUITE LES COMMANDES EXECUTABLES EN METANT A JOUR LE FICHIER STOCK ET EN EDITANT LA LISTE DES DUS PAR PRODUIT ET LES BONS DE REQUISITION A DESTINATION DU SERVICE DE GESTION DES STOCKS.
- ON VA ENFIN CALCULER ET MEMORISER LES PREFACTURES POUR CHAQUE COMMANDE PRISE EN CONSIDERATION

INSTITUT D'INFORMATIQUE FNDP

PICTURE REPORT

PR-ENREG-CDECLI-PREFACTURE

Ce graphique représente le diagramme des flux d'information d'une phase

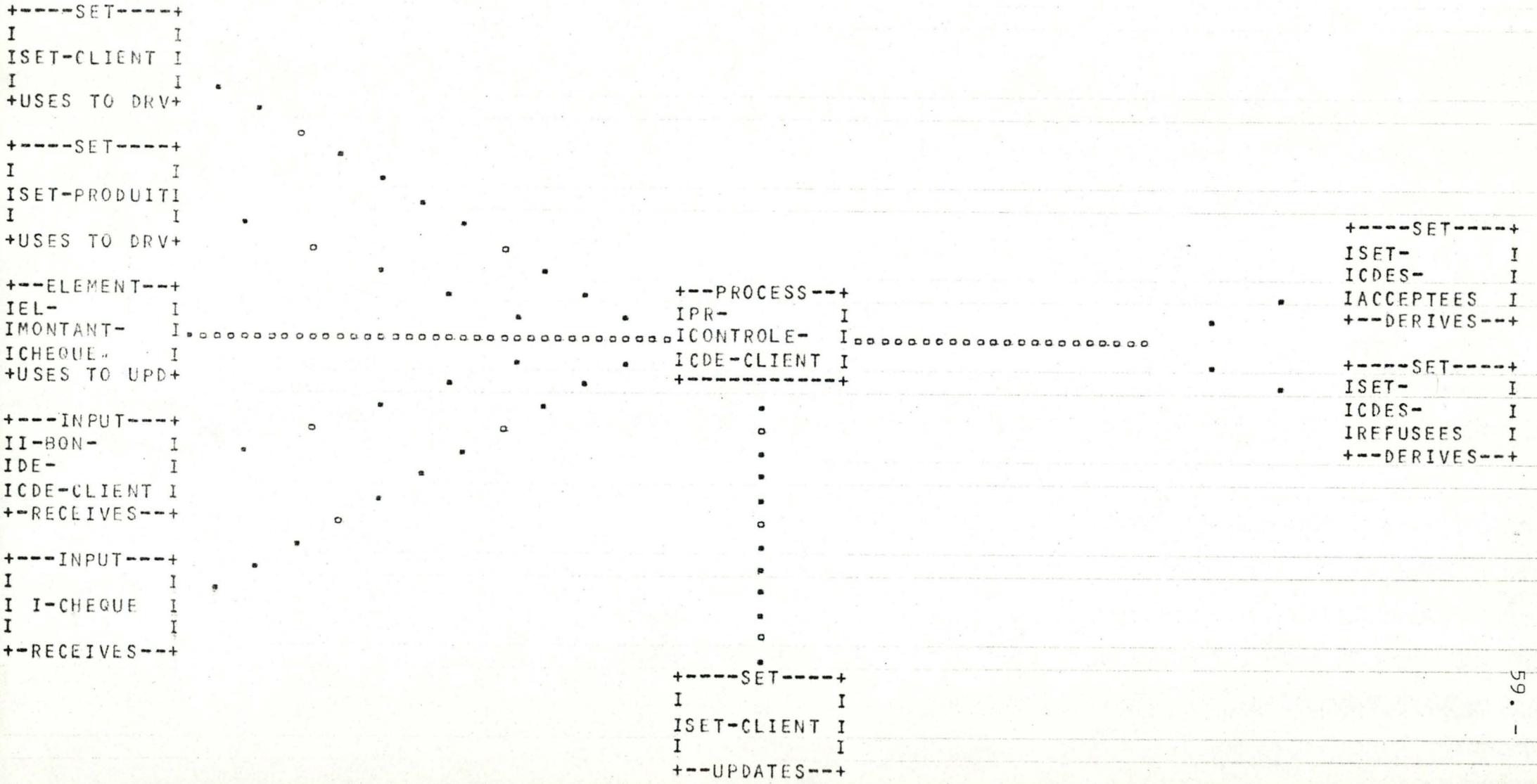


INSTITUT D'INFORMATIQUE FNDP

PICTURE REPORT

PR-CONTROLE-CDE-CLIENT

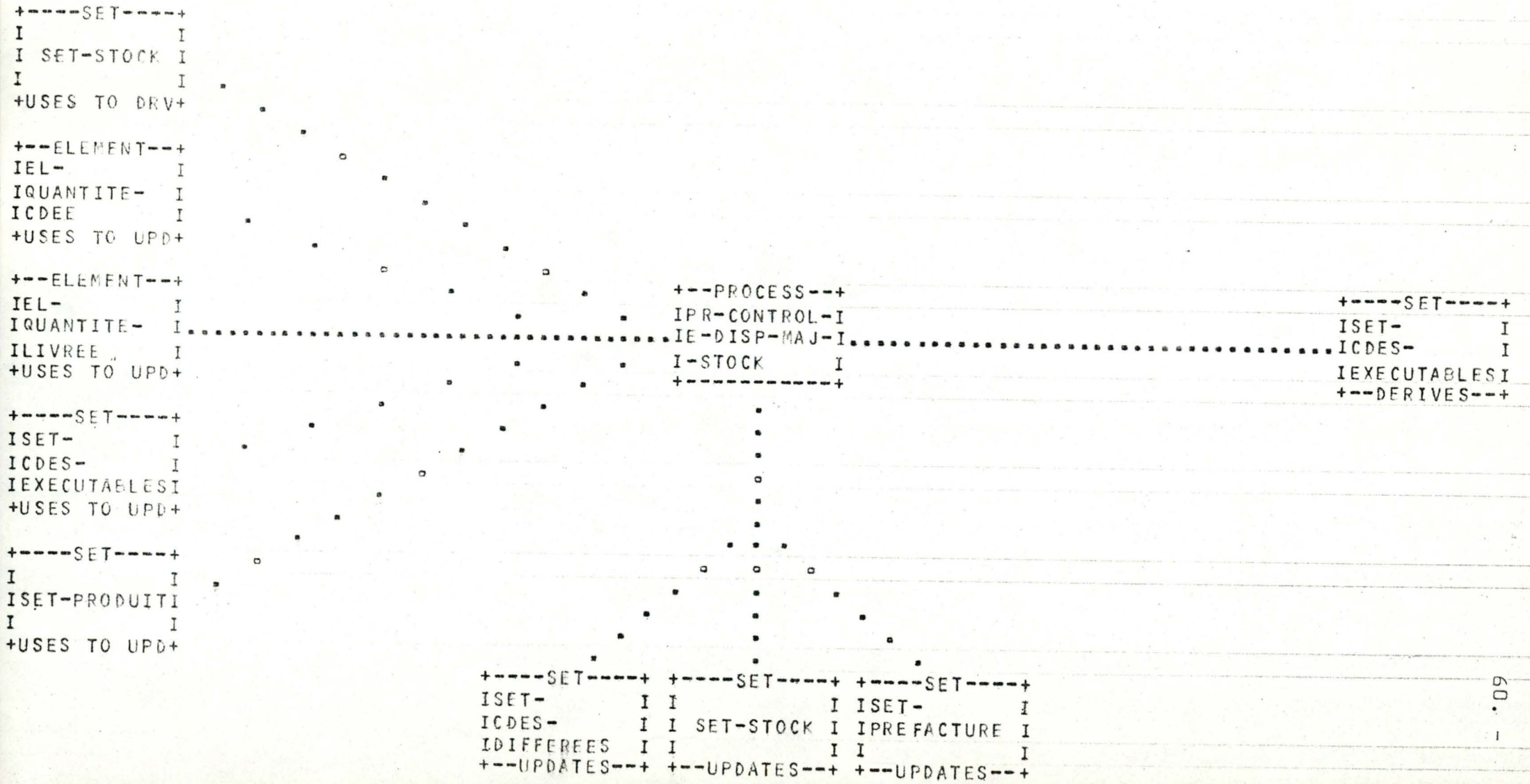
Ce diagramme peut servir ainsi que les trois suivants à assembler ou à rédiger le diagramme d'enchainement des fonctions d'une phase.



INSTITUT D'INFORMATIQUE FNDP

PICTURE REPORT

PR-CONTROLE-DISP-MAJ-STOCK

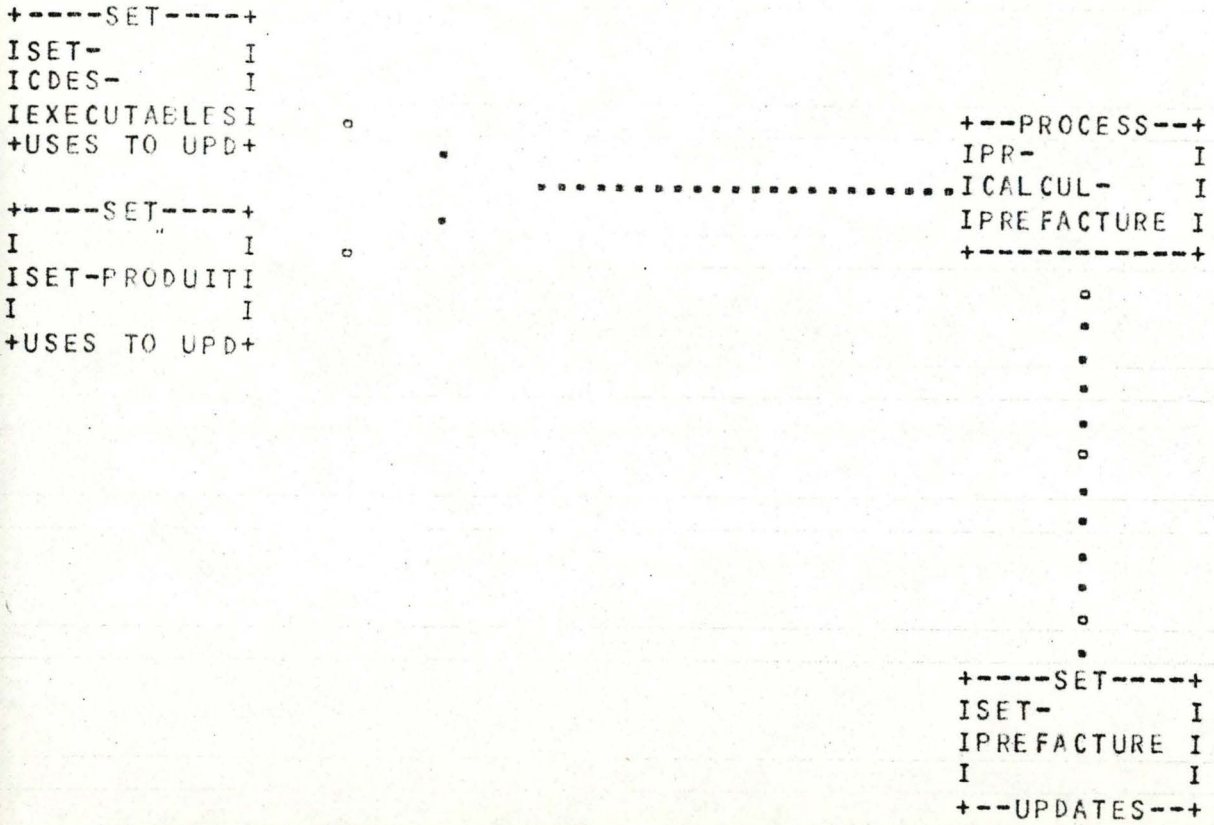


.09

INSTITUT D'INFORMATIQUE FNDP

PICTURE REPORT

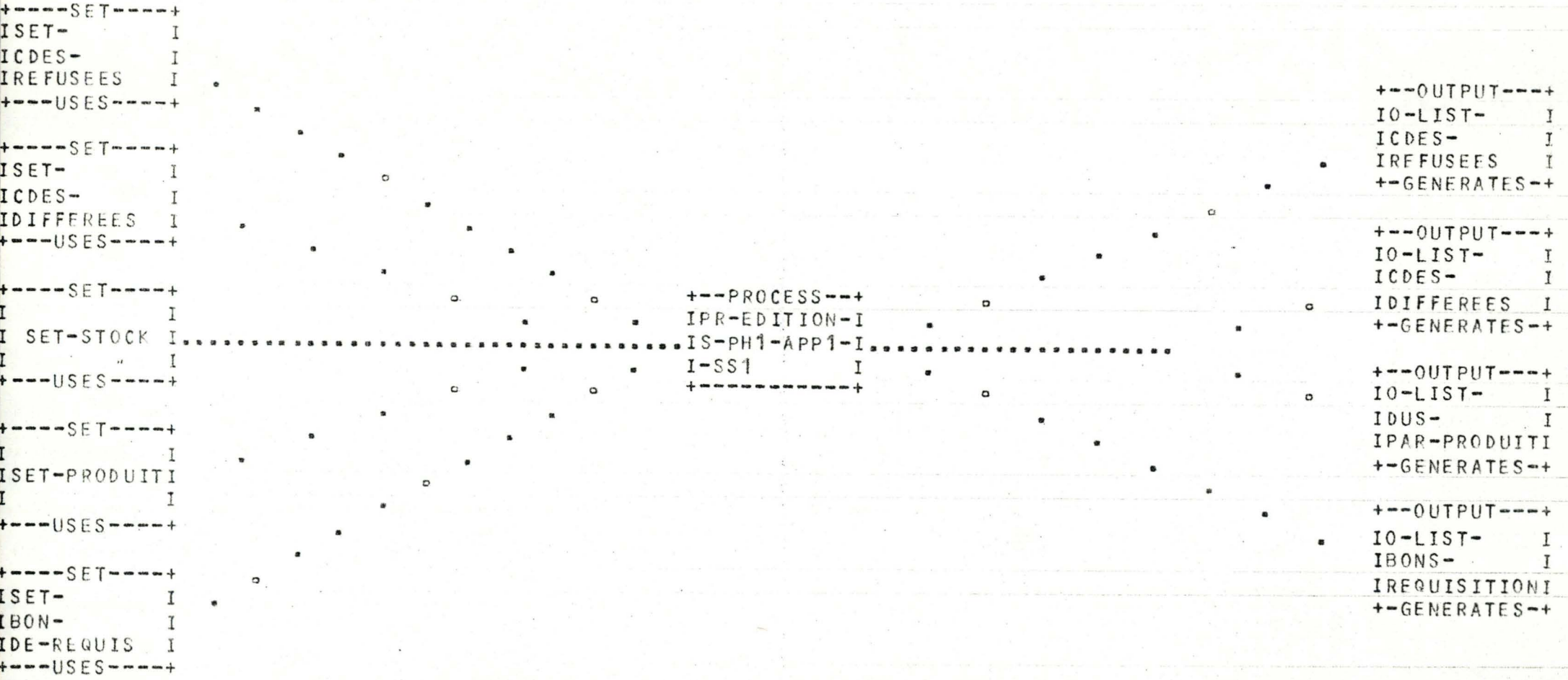
PR-CALCUL-PREFACTURE



INSTITUT D'INFORMATIQUE FNDP

PICTURE REPORT

PR-EDITIONS-PH1-APP1-SS1



INSTITUT D'INFORMATIQUE FNDP

PROCESS SUMMARY

2* PR-CONTROLE-CDE-CLIENT

CETTE FONCTION EFFECTUE LES CONTROLES SUIVANTS:
- CONTROLE DE L'EXISTANCE DU CLIENT ET DE SON ADRESSE
- CONTROLE DE SOLVABILITE DU CLIENT.
- CONTROLE DE VALIDITE DU CHEQUE.
- CONTROLE DE VALIDITE DE CHAQUE POSTE DE COMMAN
CECI PERMET DE DISTINGUER LES COMMANDES
REFUSEES DES COMMANDES ACCEPTEES.
- SI TOUS LES CONTROLES SONT BONS ON MET
A JOUR L'AVOIR DANS LE FICHIER CLIENT.

REGLES DE CONTROLE

SEVERITE :
1 = SUSPICION D'ERREUR ,ACCEPTE.
2 = ERREUR CORRIGEE D'OFFICE ,ACCEPTE.
3 = ERREUR FORMELLE ,ACCEPTE.
4 = ERREUR FATALE ,REJETE.

	CONDITION DE VALIDITE	SEVERITE	MESSAGE D'ERREUR
1	LE CLIENT N'EXISTE PAS DANS LE FICHIER	2	DECLENCHEMENT MAJ F CLIENT
2	L'ADRESSE DU CLIENT EST DIFFERENTE	2	DECLENCHEMENT MAJ F CLIENT
3	LE CODE SOLVABILITE < 0	4	CLIENT INSOLVABLE (M5)
4	LE CHEQUE EST INVALIDE	4	CHEQUE INVALIDE (M20)
5	REFERENCE PRODUIT INCORRECTE	4	PRODUIT INCONNU (M2)
6	QUANTITE INEXISTANTE	4	QUANTITE ABSENTE(M2)

POUR CHAQUE MESSAGE DE SEVERITE 4 ,ON IMPRIMERA
LA DONNEE INCORRECTE ET SON ENVIRONNEMENT

INSTITUT D'INFORMATIQUE FNDP

PROCESS SUMMARY

2* PR-CONTROLE-CDE-CLIENT

REGLES DE CONTROLES

PAS 1. SI LE CLIENT EXISTE ET SI L'ADRESSE EST
INCHANGEE ,ALLER AU PAS 2.
PAS 1.1 DECLENCHER LA PHASE DE MAJ DU SET-CLIENT.

PAS 2. SI LE CODE SOLVABILITE EST + ALLER AU PAS 3.
PAS 2.1 *IMPRIMER LE MESSAGE M5.
*ALLER AU PAS 6.

PAS 3. SI CHEQUE VALIDE (SIGNATURE ET MONTANT OK)
ALLER AU PAS 4.
PAS 3.1 *IMPRIMER LE MESSAGE M20.
*ALLER AU PAS 6.

PAS 4. POUR CHAQUE LIGNE DE LA COMMANDE:
SI LIGNE CORRECTE (QUANTITE ET REFERENCE)
ALLER AU PAS 4.2
PAS 4.1 ENREGISTRER LE REFUS DE LA LIGNE.
PAS 4.2 FIN DE TRAITEMENT D'UNE LIGNE.

PAS 5. SI IL N'Y A PAS DE LIGNE REFUSEE:
METTRE LE FICHIER CLIENT A JOUR:
NOUVEL AVOIR = ANCIEN AVOIR + MONTANT DU
CHEQUE.
APPEL A LA FONCTION PR-CONTROLE-DISP-MAJ-STOCK
PAS 5.1 SINON ALLER AU PAS 6.

PAS 6. FIN DE TRAITEMENT DE LA COMMANDE

ANALYZER UTILITY PROGRAM

CONTENTS REPORT

PARAMETERS: DB=UNKNOWN FILE=UNKNOWN NOCOMPLETENESS-CHECK NOINDEX NOPUNCHED-NAMES
LEVELS=ALL LINE-NUMBERS LEVEL-NUMBERS NAME-TYPES PRINT NONEW-PAGE

				IMAGE	LGMOY	LGMAX	DIMOY	DIMAX
1*	(ENTITY)	1	ENT-CLIENT					
1	(GROUP)	2	GR-DESIGN-CLIENT					
2	(ELEMENT)	3	EL-NO-CLIENT	N6	3	6		
3	(ELEMENT)	3	EL-NOM-CLIENT	X40	20	40		
4	(ELEMENT)	3	EL-ADRESSE	X70	35	70		
5	(ELEMENT)	2	EL-INDIC-SOLVABILITE	N1	1	1		
6	(ELEMENT)	2	EL-ANC-AVOIR	N12	6	12		
7	(ELEMENT)	2	EL-NOUV-AVOIR	N12	6	12		
2*	(ENTITY)	1	ENT-PREFACTURE					
1	(GROUP)	2	GR-DESIGN-PREFACTURE					
2	(ELEMENT)	3	EL-NO-PREFACTURE	N10	5	10		
3	(ELEMENT)	3	EL-DATE-PREFACTURE	N6	6	6		
4	(ELEMENT)	2	EL-MONTANT-HTVA	N12	6	12		
5	(ELEMENT)	2	EL-MONTANT-TVA	N12	6	12		
6	(ELEMENT)	2	EL-MONTANT-TOT	N12	6	12		
3*	(INPUT)	1	I-CHEQUE					
1	(GROUP)	2	GR-DESIGN-CLIENT					
2	(ELEMENT)	3	EL-NO-CLIENT	N6	3	6		
3	(ELEMENT)	3	EL-NOM-CLIENT	X40	20	40		
4	(ELEMENT)	3	EL-ADRESSE	X70	35	70		
5	(GROUP)	2	GR-DESIGN-CHEQUE					
6	(ELEMENT)	3	EL-NO-CHEQUE	N10	10	10		
7	(ELEMENT)	3	EL-DATE-CHEQUE	N6	6	6		
8	(ELEMENT)	3	EL-MONTANT-CHEQUE	N12	6	12		

INSTITUT D'INFORMATIQUE FNDP

DATA PROCESS INTERACTION REPORT

PARAMETERS: DB=BOD.DURI.PAS FILE=BOD.PROCESS PROCESS DATA-PROCESS-INTERACTION-MATRIX
DATA-PROCESS-INTERACTION-ANALYSIS PROCESS-INTERACTION-MATRIX PROCESS-INTERACTION-ANALYSIS
EXPLANATION

THE ROWS ARE DATA NAMES, THE COLUMNS ARE PROCESS NAMES.

	ROW NAMES	
1	SET-PREFACTURE	SET
2	SET-CLIENT	SET
3	SET-PRODUIT	SET
4	SET-STOCK	SET
5	SET-CDES-DIFFEREES	SET
6	SET-PH1-APP1-SS1	SET
7	I-BON-DE-CDE-CLIENT	INPUT
8	I-CHEQUE	INPUT
9	O-LIST-CDES-REFUSEES	OUTPUT
10	O-LIST-CDES-DIFFEREES	OUTPUT
11	O-LIST-DUS-PAR-PRODUIT	OUTPUT
12	O-BONS-REQUISITION	OUTPUT
13	SET-CDES-ACCEPTEES	SET
14	SET-CDES-REFUSEES	SET
15	EL-MONTANT-CHEQUE	ELEMENT
16	SET-CDES-EXECUTABLES	SET
17	EL-QUANTITE-CDEE	ELEMENT
18	EL-QUANTITE-LIVREE	ELEMENT
19	SET-BON-DE-REQUIS	SET
20	O-LIST-BONS-REQUISITION	OUTPUT
21	SET-APP1-SS1	SET
22	SET-PH3-APP1-SS1	SET
23	I-BON-DE-RECEPTION	INPUT
24	O-LIST-BONS-REQUIS-DIFF	OUTPUT
25	SET-PH4-APP1-SS1	SET
26	I-BON-DE-LIVR-CDES-FIFF	INPUT
27	O-LIST-BON-LIVRAISON-REFUSE	OUTPUT
28	O-LIST-FACTURE	OUTPUT
29	O-LIST-BORDEREAU-EXPEDITION	OUTPUT
30	O-LIST-DUS-PAR-CLIENT	OUTPUT

	COLUMN NAMES	
1	PR-ENREG-CDECLI-PREFACTURE	PROCESS
2	PR-CONTROLE-CDE-CLIENT	PROCESS
3	PR-CONTROLE-DISP-MAJ-STOCK	PROCESS
4	PR-CALCUL-PREFACTURE	PROCESS
5	PR-EDITIONS-PH1-APP1-SS1	PROCESS
6	PR-APPLICATION-FACTUR-REAPPROV	PROCESS
7	PR-ENREG-LIVR-FOUR-APUREM-DUS	PROCESS
8	PR-FACTUR-EXPEDIT-CDES-DIFF	PROCESS
9	PR-FACTUR-EXPEDITION	PROCESS
10	PR-GESTION-STOCK-REAPPROV	PROCESS
31	SET-PH2-APP1-SS1	SET
32	I-BON-DE-LIVRAISON	INPUT
33	SET-PH5-APP1-SS1	SET
34	O-MESS-ERREUR-STOCK	OUTPUT
35	O-LIST-ETAT-STOCK	OUTPUT
36	O-LIST-BON-REAPPROV	OUTPUT

INSTITUT D'INFORMATIQUE FNDP

DATA PROCESS INTERACTION REPORT

DATA PROCESS INTERACTION MATRIX

```
                    1
   1234567890
  +-----+-----+
1 :U UU:     :
2 :R1  :     :
3 :RRRR      :
4 :1 1 R 1  1:
5 +U-U-R-1---+
6 :R   :     :
7 :RR  :     :
8 :RR  :     :
9 :D   D     :
10+D---D-----+
11:D   D     :
12:D   :     :
13: D  :     :
14: D  R     :
15+-R--+-----+
16:   FR:    :
17:   R :    :
18:   R :    :
19:    R     :
20+----D-----+
21:    :R    :
22:    : R   :
23:    : R   :
24:    : D   :
25:    :  R  :
  +----+-----+
```

(I,J) VALUE	MEANING
R	ROW I IS RECEIVED OR USED BY COLUMN J (INPUT)
U	ROW I IS UPDATED BY COLUMN J
D	ROW I IS DERIVED OR GENERATED BY COLUMN J (OUTPUT)
A	ROW I IS INPUT TO, UPDATED BY, AND OUTPUT OF COLUMN J (ALL)
F	ROW I IS INPUT TO AND OUTPUT OF COLUMN J (FLOW)
1	ROW I IS INPUT TO AND UPDATED BY COLUMN J
2	ROW I IS UPDATED BY AND OUTPUT OF COLUMN J

INSTITUT D'INFORMATIQUE FNDP

DATA PROCESS INTERACTION REPORT

DATA PROCESS INTERACTION MATRIX ANALYSIS

DATA

SET-PREFACTURE	(SET)	(ROW 1)	NOT DERIVED BY ANY PROCESS
SET-PREFACTURE	(SET)	(ROW 1)	UPDATED, BUT NOT USED BY ANY PROCESS
SET-CLIENT	(SET)	(ROW 2)	NOT DERIVED BY ANY PROCESS
SET-PRODUIT	(SET)	(ROW 3)	NOT DERIVED BY ANY PROCESS
SET-STOCK	(SET)	(ROW 4)	NOT DERIVED BY ANY PROCESS
SET-CDES-DIFFEREES	(SET)	(ROW 5)	NOT DERIVED BY ANY PROCESS
SET-PH1-APP1-SS1	(SET)	(ROW 6)	NOT DERIVED BY ANY PROCESS
I-BON-DE-CDE-CLIENT	(INPUT)	(ROW 7)	NOT USED BY ANY PROCESS
I-CHEQUE	(INPUT)	(ROW 8)	NOT USED BY ANY PROCESS
O-LIST-CDES-REFUSEES	(OUTPUT)	(ROW 9)	NOT DERIVED BY ANY PROCESS
O-LIST-CDES-DIFFEREES	(OUTPUT)	(ROW 10)	NOT DERIVED BY ANY PROCESS
O-LIST-DUS-PAR-PRODUIT	(OUTPUT)	(ROW 11)	NOT DERIVED BY ANY PROCESS
O-BONS-REQUISITION	(OUTPUT)	(ROW 12)	NOT DERIVED BY ANY PROCESS
SET-CDES-ACCEPTEES	(SET)	(ROW 13)	DERIVED, BUT NOT USED BY ANY PROCESS
SET-BON-DE-REQUIS	(SET)	(ROW 19)	NOT DERIVED BY ANY PROCESS
O-LIST-BONS-REQUISITION	(OUTPUT)	(ROW 20)	NOT DERIVED BY ANY PROCESS
SET-APP1-SS1	(SET)	(ROW 21)	NOT DERIVED BY ANY PROCESS
SET-PH3-APP1-SS1	(SET)	(ROW 22)	NOT DERIVED BY ANY PROCESS
I-BON-DE-RECEPTION	(INPUT)	(ROW 23)	NOT USED BY ANY PROCESS
O-LIST-BONS-REQUIS-DIFF	(OUTPUT)	(ROW 24)	NOT DERIVED BY ANY PROCESS
SET-PH4-APP1-SS1	(SET)	(ROW 25)	NOT DERIVED BY ANY PROCESS
I-BON-DE-LIVR-CDES-FIFF	(INPUT)	(ROW 26)	NOT USED BY ANY PROCESS
O-LIST-BON-LIVRAISON-REFUSE	(OUTPUT)	(ROW 27)	NOT DERIVED BY ANY PROCESS
O-LIST-FACTURE	(OUTPUT)	(ROW 28)	NOT DERIVED BY ANY PROCESS
O-LIST-BORDEREAU-EXPEDITION	(OUTPUT)	(ROW 29)	NOT DERIVED BY ANY PROCESS
O-LIST-DUS-PAR-CLIENT	(OUTPUT)	(ROW 30)	NOT DERIVED BY ANY PROCESS
SET-PH2-APP1-SS1	(SET)	(ROW 31)	NOT DERIVED BY ANY PROCESS
I-BON-DE-LIVRAISON	(INPUT)	(ROW 32)	NOT USED BY ANY PROCESS
SET-PH5-APP1-SS1	(SET)	(ROW 33)	NOT DERIVED BY ANY PROCESS
O-MESS-ERREUR-STOCK	(OUTPUT)	(ROW 34)	NOT DERIVED BY ANY PROCESS
O-LIST-ETAT-STOCK	(OUTPUT)	(ROW 35)	NOT DERIVED BY ANY PROCESS
O-LIST-BON-REAPPROV	(OUTPUT)	(ROW 36)	NOT DERIVED BY ANY PROCESS

PROCESSES

PR-EDITIONS-PH1-APP1-SS1	(COLUMN 5)	USES DATA, BUT DOES NOT DERIVE OR UPDATE ANYTHING
PR-APPLICATION-FACTUR-REAPPROV	(COLUMN 6)	USES DATA, BUT DOES NOT DERIVE OR UPDATE ANYTHING

INSTITUT D'INFORMATIQUE FNDP

RELATION STRUCTURE REPORT

RELATION - ENTITY MATRIX

EACH RELATION (ROW) HAS ENTRIES INDICATING WHICH ENTITIES IT CONNECTS. THE TYPE OF ENTRY DENOTES HOW THE TWO ENTITIES ARE RELATED. THIS MATRIX IS BASED ON THE BETWEEN STATEMENT IN THE RELATION SECTION. FOR EXAMPLE: BETWEEN A AND B; AN L, LEFT, MEANS THAT THE ENTITY IS ON THE "LEFT HAND SIDE" OF THE RELATION. IN THE ABOVE EXAMPLE, A IS THE "LEFT" ENTITY. AN R, RIGHT, INDICATES THAT THE ENTITY IS ON THE "RIGHT HAND SIDE" OF THE RELATION. IN THE ABOVE EXAMPLE B IS THE "RIGHT" ENTITY. THUS AN R IN POSITION (I,J) AND AN R IN POSITION (I,K) INDICATES THAT ENTITY J IS RELATED TO ENTITY K VIA RELATION I. THE MAPPING BETWEEN ENTITIES I AND J IS ASSUMED TO BE DIRECTIONAL, I.E., FROM I TO J.

	1	2	3	4	5
1	L	R			
2		L	R		
3		L		R	
4			R	L	
5			L		R

	ROW NAMES		COLUMN NAMES	
1	REL-CLIENT-COMMANDE	RELATION	1 ENT-CLIENT	ENTITY
2	REL-COMMANDE-PRODUIT	RELATION	2 ENT-COMMANDE	ENTITY
3	REL-COMMANDE-PREFACTURE	RELATION	3 ENT-PRODUIT	ENTITY
4	REL-PREFACTURE-PRODUIT	RELATION	4 ENT-PREFACTURE	ENTITY
5	REL-PRODUIT-STOCK	RELATION	5 ENT-STOCK	ENTITY

INSTITUT D'INFORMATIQUE FNDP

RELATION STRUCTURE REPORT

RELATION SUMMARY

REL-CLIENT-COMMANDE	BETWEEN	ENT-CLIENT	AND	ENT-COMMANDE
REL-COMMANDE-PRODUIT	BETWEEN	ENT-COMMANDE	AND	ENT-PRODUIT

ASSOCIATED DATA ARE
(ELEMENT) EL-QUANTITE-CDEE

REL-COMMANDE-PREFACTURE	BETWEEN	ENT-COMMANDE	AND	ENT-PREFACTURE
REL-PREFACTURE-PRODUIT	BETWEEN	ENT-PREFACTURE	AND	ENT-PRODUIT

ASSOCIATED DATA ARE
(ELEMENT) EL-QUANTITE-LIVREE
(ELEMENT) EL-MONTANT-PRODUIT

REL-PRODUIT-STOCK	BETWEEN	ENT-PRODUIT	AND	ENT-STOCK

Voici un exemple de résultat du relation-summary que nous avons dû écrire dans le cadre de notre travail.

INSTITUT D'INFORMATIQUE FNDP

PROCESS CHAIN

PARAMETERS: DB=BOD.DURI.PAS FILE=BOD.PROCESS LINKS=10 PRINT NOINDEX NOPUNCHED-NAMES
EXPLANATION PROCESSES EVENTS BOXES DATA-BASE-ORDER COLUMNS=119 ROWS=39
HORIZONTAL-BOXES=6 VERTICAL-BOXES=6

THIS REPORT SHOWS SEQUENCE RELATIONSHIPS INVOLVING PROCESSES AND/OR EVENTS DERIVED FROM THE TRIGGERS, INCEPTION-CAUSES AND TERMINATION-CAUSES STATEMENTS.

FOR EACH NAME SPECIFIED IN THE INPUT TO THE COMMAND, A TREE IS CONSTRUCTED. THE TREE BEGINS WITH THE INITIAL OBJECT AND PROCEEDS TO THE RIGHT USING ALL THE APPROPRIATE RELATIONSHIPS. THE TREE IS PRINTED LYING ON ITS SIDE RATHER THAN TOP TO BOTTOM AS USUAL.

IN THE TREE, PROCESSES AND EVENTS ARE REPRESENTED BY BOXES. EACH NAME APPEARS IN A BOX, WITH ITS TYPE SHOWN AT THE TOP OF THE BOX. RELATIONSHIPS BETWEEN NAMES ARE REPRESENTED BY DOTTED LINES CONNECTING THE CORRESPONDING BOXES. THE RELATIONSHIP OF A NAME TO THE NAMES TO WHICH IT IS CONNECTED ON THE LEFT IS SHOWN AT THE BOTTOM OF THE BOX.

A TREE MAY REQUIRE MORE THAN ONE PAGE. THEREFORE THE PROGRAM CONTINUES A TREE ON SUCCEEDING PAGES TO THE RIGHT AND/OR DOWNWARD. IF A PAGE IS CONTINUED TO THE RIGHT, THE PAGE NUMBER OF THE CONTINUATION PAGE IS SHOWN ON THE RIGHT OF THE PAGE. THE RIGHTMOST COLUMN OF BOXES ON THE PAGE WILL APPEAR AS THE LEFTMOST COLUMN OF BOXES ON THE CONTINUATION PAGE, TO FACILITATE MATCHING THE TWO PAGES. IF A CONTINUATION OF A PAGE DOWNWARD IS REQUIRED, THE CONTINUATION PAGE NUMBER APPEARS AT THE BOTTOM OF THE PAGE. SINCE IT IS NOT POSSIBLE TO DRAW CONNECTIONS ACROSS THE PAGE BOUNDARY AS DOTTED LINES, THE POSITIONS OF THE BOXES MUST BE USED TO FIND THE FIRST CONNECTION ON THE LEFT SIDE OF THE NEW PAGE. SINCE THE HORIZONTAL POSITIONS OF THE BOXES ARE FIXED, THE BOXES ON THE CONTINUATION PAGE WILL LINE UP PRECISELY WITH THOSE ON THE ORIGINAL PAGE.

IN ADDITION TO BOXES FOR USER NAMES AND THE
CONTINUED ON PAGE N MESSAGES, THREE OTHER MESSAGES MAY APPEAR ON THE REPORT:

NOTHING FOLLOWING IN THE DATA BASE INDICATES THAT THE NAME ON THE LEFT OF THE MESSAGE HAS NO FURTHER RELATIONSHIPS TO BE SHOWN.

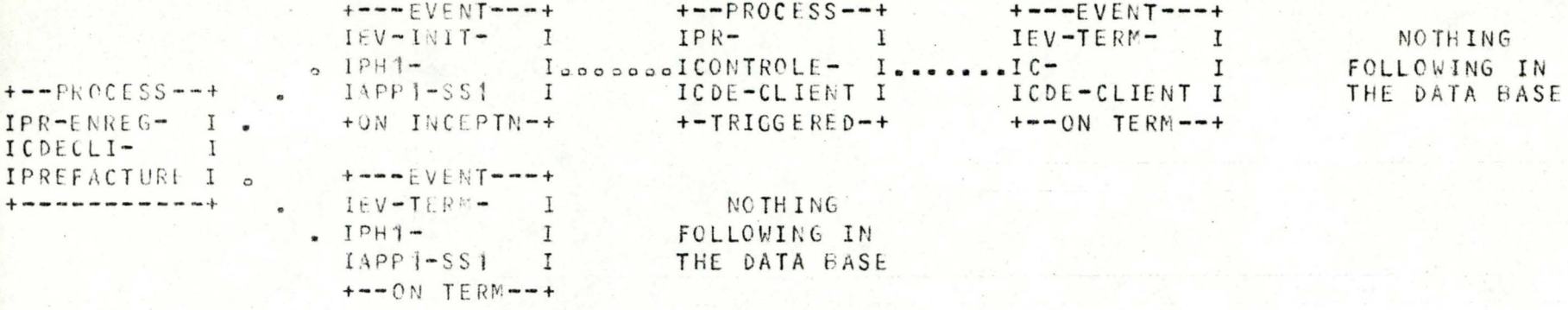
NAME OCCURS ELSEWHERE. SEE INDEX. INDICATES A CONNECTION TO SOME OTHER BOX IN THE PICTURE. THE NAME ON THE LEFT WILL APPEAR ELSEWHERE IN THE REPORT, AND MAY BE MOST EASILY LOCATED BY USING THE INDEX TO THE REPORT, WHICH IS PRODUCED WHENEVER THE INDEX PARAMETER IS GIVEN.

USER LINK LIMIT OF N REACHED INDICATES THAT THE NAME ON THE LEFT HAS FURTHER RELATIONSHIPS WHICH WILL NOT BE SHOWN BECAUSE THE LIMIT, N, SPECIFIED VIA THE LINKS PARAMETER HAS BEEN REACHED.

INSTITUT D'INFORMATIQUE FNDP

PROCESS CHAIN

INITIAL NAME = PR-ENREG-CDECLI-PREFACTURE



INSTITUT D'INFORMATIQUE FNDP

EXTENDED PICTURE

PARAMETERS: DB=BOD.DURI.PAS NAME=PR-CONTROLE-DISP-MAJ-STOCK DATA-FLOW BACKWARD LINKS=14
PRINT NOINDEX NOPUNCHED-NAMES EXPLANATION PROCESSES DATA BOXES DATA-BASE-ORDER
COLUMNS=119 ROWS=39 HORIZONTAL-BOXES=6 VERTICAL-BOXES=6

THIS REPORT SHOWS ONE OF THE FOLLOWING:

STRUCTURE INFORMATION FOR AN ENTITY, GROUP, INPUT, INTERFACE, OUTPUT, PROCESS, OR SET DERIVED FROM SUBPARTS, SUBSETS, AND CONSISTS STATEMENTS.

DATA FLOW RELATIONSHIPS INVOLVING ELEMENTS, ENTITIES, GROUPS, INPUTS, INTERFACES, OUTPUTS, PROCESSES, AND SETS DERIVED FROM USES, DERIVES, UPDATES, RECEIVES, AND GENERATES STATEMENTS.

FOR EACH NAME SPECIFIED IN THE INPUT TO THE COMMAND, A TREE IS CONSTRUCTED. THE TREE BEGINS WITH THE INITIAL OBJECT AND PROCEEDS TO THE RIGHT USING ALL THE APPROPRIATE RELATIONSHIPS. THE TREE IS PRINTED LYING ON ITS SIDE RATHER THAN TOP TO BOTTOM AS USUAL.

IN THE TREE, ELEMENTS, ENTITIES, GROUPS, INPUTS, INTERFACES, OUTPUTS, PROCESSES, AND SETS ARE REPRESENTED BY BOXES. EACH NAME APPEARS IN A BOX, WITH ITS TYPE SHOWN AT THE TOP OF THE BOX. RELATIONSHIPS BETWEEN NAMES ARE REPRESENTED BY DOTTED LINES CONNECTING THE CORRESPONDING BOXES. THE RELATIONSHIP OF A NAME TO THE NAMES TO WHICH IT IS CONNECTED ON THE LEFT IS SHOWN AT THE BOTTOM OF THE BOX.

A TREE MAY REQUIRE MORE THAN ONE PAGE. THEREFORE THE PROGRAM CONTINUES A TREE ON SUCCEEDING PAGES TO THE RIGHT AND/OR DOWNWARD. IF A PAGE IS CONTINUED TO THE RIGHT, THE PAGE NUMBER OF THE CONTINUATION PAGE IS SHOWN ON THE RIGHT OF THE PAGE. THE RIGHTMOST COLUMN OF BOXES ON THE PAGE WILL APPEAR AS THE LEFTMOST COLUMN OF BOXES ON THE CONTINUATION PAGE, TO FACILITATE MATCHING THE TWO PAGES. IF A CONTINUATION OF A PAGE DOWNWARD IS REQUIRED, THE CONTINUATION PAGE NUMBER APPEARS AT

EXTENDED PICTURE

THE BOTTOM OF THE PAGE. SINCE IT IS NOT POSSIBLE TO DRAW CONNECTIONS ACROSS THE PAGE BOUNDARY AS DOTTED LINES, THE POSITIONS OF THE BOXES MUST BE USED TO FIND THE FIRST CONNECTION ON THE LEFT SIDE OF THE NEW PAGE. SINCE THE HORIZONTAL POSITIONS OF THE BOXES ARE FIXED, THE BOXES ON THE CONTINUATION PAGE WILL LINE UP PRECISELY WITH THOSE ON THE ORIGINAL PAGE.

IN ADDITION TO BOXES FOR USER NAMES AND THE CONTINUED ON PAGE N MESSAGES, THREE OTHER MESSAGES MAY APPEAR ON THE REPORT:

NOTHING FOLLOWING IN THE DATA BASE INDICATES THAT THE NAME ON THE LEFT OF THE MESSAGE HAS NO FURTHER RELATIONSHIPS TO BE SHOWN.

NAME OCCURS ELSEWHERE. SEE INDEX. INDICATES A CONNECTION TO SOME OTHER BOX IN THE PICTURE. THE NAME ON THE LEFT WILL APPEAR ELSEWHERE IN THE REPORT, AND MAY BE MOST EASILY LOCATED BY USING THE INDEX TO THE REPORT, WHICH IS PRODUCED WHENEVER THE INDEX PARAMETER IS GIVEN.

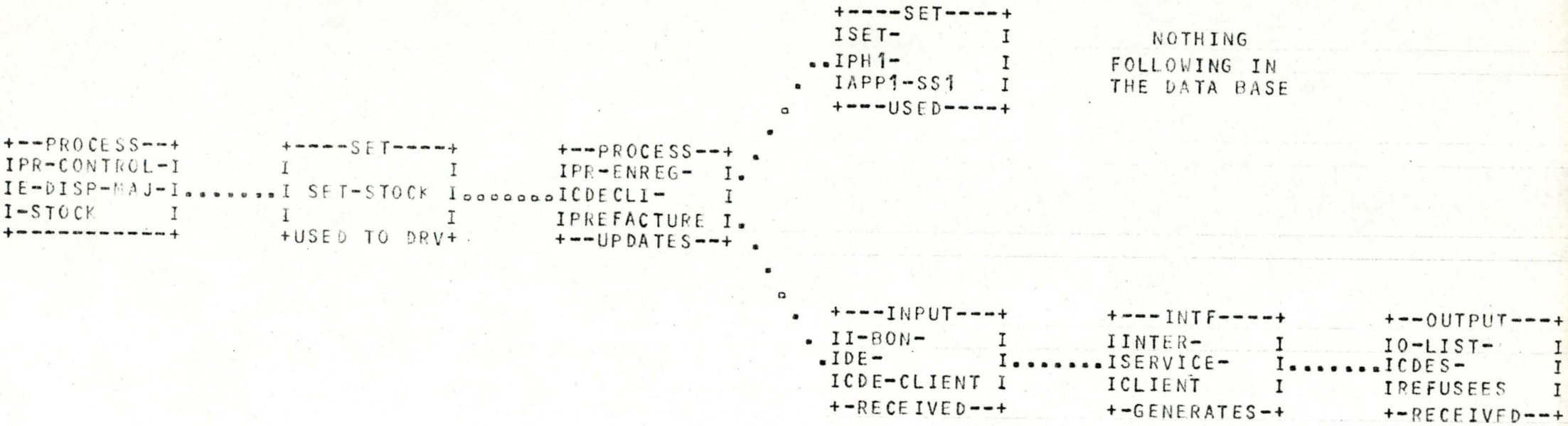
USER LINK LIMIT OF N REACHED INDICATES THAT THE NAME ON THE LEFT HAS FURTHER RELATIONSHIPS WHICH WILL NOT BE SHOWN BECAUSE THE LIMIT, N, SPECIFIED VIA THE LINKS PARAMETER HAS BEEN REACHED.

EXPLANATION OF THE PROCEDURE DEFINITION ANALYSIS

INSTITUT D'INFORMATIQUE FNDP

EXTENDED PICTURE

INITIAL NAME = PR-CONTROLE-DISP-MAJ-STOCK

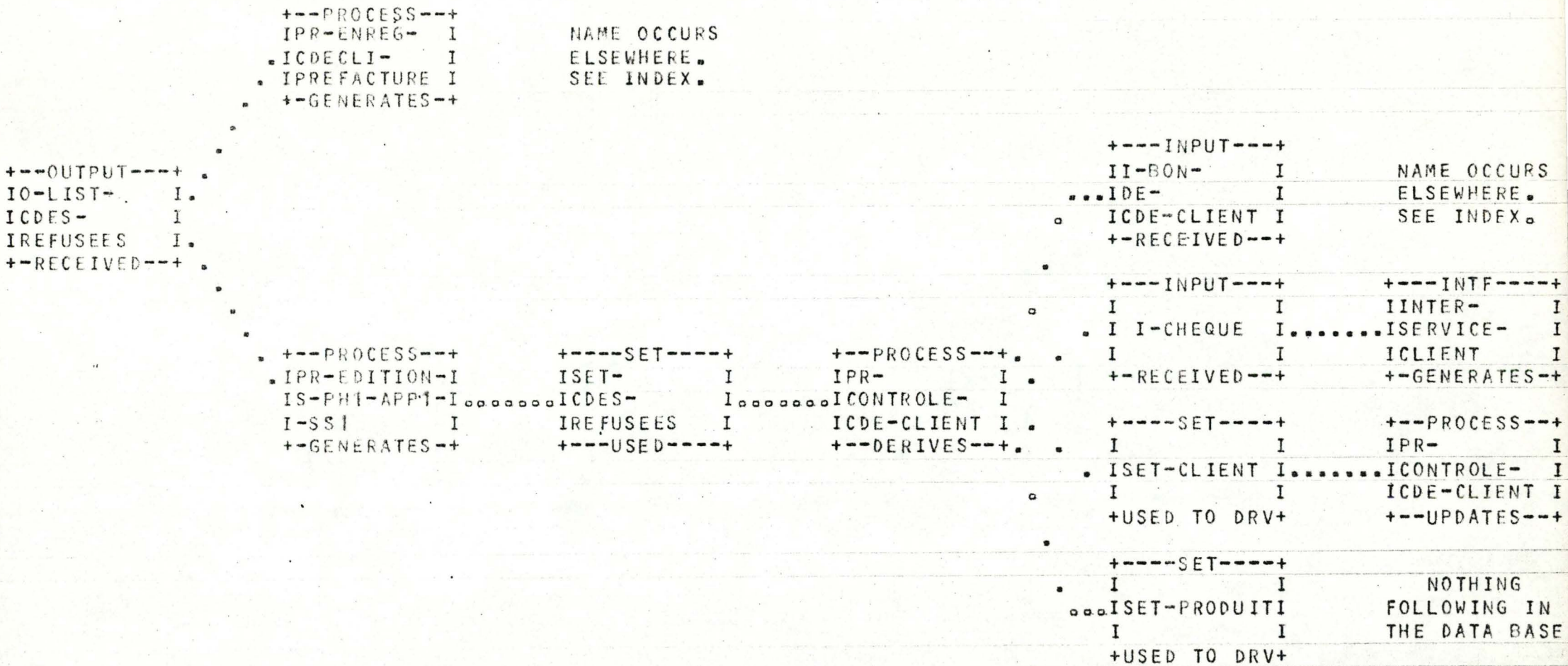


CONTINUED ON PAGE 26

INSTITUT D'INFORMATIQUE FNDP

EXTENDED PICTURE

INITIAL NAME = PR-CONTROLE-DISP-MAJ-STOCK



CONTINUED ON PAGE 27

BUMP
0 0 3 2 1 3 3 8 6
*FM B16/1978/08/2