

## THESIS / THÈSE

### MASTER EN SCIENCES INFORMATIQUES

#### Contenus et usages des dictionnaires de données. Essai de synthèse critique

Zimmer, Hugues

*Award date:*  
1977

*Awarding institution:*  
Universite de Namur

[Link to publication](#)

#### General rights

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

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

#### Take down policy

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

FACULTES UNIVERSITAIRES NOTRE-DAME DE LA PAIX - NAMUR

INSTITUT D'INFORMATIQUE

CONTENUS ET USAGES  
DES  
DICTIONNAIRES DE DONNEES  
ESSAI DE SYNTHESE CRITIQUE

Hugues ZIMMER

Mémoire présenté  
en vue de l'obtention du grade de  
Licencié et Maître en Informatique

ANNEE ACADEMIQUE 1976-1977

## REMERCIEMENTS

L'idée d'une recherche sur les dictionnaires de données nous a été suggérée par Monsieur André CLARINVAL ; elle a été développée, pour l'essentiel, sous la conduite de Monsieur Roland LESUISSE.

Nous les remercions l'un et l'autre pour l'aide et les conseils qu'ils nous ont prodigés.

Nous tenons aussi à remercier tout spécialement Monsieur François BODART, Madame Colette ROLLAND, Directeur de l'U.E.R. Informatique de l'Université de Nancy, le Révérend Père Jacques BERLEUR ainsi que Monsieur Henry LEROY pour tous les conseils qu'ils nous ont donnés et les critiques qu'ils ont faites à une première version de notre travail.

Nous tenons également à remercier tout particulièrement Madame Monique NOIRHOMME, Messieurs Jean FICHEFET, François BODART, Claude CHERTON, Jacques BERLEUR et Jean RAMAEKERS pour la confiance qu'ils nous ont témoignée lors de notre examen d'entrée.

Nos remerciements vont également à tous les professeurs et assistants qui nous ont permis d'acquérir une formation de base permettant de réaliser ce travail dans d'excellentes conditions.

Nous témoignons également notre gratitude aux personnes qui ont permis la réalisation matérielle de ce travail et tout particulièrement Madame Christiane OGER-LE ROY.

## TABLE DES MATIERES

INTRODUCTION	1
1. OBJECTIF POURSUIVI	1
2. CHOIX DE DICTIONNAIRES REPRESENTATIFS	1
3. LA METHODE D'ANALYSE	2
4. ARTICULATION DU TRAVAIL	2
<u>CHAPITRE I</u>	
GENERALITES	4
I.1. DEFINITION D'UN DICTIONNAIRE DE DONNEES	4
I.2. OBJECTIF RECHERCHE	4
I.3. CONTENU D'UN DICTIONNAIRE DE DONNEES	5
I.3.1. Que décrit un dictionnaire de données ?	5
I.3.2. Comment le décrit-il ?	6
I.3.2.1. Les caractéristiques identifiantes	6
I.3.2.2. Les caractéristiques intrinsèques	6
I.3.2.3. Les caractéristiques relationnelles	6
I.4. ORGANISATION D'UN DICTIONNAIRE DE DONNEES	8
I.5. USAGES DU DICTIONNAIRE DE DONNEES	9
I.5.1. Utilisation dans le management d'un projet	9
I.5.2. L'utilisation par les programmes	9
I.6. L'ADMINISTRATEUR DES DONNEES	10
<u>CHAPITRE II - ETUDE DE L'EXISTANT</u>	
II.1. INTRODUCTION	14
II.1.1. Objectif du chapitre II	14
II.1.2. Les dictionnaires sous étude	14
II.1.3. Rappels sur la démarche	15
II.1.4. Description de la structure relationnelle des dictionnaires	15
II.2. CONTENU ET USAGES DU DICTIONNAIRE DE DATAMANAGER	16
II.2.1. Introduction à l'analyse	16
II.2.2. Analyse du contenu du dictionnaire	16
II.2.3. Analyse des usages du dictionnaire	22

II.3. CONTENU ET USAGES DU DICTIONNAIRE DE LA BRITISH RAIL	25
II.3.1. Introduction à l'analyse	25
II.3.2. Analyse du contenu du dictionnaire	25
II.3.3. Analyse des usages du dictionnaire	31
II.4. CONTENU ET USAGES DU DICTIONNAIRE ASSOCIE AU PROJET REMORA	33
II.4.1. But du projet, rôle du dictionnaire	33
II.4.2. Analyse du contenu du dictionnaire	34
II.4.3. Analyse des usages du dictionnaire	39
II.5. CONTENU ET USAGES DU DICTIONNAIRE D'UHROWCZIK	40
II.5.1. Introduction à l'analyse	40
II.5.2. Analyse du contenu du dictionnaire	40
II.5.3. Analyse des usages du dictionnaire	49

### CHAPITRE III

SYNTHESE ET CRITIQUES	51
III.1. INTRODUCTION	51
III.2. ANALYSE COMPARATIVE DES DIFFERENTS CONTENUS	51
III.2.1. sur les noms des classes de données	51
III.2.2. sur les définitions des classes de données	52
III.2.3. sur les caractéristiques descriptives des classes de données	55
III.2.4. Conclusions et critiques	55
III.3. ANALYSE COMPARATIVE DES DIFFERENTS USAGES	66
III.3.1. Le "management use"	66
III.3.2. Le "Program use"	72
III.4. LE CYCLE DE VIE	75
III.4.1. Etape de structuration	75
III.4.2. Etape de remplissage et de gestion	76
III.4.3. Etape d'exploitation	78
III.4.4. Conclusions	78
CONCLUSIONS	81
BIBLIOGRAPHIE	83

## INTRODUCTION

### 1. OBJECTIF POURSUIVI

L'objectif poursuivi par cette étude sur les dictionnaires de données est de réaliser une synthèse critique de l'existant, basée sur une analyse méthodique des contenus et usages de quatre dictionnaires choisis.

Deux raisons majeures sont à la base de ce travail. Premièrement, les dictionnaires de données tendent de plus en plus à jouer un rôle essentiel au sein des organisations, car ils essayent de remédier - partiellement - à une insuffisance reconnue de communication de la documentation entre ceux qui sont attachés au développement d'un projet informatique. La seconde raison part du fait qu'à l'heure actuelle, il existe un nombre impressionnant de dictionnaires de données qui ont été ou sont créés sans souci de concertation entre leurs concepteurs. Cela engendre une débauche d'efforts souvent inutiles qu'il serait possible de limiter en faisant le point d'une manière critique sur l'état de l'art en cette matière. De là, notre démarche.

### 2. CHOIX DE DICTIONNAIRES REPRESENTATIFS

Le nombre important de dictionnaires de données existants rend quasi impossible une étude exhaustive dans ce domaine. Aussi avons-nous restreint notre analyse à un échantillon de dictionnaires de données utilisés et utilisables, représentatifs selon nous de leurs secteurs d'utilisation.

Cet échantillon se composera de quatre dictionnaires :

- un dictionnaire adaptable à diverses organisations, créé pour être commercialisé sous forme de package : ce sera celui de la firme de software Datamanager.
- un dictionnaire créé et utilisé par une grosse firme pour ses besoins propres : ce sera celui de la British Rail.
- un dictionnaire se situant à l'avant-garde de la recherche: ce sera le dictionnaire associé au projet REMORA.
- un dictionnaire "idéal", conçu par Uhrowczik chez IBM.

### 3. LA METHODE D'ANALYSE

Pour que l'on puisse se livrer à un travail de comparaison et de synthèse, il faut que cette analyse soit faite pour tous les dictionnaires de la même manière. Mais cette analyse, selon un moule commun ne va pas sans poser quelques problèmes :

- La présentation qui est faite des dictionnaires dans la littérature est souvent disparate ; la documentation se trouve la plupart du temps à des degrés divers d'élaboration ;
- Le vocabulaire employé manque tout à fait de normalisation, à tel point, on le verra, que deux noms identiques désignent dans deux dictionnaires différents des réalités différentes ;
- Enfin, il est difficile de mettre dans un même schéma des manières fondamentalement différentes d'appréhender un système d'information.

Malgré ces handicaps, nous recourrons à un schéma d'analyse commun qui, en gros, s'articule ainsi :

- a. Analyse des contenus d'un dictionnaire de données ;
- b. Analyse des usages qu'on fait d'un dictionnaire de données.

Cela fait, notre synthèse consistera à mettre en présence les contenus, les usages de chaque dictionnaire et en fonction de cette comparaison à souligner les avantages, les inconvénients et les lacunes des uns par rapport aux autres et surtout à proposer des améliorations.

### 4. ARTICULATION DU TRAVAIL

Notre étude débutera par un chapitre de généralités, où nous définirons le dictionnaire de données, les objectifs qu'il poursuit, son contenu et ses usages. Nous évoquerons également la fonction nouvelle d'administrateur des données qui résulte de l'implantation des dictionnaires dans les organisations.

Ensuite, dans un second chapitre, nous analyserons les contenus et les usages des quatre dictionnaires retenus selon la méthode que nous aurons décrite au chapitre un.

Notre troisième et dernier chapitre effectuera, au point de vue contenu et usages, une synthèse critique, dégagant les concepts de base communs à tous les dictionnaires et mettant en évidence leurs lacunes, e.a. managériales.

°  
° °

## CHAPITRE I

### GENERALITES

Lorsqu'il s'agit de définir un dictionnaire de données, la littérature informatique fait apparaître des divergences d'opinions assez prononcées parmi les différents auteurs. Aussi, allons-nous commencer notre travail par une définition à laquelle nous nous référerons dans la suite de l'exposé.

#### I.1. DEFINITION D'UN DICTIONNAIRE DE DONNEES

Nous conviendrons donc qu'un DICTIONNAIRE DE DONNEES est :  
*"un répertoire centralisé de descriptions de données utilisables par un projet, une entreprise ou une organisation."*

A partir de là, nous allons tenter de préciser l'objectif, le contenu, ainsi que les usages d'un dictionnaire de données ; nous évoquerons également les changements que suscite son apparition dans une organisation, en décrivant la fonction nouvelle d'administrateur des données. (Data Administrator (1).)

#### I.2. OBJECTIF RECHERCHE

L'objectif que poursuit une organisation lorsqu'elle se dote ainsi d'un dictionnaire de données est avant tout d'*ordre économique* : elle vise à accroître la qualité du travail de son centre de traitement de l'information et, par là même, sa *productivité*.

La question se pose tout naturellement de savoir comment un dictionnaire de données peut répondre à cet objectif (par exemple : diminuer les coûts dans le développement d'une application) ou, ce qui revient au même, à quelle(s) insuffisance(s) constatée(s) un dictionnaire met-il fin ?

---

(1) E.D.P. Analyser : *The 'Data Administrator' Functions*, pp. 12-14.

La lacune essentielle que l'on peut constater nous paraît être la suivante : il existe dans presque tous les centres de traitement de l'information une insuffisance de *communication* de la documentation entre tous ceux qui sont attachés au développement d'un même projet.

Cette insuffisance entraîne :

- a. des *redondances* et des *inconsistances* dans la mise au point des projets et, par voie de conséquence, des incompréhensions. Par exemple, cela se produit lorsque :
  - deux données identiques sont décrites de façon différentes ;
  - deux données différentes sont identifiées de la même façon (1).
- b. les *effets des changements* apportés dans une application ne peuvent être *déterminés*, ce qui rend la maintenance pénible. Par exemple, les modifications d'un enregistrement signalétique d'un employé (état civil, passage de vie à trépas, enfants à charge, etc.) entraînent un ensemble de mesures à prendre à court, moyen et long terme, qui peuvent amener des complications de tout ordre.

Pour résoudre la multiplicité des problèmes ainsi posés, le dictionnaire de données propose la *standardisation* de la documentation. Cette technique offre les avantages suivants, correspondant aux défauts soulignés :

- l'usage de standards facilite la description des données, leur utilisation, leur contrôle et leur maintenance.
- l'usage de standards permet de prévoir l'effet de changements. En effet, en précisant, par exemple dans la description d'une donnée, l'endroit où elle est utilisée, les effets d'un changement peuvent toujours être prévus.

Ceci revient à dire (2) que le dictionnaire de données permet une vue cohérente du système informatique d'une organisation par chacun de ses membres.

### I.3. CONTENU D'UN DICTIONNAIRE DE DONNEES

#### I.3.1. Que décrit un dictionnaire de données ?

Par définition, un dictionnaire de données décrit des données. Une

---

(1) British Rail : *Data Dictionary*.

(2) Automobile Peugeot : *Package Minos III A*.  
BODART, F. : *Structuration fonctionnelle du système informatique d'une organisation*, pp. 349 - 350.

donnée est une entité (1) ou plus explicitement un élément fonctionnel de la structure d'un système informatique (2).

L'identification de ces éléments fonctionnels est généralement liée à une méthode d'analyse. Ainsi dans le cadre du dictionnaire de Uhrowczik, les entités identifiées dans sa méthode d'analyse se résument comme suit (3) :

- pour ce qui concerne les traitements à réaliser dans une application, il distingue des programmes, des transactions, des jobs (4).
- pour ce qui concerne les données d'une application, il discrimine des éléments, des articles, des segments physiques, des fichiers, des bases de données (4).

Ces relations entre les éléments fonctionnels procurent au dictionnaire de données une structure semblable à celle du système informatique. (voir figure I.1)

### I.3.2. Comment le décrit-il ?

Un dictionnaire de données est donc un répertoire centralisé de descriptions d'entités. Les caractéristiques descriptives de données peuvent être discriminées en trois grands groupes (5) :

#### I.3.2.1. Les caractéristiques identifiantes

Elles ont pour but d'identifier les données répertoriées en donnant, par exemple, pour chacune d'elles : son nom, sa signification, son responsable, sa version, etc.

#### I.3.2.2. Les caractéristiques intrinsèques

Elles sont propres à la nature de la donnée enregistrée. Par exemple, ce sera sa longueur, son format, son mode, etc.

#### I.3.2.3. Les caractéristiques relationnelles

Elles lient logiquement et physiquement la donnée à son environne-

(1) UHROWCZIK, P.P. : *Data Dictionary/Directory*.

(2) Automobile Peugeot : *Package Minos III A*.

(3) pour plus amples informations, cfr. notre chapitre II.

(4) University Computing Company : *Data Dictionary and Manager*.

(5) BODART, F. : *Introduction à l'analyse fonctionnelle des systèmes informatiques de gestion*.

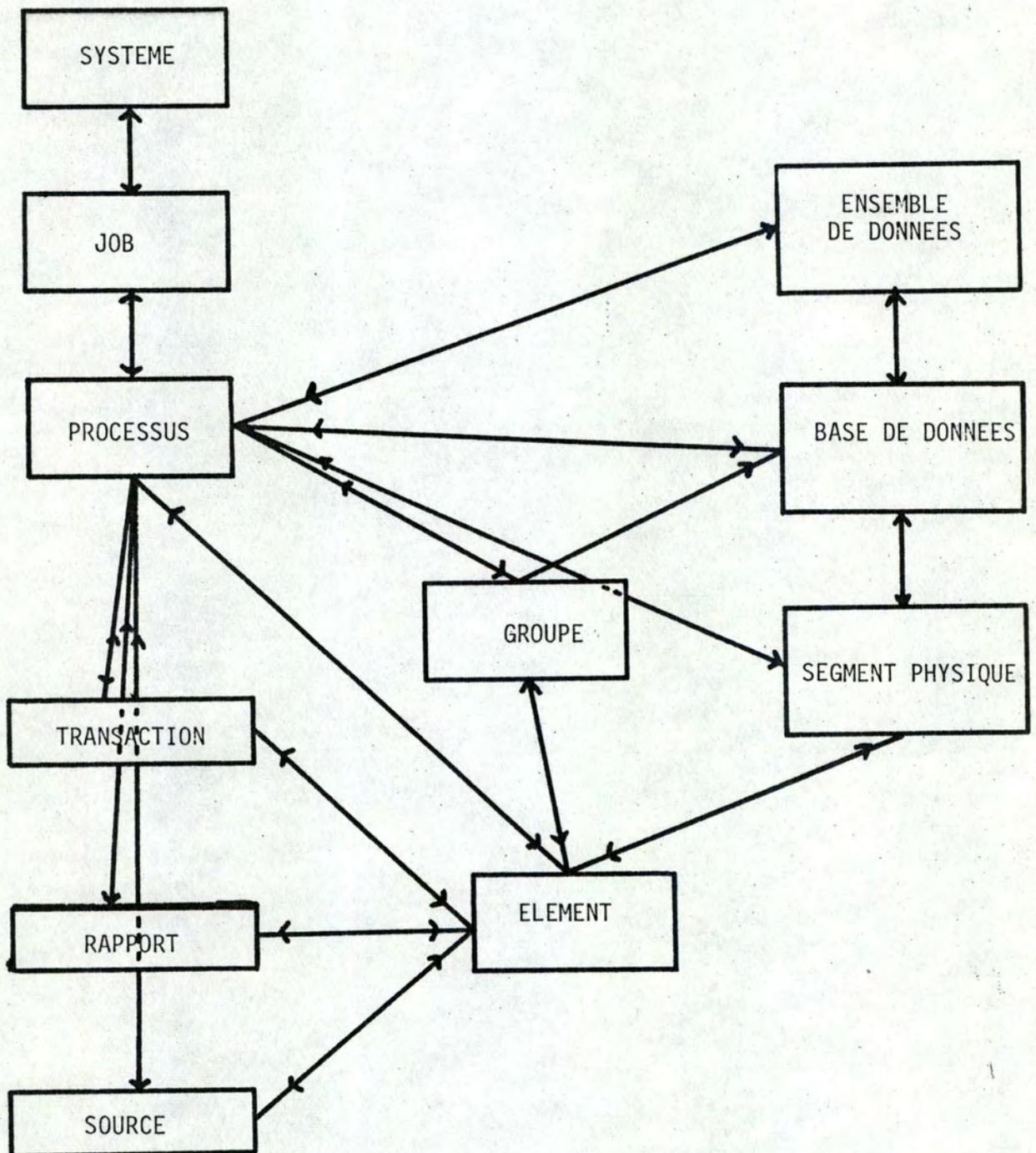


Figure 1.1. : Structure du dictionnaire d'Uhrowczik

ment. Ce sont, par exemple, les règles d'appartenance, de recouvrement, les relations explicites et/ou implicites, le mode d'accès, etc.

Par exemple, les caractéristiques descriptives d'un enregistrement de fichier dans le dictionnaire de la British Rail (1) sont les suivantes :

- Caractéristiques identifiantes :
  - User name,
  - Version/Date,
  - Assembler name,
  - Informal description,
  - Version name.
- Caractéristiques intrinsèques :
  - Type d'enregistrement (maître, ...),
  - Caractéristiques de longueur (fixe, variable) (min, moyenne, max),
  - Taux de croissance,
  - Taux de changement,
  - Volume,
  - Critère de création, de suppression.
- Caractéristiques relationnelles :
  - Description des relations qui lient un enregistrement
    - aux programmes : type d'usage (recherche, mise à jour, etc.), méthode d'accès, séquence d'accès,
    - aux articles : format de mémorisation, position de départ, numéro de niveau cobol.

L'ensemble des relations que l'on crée entre les différents éléments forme, nous l'avons dit, la structure du dictionnaire de données.

#### I.4. ORGANISATION D'UN DICTIONNAIRE DE DONNEES

Etant donné l'importance de la fonction de mémorisation (en effet, le dictionnaire de données se situe au niveau d'un projet, d'une organisation ou d'une entreprise), le dictionnaire de données devra exister sur un support permettant l'exploitation de la collection de données mémorisées. Généralement, le dictionnaire sera implanté sous forme de base de données. Cet outil offre les avantages d'être manipulé à l'aide de langages dont les actions répondent

---

(1) British Rail : *Data Dictionary*

aux impératifs de la gestion d'un dictionnaire de données. Citons, par exemple, les actions de création, de suppression, de mise à jour, de consultation, etc., que possèdent généralement tous les langages de manipulation (encore appelés langages de commande) d'une base de données.

### I.5. USAGES DU DICTIONNAIRE DE DONNEES

Il n'est pas possible, dans le cadre de cette recherche, de relater en détail tous les usages possibles du dictionnaire de données. Tout au plus pourra-t-on concentrer l'attention du lecteur sur les deux grandes classes d'utilisations possibles : les utilisations par le management et les utilisations par les programmes.

#### I.5.1. Utilisation dans le management d'un projet

L'utilisation du dictionnaire dans la gestion est faite par deux catégories d'utilisateurs (1).

##### - Les utilisateurs indépendants du système informatique

Ceux-ci se servent uniquement de ses résultats. Ainsi, par exemple, le directeur du centre de traitement de l'information, qui a, entre autres fonctions, celle de planifier les différents projets, pourrait-il, par le biais du dictionnaire de données, avoir un contrôle sur leur état d'avancement.

##### - Les utilisateurs informaticiens

Ceux-ci créent et gèrent les opérations nécessaires à l'obtention de résultats en intervenant dans le système.

Ainsi l'analyste ou le programmeur voudront-ils savoir comment est décrite telle structure de données ?, où elle est déjà utilisée ?, etc.

Quelquefois des utilisations plus sophistiquées lui sont dévolues : c'est le cas de softwares écrits une fois pour toutes, destinés à réaliser de façon automatique certaines fonctions commandées par programmes.

#### I.5.2. L'utilisation par les programmes

Le programme utilise les informations contenues dans le dictionnaire

---

(1) UHROWCZIK, P.P. : *Data Dictionary/Directory*.

de données de quatre grandes façons (1) :

1. L'inclusion des données à la compilation.

Par exemple, la description d'article de la Data Division à l'intérieur des programmes COBOL peut être directement extraite du dictionnaire à l'aide d'un ordre semblable à la directive de compilation "COPY" du langage COBOL.

2. Génération automatique de modules standards d'entrée/sortie pour les fichiers conventionnels.

3. Ecriture directe de programmes de contrôle de validité.

4. Ecriture de programmes de tests de données.

A titre illustratif, nous synthétisons sur la figure I.2. différentes utilisations d'un dictionnaire de données (2).

#### I.6. L'ADMINISTRATEUR DES DONNEES

Pour coordonner le tout (l'objectif, le contenu, les usages) et assurer la cohérence du système, on a créé, voici peu, la fonction d'administrateur des données. On distinguera dans sa mission quatre grands groupes d'activités (3).

1. Activité de conception

Il revient normalement à l'administrateur de données de concevoir les standards de description des données ainsi que la structure du dictionnaire lui-même et de définir les règles d'utilisation du dictionnaire ainsi que celles qui peuvent assurer la sécurité du système.

2. Activité administrative

Par là, nous entendons toutes les activités concernant l'administration des standards de description des données incluant, par exemple, l'approbation pour conformité des descriptions nouvelles ainsi que l'assistance et la guidance des utilisations possibles du dictionnaire de données

---

(1) E.D.P. Analyser : *The Data Dictionary/Directory Functions*, pp. 7 - 13.

(2) UHROWCZIK, P.P. : *Data Dictionary/Directory*.

(3) E.D.P. Analyser : *The 'Data Administrator' Functions*, pp. 12 - 14.

					UTILISATEURS OCCASIONNELS
					MANAGEMENT DU DEPARTEMENT
					ANALYSTE SYSTEME
					PROGRAMMEUR
					PROGRAMME
		X			DEFINITION GENERALISEE DE DONNEES
X		X			DETECTION DE DONNEES <span style="font-size: 0.8em; vertical-align: middle;">{ REDONDANTES INCONSISTANTES</span>
		X	X		LIEU D'UTILISATION
	X	X			ENQUETE DE PLANNING
X	X	X	X		DOCUMENTATION COMMUNE
			X	X	GENERATION DE DEFINITION SOURCE DE LA DATA DIVISION
MA NA GE ME NT.				PR OG RA MM E.	

Figure 1.2. : Exemple d'utilisations d'un dictionnaire de données

### 3. Activité opérationnelle

Cette activité s'identifie à la création de la structure du dictionnaire lui-même, à l'initialisation et au contrôle des procédures qui en assurent l'intégrité et la sécurité.

### 4. Activité prospective

Elle permet la mesure des performances du système, le relèvement des redondances en vue d'effectuer une amélioration éventuelle du système.

L'action et les interventions de l'administrateur peuvent être schématisés au moyen de la figure I.3. (1).

En conclusion, ce premier chapitre nous a essentiellement servi à mettre à jour une "grille de lecture" des différents dictionnaires.

Cette grille commune n'est pas le fruit d'une décision arbitraire de notre part ; elle résulte de notre lecture approfondie de la littérature concernant les dictionnaires de données.

o  
o o

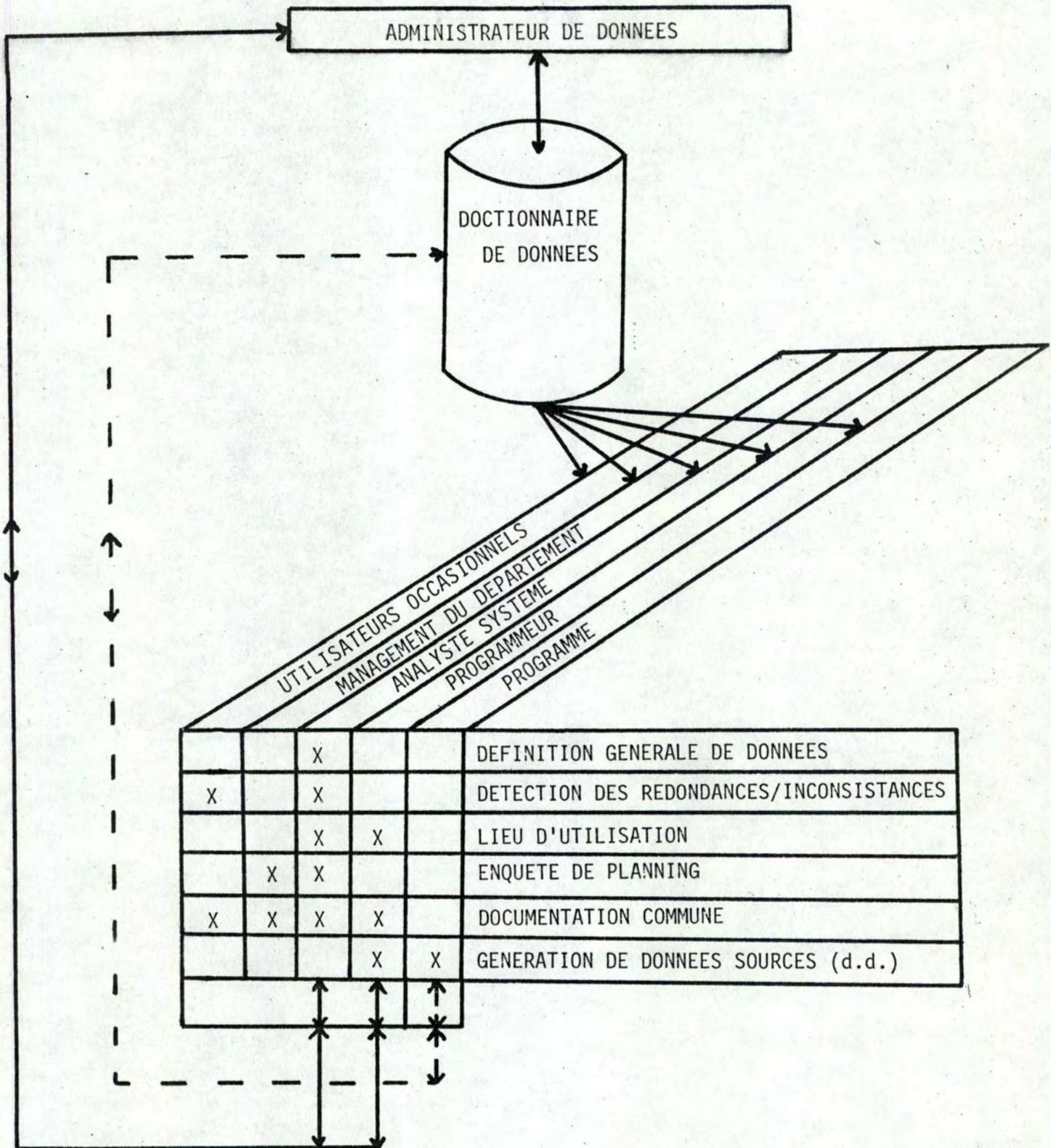


Figure 1.3. : Situation et rôle de l'administrateur de données

CHAPITRE II  
ETUDE DE L'EXISTANT

II.1. INTRODUCTION

Dans cette introduction, nous allons successivement

- préciser l'objectif du chapitre II ;
- présenter brièvement les dictionnaires sous étude ;
- rappeler la démarche suivie pour décrire les différents dictionnaires ;
- expliquer les symboles utilisés dans la description de la structure relationnelle des dictionnaires.

II.1.1. Objectif du chapitre II

Ce chapitre se propose d'effectuer une étude de l'existant en matière de dictionnaire de données. La démarche suivie dans l'analyse de chaque dictionnaire retenu est celle qui résulte du chapitre I. Ainsi sera permise ultérieurement une comparaison des contenus et des usages des différents dictionnaires. (1)

II.1.2. Les dictionnaires sous étude

Quatre dictionnaires de données composeront l'existant auquel nous nous référerons. Chacun de ces dictionnaires de données est représentatif d'un certain état de l'art en la matière.

Nous analyserons d'abord deux dictionnaires opérationnels :

- le dictionnaire de Datamanager, firme de software. Datamanager qui a créé un package sur les dictionnaires de données et l'a réalisé sous la forme d'un fichier ou d'une base de données.
- le dictionnaire de la British Rail, créé par une firme privée pour ses besoins propres et l'ayant implémenté sous la forme de fichiers conventionnels.

Nous nous pencherons ensuite sur un dictionnaire se situant encore au stade de la recherche : ce sera le dictionnaire mis en oeuvre dans le cadre du projet Rémora.

Pour terminer, nous étudierons le dictionnaire d'Uhrowczik, dictionnaire purement conceptuel et fruit d'une étude théorique d'un constructeur sur la question.

---

(1) L'utilisation d'une grille commune d'analyse des contenus permet une comparaison ultérieure aisée entre les différents dictionnaires. En revanche, elle enlève à chaque dictionnaire une partie de sa richesse. Le chapitre III - et particulièrement les § qui ont trait aux usages par le management et les programmes - comble, en bonne partie, cette lacune.

### II.1.3. Rappels sur la démarche

Conformément aux conclusions de notre premier chapitre, nous procéderons pour la description de chaque dictionnaire de données de la manière suivante :

#### a. Une analyse du contenu

A partir du schéma de la structure relationnelle du dictionnaire, nous identifierons et analyserons d'une part, les descripteurs de traitements et d'autre part, les descripteurs de données ainsi que les relations existant entre eux.

#### b. Une analyse des usages

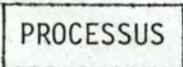
Nous nous efforcerons d'identifier et d'analyser les utilisations possibles de chaque dictionnaire à destination du management et à destination des programmes.

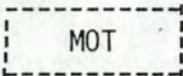
Etant donné la diversité et le manque d'uniformité de la documentation étudiée, la présentation que nous ferons de certains descripteurs (notamment pour Datamanager, et de certains usages (notamment pour Rémora) sera incomplète ou sujette à caution ; une note en bas de page en avertira le lecteur.

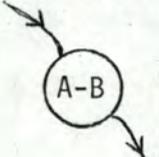
Dernière remarque : Nous ne nous intéresserons que partiellement aux langages de consultation de ces dictionnaires, car tel n'est pas l'objectif que nous poursuivons.

### II.1.4. Description de la structure relationnelle des dictionnaires

Pour exprimer la structure relationnelle d'un dictionnaire, nous aurons recours aux symboles suivants :

 = Entité du système informatique pour laquelle il existe un descripteur de traitement (en l'occurrence, PROCESSUS).

 = Entité du système informatique pour laquelle il existe un descripteur de données (dans ce cas, MOT).

 = Relation d'accès entre les entités du système A → B. A accède à B.

## II.2. CONTENU ET USAGES DU DICTIONNAIRE DE DATAMANAGER

### II.2.1. Introduction à l'analyse

#### a. Le but du dictionnaire

Le but de Datamanager est d'une part, de centraliser et de standardiser les descriptions de chaque unité individuelle de données dans une organisation et d'autre part, d'établir et de contrôler les relations entre toutes ces unités.

Datamanager est donc un software conçu pour

- agir comme véhicule de standardisation des descriptions de données ;
- agencer les relations existant entre les données et améliorer le contrôle de leurs utilisations ;
- offrir une documentation sur les traitements.

#### b. La documentation de Datamanager (1)

Il n'est pas possible de donner une liste complète des caractéristiques attachées à chaque descripteur de traitement ou de données, car Datamanager est un software commercialisé, adaptable aux diverses organisations où il doit s'intégrer. Dès lors, les caractéristiques intrinsèques varient selon les désirs des organisations dans lesquelles se trouve Datamanager.

#### c. Remarque sur l'implémentation

L'implémentation de Datamanager peut être réalisée sous la forme d'une base de données. De ce fait, et grâce à un interface avec des moniteurs de télé-traitement, Datamanager permet à tout utilisateur de travailler à tout moment sur le dictionnaire de données.

### II.2.2. Analyse du contenu du dictionnaire

#### a. Structure du dictionnaire

La structure relationnelle du dictionnaire de Datamanager se présente comme sur le schéma 2.2. ; on y reconnaît :

- les descripteurs de traitements : programme, système et module ;
- les descripteurs de données : fichiers, groupes de données et données élémentaires.

---

(1) Management Systems and Programming Ltd.: *Datamanager Product Description*.

b. Etude des descripteurs du dictionnaire

b.1. Les descripteurs de traitements

Pour aucun des descripteurs de traitements, il ne nous sera possible (1) de donner d'autres caractéristiques que les caractéristiques relationnelles.

LE MODULE

Définition

Le module est une combinaison de fichiers et/ou de groupes et/ou de données élémentaires sur lesquelles s'effectue un traitement qui accomplit un travail spécifique au sein d'un projet. Ce sera, par exemple, le calcul du montant dû à l'employé pour la prestation de ses heures supplémentaires.

Exemple de définition possible de module

A la demande : "Ajouter un module de lecture du fichier Personnel".

NOM DU MODULE : lec. fich. pers. ;

INPUT : fichier personnel ;

OUTPUT : groupe personnel ; indicatif de fin de fichier ;

APPEL DE : "contrôle du chiffre de contrôle" ;

BIBLIOTHEQUE : Input-output, COBOL ;

REMARQUES : Après avoir lu les groupes et après avoir fait le contrôle du chiffre de contrôle, il passe en retour au programme le chiffre 8, si le groupe est invalide, ou 9, si c'est la fin du fichier. Le chiffre de contrôle est un chiffre joint à chaque groupe et permettant de montrer l'appartenance du groupe à un fichier particulier ;

FIN.

Caractéristiques relationnelles

On voit sur le graphe que le module a des relations

- avec le groupe. En effet, un module travaille sur des données élémentaires et sur des groupes de données.
- avec les programmes. En effet, un module s'intègre dans une chaîne de modules formant un programme.
- avec d'autres modules, car il existe des modules-maîtres auxquels sont liés des modules-esclaves (chaîne des appels).

(1) cfr. la remarque p. 16, § b.

STRUCTURE RELATIONNELLE DU DICTIONNAIRE DE DATAMANAGER

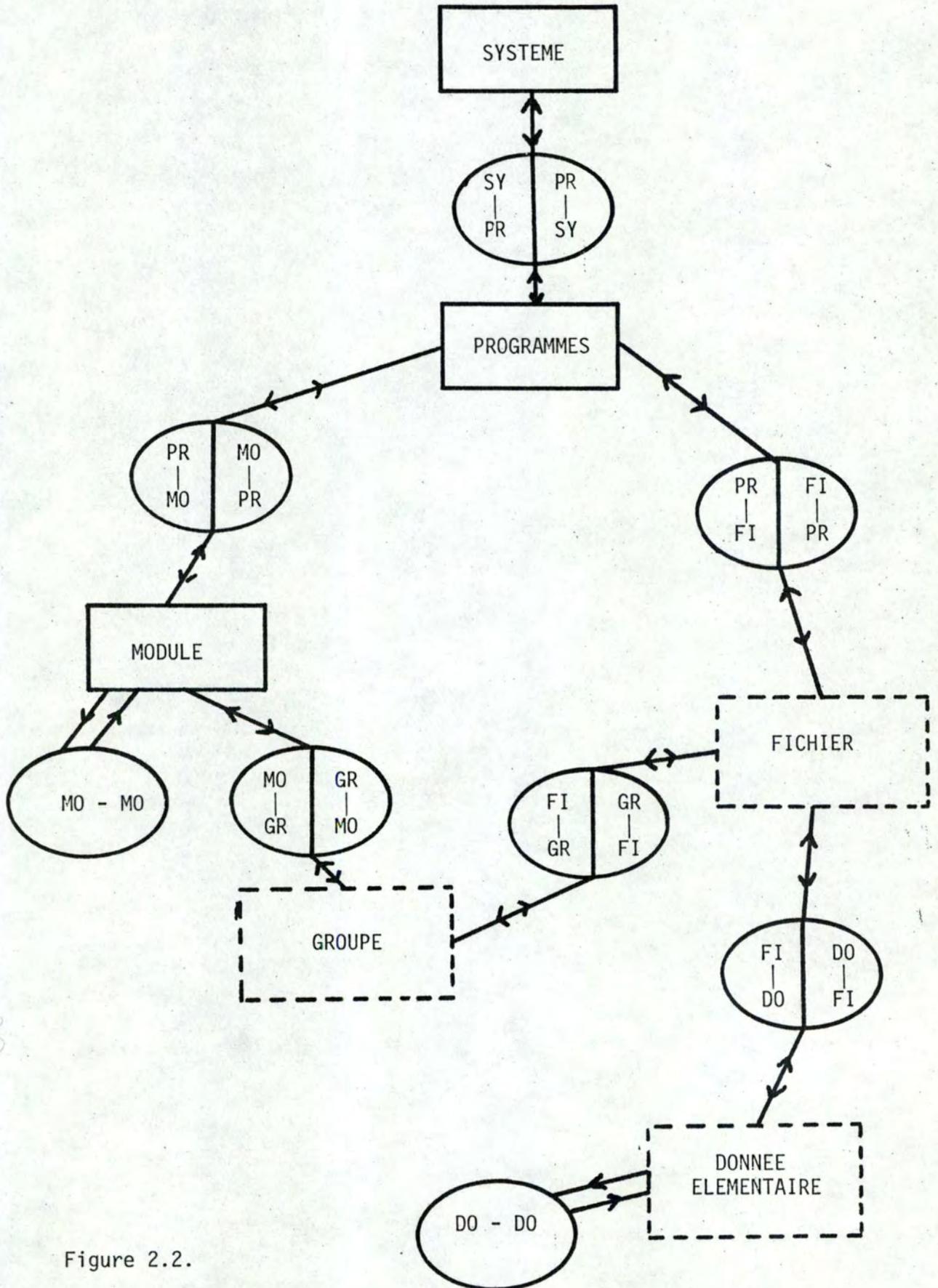


Figure 2.2.

## LE PROGRAMME

### Définition

C'est une combinaison de fichiers et modules formant une unité opérationnelle indépendante. Nous citerons, pour exemple, le programme des salaires et traitements du personnel d'une compagnie.

### Caractéristiques relationnelles

Ainsi qu'il apparaît sur le graphe (2.2.) le programme est en relation avec

- les modules : en effet, un programme est la combinaison de plusieurs modules.
- les fichiers : en effet, au niveau programme, il faut spécifier l'ensemble de données sur lequel les programmes vont travailler.
- le système, car un programme représente un secteur d'activité du système d'information.

## LE SYSTEME

### Définition

Pour Datamanager, un système est un ensemble de fichiers et de programmes qui concourent à satisfaire tout un secteur d'une organisation. Nous citerons, dans le cas d'une entreprise, le système "Gestion du personnel", "Gestion des stocks".

### Caractéristiques relationnelles

Le graphe (2.2.) nous montre l'existence d'une relation entre un système et les programmes qui le composent.

### b.2. Les descripteurs de données

## LA DONNEE ELEMENTAIRE

### Définition

Elle est un atome élémentaire d'information. La description d'une donnée élémentaire peut inclure des caractéristiques telles que son format, son type, etc.

Caractéristique identifiante

- Le nom

Caractéristiques intrinsèques

- Le format d'entrée,
- Le format de traitement,
- Le dessin d'édition,
- Le type de donnée,
- La longueur,
- La précision,
- Le contenu actuel et prévisible.

Caractéristiques relationnelles

La donnée élémentaire est en relation avec

- Les groupes et les fichiers dont elle fait partie ;
- d'autres données élémentaires, par exemple : date = JJ + MM + AA.

Exemple de description possible de donnée élémentaire

- NOM DE LA DONNEE ELEMENTAIRE : montant ;
- FORMAT D'ENTREE : TYPE DE DONNEE : numérique ;  
LONGUEUR : 7 ;  
CONTENU ACTUEL : 000 000 1 ;  
CONTENU PREVISIBLE : 999 999 9 ;
- FORMAT DE TRAITEMENT : TYPE : décimal packé ;  
FORMAT : 5.2 ;  
CONTENU ACTUEL : 0.01 ;  
CONTENU PREVISIBLE : 99999.99 ;
- DESSIN D'EDITION : FORMAT : \$\$\$\$9.99 ;  
CONTENU ACTUEL : \$ 0.01 ;  
CONTENU PREVISIBLE : \$ 99999.99 ;
- REMARQUE : Cette description montre comment une donnée élémentaire peut différer dans son apparence, lorsqu'elle passe à travers le système de l'input vers l'output ;
- FIN.

## LE GROUPE

### Definition

C'est une combinaison de données élémentaires et/ou de groupes. Cela permet à des atomes élémentaires d'information d'être logiquement groupés en enregistrements, sous-enregistrements ou groupes d'information.

### Caractéristique identifiante

- Le nom

### Caractéristiques intrinsèques (1)

#### Caractéristiques relationnelles

Le groupe est en relation avec

- les fichiers dont il fait partie, et les modules qui l'utilisent ;
- les données élémentaires qui le constituent.

## LE FICHIER

### Définition

C'est une combinaison de groupes et/ou de données élémentaires.

### Caractéristique identifiante

- Le nom

### Caractéristiques intrinsèques (1)

#### Caractéristiques relationnelles

Le fichier est en relation avec

- les programmes et les modules qui y puisent les informations nécessaires à leur exécution.
- le groupe et la donnée élémentaire qui sont les constituants du fichier.

---

(1) non spécifié.

### II.2.3. Analyse des usages du dictionnaire

#### a. A destination du management

Deux types distincts d'information sont fournis aux utilisateurs.

##### a.1. Les rapports sur l'état d'un élément du dictionnaire

Ce sont des rapports imprimés contenant toutes les informations sur un membre spécifique de la structure d'un système, tel que la donnée élémentaire, le groupe, etc.

Ces rapports font également apparaître les relations que possède ce membre spécifique avec les autres membres de la structure.

A titre d'exemple, voici le rapport sur l'état d'un module :

<u>RAPPORT DU MODULE PMACSDATE</u>	
1 :	EDITION 1 : enregistré par le propriétaire alpha le 02/01/76.
2 :	PROTECTION : aucune.
3 :	NOTE : Accepte jour, mois et année et fournit une date pour l'impression et les traitements.
4 :	LE MODULE : a en <i>INPUT</i> : les données élémentaires TRA-PCFDATE/TRA-PCS-DATE ne met à jour aucune donnée élémentaire. a en <i>OUTPUT</i> : le groupe standard : DATE appelle les modules PMCAS-SDC et PMCAS-DP.
5 :	FIN

##### a.2. Réponse à des questions sur les relations entre éléments du dictionnaire

Lorsque l'on veut, par exemple, déterminer les effets d'un changement de caractéristique d'une entité, il est intéressant de savoir quelles sont les répercussions de ce changement sur les autres entités du dictionnaire.

Aussi, à la question "Qui utilise cette entité ?", Datamanager répondra par une liste de données élémentaires, de groupes, de fichiers, de modules, et systèmes qui possèdent des liens avec l'entité modifiée. A l'utilisateur de constater l'impact de la modification qu'il veut apporter et, de prendre ensuite les mesures adéquates !

D'autres questions sont susceptibles d'être posées, qu'il serait trop long d'énumérer en détail. Nous citerons par exemple :

- quels sont les utilisateurs qui ont accès à une entité spécifique ?
- quelles sont les entités-esclaves d'une entité-maître au sein du dictionnaire ? (voir écran ci-après)

QUESTION : QUELS PROGRAMMES UTILISENT LA DONNEE ELEMENTAIRE  
"LONGUEUR-D'ENREGISTREMENT-STANDARD" ?

REPONSE : les programmes suivants utilisent la donnée élémentaire  
"longueur-d'enregistrement-standard" :  
+ PMACS-ACTIVITE-DE-M.A.J.-DES-GROUPES ;  
+ PMACS-M.A.J.-DES-PROJETS ;  
+ PMACS-IMPRESSIION ;  
+ PMACS-ANALYSE ;  
+ PMACS-INITIALISATION,

QUESTION : QUI UTILISE LE FICHER PMACS-RESSOURCE ?

REPONSE : le fichier est utilisé par les programmes: PMACS-ANALYSE ;  
PMACS-M.A.J.-RESSOURCES .  
le programme PMACS-M.A.J.-RESSOURCES est utilisé par le SYSTEME,  
LE MANAGER et LE CONTROLEUR DU PROJET.  
le programme PMACS-ANALYSE est utilisé ...

DATAMANAGER peut donner la chaîne complète

Exemple de dialogue entre Datamanager et un utilisateur

## b. A destination des programmes

### b.1. Génération de langage source

A partir des descriptions de données contenues dans le dictionnaire, il est possible de générer automatiquement des descriptions de données dans un langage de programmation choisi (COBOL, RPG, PL/1, etc.).

### b.2. Génération de fichiers contenant des données-tests

A partir d'une combinaison de données spécifiées par l'utilisateur d'une part, et des descriptions contenues dans le dictionnaire d'autre part, il est possible de générer des fichiers structurés contenant des données-tests.

### b.3. Impression du contenu d'un fichier

Le fichier ainsi que ses groupes et ses données élémentaires sont décrits au sein du dictionnaire. Il est dès lors possible d'utiliser ces renseignements pour initialiser et réaliser l'impression du fichier.

Le software associé au dictionnaire travaille logiquement de la manière suivante :

1ère étape : *Consultation dans le dictionnaire de la description du fichier.*  
Cela procure un certain nombre de renseignements tels que le format du fichier, son volume, ses liens avec les groupes ou les données élémentaires qui le composent, sa localisation physique, etc.

2ème étape : *Impression du fichier.*

A partir des caractéristiques qu'il a trouvées dans le dictionnaire, le module peut être initialisé et le vidage du fichier se réaliser.

### b.4. Génération de modules d'entrée/sortie

A partir de descriptions de fichier, on pourra produire un module assemblable pour gérer les entrées/sorties des fichiers. Le module observera les conventions standards d'interface ; il pourra donc être appelé par un module écrit dans d'autres langages de programmation.

### b.5. Mise à jour automatique du dictionnaire

A partir de la description de données dans un langage de programmation, (COBOL ou PL/1), Datamanager peut automatiquement générer un descripteur de données dans son propre dictionnaire.

Supposons, par exemple, que dans la Data Division d'un programme COBOL, on ait écrit : 01 NOMBRE PIC 99 'DIC "DONNEE ELEMENTAIRE" '

La phrase : 'DIC "DONNEE ELEMENTAIRE" ' incorporée dans le programme COBOL permet la génération automatique au sein du dictionnaire d'une description de la donnée élémentaire "NOMBRE", à la suite des autres descriptions de données élémentaires.

#### b.6. Auto-contrôle des commandes de création, de manipulation, de sécurité

Le software de gestion du dictionnaire effectue un contrôle des manipulations que l'utilisateur veut effectuer sur le dictionnaire et, sous forme de rapports édités, signifie à l'utilisateur la validité de sa demande ou son incorrection. Dans ce dernier cas,

- ou bien, il lui pose des questions afin de rendre sa demande conforme au standard ;
- ou bien, il l'annule.

### II.3. CONTENU ET USAGES DU DICTIONNAIRE DE LA BRITISH RAIL

#### II.3.1. Introduction à l'analyse

Le but de ce dictionnaire de données est d'être un répertoire central de descriptions d'informations utilisées par un projet, une compagnie, une organisation.

A la différence de Datamanager, ce dictionnaire a été créé pour un besoin propre de la firme. En raison de son succès, il a été commercialisé.

#### II.3.2. Analyse du contenu du dictionnaire (1)

##### a. Structure du dictionnaire

La structure relationnelle du dictionnaire de la British Rail se présente comme sur le schéma 2.3. ; on y reconnaît

- le descripteur de traitement : le programme,
- les descripteurs de données : la donnée élémentaire, l'enregistrement et le fichier,
- le descripteur de l'environnement.

---

(1) British Rail : *Data Dictionary*.

## b. Etude des descripteurs du dictionnaire

### b.1. Le descripteur de traitement

#### LE PROGRAMME

##### Définition

Un programme identifie tout traitement effectué sur des données, sans discrimination de niveau. En effet, là où Datamanager distingue des modules, des programmes, des systèmes, le dictionnaire de la British Rail reconnaît uniquement des programmes.

##### Caractéristiques identifiantes

- le nom du programme,
- le numéro de la version et la date,
- une description informelle.

##### Caractéristiques intrinsèques

- le langage de programmation dans lequel le programme est écrit,
- la fréquence d'utilisation,
- sa taille mémoire,
- son type (validation, édition, suppression, etc).

##### Caractéristiques relationnelles

Le programme est en relation avec

- les enregistrements et les données élémentaires dont il se sert pour son exécution.

Les paramètres essentiels que nécessite, par exemple, la relation *programme-données élémentaires* sont

- le type d'utilisation de la donnée (en consultation, etc.) ;
- la description de son utilisation (programme va chercher la "date" pour dater le travail.) ;
- les détails permettant les contrôles de validité (date à 6 caractères JJ MM AA, etc.) ;
- la fréquence d'utilisation de chaque donnée élémentaire.

STRUCTURE RELATIONNELLE DU DICTIONNAIRE DE LA BRITISH RAIL

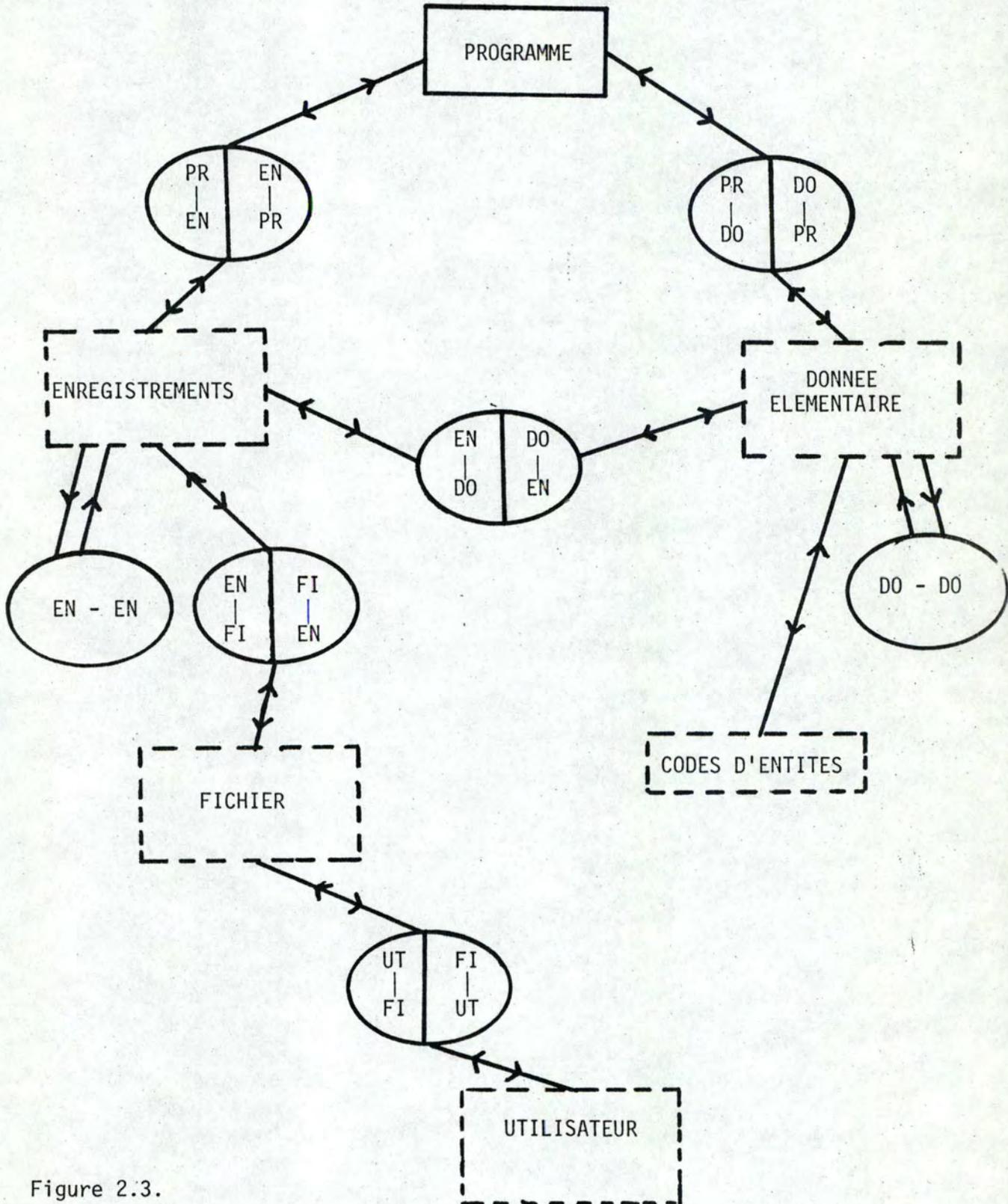


Figure 2.3.

## b.2. Les descripteurs de données

### LA DONNEE ELEMENTAIRE

#### Définition

La donnée élémentaire est l'élément central de la structure, car tous les autres descripteurs de données ne font que décrire des groupements de ces données élémentaires. Ces données élémentaires peuvent représenter 'des codes' ; dans ce cas, des dispositions sont prises pour mémoriser les valeurs et les descriptions des codes associés.

#### Caractéristiques identifiantes

- le nom usuel,
- le nom du "propriétaire",
- les synonymes,
- la version et la date,
- son numéro,
- le nom de l'enregistrement dont elle est composante,
- une description informelle,
- une description formelle.

#### Caractéristiques intrinsèques

- le type de donnée élémentaire (texte, code, etc.),
- caractéristiques de longueur (fixe, variable, minimum, maximum, moyenne),
- le type de code,
- le code de référence (standard),
- les unités,
- les valeurs nulles,
- les valeurs par défaut,
- l'en-tête de rapport,
- les critères de création et de destruction,
- les nécessités d'archivage,
- les valeurs pour le contrôle d'intégrité.

#### Caractéristiques relationnelles

Il existe une relation avec

- les programmes, lien inverse du lien programme-donnée élémentaire ;

- les enregistrements, dont les données élémentaires sont constitutives ;
- les codes ou tables de codes qui codifient ces données élémentaires ;
- d'autres données élémentaires, car il peut exister une relation entre deux données élémentaires. Les facteurs de cette dernière liaison sont :
  - le type de relation (par exemple : les relations d'appartenance) ;
  - une description informelle (le jour  $\in$  au mois  $\in$  à l'année).

## L'ENREGISTREMENT

### Définition

L'enregistrement est le regroupement de plusieurs données élémentaires. Ce regroupement permet de donner un nom commun à un ensemble de données élémentaires.

### Caractéristiques identifiantes

- le nom usuel,
- les noms des données élémentaires qui le constituent,
- le numéro de version et la date,
- le nom d'assemblage,
- la description informelle.

### Caractéristiques intrinsèques (1)

#### Caractéristiques relationnelles

Il y a une relation avec

- le programme, c'est le lien inverse de la relation programme-enregistrement ;
- les données élémentaires. Les caractéristiques associées à cette relation sont :
  - le format de mémorisation,
  - la longueur en mémoire,
  - le numéro de niveau COBOL,
  - l'alignement à droite et/ou à gauche.
- les autres enregistrements dont les caractéristiques sont :
  - le type de relation (de maître à esclave ou inverse),
  - un numéro de lien hiérarchique,
  - une description informelle.
- le fichier qu'ils constituent.

---

(1) non spécifié.

## LE FICHIER

### Définition

Des types d'enregistrements peuvent être reliés entre eux pour former un fichier.

Par exemple, soit les types d'enregistrements "NOM" (regroupant toutes les "données élémentaires" d'une personne comme son nom, prénoms, âge et adresse) et "BAREME" (regroupant les informations relatives aux barèmes des personnes). Si on rassemble ces deux enregistrements pour toutes les personnes de l'entreprise, on constitue, par exemple, le fichier "PERSONNE".

### Caractéristiques identifiantes

- le nom usuel,
- le nom des enregistrements constituant le fichier,
- la date et le numéro de version,
- une description informelle.

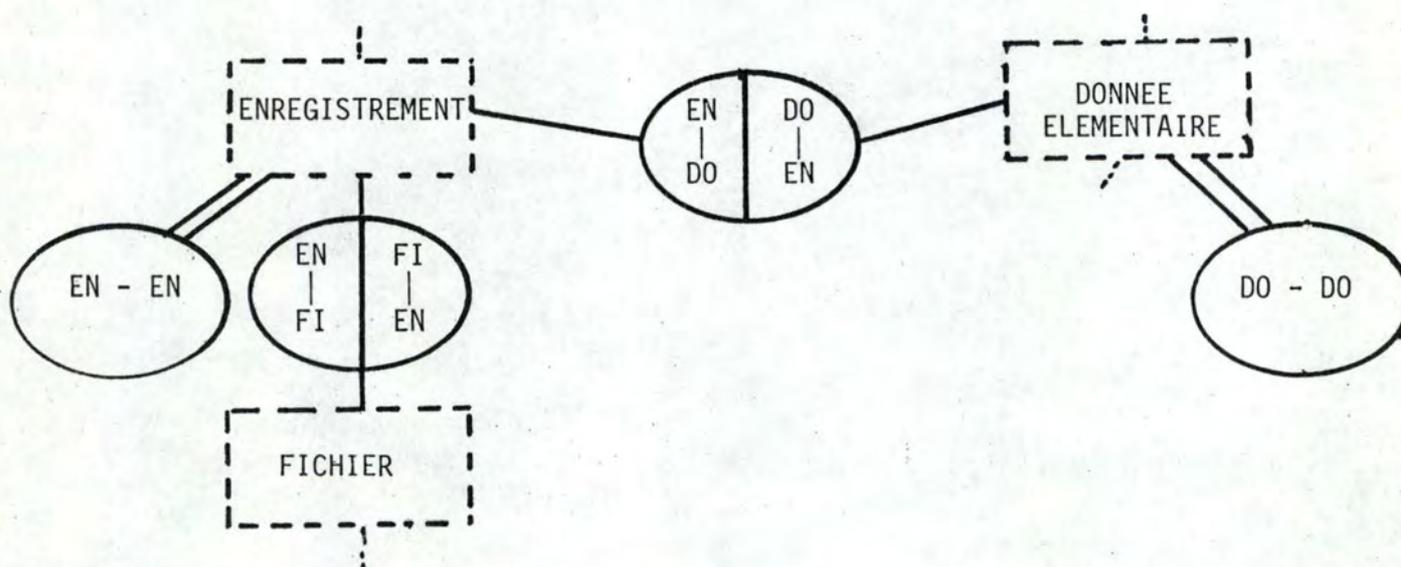
### Caractéristiques intrinsèques

- l'organisation du fichier,
- le support,
- la localisation physique (endroit de mémorisation),
- les besoins d'archivage.

### Caractéristiques relationnelles

Il y a une relation avec

- les enregistrements qui constituent le fichier,
- les utilisateurs qui emploient le fichier.



## L'UTILISATEUR

### Définition

Les utilisateurs qui interrogent le système de documentation, ou qui sont à l'origine des documents d'entrée, peuvent être décrits dans le dictionnaire de données. Chaque utilisateur est relié aux fichiers des états imprimés qu'il peut demander et chaque type d'état imprimé est relié à ses différents utilisateurs.

### Caractéristiques identifiantes

- Le nom de l'utilisateur,
- La spécification de ses responsabilités.

### Caractéristiques intrinsèques (1)

#### Caractéristiques relationnelles

Il y a une relation avec

- les fichiers des états édités qu'il est susceptible de demander. Les caractéristiques de cette relation sont :
  - une description informelle des états auxquels on veut se référer.
  - le nombre de copies.

### II.3.3. Analyse des usages du dictionnaire

#### a. A destination du management

Quatre grands types de rapports sont produits.

##### a.1. Les rapports décrivant l'état du dictionnaire

Ils ont pour but de donner des informations détaillées sur les caractéristiques des données utilisées. On trouve dans cette catégorie un rapport décrivant les caractéristiques d'une donnée élémentaire, d'un enregistrement, d'un fichier ou d'un programme.

##### a.2. Les rapports énumérant des endroits d'utilisation d'une entité de la structure relationnelle

Par exemple - et ceci est aussi vrai pour toutes les relations du graphe (2.3.) -, le rapport décrivant les relations entre les données élémentaires

---

(1) non spécifié.

et les enregistrements donne toutes les informations sur toutes les données élémentaires constitutives d'un enregistrement donné.

### a.3. Glossaire pour l'utilisateur

Comme son nom l'indique, ce glossaire permet, à la demande, de constituer, pour un utilisateur, un répertoire particulier des descriptions de données qu'il emploie.

### a.4. Les listings des codes

Cet état imprimé donne la liste des codes associés aux données élémentaires ainsi que les significations de leurs valeurs.

#### EXEMPLE DE LISTING DE DONNEES ELEMENTAIRES

CAR ID	: DONNEE ELEMENTAIRE (NOM)	: fréquence ;
	NOM USUEL	: fréquence de traitement ;
	PROPRIETAIRE	: Alpha ;
	DESCRIPTION INFORMELLE	: indique la fréquence du passage des programmes ;
CAR IT	: TYPE DE LONGUEUR	: fixe ;
	LONGUEUR MAXIMALE	: 005 ;
	VALEUR REQUISE	: aucune ;
	TYPE DE CODE	: code par table ;
	VALEUR NULLE	: espaces ;
	EN-TETE	: "Fréquence de Traitement".
CAR RL	: APPELLE PAR L'ENTITE	: programme.

#### EXEMPLE DE LISTING DE FICHER

CAR ID	: FICHER (NOM)	: route-fichier ;
	DESCRIPTION INFORMELLE	: Ce fichier contient les détails des routes suivies par les trains. Le dernier enregistrement est suivi par un enregistrement dont la valeur de clé est maximale. Une copie du fichier réside en permanence en mémoire centrale ;
CAR IT	: ORGANISATION	: séquentielle, enregistrements triés selon un ordre spécifique.
	TYPE	: maître
	MEMORISE SUR	: disque.

CAR RL : RELIE A

1. CAR ID NOM D'ENREGISTREMENT : enregistrements-route ;  
DESCRIPTION INFORMELLE : l'enregistrement contient les codes ou numéros des routes et noms des stations pour chaque direction de la route ; un code = 0 indique que l'enregistrement est effacé ;
2. CAR ID NOM D'ENREGISTREMENT : rtman ;  
DESCRIPTION INFORMELLE : enregistrement d'en-tête pour le route-fichier utilisé par le moniteur.

b. A destination des programmes

b.1. Génération automatique de descriptions de données en langage COBOL

A partir d'un ordre semblable à la directive de compilation "COPY", des descriptions de données présentes dans le dictionnaire peuvent être extraites, reformatées et incluses dans un programme COBOL de façon dite "transparente" à l'utilisateur.

b.2. Auto-enrichissement des standards de description

Dès qu'une donnée se trouve décrite dans le dictionnaire, elle y est décrite selon un moule standard. Or, cette donnée subit des modifications au cours de sa vie dans le système. Ces modifications doivent se répercuter au niveau de la description de la donnée dans le dictionnaire, afin que celui-ci soit toujours correct. Le logiciel associé au dictionnaire réalise automatiquement ces modifications, dès que la donnée se trouve modifiée dans le système, de la même façon que le logiciel associé à Datamanager (voir p. 24).

II.4. CONTENU ET USAGES DU DICTIONNAIRE ASSOCIE AU PROJET REMORA

II.4.1. But du projet, rôle du dictionnaire

Le projet Rémora a pour objectif la conception et la réalisation d'un système de pilotage des projets informatiques.

Ce système s'appuie sur des outils qualifiés d'orthogonaux, parce qu'ils peuvent être utilisés isolément, ponctuellement ou, au contraire, combinés les uns aux autres selon la volonté et le besoin des utilisateurs.

En particulier, leur combinaison permet, dans un ordre déterminé, de passer sans heurt dans la conduite d'une application informatique, de l'analyse à la programmation et à la maintenance, en assurant une automatisation maximale des tâches par la génération des modules de traitement, la déduction automatique des classes de données et la génération de leurs descriptions dans un dictionnaire de données appelé "BASE DES OBJETS ELEMENTAIRES", que nous allons présenter.

Ce projet est encore à l'étude. Si ses principes et ses objectifs sont définis, les tests, eux, sont seulement en cours.

#### II.4.2. Analyse du contenu du dictionnaire (1)

##### a. Structure du dictionnaire

La structure relationnelle du dictionnaire associé à Rémora se présente comme sur le schéma 2.4. ; on y reconnaît :

- les descripteurs de traitement : application, procédure, module ;
- les descripteurs de données : structures, mots.

##### b. Etude des descripteurs du dictionnaire

###### b.1. Les descripteurs de traitement

### LES PROCEDURES

#### Définition

On appelle procédure un ensemble de traitements d'informations donnant réponse à un événement tendant à une fin utile, et d'exécution assez fréquente pour qu'on ait pu en arrêter les modalités.

Par exemple, une procédure correspond à une activité de gestion, comme la paie du personnel ou la gestion d'un stock de marchandises, qui est déclenchée par un événement tel que la fin du mois ou la rupture de stock.

#### Caractéristiques identifiantes

- Identificateur de procédure qui vérifie la règle d'unicité ;
- une désignation ou un titre ;
- un résumé précisant la finalité ;
- la périodicité ou la fréquence.

(1) ROLLAND, C. et al. : *La base des objets élémentaires de Rémora.*

STRUCTURE RELATIONNELLE DU DICTIONNAIRE DE REMORA

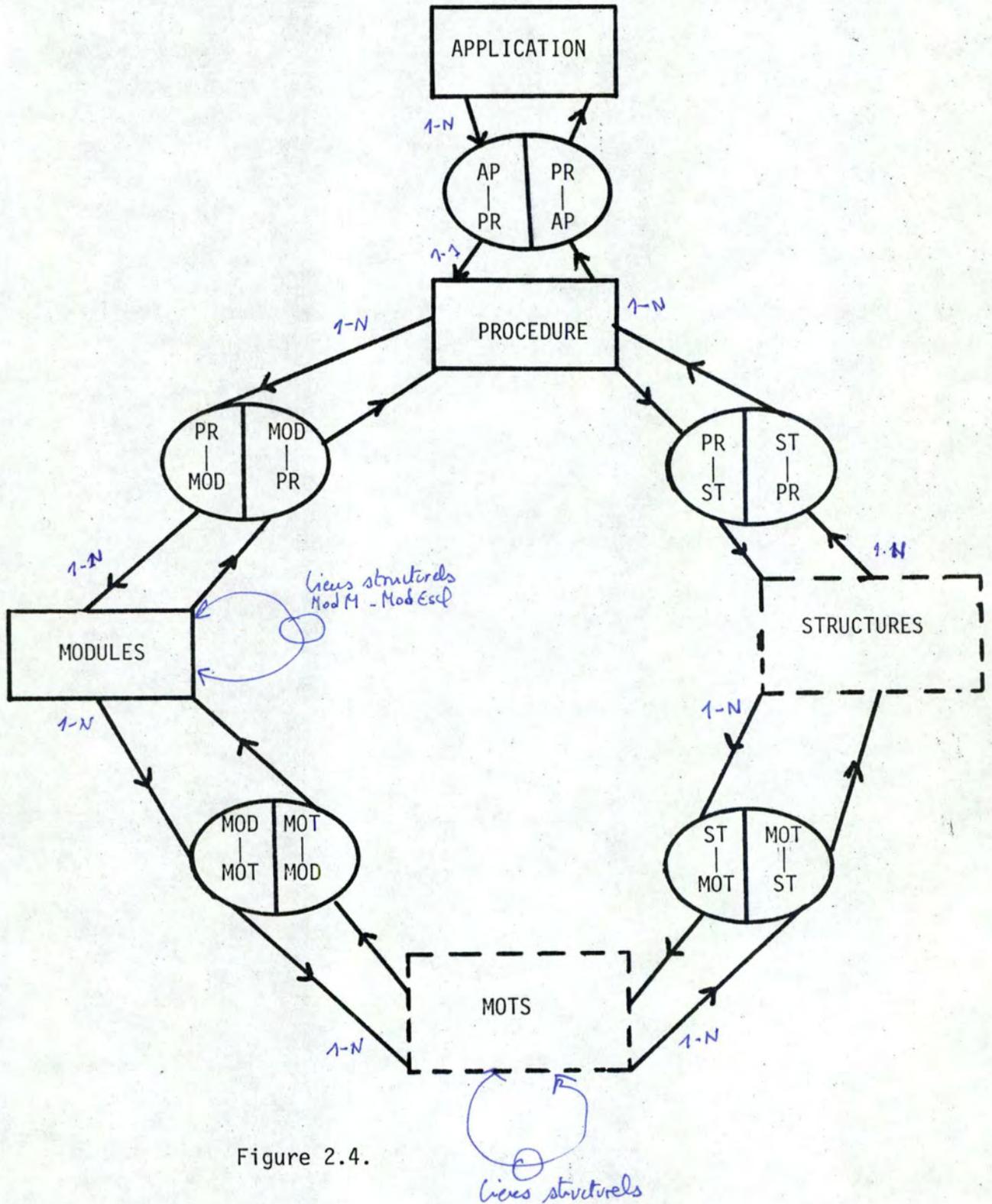


Figure 2.4.

Caractéristique intrinsèque

Aucune

Caractéristiques relationnelles

Elles concernent les liens entre la procédure et les autres objets.

Il existe, d'une part, deux liens hiérarchiques inférieurs entre la procédure et les modules, et entre la procédure et les structures. Ce sont des liens (1) - (N) car une procédure est composée de plusieurs modules et utilise un certain nombre de lots d'informations qui sont représentés par les structures.

Il existe, d'autre part, un lien hiérarchique supérieur entre la procédure et l'application. On fait l'hypothèse qu'une procédure appartient à une seule application.

L'APPLICATION
---------------

Définition

L'application est composée d'une famille de procédures appartenant à un même domaine d'activité de l'entreprise, traitant d'informations concernant une même population, déclenchée par des événements de même nature et nécessaires pour l'obtention de mêmes sorties. Elle est définie par l'unicité des informations que ses procédures manipulent.

Par exemple, la gestion du personnel est une application dont fait partie la procédure 'paie'.

Caractéristiques identifiantes

- l'identificateur qui est le nom de l'application et qui vérifie la règle d'unicité ;
- une désignation ou un titre ;
- un résumé expliquant l'objectif de chacune des procédures qui la composent.

Caractéristique intrinsèque

Aucune.

Caractéristiques relationnelles

Il y a un lien hiérarchique inférieur avec la procédure. A une application sont liées une ou plusieurs procédures.

## LE MODULE

### Définition

Le module de traitement est un sous-ensemble de la règle du jeu de la procédure.

Par exemple, le calcul du montant dû aux heures supplémentaires prestées est un module appartenant à la procédure de la paie du personnel.

### Caractéristiques identifiantes

- un identificateur qui obéit à la règle d'unicité car il est permis d'utiliser un même module dans différentes procédures ;
- une désignation en clair ou titre ;
- un résumé des traitements effectués par le module ;
- les paramètres du module lorsqu'ils existent (maître-esclave, etc.) ;
- des noms synonymes, si le module est utilisé dans plusieurs procédures.

### Caractéristique intrinsèque

Aucune.

### Caractéristiques relationnelles

Un module est lié par un lien hiérarchique supérieur avec la procédure (inverse du lien procédure-module). Il peut exister des traitements communs à plusieurs procédures dont la description sous forme de module ne sera donnée qu'une seule fois.

Un module est ensuite en liaison avec les mots par un lien hiérarchique inférieur.

Les modules ne sont pas indépendants entre eux, ils sont liés par un lien structurel. Ce lien indique pour un module-maître tous les modules esclaves qu'il appelle et inversement.

### b.2. Les descripteurs de données

## LE MOT

### Définition

Le mot repère une entité intervenant dans une application. L'ensemble des

mots constitue le vocabulaire de l'application. C'est celui créé par l'utilisateur lors de la description de son application.

#### Caractéristiques identifiantes

- l'identificateur qui n'obéit pas forcément à la règle d'unicité ;
- une désignation en clair exprimant le rôle ou la signification du mot.

#### Caractéristiques intrinsèques

- la taille, c'est-à-dire la précision ;
- le type (numérique, alphabétique, etc.).

#### Caractéristiques relationnelles

Il existe un lien hiérarchique supérieur avec les modules, c'est un lien (1) - (N), un mot pouvant apparaître dans plusieurs modules.

Il existe aussi un lien hiérarchique supérieur avec les structures, également de la forme (1) - (N), un mot pouvant appartenir à plusieurs documents de sortie et donc plusieurs structures.

Il existe des liens structurels inter-mots.

### LA STRUCTURE

#### Définition

Une structure repère un ensemble d'entités apparaissant, en sortie de l'application, sur un même document. Le système élabore des structures logiques représentatives à la fois du contenu et de la forme du document (pour l'édition). Il élabore aussi des structures logiques de travail, en formatant les données circulant entre les divers modules de traitement.

#### Caractéristiques identifiantes

- l'identificateur de structure qui obéit à la règle d'unicité ;
- la désignation en clair.

#### Caractéristiques intrinsèques

Type de structures. Une structure est soit une

- structure d'édition : associe un document obtenu comme résultat de l'application
- structure de travail : associée à une unité opérationnelle. Elle contient toutes les déclarations de résultats.

- structure logique : générée par les opérateurs de classification (1).
- texte objet : déclaration de la structure en langage de programmation (COBOL, PL/1, etc.).

#### Caractéristiques relationnelles

Ce sont les liens inverses de ceux déjà repérés entre procédure et structure d'une part, structure et mots d'autre part.

Les liens vers la procédure sont des liens hiérarchiques inférieurs de type (1) – (N), puisque nous rattachons la structure au niveau application. Une même structure peut être utilisée par plusieurs procédures.

Le lien structure-mot est un lien hiérarchique supérieur de type (1) – (N), une même structure étant par définition composée de plusieurs mots.

#### II.4.3. Analyse des usages du dictionnaire (2)

Nous ne dissociérons pas, cette fois, l'usage du dictionnaire fait dans le management et dans les programmes. La raison en est que ces deux usages sont intimement liés. En effet, l'usage du dictionnaire dans les programmes fournit l'usage dans le management et inversement.

L'idée-force du projet est une tentative de normalisation des faits de gestion (point de vue management) pour en obtenir une traduction automatique (point de vue programme).

Le concepteur travaille en mode déclaratif, c'est-à-dire qu'il se contente de décrire sur un descripteur ce qu'il veut voir se réaliser. Il décrit les actions dans l'ordre où il les pense, à l'image des mécanismes de sa pensée selon une méthode de conception descendante (ainsi en partant des résultats qu'il désire obtenir, il définit par étapes une solution conduisant à ces résultats).

A partir de cet instant un traducteur utilisant la charpente pré-établie du descripteur traduit ce descripteur en instructions et génère les modules de traitement.

---

(1) ROLLAND, C. et al. : *Engineering of Management Information System*.

En bref, c'est un software qui répartit les données en classes homogènes. Trois types de classes existent : les données en tant que telles, les résultats intermédiaires, les résultats.

(2) Tous les articles de ROLLAND, C et al. sur le projet REMORA.

Pour ce faire, un gestionnaire tient à jour un dictionnaire de données où sont rangées toutes les informations précisées sur les descripteurs. Il les classe et crée des relations entre les descripteurs d'informations. Ce dictionnaire appelé "Base des objets élémentaires" sert au générateur de programmes qui y puise les renseignements nécessaires pour l'élaboration de modules. Dès que le module est établi par Rémora, les résultats sont utilisés par le management du projet pris en compte. De plus, comme nous le voyons, le dictionnaire est vraiment un outil centralisateur et central dans un projet : à tout instant et à tout stade de l'évolution de la méthode, il fournit des renseignements aux concepteurs du projet.

## II.5. CONTENU ET USAGES DU DICTIONNAIRE D'UHROWCZIK

### II.5.1. Introduction à l'analyse

Le dictionnaire d'Uhrowczik est le fruit d'une étude théorique menée chez IBM. Le but de l'auteur est principalement de faire, dans ce secteur connaissant une évolution rapide, une synthèse des connaissances actuelles.

Par ailleurs, il passe volontairement sous silence l'aspect d'aide au management du dictionnaire ; il apparaît en effet que cet aspect est déjà abondamment traité dans d'autres articles édités sous le patronage d'IBM.

### II.5.2. Analyse du contenu du dictionnaire (1)

#### a. Structure du dictionnaire

La structure relationnelle du dictionnaire d'Uhrowczik se présente comme sur le schéma 2.5. On y reconnaît

- les descripteurs de traitements : système, job, processus ;
- les descripteurs de données : éléments, groupe, base de données, ensemble de données, segment physique ;
- des données de l'environnement : transaction, rapport, source.

#### b. Etude des descripteurs du dictionnaire

##### b.1. Les descripteurs de traitement

---

(1) UHROWCZIK, P.P. : *Data Dictionary/Directory*.

STRUCTURE RELATIONNELLE DU DICTIONNAIRE D'UHROWCZIK

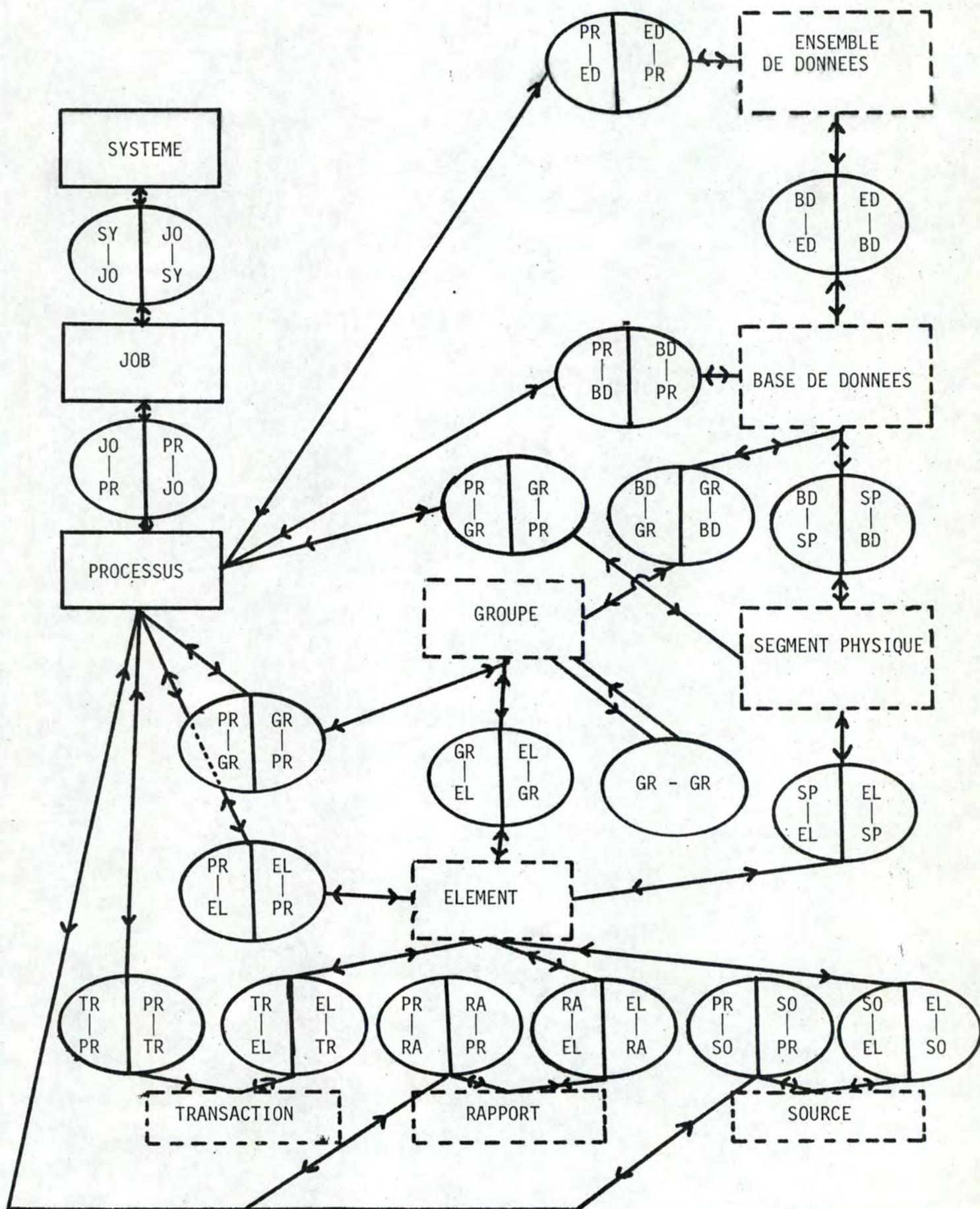


Figure 2.5.

LE PROCESSUS
--------------

Définition

C'est une procédure ou une étape de Job, ou un programme, ou une unité de procédure qui accomplit une tâche spécifique de traitement des données. Elle peut être soit manuelle, soit mécanisée. Tel est le cas, par exemple, d'un programme de vidage d'un fichier central de paie.

Caractéristiques identifiantes

- Le nom ou étiquette (1) ;
- les spécifications du responsable (1) ;
- le contenu de ses responsabilités (1) ;
- la date du dernier changement (1) ;
- une description textuelle (nom commun, signification, but) (1) ;
- synonymie (autre description de la même donnée ayant une étiquette différente) (1) ;
- la version ;
- le statut (proposé, concurrent, approuvé, effectif).

Caractéristique intrinsèque

- la longueur en caractères.

Caractéristiques relationnelles

Il y a une relation avec

- le job auquel il appartient ;
- les transactions qui agencent son exécution ;
- les groupes, les bases de données, les ensembles de données et les segments physiques qui constituent les pools de données nécessaires à l'exécution du processus ;
- la source qui fournit des informations sur les fichiers non résidant sur mémoire secondaire (2) ;
- le rapport d'édition qui le relie avec les utilisateurs.

---

(1) Toutes ces caractéristiques se rencontrent partout ailleurs. Nous ne les répétons donc pas.

(2) Voir définition p. 48.

LE JOB
--------

Définition

C'est un ensemble de processus formant un tout en lui-même. Par exemple, ce sera le job "Salaire et traitement du personnel".

Caractéristiques identifiantes (1)Caractéristique intrinsèque (2)Caractéristiques relationnelles

Le Job possède une relation avec

- le système dont il représente tout un secteur d'activité ;
- les processus qui le constituent.

LE SYSTEME
------------

Définition

Il est une combinaison de jobs aussi appelé système d'exploitation (ou informatique) et il satisfait le traitement de l'information de tout un département.

Nous citerons l'exemple du système "Personnel".

Caractéristiques identifiantes (1)Caractéristique intrinsèque (2)Caractéristiques relationnelles

Le système est en relation avec

- les jobs qui représentent les secteurs principaux sur lesquels le système agit.

---

(1) cfr. note n° 1, p. 42.

(2) aucune.

## b.2. Les descripteurs de données

### L'ELEMENT

#### Définition

C'est la plus petite unité indépendante de donnée qui peut être référencée par un processus. Ce sera, par exemple, le nom de l'employé, son sexe, etc.

#### Caractéristiques identifiantes (1)

- son synonyme dans un autre langage de programmation ;
- la destination.

#### Caractéristiques intrinsèques

- la longueur en caractères ;
- le mode (string de bits, de caractères, etc.) ;
- l'alignement (à droite, à gauche) ;
- le format d'édition ;
- les règles d'édition (masques, etc.) ;
- l'algorithme dérivé (pour le calcul de certains éléments) ;
- la clé ;
- l'index ;
- l'unité (poids, longueur, argent, etc) ;
- la séquence (la position séquentielle par rapport aux autres éléments) ;
- les sécurités.

#### Caractéristiques relationnelles

L'élément est en relation avec

- les processus, les rapports, les transactions, les sources, les segments physiques, et les groupes auxquels il fournit les caractéristiques des éléments dont ils sont constitués.

### LE GROUPE

#### Définition

C'est le groupement logique d'éléments et/ou groupes. Ceux-ci sont aussi

---

(1) Cfr. note n° 1, P. 42.

appelés articles de données groupées, segments logiques, enregistrements logiques. Typiquement, un groupe représente la vision que l'utilisateur a de ses données. Par exemple, le groupe 'date' est le groupement de "JJ/MM/AA".

### Caractéristiques identifiantes (1)

#### Caractéristiques intrinsèques

- la longueur,
- le format d'édition,
- les règles d'édition,
- la clé,
- l'index,
- la séquence,
- les sécurités,
- les options de traitement I.M.S. (2).

#### Caractéristiques relationnelles

Il existe une relation avec

- les processus, les segments physiques et les bases de données dont le groupe est constituant.
- les éléments, qui, eux, lui fournissent des caractéristiques des données qu'il regroupe.
- les autres groupes, par exemple, un groupe-maître est lié à ses groupes-esclaves.

### LE SEGMENT PHYSIQUE

#### Définition

C'est la plus petite unité de données accessible directement et résidant sur mémoire secondaire. C'est, par exemple, un type d'enregistrement d'un fichier signalétique personne.

#### Caractéristiques identifiantes (1)

---

(1) Cfr. note n° 1, p. 42.

(2) Pour plus d'explication, le lecteur voudra bien se référer à l'article de Uhrowczik : *Data Dictionary/Directory*.

Caractéristiques intrinsèques

- la longueur,
- les règles d'édition,
- la séquence,
- les règles de mise à jour I.M.S. (1),
- les options de traitement I.M.S. (1).

Caractéristiques relationnelles

Le segment physique est en relation avec

- les éléments qui le constituent,
- les bases de données dont il est constitutif,
- les processus dont il est outil de travail.

L'ENSEMBLE DE DONNEES

Définition

Il peut aussi être appelé fichier physique. Il regroupe les segments physiques sur mémoire secondaire. Normalement, un ensemble de données consiste en un ou plusieurs type(s) de segment(s) physique(s) organisé(s) de manière logique.

Ce sera le cas du fichier 'personnel', 'approvisionnement'.

Caractéristiques identifiantes (2)Caractéristiques intrinsèques

- le volume,
- le facteur de croissance,
- le moyen de stockage (cartes, bandes, etc.),
- la séquence de tri.

Caractéristiques relationnelles

L'ensemble de données est en relation avec

- le processus dont il est outil de travail ;
- la base de données dont il est constituant.

---

(1) Cfr. note n° 2, p. 45.

(2) Cfr. note n° 1, p. 42.

## LA BASE DE DONNEES

### Définition

Elle regroupe un ou plusieurs ensemble(s) de données ou base(s) de données. Par exemple, ce sera la base de données du système "Personnel".

### Caractéristiques identifiantes (1)

#### Caractéristiques intrinsèques

- le volume,
- le facteur de croissance,
- l'organisation,
- les sécurités,
- la séquence de tri,
- les règles de mise à jour I.M.S. (2)
- les options de traitement I.M.S. (2).

#### Caractéristiques relationnelles

Elle est en relation avec

- les processus qui l'utilisent,
- les groupes, les ensembles de données et les segments physiques qui la constituent.

## LES TRANSACTIONS

### Définition

C'est un ensemble spécifique de données paramétrant l'exécution d'un processus ou d'un job. Les transactions peuvent d'abord être pré-mémorisées, puis présentées au processus, ou bien chaque transaction peut paramétrer un processus en temps réel. Ce sera par exemple les cartes contrôles d'un "Job Control Language".

### Caractéristiques identifiantes (1)

---

(1) Cfr. note n° 1, p. 42.

(2) Cfr. note n° 2, p. 45.

Caractéristiques intrinsèques

- la longueur,
- la sécurité,
- le support (disque, carte, bande, vidéo, etc.).

Caractéristiques relationnelles

Les transactions sont en relation avec

- les processus, en effet les transactions commandent leur exécution.
- les éléments qui sont les données nécessaires pour paramétrer le déroulement des processus.

LE RAPPORT

Définition

C'est l'information telle qu'elle est modelée pour être présentée aux utilisateurs. Le rapport est typiquement le résultat de certains processus et peut prendre forme sur un listing, sur un vidéo, etc.

Caractéristiques identifiantes (1)Caractéristiques intrinsèques

- la longueur,
- les sécurités,
- les supports.

Caractéristiques relationnelles

Le rapport est en relation avec

- les processus qui commandent leur impression.
- les éléments qui caractérisent les données à imprimer.

LA SOURCE

Définition

La source est un fichier de données qui n'existe pas sur une mémoire secondaire du système. On ne peut donc stocker une source ni sur une bande

---

(1) Cfr. note n° 1, p. 42.

magnétique, ni sur un disque. La source contient des informations qui dès leur entrée dans le système sont exploitées puis rejetées à l'extérieur. Ce sera par exemple le cas des résultats d'enquête qui chaque jour, en grand nombre, sont traités mais non stockés.

### Caractéristiques identifiantes (1)

#### Caractéristiques intrinsèques

- la longueur,
- le support (cartes, vidéo, ...).

#### Caractéristiques relationnelles

La source est en relation avec

- le processus auquel la source fournit des renseignements ;
- les éléments qui décrivent les données qui vont être lues et traitées par les processus.

## II.5.3. Analyse des usages du dictionnaire

### a. A destination des programmes

Les traitements pour développer et maintenir des applications peuvent être améliorés en étendant l'usage du dictionnaire de données aux systèmes de programmations (compilateur, S.G.B.D., etc.).

L'objectif consiste à supprimer de la phase de programmation des applications la plupart des efforts nécessaires pour définir et manipuler les données. Ceci est possible si on pré-mémorise les descriptions des données dans un dictionnaire, de façon telle que le compilateur et/ou le S.G.B.D. puisse puiser dans ce pool d'informations les informations nécessaires aux manipulations que le compilateur et/ou le S.G.B.D. effectuent.

#### a.1. Conversion de données

Par un processus de mapping (passage de données d'une structure A dans une structure B), les données peuvent être converties dans des formats différents. Par exemple, une donnée peut passer du format condensé au format non condensé.

---

(1) Cfr. note n° 1, p. 42.

a.2. Compactage des données

Des données peuvent être mémorisées en une forme compacte mais présentées à l'utilisateur dans un format plus aéré et plus significatif.

a.3. Validation d'entrée/sortie

Les données entrantes dans un programme ou dans une mémoire physique peuvent être contrôlées en les comparant avec des standards pré-établis mémorisés dans le dictionnaire.

a.4. Génération de données tests

A partir des descriptions des données faites dans le dictionnaire et les demandes de l'utilisateur, il est possible de générer des données-tests. (cfr. Datamanager p. 24)

°  
° °

## CHAPITRE III

### SYNTHESE ET CRITIQUES

#### III.1. INTRODUCTION

Cette étude de l'existant, quoique assez succincte, nous donne, grâce à la représentativité des sujets étudiés, une bonne idée des dictionnaires de données utilisés ou utilisables dans les organisations.

Nous allons maintenant faire une analyse comparative de leurs contenus et de leurs usages, avant de montrer que si contenus et usages sont différents, en tout cas, *le cycle de vie* de tous les dictionnaires est lui identique.

#### III.2. ANALYSE COMPARATIVE DES DIFFERENTS CONTENUS

Chaque dictionnaire de données est constitué, avons-nous écrit, d'un ensemble de classes de données, mises en relation. Chaque classe a reçu un nom, une définition du rôle joué dans le système informatique par les données qui y sont rassemblées, et un ensemble de caractéristiques descriptives de ces données.

En mettant en présence dans un premier stade tous les noms de classes et les définitions du rôle qu'elles jouent, plusieurs conclusions s'imposent :

##### III.2.1. sur les noms des classes de données

Il est difficile de comparer des dictionnaires entre eux avec pour seul point de comparaison les noms des classes de données. En effet, il n'y a aucune uniformité dans l'appellation des classes. Tout au plus peut-on constater la présence de six noms, communs à deux dictionnaires au maximum. Ce sont :

- le *système* que l'on retrouve dans Datamanager et Uhrowczik ;
- le *programme* que l'on retrouve dans Datamanager et British Rail ;
- le *module* que l'on retrouve dans Datamanager et Rémora ;
- le *fichier* que l'on retrouve dans Datamanager et British Rail ;
- le *groupe* que l'on retrouve dans Datamanager et Uhrowczik ;
- la *donnée élémentaire* que l'on retrouve dans Datamanager et British Rail.

On trouvera sur la figure 3.1. un tableau récapitulatif des noms des différentes classes de données, décrites dans chacun des quatre dictionnaires.

Il se peut pourtant, que deux noms de classes différentes, réfèrent au même objet ; c'est pourquoi il est utile de comparer entre elles les définitions de classes.

### III.2.2. sur les définitions des classes de données

Si l'on compare entre elles les définitions attachées à chacune des classes de données, le résultat n'est pas tellement meilleur. En effet, un comparaison de définitions est difficile car les définitions tiennent compte dans leur énoncé du contexte particulier dans lequel elles ont été élaborées. Il faut donc pour que cette comparaison soit valable faire abstraction de ce contexte ou encore trouver des critères de comparaison indépendants de ce contexte. Deux critères discriminants peuvent être établis :

- *les niveaux d'imbrication hiérarchique des données et des traitements.*  
En effet, il existe une hiérarchie des traitements d'une part, et une hiérarchie des données d'autre part, utilisés dans un système informatique. Par exemple, un module est un sous-ensemble de la règle du jeu des procédures dans Rémora ; des données élémentaires rassemblées constituent des enregistrements, ceux-ci regroupés formeront un fichier pour la British Rail. Il est donc tentant de comparer entre elles les classes qui se situent au même niveau de hiérarchie à partir des feuilles.
- *la représentation, par l'exemple, de ce que la définition implique, aux yeux de son concepteur.*  
Ainsi, dans le dictionnaire de Datamanager et Rémora, l'exemple illustrant la définition d'un travail effectué par un module est le traitement du calcul du montant des heures supplémentaires.

3.1. Tableau comparatif des différents noms de classes de données

Dictionnaire	Datamanager	British Rail	Rémora	Uhrowczik
Classe de données	ystème		application	ystème
	programme		procédure	job
	module		module	
		programme		
				processus
TRAITEMENT				
DONNEES	donnée élémentaire	donnée élémentaire	mot	élément
	groupe	enregistrement		groupe segment physique
	fichier	fichier	structure	ensemble de données
				base de données

En fonction de ces deux critères, on voit apparaître, pour ce qui concerne les traitements et les données, les similitudes suivantes (cfr. pour s'en convaincre, leurs définitions réparties sur le tableau 3.2., où est également synthétisée l'imbrication des différentes descriptions dans les différents dictionnaires.) :

- Au point de vue des traitements

- Le module de Datamanager,
- le module de Rémora.
  
- Le programme de Datamanager,
- la procédure de Rémora,
- le job de Uhrowczik.
  
- Le système de Datamanager,
- le système de Uhrowczik,
- l'application de Rémora.

- Au point de vue des données

- La donnée élémentaire de Datamanager,
- la donnée élémentaire de British Rail,
- le mot de Rémora,
- l'élément d'Uhrowczik.
  
- Le groupe de Datamanager,
- l'enregistrement de British Rail,
- le groupe de Uhrowczik.
  
- Le fichier de Datamanager,
- le fichier de British Rail,
- la structure de Rémora,
- l'ensemble de données de Uhrowczik.

D'abord, il faut remarquer que les traitements présentent un moins grand accord entre eux que les données.

Ensuite, nous constaterons que cette comparaison ne permet pas d'arriver à une conclusion généralisée. En effet, la notion de programme pour la British Rail ainsi que celle de base de données pour Uhrowczik, par exemple, restent sans correspondance ailleurs. Pour la base de données, ce n'est pas

grave, car elle se situe à un niveau plus englobant encore que celui des fichiers, mais la notion de programme, pour ne citer que celle-la, est sans correspondance parce que le dictionnaire de la British Rail, à la différence des autres dictionnaires ne discrimine pas de niveau dans les traitements.

Poursuivant plus avant nos investigations, nous avons effectué une comparaison des caractéristiques descriptives de chaque classe pour voir, s'il n'existe pas pour les classes de données ayant une définition similaire, des noyaux de caractéristiques identiques.

### III.2.3. sur les caractéristiques descriptives des classes de données

D'abord, vérifions au moyen du tableau 3.3. l'existence ou la non-existence de noyaux pour les classes ayant une définition semblable. On ne peut rien conclure car si noyaux il y a, aussi bien pour les caractéristiques identifiantes que pour les caractéristiques intrinsèques, ils sont petits et quasi sans intérêt (1).

Comment expliquer que les noyaux sont petits et sans intérêt ?

Il est intéressant de remarquer que la richesse de chaque dictionnaire en caractéristiques descriptives varie selon que les usages pour lesquels ils ont été réalisés, sont plus ou moins élaborés. Il y a donc une interdépendance entre la richesse en caractéristiques descriptives présentes au sein du dictionnaire et la sophistication des usages qu'on en fait. Il n'y a rien d'étonnant alors si les noyaux sont petits et banals, car pour éviter cet état de fait, il faudrait que tous les dictionnaires aient eu un même ensemble d'usages, or ce n'est pas le cas (2).

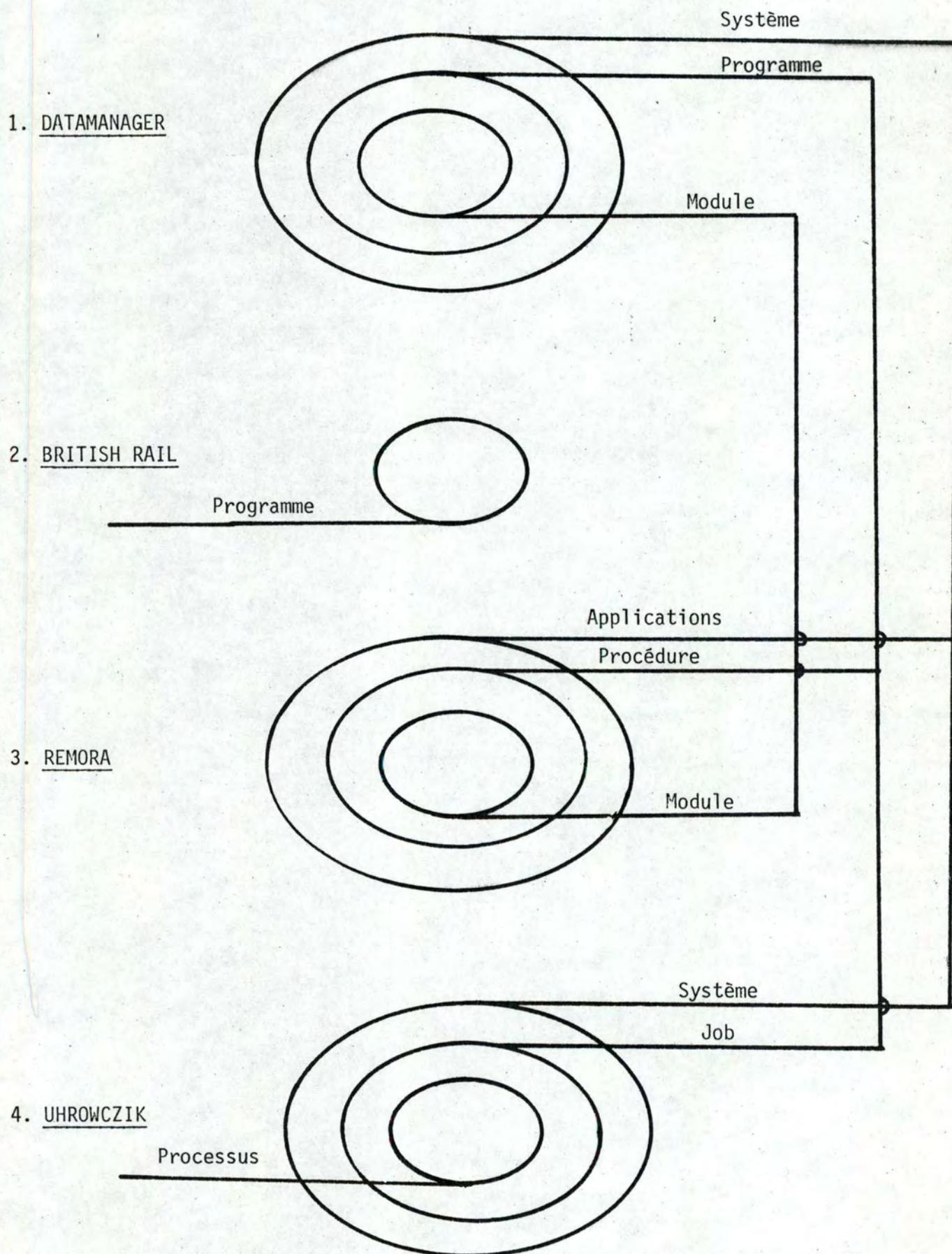
### III.2.4. Conclusions et critiques

A l'issue de cette synthèse basée sur une analyse comparative des différents contenus d'un dictionnaire de données on constatera d'abord, le caractère ardu du travail de comparaison. Nous ferons ensuite les remarques suivantes :

- 
- (1) On voit cependant qu'une analyse globale permet de déceler un ensemble de *caractéristiques identifiantes* très complet chez Uhrowczik; il est le seul qui à chaque niveau fasse intervenir la notion de responsabilité de la donnée décrite.
- (2) Cfr. notre analyse des usages au § III.3.

Tableau 3.2. : Similitude des définitions attachées aux classes de traitements

Dictionnaire	DATAMANAGER	REMORA	UHWOCZIK
CLASSES	Module	Module	
Modules	C'est une combinaison de fichiers et/ou groupes et/ou données élémentaires sur lesquels s'effectue un traitement spécifique.	C'est un sous-ensemble de la règle du jeu de la procédure.	
	Programme	Procédure	Job
Programme/Procédure/Job	C'est une combinaison de fichiers et/ou modules formant une unité opérationnelle indépendante.	C'est un ensemble de traitements donnant réponse à un événement tendant à une fin utile et d'exécution assez fréquente pour qu'on ait pu en arrêter les modalités.	C'est un ensemble de processus formant un tout en lui-même.
	Système	Application	Système
Système/Application	C'est une combinaison de fichiers et programmes concourant à satisfaire tout un secteur d'une entreprise.	C'est une famille de procédures appartenant à un même domaine d'application, traitant des informations appartenant à une même population, déclenchées par des événements de même nature pour obtenir les mêmes résultats.	C'est une combinaison de Jobs qui satisfont le traitement de l'information de tout un secteur.



3.2. suite : Présentation hiérarchique des classes de traitement et correspondance entre les classes semblables

3.2. suite : Similitude des définitions attachées aux classes de données

Dictionnaire	DATAMANAGER	BRITISH RAIL	REMORA	UHROWCZIK
CLASSE	Données élémentaires	Données élémentaires	Mot	Elément
Données élémentaires/Mots/Eléments	C'est un atome élémentaire d'information.	C'est l'élément central et le plus petit de la structure.	Répère une entité intervenant dans la structure.	La plus petite unité indépendante de donnée.
	Groupe	Enregistrements		Groupe
Groupe/Enregistrements	C'est une combinaison de données élémentaires et/ou de groupes.	C'est le regroupement de données élémentaires.		C'est le groupement logique d'éléments et/ou de groupes.
Fichier/Structure/Ensemble de données	Fichier	Fichier	Structure	Ensemble de données
	C'est une combinaison de groupes.	C'est le regroupement de types d'enregistrements.	C'est un ensemble d'entités groupées sur un document.	C'est le fichier physique qui regroupe les segments physiques.

3.2. suite : Présentation hiérarchique des classes de données et correspondance entre les classes semblables

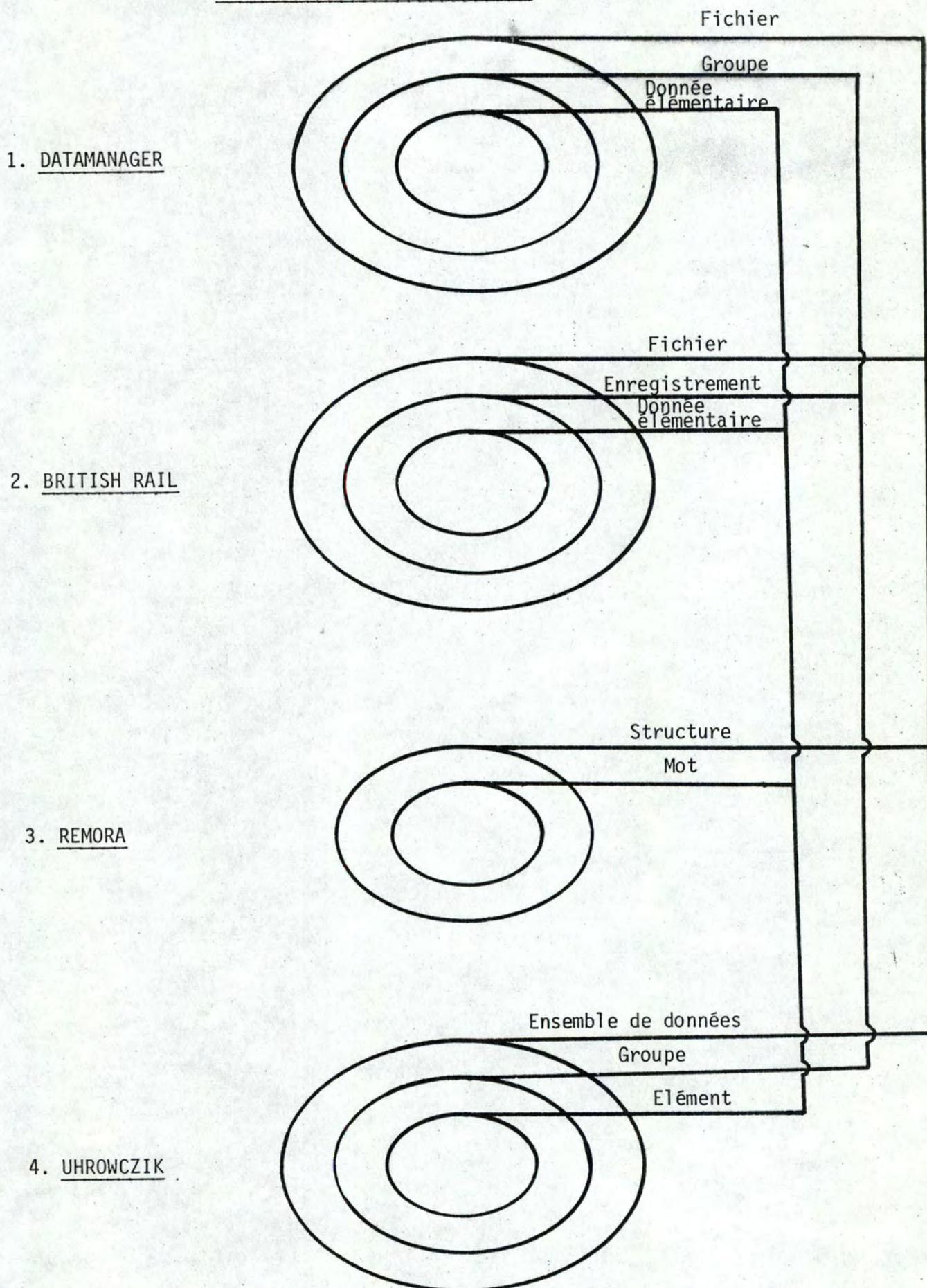


Tableau 3.3. : Présentation du noyau de caractéristiques descriptives pour les quatre dictionnaires

Dictionnaire	DATAMANAGER	REMORA	UHROWCZIK.
CLASSES	Système	Application	Système
Caractéristiques identifiantes	<input type="checkbox"/> <input type="checkbox"/> (Remarque : <input type="checkbox"/> = comme Datamanager est un package, vu son adaptabilité on considérera qu'il possède le même noyau que les autres dictionnaires)	IDENTIFICATEUR RESUME Désignation	ETIQUETTE DESCRIPTION TEXTUELLE Spécification du responsable. Contenu des responsabilités. Date du dernier changement. Synonymie. Version - status.
Caractéristiques intrinsèques			Longueur
CLASSES	Programme	Procédure	Job
Caractéristiques identifiantes	<input type="checkbox"/> <input type="checkbox"/>	IDENTIFICATEUR RESUME	ETIQUETTE DESCRIPTION TEXTUELLE (1)
Caractéristiques intrinsèques			Longueur
CLASSES	Module	Module	Processus
Caractéristiques identifiantes	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	IDENTIFICATEUR RESUME SYNONYMIE	ETIQUETTE DESCRIPTION TEXTUELLE SYNONYMIE (1)
Caractéristiques intrinsèques			Longueur

(1) : idem que pour le système.

Tableau 3.3. suite

Dictionnaire	DATAMANAGER	BRITISH RAIL	REMORA	UHROWCZIK
CLASSE	Fichier	Fichier	Structure	Ensemble de données
Caractéristiques identifiantes	NOM <input type="checkbox"/>	NOM (USUEL, ELEMENTS) DESCRIPTION INFORMELLE Date et version	IDENTIFICATEUR DESIGNATION EN CLAIR	ETIQUETTE DESCRIPTION TEXTUELLE (1)
Caractéristiques intrinsèques	<input type="checkbox"/> <input type="checkbox"/>	ORGANISATION  LOCALISATION - SUPPORT Archivage	STRUCTURE DE TRAVAIL, D'EDITION GLOSSAIRE Texte objet	ORGANISATION  STOCKAGE Volume, taux de croissance séquence de tri.
CLASSE	Groupe	Enregistrement		Groupe
Caractéristiques identifiantes	NOM <input type="checkbox"/>	NOM (USUEL, ASSEMBLAGE, ELEMENT) DESCRIPTION INFORMELLE Date et version		ETIQUETTE  DESCRIPTION TEXTUELLE (1)
Caractéristiques intrinsèques				Longueur, règle d'édition, séquence, I.M.S., format d'édition, clé, index, sécurités.

(1) idem pour le système.

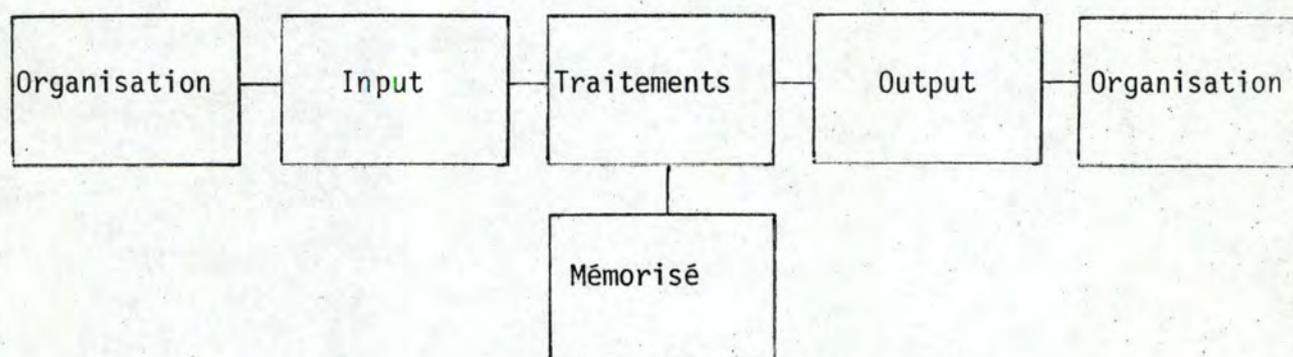
Tableau 3.3. suite

Dictionnaire	DATAMANAGER	BRITISH RAIL	REMORA	UHROWCZIK
CLASSE	Donnée élémentaire	Donnée élémentaire	Mot	Eléments
Caractéristiques identifiantes	NOM  □	NOM (USUEL, ENREGISTREMENTS)  DESCRIPTION (FORMELLE, INFORMELLE)  Synonymie  Date et version	IDENTIFICATEURS  DESIGNATION EN CLAIR	ETIQUETTE  DESCRIPTION TEXTUELLE  Synonymie destinataire + (1)
	LONGUEUR TYPE  format d'entrée, de traitement. Dessin d'édition. Précision. Contenu actuel,	LONGUEUR TYPE (Code)  en-rête rapport Critères de création/ destruction Contrôle d'intégrité Unité, archivage Valeur (nulle ou par défaut) Code de référence.	TAILLE TYPE	LONGUEUR (ALGORITHME, REGLE D'EDITION, MODE) format d'édition, alignement, unité, clé, index, séquence, I.M.S.

(1) idem pour le système.

1. Il existe un manque de normalisation des objets considérés comme appartenant à un système informatique. Cette absence de normalisation provient évidemment des modèles de systèmes informatiques différents selon les concepteurs. Une question intéressante serait : "N'y a-t-il pas un modèle simple qui pourrait être accepté par tous ceux qui ont conçu les dictionnaires sous étude ?"

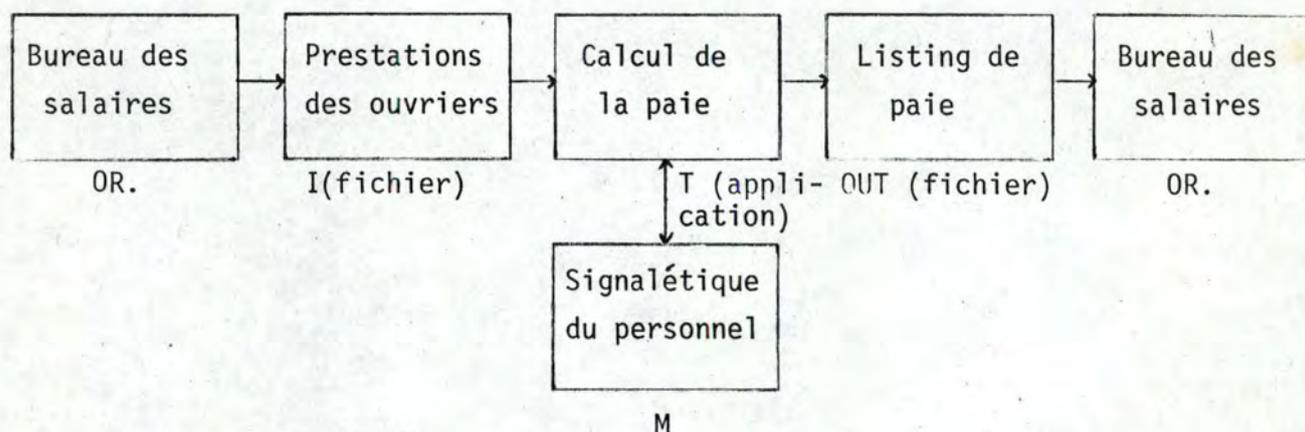
A notre avis, il en existe un : nous référant aux travaux de Wirth et de Dijkstra, e.a., sur la programmation structurée, on peut dire que "pour maîtriser le problème, l'esprit humain en organise la solution *par niveau* et à *tout niveau*, on retrouve un schéma pareil à celui-ci :



Ce modèle est celui qui sert de base au projet ISDOS (1); on le retrouve encore avec des variantes, dans le projet HYPO (2) développé chez IBM.

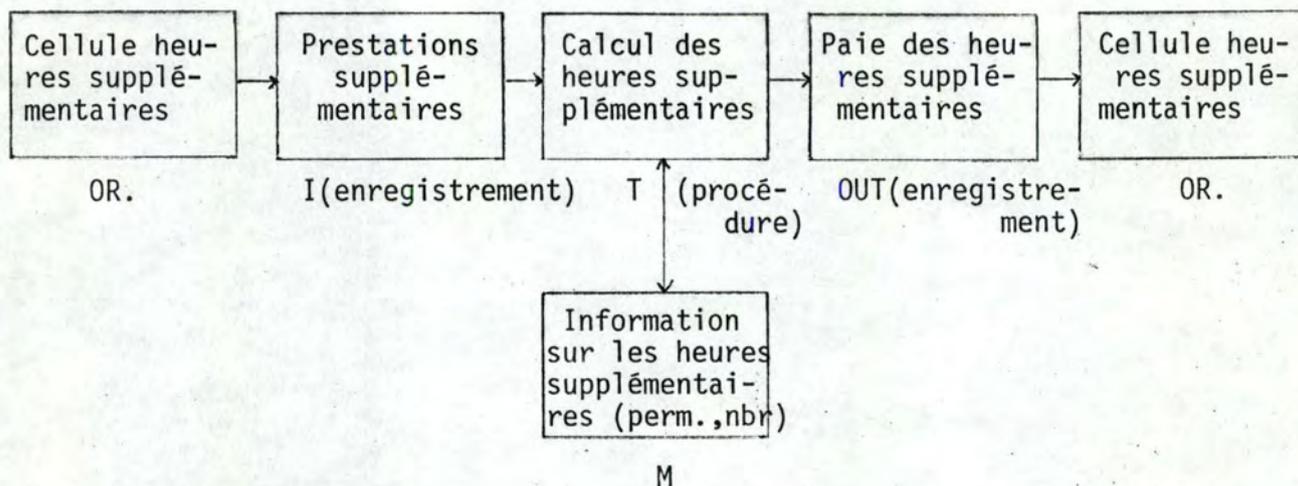
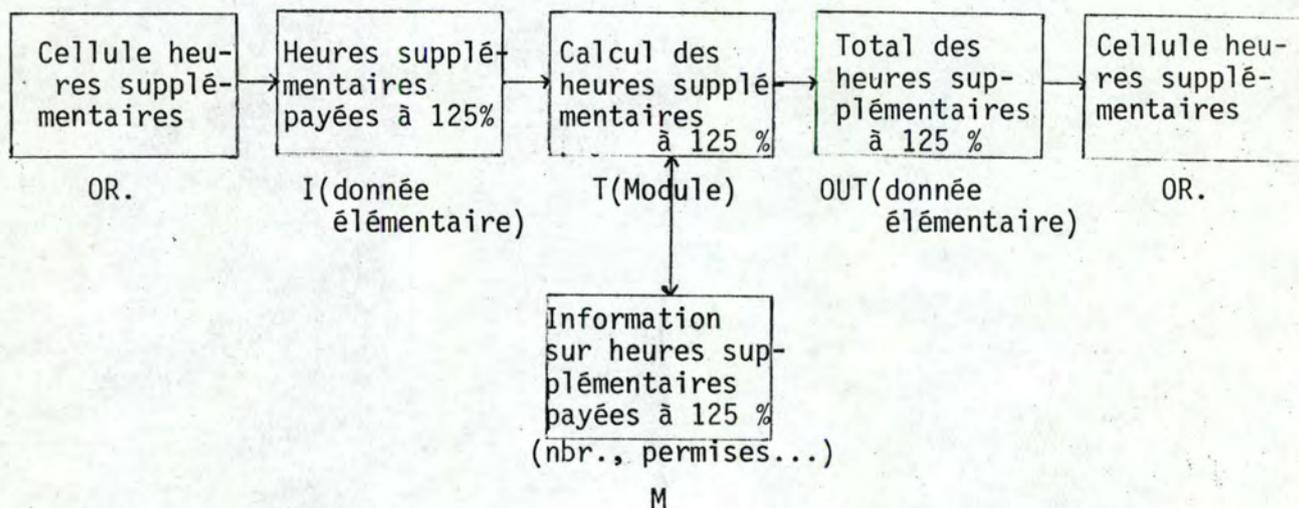
Pour illustrer notre propos, voici un exemple tiré d'une application de calcul des salaires. Nous présentons ici trois niveaux correspondant respectivement pour les input/output à la découpe en Fichier, Enregistrement, Donnée élémentaire, et les traitements à Application, Procédure, Module.

#### Premier niveau



(1) LESUISSE, R. : *Exposé introductif au projet ISDOS.*

(2) LABOUREUR, L. : *Le développement de la documentation des projets et des programmes : application des concepts.*

Deuxième niveauTroisième niveau

Les dictionnaires ainsi présentés auraient cinq grandes classes de données : les *Inputs*, les *Outputs*, les *Traitements*, les *Mémoires* et l'*Environnement* (ou l'*Organisation*).

Que décrirait chacune de ces classes ?

- Les *Inputs* seraient constitués de caractéristiques descriptives décrivant
  - les niveaux d'agrégation (fichier, enregistrement, donnée élémentaire)
  - des données en entrée.

- On pourrait les caractériser par l'adjonction de certains renseignements :
    - d'où vient cet input ?
    - ou va t'il ?
    - qui peut y accéder ?
    - les traitements qui l'utilisent ;
    - les modifications que l'on peut y apporter.
  - Les Outputs seraient constitués de caractéristiques descriptives semblables à celles des Inputs mais décrivant des données qui sont en sortie.
  - Les Traitements seraient décrits au moyen de caractéristiques descriptives décrivant les programmes qui effectuent des travaux sur les Inputs pour obtenir les outputs (comme le descripteur du plus bas niveau dans le dictionnaire).
  - Les mémorisés décriraient des données qui ne sont ni en entrée, ni en sortie, mais auxquelles les traitements font référence (exemple : un fichier barème).
  - L'environnement décrit tout ce qui à trait au système d'information mais qui n'est pas le système informatique. Ce sera par exemple les utilisateurs, la cellule d'activité, ...
2. Tous les dictionnaires ne donnent qu'une *image du système informatique et non du système d'information*. Il est, en effet, très peu question de l'organisation dans laquelle les systèmes informatiques s'intègrent. Par exemple, seul British Rail crée un descripteur pour les utilisateurs et seul Uhrowczik prévoit comme caractéristique descriptive appartenant à toutes les classes, la notion de "contenu et spécification" de la responsabilité de la donnée. D'une façon générale, il n'y a aucune description des cellules d'organisation. Ce qui a pour conséquence que le dictionnaire ne devient un outil utile et utilisable que fort tard dans l'analyse, pratiquement au moment de l'implémentation, lorsque les grands choix sont faits.
3. Au niveau global, la *description* d'un système informatique est essentiellement *statique*. Comment pourrait-on, à l'aide des dictionnaires étudiés, décrire les flux d'informations dans une organisation ? Seul Uhrowczik, faisant exception avec ses descripteurs Source et Transaction, nous montre comment il faut faire. On pourrait donc palier cela par la création de classes spéciales descriptives des flux.

4. Au niveau des caractéristiques attachées à une classe de données, peu de caractéristiques permettent le suivi d'un projet, son contrôle par le management. Nous montrerons plus loin, à l'aide d'exemples, que moyennant l'insertion de nouvelles caractéristiques descriptives dans les classes de données, on pourrait produire des rapports qui pallient cette lacune.
5. Seul Rémora prend véritablement en compte des systèmes informatiques complexes ; les autres, datant de quelques années se contentent de documenter des applications isolées. Pour s'en convaincre, il n'y a qu'à relire les rôles joués par ce dictionnaire au sein du projet (1).

### III.3. ANALYSE COMPARATIVE DES DIFFERENTS USAGES

#### III.3.1. Le "management use"

Le tableau 3.4. regroupe les différents usages de chaque dictionnaire sous étude à destination des responsables. En règle générale, cette utilisation vise à exploiter le contenu des dictionnaires à des fins de documentation éditée sous forme de rapport, glossaire ou listing.

En observant de plus près, on décèle deux types de rapports :

- le rapport sur l'état d'un membre du dictionnaire ;
- le rapport sur l'impact d'une modification d'un membre.

Le tableau 3.4. distingue aussi ces rapports selon leurs types et associe à chaque usage un exemple cité au chapitre II.

De toute évidence, ces deux types de rapports d'état et d'impact sont commandés par la constatation suivante et vérifiée : des changements provoquent souvent des erreurs en cascade (2). Ces erreurs peuvent être évitées pour la plupart par la production de rapports.

Mais, comme on l'a déjà dit, on peut se demander pourquoi les utilisateurs n'ont pas profité d'une manière plus à propos du caractère de répertoire centralisé de descriptions de données du dictionnaire pour en faire un outil de planification et de contrôle (sous la responsabilité du Data Administrator).

---

(1) Cfr. notre chapitre II, § II.4.

(2) H. LEROY : Cours de programmation structurée.

DICTIONNAIRES	USAGES DANS LE MANAGEMENT	TYPE	EXEMPLE
DATAMANAGER	Rapport sur l'état d'un élément du dictionnaire.	état	p. 22
	Réponse à des questions sur les relations entre éléments du dictionnaire.	impact	p. 22
BRITISH RAIL	Rapport décrivant l'état du dictionnaire.	état	p. 31
	Rapport énumérant des endroits d'utilisation d'une entité de la structure relationnelle.	impact	p. 31
	Glossaire pour l'utilisateur.	état	p. 32
	Listing des codes.	état	p. 32
REMORA	Le projet lui-même	état/impact	p. 39
UHROWCZIK	Cfr. la bibliographie.		

Tableau 3.4. : Présentation des usages à destination du management

Ainsi, par exemple, moyennant l'adjonction aux descripteurs de données de quelques caractéristiques supplémentaires, on pourrait produire trois types de rapports concernant l'utilisation des ressources au cours de la phase de développement d'un projet.

Pour chaque type de rapport, nous donnerons :

- une brève définition du rôle qu'il pourrait jouer ;
- les caractéristiques qu'il faudrait ajouter aux différents descripteurs d'un dictionnaire comparable à celui d'Uhrowczik, pour les produire ;
- un exemple.

a. Rapport sur l'utilisation des ressources - hommes

a.1. Son rôle

Ce type de rapport renseignerait le responsable sur le déroulement des activités du programmeur dans sa tâche d'implémentation des programmes. Par exemple, pour une période de référence donnée, ce rapport indiquerait le déroulement des activités du programmeur XXXX. Ces activités seraient exprimées en nombre de mises à jour effectuées, nombre de lignes de code-source, écrites en Input, supprimées, ajoutées, changées durant la période de référence.

a.2. Caractéristiques à ajouter

- La date de début et de la dernière modification du programme.
- La période de référence.
- Le nombre de mises à jour effectuées d'une part, pendant la période de référence et d'autre part, depuis le début jusqu'à la date de la dernière modification.
- Le nombre de lignes de code-source écrites en input, ajoutées, supprimées, changées pendant la période de référence et jusqu'à la date de la dernière modification.

Ces caractéristiques descriptives seraient ajoutées à celles qui décrivent déjà le programme au sein du dictionnaire.

a.3. Exemple (1)

---

(1) inspiré de l'article de R.L. SMITH : *Statement of Work of a Structured Programming System*, pp. 3.1. - 3.31.

Ce rapport pourrait se présenter de la sorte :

Demandé	RAPPORT DE STATISTIQUE	LE 28/08/77
par	SUR LA PRODUCTION	
XXXXX	DU PROGRAMME	
	"GESTION COMPTES COURANTS"	
- Identification de l'utilisateur	:	XXXX
- Date de début	:	05/02/77
- Date du dernier changement	:	09/08/77
- Période de référence	:	31/07 - 29/08/77
- Nombre de mises à jour : total	:	8
- Nombre total de lignes en code-source :		
- écrites en input	:	380
- supprimées	:	10
- ajoutées	:	34
- changées	:	14
- Nombre de lignes en code-source écrites en input pendant la période de référence :	:	30
- Nombre de mises à jour pendant la période de référence	:	3

b. Rapport sur l'utilisation des "ressources machines"

b.1. Son rôle

Ce type de rapport renseignerait le responsable sur l'utilisation de l'ordinateur faite par un programmeur de façon à aider le responsable dans ses tâches d'administrateur, d'optimisateur et d'alloueur du temps nécessaire à l'exécution de programmes.

Par exemple, pour un programme donné, ce rapport montrerait le nombre de passages en machine, le temps moyen passé en machine par le programme pendant une période de référence donnée.

b.2. Caractéristiques à ajouter

Ces caractéristiques seraient à ajouter à celles déjà existantes décrivant le programme au sein du dictionnaire. Elles sont :

- la date de début et de la dernière modification du programme ;
- la période de référence ;
- le temps moyen de passage en machine pendant la période de référence et depuis le début jusqu'à la dernière modification du programme ;
- le nombre de passages en machine pendant la période de référence et depuis la date de début jusqu'à la date de la dernière modification.

### b.3. Exemple (1)

Ce rapport pourrait se présenter de la sorte :

Demandé par XXXXX	RAPPORT D'UTILISATION DE L'ORDINATEUR PAR LE PROGRAMMEUR XXXX	LE 29/08/77
- Nom des programmes	: entrée du fichier maintenance	: traitement du fichier maintenance
- nombre de passages	: 52	: 10
- temps moyen d'ordinateur dépensé depuis le début du projet	: 6HRS.18min	: 12HRS.02min
- nombre de passages pen- dant la période	: 18	: 6
- temps moyen d'ordinateur dépensé pendant la pério- de	: 4HRS.30min	: 7HRS.28min.

### c. Rapport d'évaluation globale

#### c.1. Rôle

Le troisième et dernier rapport que nous présentons est un rapport d'évaluation globale de la manière dont les ressources ont été budgétées et dépensées. A terme, des informations de cette nature devraient permettre aux responsables d'affiner leurs prévisions de coûts des différents projets à développer.

#### c.2. Caractéristiques à ajouter

- la date de début, de fin et de la dernière modification du projet ;
- le nombre de "mois d'homme" d'efforts pour les différentes classes d'utilisateurs pour la période de référence et pour la durée du projet que l'on a budgété et que l'on a dépensé.

(1) Cfr. note n° 1., page 68.

- le budget des coûts et les dépenses réelles du personnel pour la période de référence et la durée du projet.
- le temps d'ordinateur, les coûts d'ordinateur, les coûts de voyages, les coûts du matériel que l'on a budgétés et dépensés pour la période de référence et pour la durée du projet.

Ces caractéristiques sont pour la plupart déductibles de tous les descripteurs du projet.

### c.3. Exemple (1)

Ce rapport pourrait se présenter de la sorte :

demandé par XXXX le 29/08/77	RAPPORT D'EVALUATION POUR LE SOUS-SYSTEME MAINTENANCE - FICHER DEPENSES EN RESSOURCES POUR LA PERIODE DU 31/07 - 29/08/77.			
- Date de début	:	09/01/77		
- Date de la dernière modification	:	28/08/77		
- Date de fin	:	31/12/77		
		<u>Budget</u>		<u>Réel</u>
- nombre de "mois d'homme" d'efforts pour - responsables		0.3	2.2.	0.3      2.2
- programmeurs		2.4	17.5	2.0      17.0
- analystes		0.2	1.5	0.2      1.5
- secrétaires		0.2	1.5	0.2      1.5
- coût du personnel (charge et travail) pour - responsables		48.000	336.000	47.500      335.000
- programmeurs		288.000	2.102.500	286.500      2.050.000
- analystes		25.000	177.500	24.500      170.000
- secrétaires		12.500	87.500	12.300      85.000
- Temps d'ordinateur (HRS)		10	75	8      70
- Coût d'ordinateur (FB)		100.000	750.000	80.000      720.000
- Coût des voyages (FB)		5.000	36.000	2.500      34.500
- Coût du matériel (FB)		1.250	26.000	1.100      19.200

(1) Cfr. note n° 1., page 68.

### En conclusion du Management Use

Si l'on considère que tout projet se développe en quatre étapes, une étape de conception, de développement, d'implémentation et d'évaluation, le tableau 3.5 nous montre :

- pour chaque rapport, l'étape où il est utilisé ;
- pour chaque étape, l'utilisateur concerné par le rapport ;
- pour chaque rapport, l'objet de son information.

### III.3.2. Le "Program use"

On veut exploiter le contenu du dictionnaire en vue d'effectuer un certain nombre de travaux de façon automatique. Ces travaux se divisent en quatre grandes classes.

- génération automatique de descriptions de données en langage COBOL ou PL/1.
- génération de données tests.
- génération et validation d'entrées/sorties.
- mise à jour automatique ou auto-enrichissement du dictionnaire.

On trouvera présentés sur le tableau 3.6. les différents "Program uses" associés aux différents dictionnaires. Cette utilisation est restreinte par rapport à la masse importante d'informations contenues dans le dictionnaire. Puisque l'on possède des caractéristiques descriptives des traitements ainsi que des données qui s'y rapportent, il serait possible, à notre avis, d'étendre cette utilisation à des usages un peu plus sophistiqués. Ceux-ci seraient, e.a. :

1. Génération automatique de certains programmes de contrôle de validité des données utilisées par un projet.

Par programme de contrôle de validité, il faut entendre un programme qui vérifierait e.a. que les valeurs prises par des éléments dans le système sont compatibles avec la description qui a été faite.

Exemple : la donnée élémentaire "JOUR". Tous les renseignements sont décrits au moyen de caractéristiques descriptives associées à la donnée "JOUR" dans le dictionnaire. Lors de l'impression du "jour" (dans une date par exemple) un contrôle de sa validité peut être effectué par référence aux renseignements contenus dans le dictionnaire.

Tableau 3.5. : Synthèse des rapports proposés

Type de rapports	Utilisateur	Objet de leur information	Stade de leur utilisation			
			conception	développement	Implémentation	Evaluation
Statistique sur la production de programme.	programmeur responsable en chef, responsable du projet	programme sous-système système			x x x	x x x
Utilisation de l'ordinateur	responsable en chef, responsable du projet	à un niveau d'utilisation déterminé		x	x	
Evaluation des dépenses	programmeur, responsable en chef, responsable du projet	programme sous-système système	x x	x x	x x x	x x x

Dictionnaire	Usage par les programmes
DATAMANAGER	Génération de langage source. Génération de fichiers de données-tests. Impression du contenu d'un fichier. Génération de modules d'entrée/sortie. Mise à jour automatique du dictionnaire. Auto-contrôle des commandes de manipulation.
BRITISH RAIL	Génération automatique de description COBOL. Auto-enrichissement des standards de description.
REMORA	Le projet lui-même.
UHROWCZIK	Conversion de données. Compactage de données. Validation d'entrées/sorties. Génération de données-tests.

Tableau 3.6 : Présentation des usages à destination des programmes

2. Enfin, un projet beaucoup plus ambitieux : la production à partir des descriptions du dictionnaire de toute la partie routinière d'un projet.

C'est ce que se propose de faire Rémora qui veut aller jusqu'à générer le code, mais avec quel succès ?

### III.4. LE CYCLE DE VIE

Au-delà des contenus peu comparables, des usages très diversifiés, apparaît pour tous les dictionnaires un cycle de vie identique composé de trois étapes majeures :

- une étape de structuration du dictionnaire ;
- une étape de remplissage et gestion du dictionnaire ;
- une étape d'exploitation du dictionnaire.

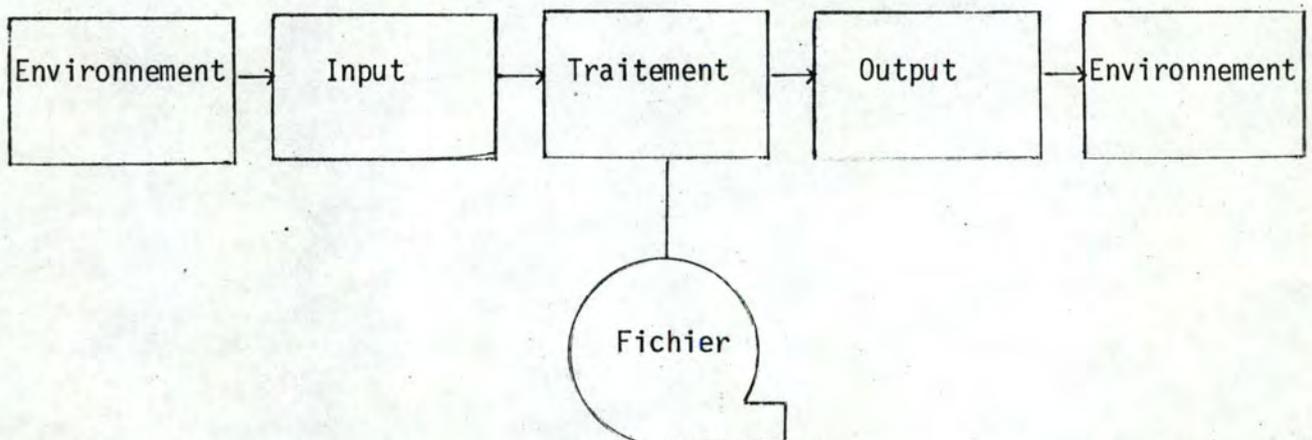
#### III.4.1. Etape de structuration

Le modèle du système informatique mis au point par le responsable de l'analyse permet de mettre en évidence les classes de données.

A partir des buts poursuivis par le projet, la compagnie, ou l'organisation, il est possible d'identifier les caractéristiques descriptives (identifiantes, intrinsèques, relationnelles) des données de chaque classe, ainsi que leurs spécifications (mais non leurs valeurs) *qui seront utiles à l'étape suivante de remplissage et de gestion.*

Cette première phase consiste à créer grâce à ces informations la structure du dictionnaire et les caractéristiques de chaque classe de données.

Par exemple, on aura déterminé le modèle suivant :



Et pour la classe de données FICHER nous aurons les caractéristiques suivantes :

NOM DE CLASSE DE DONNEES : FICHER - ELEMENT

CARACTERISTIQUES IDENTIFIANTES :

- nom usuel : = X(30) ;
- Description informelle : = texte (2,60) ;
- Responsable : = A(30) ;
- Synonymie : = X(30) ;
- Version : = (proposé, concurrent, approuvé, effectif).

CARACTERISTIQUES INTRINSEQUES :

- longueur : = 9999 ;
- mode : = (Packé, non Packé, Binaire) ;
- alignement : = (droite, gauche) ;
- unité : = (temps, mesure, argent, ...) ;
- séquence : = (SAM, ISAM, ..., ou explicite) ;
- valeurs nulles : = texte (1.60) ;
- sécurité : = texte (3.60) ;
- règle d'édition : = texte (5.60).

CARACTERISTIQUES RELATIONNELLES :

- avec les données décrites dans la classe fichier ;
- avec les traitements décrits dans la classe traitement.

#### III.4.2. Etape de remplissage et de gestion

Cette étape consiste dans une première phase à insérer au sein du dictionnaire la description des données en meublant le moule (étape 1) et dans une deuxième phase à maintenir et gérer ces descriptions en faisant des mises à jour, ajouts, modifications, suppressions, etc.

Exemple :

ACTION : Générer une donnée : "Fichier-élément".

RESULTAT : NOM DE CLASSE DE DONNEES : "Fichier-élément".

CARACTERISTIQUES IDENTIFIANTES :

- nom usuel = date ;
- description informelle = jour/mois/année ;
- responsable = opérateur ;
- synonymie = DATUM ;
- version = effectif.

CARACTERISTIQUES INTRINSEQUES :

- longueur = 0006 ;
- mode = non packé ;
- alignement = droite ;
- unité = temps ;
- séquence = jour suivi de mois suivi d'année ;
- valeurs nulles = 29/2, 30/2, 31/2, 31/4, 31/6; 31/9, 31/11 ;  
lorsque le reste de 19AA/4 = 0 ; il existe un 29/2 ;
- sécurité : JJ  $\leq$  31  $\neq$  0 ; MM  $\leq$  12  $\neq$  0 ; AA  $\leq$  99.

CARACTERISTIQUES RELATIONNELLES :

- lié à TRAITEMENT EDITION.

ACTION : Supprimer la synonymie "DATUM" de "Fichier-élément" de la donnée "DATE".

ACTION : Modifier - la longueur = 8 ;  
 - la sécurité = JJ  $\neq$  0, MM  $\neq$  0, 19AA < 1999, JJ  $\leq$  31, MM  $\leq$  12;  
 - la règle d'édition = "99/99/'19'99"  
 de la donnée "DATE" de "Fichier-élément".

RESULTAT : pour les deux actions :

NOM DE CLASSE DE DONNEES : Fichier-élément.

CARACTERISTIQUES IDENTIFIANTES :

- nom usuel = DATE ;
- description informelle = jour/mois/année ;
- responsable = opérateur ;
- version = effectif.

#### CARACTERISTIQUES INTRINSEQUES :

- longueur = 0008 ;
- mode = non packé ;
- alignement = droite ;
- unité = temps ;
- séquence = jour suivi de mois suivi d'année ;
- valeur nulle = 29/2, 30/2, 31/2; 31/4; 31/6; 31/9, 31/11 ;  
lorsque le reste de  $19AA/4 = 0$ , il existe un 29/2 ;
- sécurité =  $JJ \neq 0$ ,  $MM \neq 0$ ,  $19AA < 1999$ ,  $JJ \leq 31$ ,  $MM \leq 12$  ;
- règles d'édition = "99/99/'19'99".

#### CARACTERISTIQUES RELATIONNELLES :

lié à TRAITEMENT-EDITION.

#### III.4.3. Etape d'exploitation

Le dictionnaire doit encore être exploité à des fins de "management and program uses". Cette étape relève plus d'une exploitation normale d'une "base de données" que du fait spécifique que ce soit un dictionnaire. Il suffit pour s'en convaincre de se référer aux usages cités précédemment.

En bref, le cycle de vie détaillé d'un dictionnaire de données se trouve repris sur le schéma 3.7.

#### III.4.4. Conclusions

En fonction de ce cycle de vie, on pourrait concevoir des petits langages capables de répondre aux exigences de chacune des étapes. Ceux-ci seraient :

- un langage de structuration permettant de décrire la structure du dictionnaire (e.a. Luguët) (1) ;
- un langage de remplissage et gestion permettant de gérer le contenu de cette structure : ce langage existe dans tous les logiciels de gestion de base de

---

(1) LUGUËT, J. et al. : *Langage d'analyse et de conception des systèmes d'information automatique.*

données (NULL, SOCRATE, CODASYL, ...) (1) ;

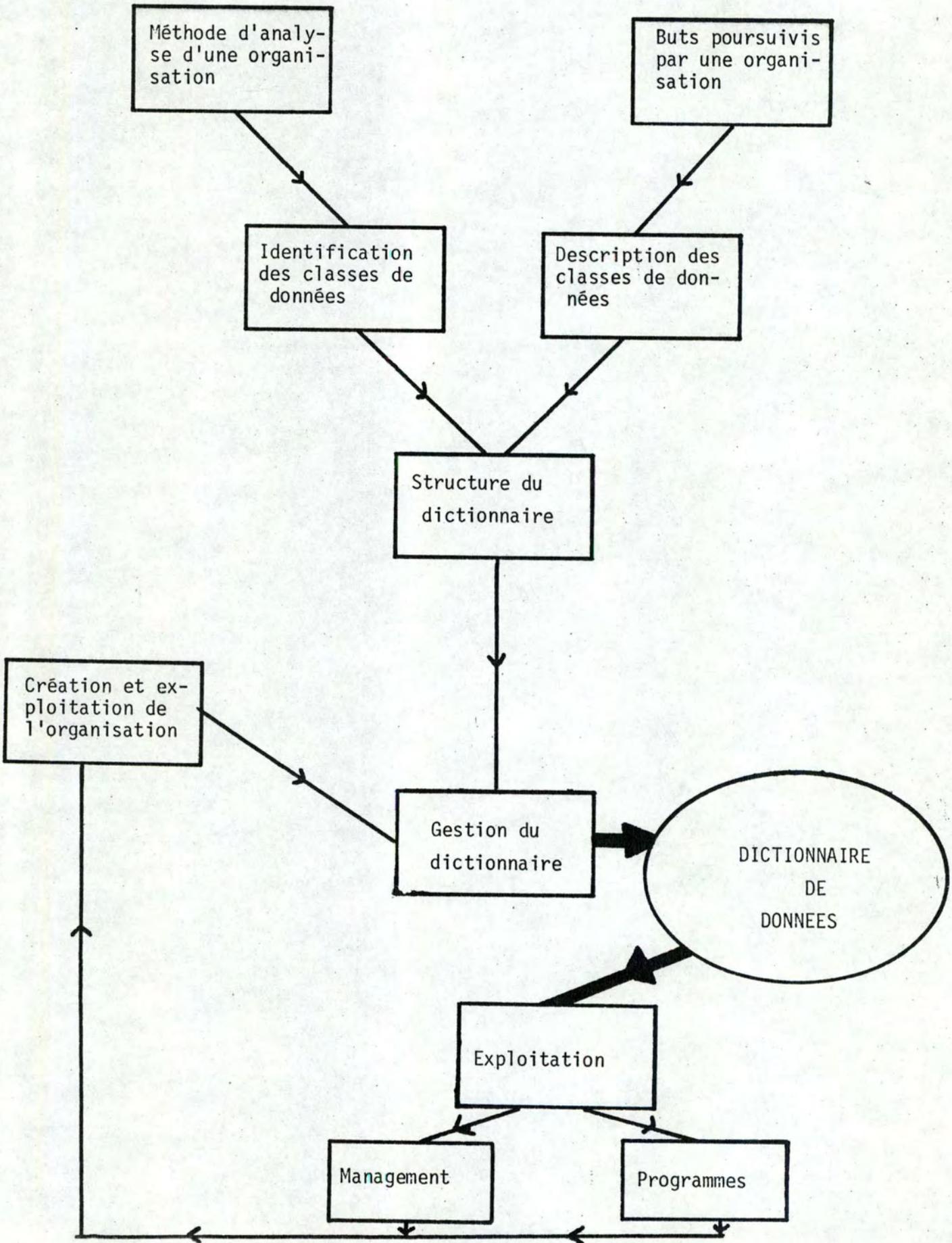
- un langage d'exploitation permettant d'interroger le dictionnaire ; de nombreux langages dits d'interrogation (query language) ont également été mis au point.

Si l'on est plus ambitieux, on peut concevoir un langage global pour ces trois étapes, c'est ce que l'on appelle un langage d'analyse dont le "P.S.L." lié au projet ISDOS est un exemple (2).

°  
° °

- 
- (1) DEHENEFFE, C. et al. : *NUL : A Navigational User's Language for a Network structured Data Base.*  
ABRIAL, J.R. et al. : *Projet Socrate.*  
RANDAL, H. et al. : *A Schema Report Facility for a Codasyl based Data Definition Language.*
- (2) LESUISSE, R. : *Exposé introductif au projet ISDOS.*  
LABOUREUR, L. : *Le développement de la documentation des projets et des programmes.*

Tableau 3.7. : Articulation générale



## CONCLUSIONS

Notre étude visait à faire une synthèse critique de l'existant en matière de dictionnaire de données.

Pour ce faire, nous avons d'abord dans un premier chapitre défini ce qu'était un dictionnaire de données et établi une "grille de lecture" commune pour les différents dictionnaires.

Ensuite - et ce fut l'objet du deuxième chapitre - nous avons effectué une analyse méthodique et détaillée des contenus d'une part et des usages d'autre part, de quatre dictionnaires représentatifs de l'ensemble des dictionnaires existants. Nous avons utilisé à cet effet une "grille de lecture" commune qui devait nous permettre ultérieurement de comparer entre eux les différents contenus et usages.

Les résultats de notre recherche peuvent se résumer en quelques points.

Pour ce qui concerne les contenus du dictionnaire, il est apparu que :

1. les modèles de systèmes informatiques sous-jacents à chaque dictionnaire étaient différents; nous avons proposé un modèle simple qui, à notre avis, les englobe tous.
2. un dictionnaire de données décrit généralement le système informatique, mais non le système d'informations.
3. les descriptions qui sont faites au sein du dictionnaire sont essentiellement statiques.
4. il est peu tenu compte dans les descriptions des informations nécessaires pour gérer le suivi d'un projet.

5. seul Rémora prend en compte des systèmes complexes, les autres dictionnaires sont basés sur une conception plus ancienne de l'informatique.
6. la plus importante découverte est celle-ci : ce sont les usages d'un dictionnaire qui en déterminent la richesse du contenu et dissocier les deux, risque de compromettre la compréhension.

En ce qui concerne les usages, nous avons, tant pour le management que pour les programmes, proposé de nouvelles utilisations du dictionnaire de données. En effet, en confrontant entre eux les usages faits dans différentes organisations, nous nous sommes aperçus que les responsables n'exploitaient qu'en partie la masse d'informations regroupée au sein du dictionnaire et qu'il était possible, par exemple, d'étendre son usage à :

- des rapports sur l'état des ressources 'homme' et 'machine' ;
- des programmes de contrôle d'intégrité, etc.

Enfin cette synthèse nous a permis de mettre à jour un cycle de vie commun à tous les dictionnaires. Ce cycle de vie contient trois étapes : une étape de structuration, une étape de remplissage et de gestion et une étape d'exploitation du dictionnaire.

°  
° °

BIBLIOGRAPHIE

ABRIAL, J.R. et al. : *Projet Socrate*, Université Scientifique et Médicale de Grenoble, Grenoble, 1972

Automobiles Peugeot : *Package Minos III A : Guide de l'utilisateur*, sl., sd.

BODART, F. : *Introduction à l'analyse fonctionnelle des systèmes informatiques de gestion*, Cours, Namur, 1976

BODART F. : *Structuration fonctionnelle du système informatique d'une organisation : Proposition méthodologique*, in Séminaire sur l'informatique d'organisation, les systèmes d'information et de décision, texte des communications IRIA, pp. 347-369, Paris, 1975

British Rail : *Data Dictionary*, Londres, 1974

CODD, E.F. : *A Relational Model of Data for Large Shared Data Bank*, IBM Research Laboratory, in *Communication of ACM*, Vol. 13, n° 6, San Jose, California, 1970

DEHENEFFE, C. et al. : *NUL : A Navigational User's Language for a Network Structured Data Base*, in SIGMOD 76 (ACM), edited by James B. Rothnie, Washington DC, June 2-4, 1976

Eastern Air Lines : *Data Base Directory : System Manual*, Miami, 1973

E.D.P. Analyzer : *The Data Dictionary/Directory Functions*, Vol. 12, N° 11, November 1974

E.D.P. Analyzer : *The 'Data Administrator' Functions*, Vol. 10, N° 11, 1972

LABOUREUR, L. : *Le développement de la documentation des projets et des programmes : application des concepts I.T.P.*, sl., mars 1975

LANGEFORS, B. : *Information Systems*, in *Information Processing*, pp. 937-945, Stockholm, 1974

LESUISSE, R. : *Exposé introductif au projet ISDOS*, in *Bulletin de Liaison du Club Banques de Données*, IRIA, sl, juin 1976

LUGUET, J. et al. : *Langage d'analyse et de conception des systèmes d'information automatique : Principe d'analyse conversationnelle dans S.C.A.P.F.A.C.E.*, in *Séminaire Inforsid Groupe II*, IRIA, Pont-à-Mousson, mai 1976

Management Systems and Programming Ltd. : *"Datamanager" Product Description*, reference DMS-001, Londres, 1974

Management Systems and Programming Ltd. : *"Datamanager" Product Description*, Londres, November 1975

RANDALL, H. et al. : *A Schema Report Facility for a Codasyl based Data Definition Language*, Sperry Univac, Roseville, Minesota, sd.

ROLLAND, C. et al. : *Engineering of Management Informatic System*, in *Congress Book*, sl., Octobre 1974

ROLLAND, C. et al. : *Méthodologie de Rémora*, Nancy, 1975

ROLLAND, C. et al. : *La génération dans Rémora*, Nancy, 1975

ROLLAND, C. et al. : *Présentation générale du projet Rémora*, Nancy, sd.

ROLLAND, C. et al. : *La base des objets élémentaires de Rémora*, Nancy, sd.

ROLLAND, C. et al. : *Etude partielle du langage de conception de Rémora*, Nancy, sd.

ROLLAND, C. et al. : *Système d'édition dans le projet Rémora*, in *BIGEN*, pp. 13-16, Nancy, sd.

ROLLAND, C. et al. : *Description d'un cas concret d'application de Rémora*, Nancy, sd.

- ROLLAND, C. et al. : *Un exemple d'application de la méthode Rémora*, Nancy, 1975
- ROLLAND, C. et al. : *La phase de conception de Rémora*, Nancy, 1975
- ROLLAND, C. et al. : *Les différentes phases de la construction d'un système d'information à l'aide de Rémora*, Nancy, 1975
- SMITH, R.L. : *Statement of Work for the Structured Programming System, for RADC, under contract n° F 30602-74-C-0186*, sd., sl.
- TEICHROEW, D. et al. : *Automation of System Building*, in *Datamation*, pp. 25-30, sl., August 15, 1971
- THULY, J. : *Présentation générale de la méthode Minos*, sl., sd.
- UHROWCZIK, P.P. : *Data Dictionary/Directory*, in *IBM System Journal*, Vol. 4, pp. 332-350, sl., 1973
- University Computing Company : *Data Dictionary and Manager*, New Jersey, 1973.