

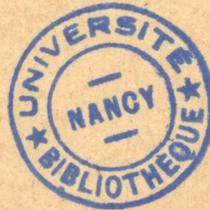
Dactyl

UNIVERSITE DE NANCY 1

U.E.R. DE SCIENCES MATHÉMATIQUES

Sc. N. 72/80 A

**SOFTWARE D'AIDE
A L'ANALYSE ET A LA
PROGRAMMATION DE GESTION**



T H E S E

pour l'obtention du

DOCTORAT DE SPECIALITE MATHÉMATIQUES (3ème CYCLE)

soutenu devant le jury le 21 octobre 1972

par

DANIEL ORY

JURY : Mr LEGRAS J. Président
Mr DEPAIX M. Examineur
Mme ROLLAND C. Examineur

UNIVERSITE DE NANCY 1

U.E.R. DE SCIENCES MATHÉMATIQUES

**SOFTWARE D'AIDE
A L'ANALYSE ET A LA
PROGRAMMATION DE GESTION**

par

DANIEL ORY

Je tiens à exprimer toute ma gratitude à Monsieur le Professeur LEGRAS, Directeur de l'Institut Universitaire de Calcul Automatique de NANCY, pour la bienveillance et l'aide constante qu'il n'a cessé de me témoigner et pour la formation que j'ai reçue à l'Institut.

Je remercie vivement Monsieur le Professeur DEPAIX pour l'honneur qu'il me fait en participant au jury.

Ce travail a été réalisé sous la direction de Madame ROLLAND à qui j'exprime ma profonde reconnaissance pour les conseils et les encouragements amicaux qu'elle m'a prodigués tout au long de cette étude.

L'étude du cas de paie a été le fruit d'un travail commun avec Monsieur AUBERT de la Société Electro-Vacher ; qu'il trouve ici l'expression de tous mes remerciements pour son aide et sa sympathique collaboration.

Je remercie également tout le personnel de l'I.U.C.A. et tout particulièrement Madame LECLERC qui a assuré avec rapidité et soin la réalisation matérielle de cette thèse.

SOMMAIRE

Introduction

Première partie : Etude d'un cas de paie

Chapitre I : Présentation du problème

- | | |
|--|-------|
| 1. Introduction | I - 1 |
| 2. La procédure manuelle existante | |
| 2.1. Calcul de la rémunération de chaque agent | I - 1 |
| 2.2. Les documents récapitulatifs | I - 6 |
| 3. Les impératifs de l'automatisation | I - 8 |

Chapitre II : La solution adoptée

- | | |
|--|---------|
| 1. Présentation générale | |
| 1.1. Le matériel utilisé | II - 10 |
| 1.2. Eléments fixes et éléments variables | II - 10 |
| 1.3. Principe général de la chaîne de traitement | II - 10 |
| 2. Mode de description de l'analyse | II - 11 |
| 3. L'analyse de la chaîne de traitement | II - 12 |

Chapitre III : Conclusion

- | | |
|--|----------|
| 1. Les difficultés rencontrées | |
| 1.1. au niveau de la connaissance du problème | III - 24 |
| 1.2. au niveau de l'analyse et de la programmation | III - 25 |
| 2. Les solutions proposées | |
| 2.1. Pour l'étude de l'existant | III - 25 |
| 2.2. Pour l'analyse et la programmation | III - 26 |

Deuxième partie : Les Outils "software":

.../...

Chapitre IV : Système d'édition

1. Introduction	
1.1. Les objectifs du système	IV - 27
1.2. Le principe du système	IV - 29
2. Schéma général du système	IV - 30
3. Descripteur d'état et générateur associé	
3.1. Le descripteur d'état	IV - 32
3.2. Perforation des bordereaux et préparation des cartes en vue de la génération	IV - 45
3.3. Le générateur de description	IV - 46
3.4. Utilisation de la description générée dans un programme d'édition	IV - 54
4. Les programmes de service	
4.1. Programme d'édition de la description générée	IV - 60
4.2. L'éditeur de maquette	IV - 62
4.3. Modification de description	IV - 69
5. Conclusion	IV - 91

Chapitre V : La sous-compilation

1. Introduction	V - 93
2. Les mécanismes de la sous-compilation	V - 94
3. Exemples d'utilisation	V - 97
4. Possibilités et limites de la sous-compilation	V - 99

Chapitre VI : La table des salaires :

1. Introduction	VI - 101
2. Le document d'analyse	
2.1. Structure de la table	VI - 102
2.2. Représentation des actions à entreprendre	VI - 105

.../...

2.3. Les documents annexes	VI - 110
2.4. Un exemple d'utilisation	VI - 111
3. Le passage à une méta - instruction	
3.1. La programmation	VI - 115
3.2. Exemple	VI - 117
4. La génération des instructions de calcul	
4.1. Les documents annexes	VI - 119
4.2. La génération du sous-programme de calcul	VI - 126
5. Conclusion	VI - 139

Conclusion

Annexes de la première partie

Annexe I : Les documents d'entrée
Annexe II : Les fichiers
Annexe III : Les documents de sortie
Annexe IV : Programmes - Organigrammes.

INTRODUCTION

Ce travail est constitué de deux parties .

La première est une étude de cas ; il s'agit de l'automatisation de la paie d'une entreprise privée. Après la présentation du problème, nous décrivons la solution qui a été adoptée puis les difficultés rencontrées pendant son élaboration.

L'étude critique de ces difficultés a servi de point de départ pour la recherche d' "outils" facilitant l'analyse et la programmation, sous la forme d'un software d'aide à l'analyse et à la programmation.

Ce software est développé dans la seconde partie. On trouve tout d'abord un système d'édition visant à faciliter la mise au point des documents, la rédaction et la maintenance des programmes d'édition.

Nous exposons ensuite une technique de programmation Cobol adaptée à la maintenance et permettant l'utilisation de sous-programmes.

Enfin nous donnons une première ébauche d'un système d'aide à l'analyse et à la programmation spécifiquement adapté à la fonction 'paie'.

P R E M I E R E P A R T I E

ETUDE D'UN CAS
DE PAIE

C H A P I T R E I

PRESENTATION DU PROBLEME

1. Introduction :

L'objet de cette étude est l'automatisation de la paie des agents d'une entreprise, en se restreignant à ce problème particulier c'est-à-dire sans gestion du personnel ni intégration dans une comptabilité automatisée.

L'entreprise intéressée est une petite entreprise d'une centaine de personnes spécialisée dans la vente et l'installation de matériel électrique ; aucun équipement électronique n'est actuellement utilisé dans cette maison, la paie étant la première procédure que l'on ait tenté d'automatiser.

L'étude de l'existant a été réalisée dans le service comptable lui-même sous forme d'enquêtes orales.

2. La procédure manuelle existante :

Elle se compose essentiellement de deux parties :

- le calcul de la rémunération due à chaque agent (calcul du salaire brut, des retenues sociales, du salaire net et de la somme à payer) et l'écriture de son bulletin de salaire ;
- l'élaboration et l'écriture des documents récapitulatifs : le journal des salaires, la fiche individuelle comptable, les récapitulatifs des retenues sociales, la liste des salaires bruts et un état bancaire donnant la liste des sommes à verser.

2.1. Calcul de la rémunération de chaque agent :2.1.1. Les grandes options adoptées dans l'entreprise

On distingue, dans cette entreprise, deux catégories d'agents :

- les salariés qui sont les agents productifs,
- les employés ;

cette distinction étant surtout utile du point de vue comptable.

Employés comme salariés sont soit des "horaires" payés au nombre d'heures effectuées par semaine sur la base d'un taux horaire fixé, soit des "mensuels" percevant une rémunération mensuelle dite salaire mensuel de base.

.../...

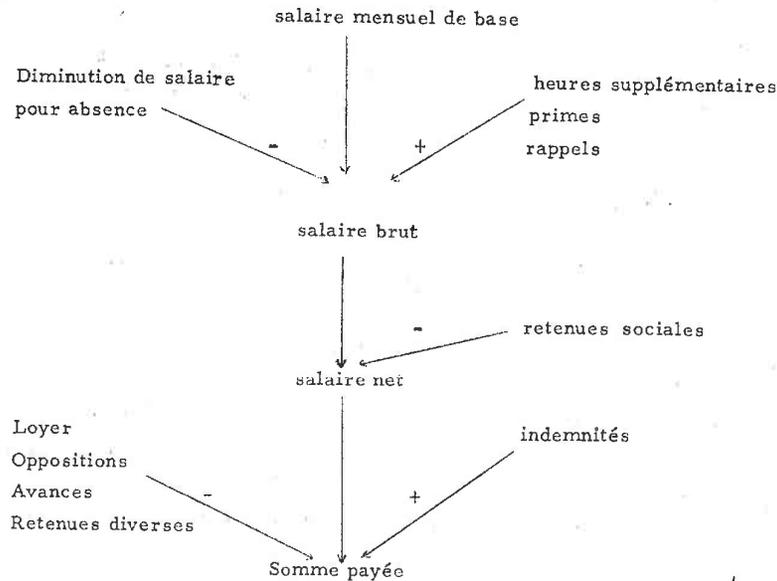
Toutefois, d'après la législation actuellement en vigueur, les agents horaires de cette entreprise sont mensualisés ; leur salaire mensuel de base, calculé sur un horaire hebdomadaire de 40 heures est égal au douzième de la rémunération annuelle,

soit $\frac{52 \times 40}{12} \times \text{taux de l'heure}$ ou $173 \text{ h } 1/3 \times \text{taux de l'heure}$.

Cependant, le salaire mensuel de base sert seulement de référence ; il est contrôlé tous les mois et peut être diminué ou augmenté selon le travail fourni ; en particulier le service de paie dispose, pour chaque agent horaire, de feuilles d'activité hebdomadaires donnant les heures effectuées pendant la semaine.

Enfin, si le mois en cours se termine au milieu d'une semaine, la semaine incomplète n'est pas prise en compte et sera intégrée au mois suivant.

2.1.2. Le schéma synoptique de ce calcul



.../...

2.1.3. Calcul du salaire brut :

- Pour un agent mensuel, le salaire brut peut se décomposer de la façon suivante :

+/-	Eléments	Calcul
+	Salaire mensuel de base	montant
-	Diminution de salaire pour absence	montant
+	Primes et heures supplémentaires	montant
+	Rappels	montant

- Pour un agent horaire :

+/-	Eléments	Calcul
+	Salaire mensuel de base	173 h 1/3 x taux horaire
-	Diminution pour absence	} ①
+	Heures supplémentaires	
+	Heures de route	nombre x taux h oraire
+	Heures de nuit et dimanche (heures à 100 %)	nombre x 2 x taux horaire
+	Primes de panier	} montants
+	Autres primes	
+	Rappels en somme	
+	Rappels en heures	

① La rémunération des heures est calculée, au niveau de chaque semaine, à partir du salaire mensuel de base grâce aux feuilles d'activité :

Pour chaque semaine du mois :

- si le nombre d'heures effectuées est inférieur à 40, on effectue une "retenue pour absence" en portant en deuxième ligne la différence entre 40 heures et le nombre d'heures effectuées.
- sinon, on indique en troisième ligne les heures supplémentaires : les huit premières sont payées au taux majoré de 25 % (heures à 25 %,) les heures suivantes au taux majoré de 50 % (heures à 50 %)

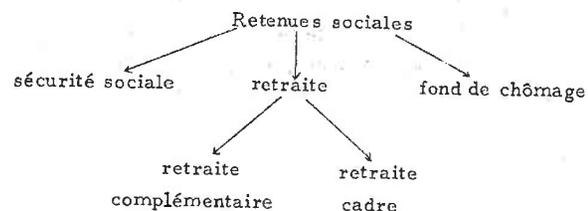
② On donne le taux de l'heure de rappel (qui n'est pas obligatoirement le taux horaire actuel) ainsi que le nombre d'heures de chaque catégorie

.../...

(heures normales, à 25 %, à 50 % et à 100 %) ; un agent peut avoir plusieurs rappels à des taux différents pour un même mois.

2.1.4. Calcul des retenues sociales :

Les retenues sociales sont de quatre types :



Elles se calculent toutes de la même manière :

on définit d'abord la base de cotisation en fonction du salaire brut puis la "base plafonnée" égale à la base si cette dernière est inférieure au plafond, égale au plafond dans le cas contraire ;

la base, éventuellement plafonnée, est ensuite multipliée par le taux de retenue correspondant.

- Sécurité Sociale :

Eléments	condition	montant/taux	Plafond
Base	employé salarié	salaire brut salaire brut - 10 %	
Maladie		1 %	
Maladie		2,5 %	1.830 F
Vieillesse		3 %	1.830 F

- Retraite complémentaire : les apprentis et les agents agés de moins de 21 ans ne cotisent pas.

.../...

Eléments	condition	montant/taux	plafond
Base		salaire brut	
Retraite complémen- taire	ouvrier	1.68 %	1.830 F
	ETAM	2.10 %	
	cadre	4 %	
Prévoyance	ouvrier	0.40 %	
	ETAM	0.60 %	

- Retraite Cadre : seuls les cadres y sont affiliés.

On introduit, comme montant de base, le "différentiel", quantité égale au salaire brut diminué du plafond de la retraite complémentaire, éventuellement plafonné.

Eléments	condition	montant/taux	plafond
Base	petit cadre	différentiel	5.680 F
	cadre supérieur	différentiel	
Retraite	petit cadre	4 %	
	cadre supérieur	6 %	

- Chômage : la base est la base sécurité sociale mais les apprentis ne cotisent pas ; le taux est unique et fixé à 0,08 %

Rappelons que le salaire net est obtenu en déduisant du salaire brut les retenues sociales que nous venons d'expliquer.

2.1.5. Calcul de la somme à payer :

Il reste à tenir compte des éléments de salaire qui ne sont pas imposables ; ils sont tous donnés sous forme de montants ; on ajoute au salaire net les indemnités ; on lui soustrait loyer, oppositions, avances et retenues diverses.

2.1.6. Le bulletin de salaire :

Légalement obligatoire, il doit porter, outre les indications sur l'entreprise et l'agent, tous les éléments constituant le salaire (il doit permettre à l'agent de rétablir le calcul de son salaire).

.../...

2.2. Les documents récapitulatifs :

Leur forme et leur contenu ont été définis par le service de paie en fonction des règles légales et des usages internes en vigueur.

2.2.1. Le journal des salaires :

Il est obtenu par carbone à partir des bulletins de salaire.

2.2.2. La fiche individuelle comptable :

Il en existe une pour chaque agent ; à usage exclusivement interne, elle porte l'image des bulletins de salaire de l'agent depuis le début de l'année elle est complétée mois par mois par carbone à partir du bulletin.

2.2.3. La liste des salaires bruts :

C'est un récapitulatif des salaires bruts des employés et des salariés dans l'ordre de leurs matricules avec séparation des deux catégories et indication de chaque cumul ; il a un intérêt exclusivement comptable.

2.2.4. L'état bancaire :

Ce document est transmis à la banque de l'entreprise pour qu'elle assure le paiement effectif des salaires ; c'est une liste des agents payés par virement postal ou bancaire, classés par mode de paiement et donnant toutes les caractéristiques nécessaires au paiement (nom, numéro de compte, agence bancaire ou C.C.P. et somme) ; elle est complétée par la liste des agents payés en liquide avec indication de la somme cumulée.

2.2.5. Récapitulatifs des retenues sociales :

Ce document porte, pour chaque type de cotisation, la base cumulée obtenue en ajoutant les bases de cotisations de tous les agents, la part salariale, la part patronale et la cotisation totale.

La cotisation totale est obtenue en appliquant à la base cumulée le taux correspondant ; la part salariale est le cumul des cotisations versées par les agents, la part patronale est obtenue par différence.

.../...

Caisse	Désignation de l'élément	Bases à cumuler	taux de cotisation totale
Sécurité Sociale	Maladie sur base totale	base S. S.	3 %
	Maladie	base plafonnée	12.95 %
	Vieillesse		8.75 %
	Allocations familiales		10.50 %
	Accidents du travail		5.47 %
Retraite Complémentaire	Retraite ouvrier	base	4.20 %
	Retraite ETAM		4.20 %
	Décès ETAM		1 %
	Prévoyance ETAM	retraite	1.30 %
	Prévoyance ouvrier		1.75 %
	Retraite agent cadre		base plafonnée
	Décès cadre	différentiel	2.8 %
	Prévoyance cadre	base plafonnée	1.5 %
Retraite cadres	Retraite petits cadres	différentiel	8 %
	Retraite cadres supérieurs	différentiel plafonné	16 %
ASSEDIC	Chômage	base chômage	0.40 %
Congés Payés	Congés payés	brut - panier	15.8 %
	Comité de prévention	brut - panier (1)	0.10 %
	Chômage intempéries	saire brut (1)	on ne donne que la base
	(1) seulement si l'agent est inscrit.		

.../...

3. Les impératifs de l'automatisation :

Utiliser l'ordinateur pour automatiser une paie, c'est lui faire transposer les méthodes anciennes même plusieurs fois améliorées mais sans modifier l'esprit de la solution, c'est-à-dire en obtenant les mêmes résultats par une démarche logique analogue ; cette transposition est d'autant plus accentuée que la paie est une procédure rigide où les méthodes de calcul sont réglementées et le contenu d'un certain nombre de documents imposé.

Les avantages d'une telle utilisation de l'ordinateur sont des avantages "administratifs" : un coût moindre et une sécurité plus grande.

Pour atteindre ces objectifs, il nous a semblé indispensable d'accorder une attention particulière à la sécurité et à la réduction du coût de la saisie :

1. En utilisant la possibilité de l'ordinateur de garder en mémoire, aussi longtemps que nécessaire, toute information constante qu'il suffira de saisir une seule fois au moment le plus propice, réduisant ainsi les éléments demandés.

2. En concevant des documents de saisie :

- d'un emploi simple et rapide,
- où la quantité d'informations à inscrire est réduite au minimum (on ne demandera tous les mois que les renseignements variables et spécifiques à ce mois),
- qui soient des bordereaux de perforation de façon à limiter les erreurs de transcription,
- qui permettent une plus grande fiabilité de la transmission en diminuant les informations parasites et les erreurs d'interprétation par la pré-impression par l'ordinateur de notions essentielles (matricules, nom, libellés explicatifs ...).

Ces dispositions prises sur les documents d'entrée ont pour but d'augmenter la qualité des données reçues de l'extérieur de façon à réduire le coût de l'opération ; en effet, si le nombre des erreurs est grand, les corrections sont nombreuses, coûteuses en personnel, documents de saisie et heures machine, les délais sont plus longs.

.../...

3. En validant les données transmises par des programmes de détection d'erreurs de façon à éviter les 'blocages' au cours de l'exécution dont les conséquences sont fâcheuses :

en effet, malgré les précautions prises en particulier par la conception des documents de saisie, il reste des risques d'erreurs. Pour éviter l'utilisation du programme de calcul sur des données erronées, nous en assurons le contrôle par programme ; les contrôles utilisés sont :

- des contrôles de dualité sur la validité des codes (correspondance entre matricule et nom par exemple) ;
- des contrôles de syntaxe ou de structure sur le type et la présence des différentes zones du message d'entrée ;
- des contrôles de vraisemblance sur les zones de l'enregistrement d'entrée (un agent est horaire ou mensuel) ;
- des contrôles de compatibilité entre les zones de l'enregistrement ou entre cet enregistrement et l'enregistrement d'un autre fichier.

Il reste néanmoins que, quelque soient les contrôles effectués, certaines erreurs ne sont pas détectables par programme (par exemple une erreur de chiffre sur un document de saisie) ; les seules vérifications possibles sont alors des contrôles à posteriori amenant des corrections coûteuses et peu sûres.

Il semble donc nécessaire de motiver le personnel du service en l'incitant à assurer une partie des responsabilités du contrôle de la saisie.

Cependant cette recherche du moindre coût ne doit pas nuire à la souplesse et à l'efficacité du système :

- les usages internes de l'entreprises doivent être respectés ; en particulier les documents d'entrée et de sortie doivent répondre aux desiderata du service de paie ;
- la paie devant être terminée à une date impérative, il est nécessaire de prévoir des procédures d'exception permettant la correction rapide d'une erreur.

C H A P I T R E I I

LA SOLUTION ADOPTÉE

1. Présentation générale :

1.1. Le matériel utilisé :

L'analyse a été réalisée en vue d'une mise en exploitation sur l'ordinateur CII - 10070 système BPM de l'Institut de Calcul ; les périphériques disponibles sont un lecteur de cartes, une imprimante, des disques et des dérouleurs de bandes magnétiques.

Nous avons écrit les programmes en langages Cobol 65, utilisant de plus les programmes standards de tri, copie de bandes, changement de support ainsi qu'un programme général de mise à jour de fichiers nommé CLAMS.

1.2. Eléments fixes et éléments variables :

Afin d'utiliser au mieux les possibilités de stockage de l'ordinateur, les informations qui permettent de calculer la paie d'un agent ont été réparties en deux sous-ensembles :

- les éléments fixes : ce sont les renseignements concernant l'état-civil de l'agent, sa situation professionnelle (horaire, cadre ...) et sa rémunération (taux horaire, salaire mensuel de base) ;

ces éléments, s'ils peuvent être modifiés au cours du temps, présentent tout de même une certaine stabilité.

Ils sont stockés dans les enregistrements du fichier du personnel.

- les éléments variables : ce sont les éléments propres au mois considéré tels que nombre d'heures, montant des primes ... ; ils sont donnés chaque mois.

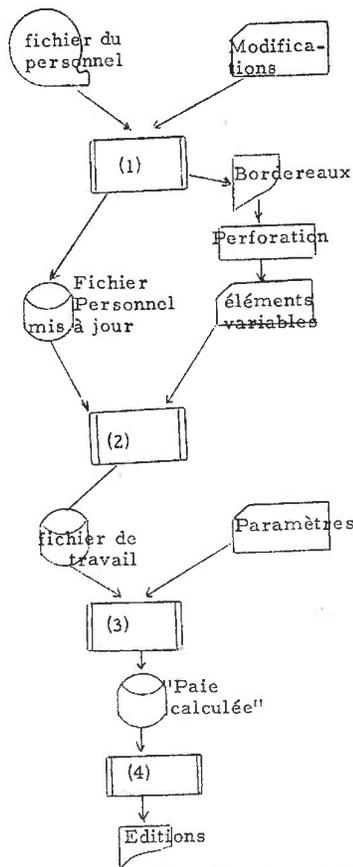
1.3. Principe général de la chaîne de traitement :

La chaîne de traitement est composée de quatre phases qui sont exécutées chronologiquement dans cet ordre :

- (1) Mise à jour du fichier du personnel,
- (2) Vérification et prise en compte des éléments variables du mois,
- (3) Calcul de la paie,
- (4) Edition des bulletins de salaire et des documents comptables ;

.../...

Elle est schématisée par l'organigramme suivant :



(1) il s'agit de prendre en compte l'arrivée ou le départ d'un agent ou bien les modifications intervenant dans sa situation ; le fichier personnel mis à jour permet alors d'éditer les bordereaux. où seront portés les éléments variables.

(2) les bordereaux précédents étant transcrits sur cartes perforées, les contrôles habituels de structure et de vraisemblance sont effectués ; on vérifie ensuite la comptabilité entre les cartes et le fichier du personnel ; simultanément on constitue un fichier de travail qui contient toutes les informations permettant le calcul de la paie .

(3) les différents taux de retenues et les plafonds ont été considérés comme des paramètres et groupés dans un 'fichier des paramètres' dont la mise à jour ne pose pas de problème . A partir de ce fichier et du fichier de travail, on calcule la paie de chaque agent pour constituer un fichier de la 'paie calculée'.

(4) Utilisant ce fichier, on édite les bulletins de salaires et tous les documents comptables exigés.

2. Mode de description de l'analyse :

L'analyse que nous proposons dans les pages suivantes est décrite de façon modulaire ; cette méthode de description est née de la remarque qu'un esprit humain ne peut maîtriser un problème complexe qu'en le décomposant en problèmes partiels élémentaires, chaque problème partiel pouvant être à son tour, pour une meilleure compréhension, partagé en tâches plus

.../...

élémentaires. La notion de description modulaire s'impose alors pour que le système de description puisse suivre les mécanismes de compréhension.

L'analyse apparaît ainsi sous forme d'une suite de modules décrivant chacun un problème partiel ou une tâche ; chaque module peut à son tour être décomposé en une suite de nouveaux modules qui en assurent une définition plus précise ; le niveau de décomposition est repéré par un nombre.

Cette méthodologie a également l'avantage d'incorporer sans heurt, dans une même description, les procédures administratives et les procédures informatiques.

L'expérience nous a montré qu'elle permettait en outre un meilleur dialogue entre informaticiens et non spécialistes.

Remarque : L'analyse fait référence à des documents ou fichiers repérés par leur code et dont les descriptions que nous n'avons pas jugé utile de donner ici ont été reportées en annexe ; notons toutefois que le code E caractérise un document d'entrée, F un fichier et S un document de sortie.

3. L'analyse de la chaîne de traitement :

.../...

1 - En utilisant les documents éventuellement préimprimés par l'ordinateur, le service de paie complète :

- les fiches d'embauche (E₁) et les fiches de modifications (E₂) concernant les enregistrements du fichier personnel (F1) du mois précédent.

- le bordereau (E₃) donnant les nouvelles valeurs des paramètres de paie ;

- le bordereau (E₄) permettant de mettre à jour la table des modes de paiement.

Envoi au Centre de Calcul.

2 - A partir des fiches (E₁) et (E₂) transcrites sur cartes, mise à jour du fichier personnel (F1) et contrôle de la validité des modifications passées ; édition, à partir du nouveau fichier (F1.1) des bordereaux de perforation (E₅) des cartes "activité" et (E₆) des cartes 'supplémentaires' ; envoi au service de paie.

3 - Mise à jour et contrôle du fichier des paramètres (F5) à partir du bordereau (E₃) ; mise à jour de la table (F9) des modes de paiement à partir du bordereau (E₄).

4 - Le service de paie remplit les bordereaux de perforation (E₅), (E₆) et (E₇) portant les éléments variables de la paie (cartes 'activité', cartes 'supplémentaires' et cartes 'rappels') ; envoi au Centre de Calcul.

5 - A partir du nouveau fichier personnel (F1.1), vérification et prise en compte des cartes perforées à partir des bordereaux précédents ; création du fichier de travail (F4) servant au calcul de la paie.

Pour chaque agent, à partir du fichier précédent et du fichier des paramètres (F5) :

6 - Calcul des éléments de salaire et création d'un enregistrement du fichier (F6) de la paie calculée et des fichiers intermédiaires (F7) et (F8).

.../...

7 - Edition, à partir des fichiers (F6), (F7) et (F8) des bulletins de salaire et des documents comptables ; le fichier (F6) est archivé.

8 - A partir des documents édités, le service de paie contrôle la paie calculée ; si une erreur est détectée dans la paie d'un agent, il calcule les éléments de salaire corrects et les porte sur le bordereau (E₈) qui permet la mise à jour du fichier (F6) de la paie calculée.

Rédition, grâce au fichier mis à jour, des nouveaux documents comptables corrigés.

.../...

Module 1 : Préparation des modifications du fichier personnel1.1. Préparation des fiches d'embauche :

Cette fiche est utilisée à chaque arrivée d'un nouvel agent ; on y porte les éléments fixes concernant cet agent, éléments qui permettent la création d'un nouvel enregistrement du fichier du personnel.

1.2. Préparation des fiches de modification :

Chaque agent possède une fiche nominative préimprimée par l'ordinateur et qui doit être remplie :

- soit pour indiquer certaines modifications des rubriques de l'enregistrement du fichier personnel (on porte alors les nouvelles valeurs des zones à modifier à l'emplacement indiqué) ;
- soit au moment du départ de l'entreprise (l'indiquer sur la dernière ligne de la fiche) ; l'enregistrement correspondant du fichier personnel sera supprimé.

1.3. Préparation du bordereau (E3) :

On porte sur cet imprimé les nouvelles valeurs des paramètres de paie qui ont été modifiés.

1.4. Préparation du bordereau (E4) :

Il s'agit d'une table manuscrite; on peut en modifier les éléments par simple indication manuelle ; lorsque l'on désire ajouter un mode de paiement, il faut d'abord lui affecter un code (ce code est utilisé dans le fichier personnel).

.../...

Module 2 : Mise à jour du fichier du personnel2.1. Perforation et vérification des fiches (E₁) et (E₂) :

2.2. A partir des cartes précédentes triées par matricule d'agent et par code de mise à jour (adjonction, modification ou suppression), mise à jour du fichier personnel (F1) du mois précédent et sortie du sous-fichier (F2) des enregistrements modifiés ; cette mise à jour effectue la recherche des erreurs de structure qui sont éditées (T1).

2.3. Vérification complémentaire du fichier (F2) (contrôles de vraisemblance) ; seuls les enregistrements corrects sont conservés dans un fichier (F3) ; édition des erreurs détectées (T2) ; simultanément calcul du salaire mensuel de base des agents horaires dont le taux de l'heure a été modifié.

2.4. A partir du fichier (F3) précédent, édition des fiches de modification vierges destinées à remplacer les fiches utilisées.

2.5. Procédure de redressement des erreurs :

- dépouillement des documents (T1) et (T2) ;
- perforation des cartes permettant d'effectuer les modifications complémentaires désirées (ces cartes sont analogues aux cartes obtenues à partir de la fiche (E2)) ;
- retour en 2.2. à partir du fichier modifié .

Pour chaque agent du nouveau fichier personnel (F1.1), dans l'ordre des matricules :

2.6. Si l'agent est horaire, édition d'une ligne du bordereau de perforation (E5) des cartes 'activité' pour chaque semaine du mois considéré.

2.7. Edition d'une ligne du bordereau de perforation (E6) des cartes 'supplémentaires'.

.../...

Module 3 : Mise à jour du fichier des paramètres et de la table des modes de paiement

3.1. Perforation, à partir du bordereau (E₃), des cartes donnant les nouvelles valeurs des paramètres de paie.

3.2. Modification manuelle du fichier des paramètres sur cartes.

3.3. Le fichier carte ainsi modifié est vérifié et sauvegardé sur bande magnétique ; édition des erreurs détectées (T3).

3.4. Procédure de correction des erreurs :

- dépouillement du document (T3) ;
- retour en 3.1.

3.5. Les cartes perforées à partir du bordereau (E₄) sont replacées dans la table (F9) des modes de paiement (fichier sur cartes).

.../...

Module 4 : Préparation des éléments variables

4.1. Préparation des bordereaux de perforation (E5) des cartes 'activité' :

Pour chaque agent horaire repéré par son matricule, son nom et son prénom qui sont préimprimés :

Pour chaque semaine du mois repérée par son numéro, on porte les renseignements suivants :

COLONNES	CONTENU
HEUR	nombre d'heures de la semaine
NUIT	nombre d'heures de nuit ou dimanche
ROUT	nombre d'heures de route
ENL	nombre d'heures à ôter
PANIER	montant de la prime de panier

4.2. Préparation du bordereau de perforation (E6) des cartes 'supplémentaires'

Pour chaque agent repéré par son matricule, son nom et son prénom également préimprimés, on porte :

COLONNES	CONTENU (montants)
RAPPEL	rappels en somme
PRIMES	primes diverses
INDEMN	indemnités
LOYER	loyer
OPPOSI	oppositions sur salaire
AVANCE	avances déjà payées
RETEN	retenues diverses non imposables
MOINS	diminution de salaire pour absence

.../...

4.3. Préparation du bordereau de perforation (E7) des cartes 'rappels' :

Pour chaque agent horaire bénéficiant de rappels en heures et pour chaque taux horaire de rappel, on porte sur une ligne du bordereau :

COLONNES	CONTENU
MATRICULE	matricule de l'agent
NOM	son nom
PRENOM	son prénom
HEURES NORMALES	nombre d'heures payées au taux normal
HEURES A 25 %	nombre d'heures payées au taux majoré de 25 %
HEURES A 50 %	nombre d'heures payées au taux majoré de 50 %
HEURES A 100 %	nombre d'heures payées double
TAUX HORAIRE	taux de l'heure de rappel

.../...

Module 5 : Vérification et prise en compte des éléments variables

5.1. Tri des cartes perforées à partir des bordereaux (E5), (E6) et (E7) sur le matricule de l'agent et le code carte ; ce tri permet le classement par agent et, pour chaque agent, d'avoir, dans l'ordre, les cartes 'activité' s'il est horaire, la carte 'supplémentaire' et éventuellement les cartes 'rappels'.

Pour chaque agent du fichier personnel (F1,1) dans l'ordre des matricules :

5.2. Vérification de l'existence d'une carte 'supplémentaire' et si l'agent est horaire, d'une carte 'activité' par semaine.

5.3. Contrôles de structure et de vraisemblance des cartes précédentes.

5.4. Vérification de la compatibilité entre les cartes et les renseignements figurant au fichier personnel.

5.5. Si aucune erreur n'est détectée, création d'un enregistrement du fichier de travail (F4).

5.6. Sinon, écriture sur (T4) des erreurs détectées.

5.7. Procédure d'erreur :

- dépouillement des documents (T4)
- remplacement des cartes erronées
- retour en 5.1.

.../...

Module 6 : Calcul de la paie

6.1. Lecture du fichier (F5) des paramètres de paie.

Pour chaque enregistrement du fichier de travail (F4) dans l'ordre des matricules d'agent :

6.2. Calcul du salaire brut.

6.3. Calcul des bases de cotisations sociales.

6.4. Calcul des retenues sociales et du salaire net.

6.5. Calcul de la somme à payer en incorporant les éléments non imposables.

6.6. Création d'un enregistrement du fichier (F6) de la paie calculée.

6.7. Ecriture du salaire brut sur le fichier intermédiaire (F7) si l'agent est salarié, sur le fichier (F8) s'il est employé.

6.8. Cumul des bases de cotisations et des retenues sociales.

6.9. Ecriture, sur le fichier (F6), de l'enregistrement 'récapitulatif' portant le cumul des bases et des montants de cotisations.

.../...

Module 7 : Edition des documents comptables

7.1. Lecture du fichier des paramètres (F5).

Pour chaque enregistrement du fichier (F6) de la paie calculée :

7.2. s'il s'agit d'un enregistrement 'agent', édition du bulletin de salaire (S1).

7.3. Pour le dernier enregistrement (récapitulatif), calcul des cotisations totales et de la part patronale ; édition des récapitulatifs des retenues sociales (S2).

7.4. A partir du fichier (F6) de la paie calculée, édition du journal des salaires (S3) et du double de ce journal pour la fiche individuelle comptable.

7.5. A partir du fichier (F6) trié par code de paiement et de la table des modes de paiement donnant la correspondance entre le code et le libellé du mode de paiement, édition de l'état bancaire (S5).

7.6. A partir des fichiers (F7) et (F8), édition de la liste des salaires bruts en séparant les agents salariés et les agents employés avec indication des deux cumuls.

.../...

Module 8 : Redressement des erreurs de la paie calculée

Pour chaque agent dont la paie s'est révélée incorrecte :

8.1. le service de paie calcule les éléments de salaire corrects ; ces éléments sont portés sur un bordereau (E8) qui donne lieu à la perforation de cinq cartes.

8.2. A partir des cartes précédentes triées par matricule d'agent et par code carte, mise à jour du fichier (F6) de la paie calculée.

8.3. Modification de l'enregistrement 'récapitulatif' en recalculant les cumuls des bases et des cotisations.

8.4. A partir du fichier (F6) ainsi mis à jour, édition des bulletins de salaire modifiés (les matricules des agents intéressés sont donnés par une carte utilisée au moment de l'édition).

8.5. Exécution des modules 7.3, 7.4 et 7.5.

C H A P I T R E III

CONCLUSION

1. Les difficultés rencontrées :

Elles sont apparues à chacun des deux niveaux de l'étude : la connaissance de l'existant, l'analyse et la programmation.

Mais c'est incontestablement au cours de l'étude de l'existant que nous avons rencontré le plus de difficultés, l'analyse et la programmation n'ayant, quant à elles, pas posé de problèmes d'ordre technique mais demandé un lourd travail, souvent fastidieux ; c'est pour alléger ce travail que nous avons été amenés à mettre au point les techniques qui seront développées dans la seconde partie.

1.1. Au niveau de la connaissance du problème :

Notre tâche n'a pas été simplifiée du fait que, d'une part, cette étude a été le premier contact de l'entreprise avec l'informatique et que, d'autre part, les enquêtes ont été réalisées par une personne extérieure à l'entreprise non spécialisée dans les problèmes de paie.

Il ne semble cependant pas que ce soient les seules raisons des difficultés suivantes :

- Il s'est révélé impossible de recenser tous les éléments intervenant dans le calcul de la paie pendant ces enquêtes ; en effet, si la procédure de calcul a été définie dans ses grandes lignes dès le début, un nombre important de détails ont été omis et ceci pour les deux raisons que nous croyons être principales :

1 - certains détails, qui ne posent manuellement aucun problème ont des répercussions importantes sur la procédure automatique ; on peut citer tous les cas particuliers (gérant cotisant uniquement à la sécurité sociale au taux de 1 %).

2 - aucun document écrit n'a été rédigé au moment de l'enquête ; ce document, signé par le responsable du service de paie, aurait servi de référence pour la suite de l'étude.

- Il a également été très difficile de définir avec précision les documents de sortie ; il semble en effet que la définition exacte de leur contenu et de leur forme ne puisse être obtenue qu'après leur édition réelle.

1. 2. Au niveau de l'analyse et de la programmation :

La connaissance imparfaite du problème lui-même a eu de nombreuses incidences sur la programmation qu'il a fallu revoir et corriger à maintes reprises (programme de calcul de paie et programme d'édition).

L'importance et le nombre des documents à imprimer ont entraîné l'écriture de longs et nombreux programmes d'édition dont la mise à jour a été délicate et peu sûre.

2. Les solutions proposées :

2. 1. Pour l'étude de l'existant :

- Les difficultés citées auraient été sans doute aplanies sinon supprimées si nous avions disposé de documents préétablis servant de "langage commun" entre le service de paie et l'informaticien.

Aussi avons-nous essayé de mettre au point un document qui, faisant apparaître clairement les arguments de la paie et leurs modalités d'intervention dans le calcul du salaire, facilite ainsi la discussion entre le responsable de la paie et le concepteur, lui apporte une connaissance plus rapide et plus sûre du problème et lui serve de document de référence au moment de la programmation.

Ce document, que nous étudions sur le cas d'une paie, est un cas particulier de "descripteurs de traitement" spécifiquement adaptés à une "fonction" déterminée dont ils doivent permettre une analyse stricte dans un langage commun au concepteur et au gestionnaire.

Nous l'appellerons, pour la "fonction paie", la "table des salaires".

- Pour accélérer la mise au point des états à éditer, nous avons élaboré un processus de simulation des sorties ; elle permet, bien avant que le programme d'édition ne soit écrit, d'éditer une maquette d'état sur laquelle le service de paie peut vérifier la forme des documents et qu'il peut modifier facilement s'il le juge utile.

.../...

2. 2. Pour l'analyse et la programmation :

- Les mises à jour incessantes nous ont incitées à rechercher une logique de programme adaptée à la maintenance ; nous avons utilisé la technique de la sous-compilation qui permet de décomposer un programme en modules tout à fait indépendants et donc d'en faciliter les modifications en les localisant.

- Pour faciliter la mise au point des documents à éditer, effectuée conjointement par l'analyste et l'utilisateur, nous avons pensé qu'il était souhaitable de mettre au point un "descripteur de document" proche du document classique et rappelant en particulier le graphisme de ce dernier par ses dispositions typographiques.

Ce document sur lequel l'analyste porte les caractéristiques de l'état sous une forme très voisine du dessin normalement élaboré par le concepteur sur une grille d'impression, n'est cependant un langage commode que s'il permet une programmation aisée.

C'est pour cette raison que nous avons inclu dans ce travail la construction d'un générateur de programme utilisant le descripteur précédent.

La deuxième partie de ce travail a pour objet l'étude détaillée des points que nous venons d'introduire :

- Descripteur de document et système d'édition,
- Sous-compilation,
- Le descripteur de traitement : table des salaires.

DEUXIEME PARTIE

LES OUTILS
"SOFTWARE"

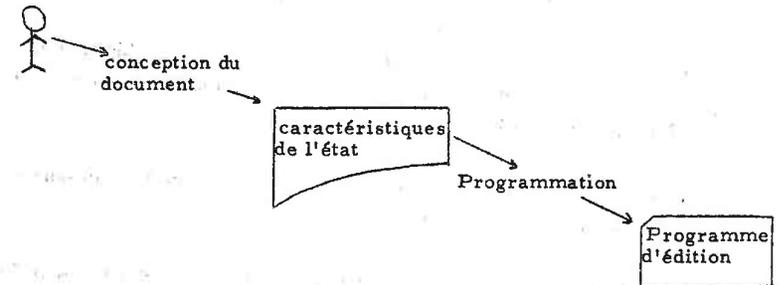
CHAPITRE IV

SYSTEME D'EDITION

1. Introduction :

1.1. Les objectifs du système :

Tout traitement de gestion se termine inévitablement par une phase d'édition dont on peut schématiser les étapes de la façon suivante :



- Le document est en général conçu conjointement par l'informaticien et le gestionnaire au moment de l'analyse du problème.

Il s'agit de définir ce que l'on peut appeler la "logique" de l'état, c'est-à-dire les différentes parties qui le composent puis le dessin de chaque partie, les informations qui y sont éditées, les totalisations partielles ou globales à effectuer ainsi que les arguments présidant au déroulement de l'impression des différentes parties.

Le résultat de cette étude est le dessin de l'état porté sur une grille d'impression et précisant le contenu des différentes zones du dessin ; le déroulement de l'impression, quant à lui, pourra être représenté par exemple à l'aide d'un organigramme.

- A partir des renseignements précédents interprétés puis codifiés, l'informaticien obtiendra le programme d'édition correspondant.

Il ne semble pas que l'on puisse intervenir de façon automatique dans la conception du document qui est une démarche purement intellectuelle ; mais on peut certainement faciliter et accélérer la mise au point de l'état en fournissant au concepteur un document préétabli où il pourra porter directe-

ment les caractéristiques de l'état ; c'est le descripteur d'état introduit précédemment.

Cependant l'efficacité de ce descripteur ne sera totale que s'il supprime les problèmes d'interprétation et de codification c'est-à-dire s'il minimise la programmation tout en la rendant plus facile et plus sûre.

Pour atteindre ce but, on peut concevoir plusieurs méthodes :

1. Mettre au point un langage spécifiquement adapté à l'édition permettant une écriture rapide et concise ainsi que le compilateur correspondant (langages d'édition paramétrés par exemple).
2. Générer tout ou partie des programmes d'édition dans un langage évolué quelconque.

C'est cette dernière solution que nous avons retenue essentiellement pour deux raisons :

- un nouveau langage exige de l'utilisateur d'en assimiler la syntaxe et l'emploi alors que celui-ci connaît généralement les langages de programmation courants.
- cette solution permet de conserver toutes les possibilités des langages généraux, possibilités qu'il est difficile de retrouver dans un langage adapté à une fonction unique telle que l'édition sous peine de l'alourdir considérablement et de lui faire perdre sa qualité première : la simplicité d'utilisation.

Ce choix a deux conséquences immédiates :

- Il nécessite de préciser le langage d'écriture utilisé ; en effet, si le principe ne dépend pas de ce langage, le générateur devra être défini en conséquence.
- Le descripteur d'état devra revêtir une forme permettant cette génération. On s'efforcera cependant à ce qu'il reste proche du dessin habituel de l'état afin de ne pas multiplier les documents à fournir et pour qu'il permette une première visualisation du document final.

.../...

1.2. Le principe du système :

Nous avons choisi le langage Cobol - 65. Dans ce langage, la programmation de la phase d'édition est souvent lourde parce qu'il faut décrire chaque catégorie de lignes éditées et, pour chaque ligne décrite, détailler le contenu de chaque colonne.

Une première amélioration a été apportée par l'éditeur Cobol inclus dans de nombreux compilateurs et qui permet une gestion automatique des mises en page, des débordements et des ruptures de contrôle ainsi que certaines impressions automatiques. La partie exécutive du programme (division des traitements) est alors considérablement allégée mais au contraire l'éditeur exige de la part du programmeur une lourde description de chaque groupe de lignes à éditer (section édition de la division des données).

Notre but est de faire disparaître cette phase de transcription et de codification du dessin, tout en conservant les possibilités offertes par l'éditeur ; le générateur construit, directement à partir du descripteur, la partie déclarative du programme suivant les normes de l'éditeur Cobol (sans intervenir dans la division des traitements).

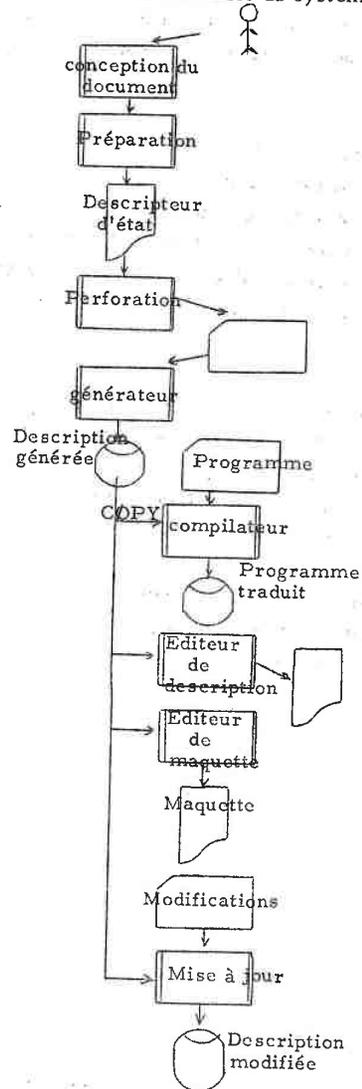
Nous avons complété ce système d'édition par un programme de simulation de l'édition qui permet, à partir de la description générée, l'impression d'une maquette de l'état bien avant l'écriture du programme d'édition et la mise en place des fichiers d'essais.

Enfin, pour permettre une maintenance facile et fiable des programmes d'édition, nous avons mis au point un programme de mise à jour automatique des descriptions générées.

.../...

2. Le schéma général du système :

L'ensemble du système peut être schématisé de la façon suivante :



Les caractéristiques générales de l'état (différents groupes, mise en page ...) et le détail de chaque ligne sont portés sur le descripteur d'état .

Le document résultant est transcrit sur cartes perforées .

Ces cartes sont confiées au générateur qui engendre la description de l'état selon les normes de l'éditeur . Cobol.

Le programme d'édition utilisera cette description grâce au verbe COPY qui permet d'inclure celle-ci dans le texte source en langage Cobol au moment de la compilation.

On a joint au système un programme permettant d'éditer la description.

L'éditeur de maquette permet d'obtenir une image exacte du résultat imprimé sans programme d'édition ni jeu d'essais.

A partir des modifications à effectuer sur la description, le programme de mise à jour génère la description modifiée ; celle-ci pourra être utilisée comme la description initiale.

.../...

Les chapitres suivants sont consacrés à l'étude détaillée de chacune des deux grandes parties du système :

1. Le descripteur d'état et le générateur de description.

2. "Les programmes de service" :

- l'éditeur de description
- l'éditeur de maquette
- le programme de mise à jour de description.

Afin d'en permettre une large diffusion sur d'autres matériels, tous les programmes de ce système ont été rédigés en langage Cobol - 65.

.../...

3. Descripteur d'état et générateur associé :3. 1. Le descripteur d'état :3. 1. 1. Présentation générale :

Le descripteur d'état est le document donnant les caractéristiques de l'état à l'issue de la phase de conception lorsque la "logique" de l'état a été définie, c'est-à-dire, si l'on se place dans le cadre de programmes d'édition en langage Cobol avec éditeur, lorsque les caractéristiques de mise en page, les contrôles, la décomposition en groupes d'impression, leur type et leur description ligne par ligne ont été déterminés.

Le descripteur est essentiellement constitué d'un bordereau (bordereau 2) portant le dessin de chaque groupe de l'état ; ce bordereau est en fait une grille d'impression habituellement utilisée où le dessin porté se rapproche le plus possible du vrai dessin de l'imprimé.

Toutes les caractéristiques de l'état ne peuvent apparaître sur ce dessin aussi le bordereau 2 est - il complété par trois autres bordereaux :

- Pour des raisons de clarté de présentation, les données sur la mise en page et les contrôles automatiques ont été placés dans la première partie du bordereau 1 ; pour la même raison, les en-têtes de chacun des groupes qui ont été dessinés sur le bordereau 2, donnant le nom du groupe et son type, constituent la seconde partie du bordereau 1.

- Pour des raisons typographiques évidentes, il était impossible de porter sur le dessin les identificateurs donnant l'origine des valeurs à éditer ; les bordereaux 3 et 4 ont été prévus pour recevoir ces informations complémentaires.

Remarques :

- Les bordereaux ont été conçus pour être saisis sur cartes perforées ;
- On peut utiliser plusieurs bordereaux de chaque type ;
- Nous avons limité la longueur des identificateurs à huit caractères, la qualification reste possible ;

... / ...

- L'utilisation du caractère £ dans les libellés du bordereau 2 est impossible, ce caractère étant utilisé en tant que code dans les programmes du générateur.

3. 1. 2. Un exemple d'utilisation :

Pour faciliter la compréhension de l'exposé technique, nous nous référerons à un exemple concret tiré de la chaîne de traitement de la paie.

Il s'agit de l'état bancaire introduit dans la première partie ou état récapitulatif des sommes versées aux agents.

... / ...

On dispose d'un fichier de la paie calculée trié par mode de paiement à partir duquel il faut éditer, pour chaque mode de paiement (cette édition est en fait celle que l'on obtient réellement comme résultat du programme d'édition) :

```
*****
* SOCIETE NANCEENNE ET VARIN BERNIER      MAY 30 72 *
*****
*MATR   NOM      | PRENOM  | NO DE COMPTE | SOMMES *
*****
*0104| | JEAN      | 191052-6042-E | 1319.24*
*0135| | PIERRE    | 150748-6041-M | 1980.00*
*0463| | IRENE     | 132763 6041-T | 2113.97*
*0575| | MARIE     | 148970 6041-Z | 1881.48*
*0585| | YVES      | 135834 6041-H | 1668.82*
*0612| | HENRI     | 199384 6041-N | 318.75*
*0655| | DANIEL    | 144290 6041-P | 1042.42*
*0657| | ALAIN     | 148899 6041-N | 745.21*
*0672| | ROBERT    | 134199 6041-X | 1685.50*
*0808| | GUY       | 178737 6041-B | 933.89*
*0877| | BERNARD   | 159311 6041-A | 1943.96*
*0890| | FRANCOISE | 16957 6041-R | 1086.64*
*****
*                               SOMME TOTALE A PAYER : 16689.88*
*****
```

un groupe de haut de contrôle (la zone de contrôle est le code de paiement)

un groupe DETAIL pour chaque agent payé dans ce mode de paiement (groupe BAGENT pour banque ou CCP, LAGENT pour liquide).

un groupe de bas de contrôle donnant le cumul des sommes payées (ce cumul est obtenu grâce à une proposition SUM).

Cet état est complété, à la fin, par un groupe de bas de contrôle au niveau FINAL :

```
*****
* CUMUL DES SOMMES PAYEES PAR BANQUE OU CCP 51702.59*
*****
* CUMUL DES SOMMES PAYEES EN LIQUIDE      20303.06*
*****
* CUMUL TOTAL DES SOMMES PAYEES          72005.65*
*****
```

Les trois cumuls figurant dans ce groupe sont obtenus grâce à des compteurs SUM ; dans le premier sont cumulées les sommes payées par banque ou CCP (option UPON BAGENT), dans le second les sommes payées en liquide (option UPON LAGENT), le troisième est la somme des deux compteurs précédents.

3.1.3. Bordereau 1 :

Il comporte deux parties :

1. la première permet d'indiquer les caractéristiques générales de l'état ; c'est la phrase RD de la section édition.

- les lignes RD1 et RD2 donnent les renseignements habituels de mise en page ;

- les lignes RDC portent les noms des zones de contrôle utilisées ; ces zones sont notées dans l'ordre de la hiérarchie de contrôle.

2. la seconde partie donne la liste des en-têtes de groupe (niveau 01 de la section édition) ; chaque groupe est repéré par un code qui sert à reconnaître l'appartenance des éléments décrits sur les autres bordereaux aux différents groupes. Il est également utilisé pour classer les groupes entre eux et permettre ainsi un tri des informations d'entrée indispensable à la génération.

Ce code est placé en premier caractère de la ligne GRP01 et de la ligne GRP02 si elle est utilisée.

Exemple :

Les cartes RD1 et RD2 donnent le nom de l'état ainsi que les valeurs utilisées pour la mise en page ; il n'y a qu'une seule carte RDC donnant les zones de contrôle (niveau FINAL et code de paiement CØDP).

L'état comporte 5 groupes : 2 groupes de contrôle relatifs au code de paiement (1 : haut de contrôle, 4 : bas de contrôle), 2 groupes de type DETAIL (groupe 2 pour les agents payés en liquide, 3 pour les autres) et enfin un groupe de bas de contrôle au niveau FINAL (groupe 5).

.../...

DESCRIPTEUR D'ETAT															BORDEREAU 1		IV-37			
PERFORATION																				
1	51	101	151	201	251	301	351	401	451	501	551	601	651	701	751	80				
RAD1 RD1 ETAT PAGE 000 LINES HEADING 001 FIRST DETAIL 001																				
RAD2 LAST DETAIL 000 FOOTING 000																				
RAD3 CONTROL FINAL 00 00 CØDP 00 00																				
RAD4 00																				
RAD5 00																				
RAD6 00																				
RAD7 00																				
RAD8 00																				
RAD9 00																				
RAD10 00																				
RAD11 00																				
RAD12 00																				
RAD13 00																				
RAD14 00																				
RAD15 00																				
RAD16 00																				
RAD17 00																				
RAD18 00																				
RAD19 00																				
RAD20 00																				
RAD21 00																				
RAD22 00																				
RAD23 00																				
RAD24 00																				
RAD25 00																				
RAD26 00																				
RAD27 00																				
RAD28 00																				
RAD29 00																				
RAD30 00																				
RAD31 00																				
RAD32 00																				
RAD33 00																				
RAD34 00																				
RAD35 00																				
RAD36 00																				
RAD37 00																				
RAD38 00																				
RAD39 00																				
RAD40 00																				
RAD41 00																				
RAD42 00																				
RAD43 00																				
RAD44 00																				
RAD45 00																				
RAD46 00																				
RAD47 00																				
RAD48 00																				
RAD49 00																				
RAD50 00																				
RAD51 00																				
RAD52 00																				
RAD53 00																				
RAD54 00																				
RAD55 00																				
RAD56 00																				
RAD57 00																				
RAD58 00																				
RAD59 00																				
RAD60 00																				
RAD61 00																				
RAD62 00																				
RAD63 00																				
RAD64 00																				
RAD65 00																				
RAD66 00																				
RAD67 00																				
RAD68 00																				
RAD69 00																				
RAD70 00																				
RAD71 00																				
RAD72 00																				
RAD73 00																				
RAD74 00																				
RAD75 00																				
RAD76 00																				
RAD77 00																				
RAD78 00																				
RAD79 00																				
RAD80 00																				

3.1.4. Bordereau 2 :

C'est le corps de l'état. Le concepteur y dessine les groupes déclarés sur le bordereau 1 ligne par ligne et dans l'ordre logique où elles seront imprimées ; chaque ligne a 132 caractères (largeur de la ligne d'imprimante) de la taille standard d'un caractère imprimante de façon à ce que l'utilisateur puisse, dès ce moment là, se rendre compte des formes et des dimensions de l'état.

Chaque ligne est précédée de sept caractères : les quatre premiers sont des codes permettant d'associer les lignes dessinées aux en-têtes de groupe et aux propositions SOURCE et SUM correspondantes ; les trois suivants servent à indiquer l'espacement souhaité entre les lignes à l'impression.

Une ligne du bordereau sera donc composée de trois zones :

a. Les codes de référence : le premier caractère indique le code du groupe auquel appartient la ligne ; les trois autres servent à référencer la ligne.

A chaque ligne en effet est associée une référence de ligne permettant, d'une part, de reconnaître cette ligne sur les bordereaux 3 et 4 et, d'autre part, de classer les lignes d'un groupe (les références doivent être croissantes à l'intérieur d'un même groupe) ; les références ont été prémarquées, de cinq en cinq afin de permettre l'insertion d'une ligne dans un groupe déjà dessiné ; on utilise pour cela la partie vierge du bordereau (sans référence prémarquée) en affectant à la ligne une référence correspondant à la place qu'elle doit occuper dans son groupe.

b. L'espacement de lignes :

cette zone permet la gestion automatique des sauts de papier (proposition LINE de la section édition), elle n'est pas garnie dans le cas d'un espacement simple.

Dans les autres cas, l'espacement est indiqué de la façon suivante :

[P] pour un saut de page,

[c] pour un numéro de ligne absolu ;

[+ c] pour un espacement de lignes relatif.

c. Le dessin de la ligne :

certaines facilités d'écriture sont offertes à l'utilisateur pour alléger ce dessin :

.../...

- Répétition d'un même caractère sur plusieurs colonnes :

Si l'on veut répéter un même caractère d'une colonne i à une colonne j, il suffit de placer le caractère à reproduire dans chacune de ces colonnes puis de porter, à un endroit quelconque de cet intervalle, le symbole \$JUSQUE\$.

Une même ligne peut comporter un nombre quelconque de répétitions de ce type mais ce symbole ne permet pas la répétition d'un groupe de caractères.

- Répétition d'une ligne :

Si une ligne comporte les mêmes libellés qu'une ligne déjà dessinée, on peut ne pas recommencer le dessin mais porter, à un emplacement quelconque de la ligne, le symbole \$CÔMME\$ immédiatement suivi du code du groupe et de la référence de la ligne modèle :

[\$ | C | Ø | M | M | E | \$ | g | l | l | l]

↑ ↑
référence de ligne

↑
code de groupe

Ce symbole sera remplacé par le dessin de la ligne modèle ; celle-ci doit donc être dessinée explicitement.

Sur l'exemple cité :

- la ligne 1005 (groupe 1, référence de ligne 005) dont l'espacement est + 03 (équivalent à LINE PLUS 3) sera imprimée sur la troisième ligne après la dernière impression ; elle est composée de caractères * répétés de la colonne 40 à la colonne 94.

- les lignes 1015 et 5085, définies par \$CÔMME\$1005, sont identiques à la précédente.

.../...

3.1.5. Bordereaux 3 et 4 :

Ils donnent les renseignements complémentaires sur l'origine des données à éditer :

1. Les données émettrices des propositions SØURCE et les données à totaliser des propositions SUM (bordereau 3) :

Le code du groupe et la référence de ligne permettent de préciser la ligne sur laquelle va apparaître la donnée, le numéro de colonne donne sa position sur cette ligne.

Les renseignements portés dans les zones préimprimées sont ceux qu'exige l'éditeur Cobol c'est-à-dire :

- un nom [optionnel],
- l'image (clause PICTURE),
- l'indication SØURCE ou SUM,
- le nom de la donnée émettrice dans le cas de SØURCE ou de la première donnée à totaliser pour SUM,
- les options associées (GRØUP INDICATE, BLANK WHEN ZERO, JUSTIFIED RIGHT) ;

SUM peut être suivi de plusieurs données à totaliser ; il suffit alors de porter les identificateurs sur les lignes suivantes, à raison de un par ligne en répétant seulement le code du groupe, la référence de ligne et le numéro de colonne.

Exemple :

- en colonne 46 de la ligne 2030 sera éditée la zone NØM sur 15 caractères alphanumériques ;
- en colonne 87 de cette même ligne sera éditée la zone SØMPIE sous le format Z(4).ZZ ; l'option BLANK WHEN ZERO a été demandée ;
- sur la ligne 5080, on éditera en colonne 85 la somme des deux compteurs CPTB et CPTL

The image shows a COBOL data card with columns for 'LIGNE', 'GROUPE', and 'COLONNE'. The card is divided into sections for 'Bordereau 2' and 'Bordereau 3'. Handwritten entries include:

- Line 2030: SØURCE NØM (15 characters), SØMPIE (4 characters, Z format, BLANK WHEN ZERO), CPTB (2 characters), CPTL (2 characters).
- Line 5080: SUM CPTB (2 characters), CPTL (2 characters).

IV-42 DESCRIPTEUR D'ETAT

BORDEREAU 3

CODE DE GROUPE
 REFERENCE DE LIGNE
 NUMERUS DE COLONNE

OPTION JUSTIFIED RIGHT PATER 1
 OPTION BLANK WHEN ZERO PATER 1
 OPTION GROUP INDICATE PATER 1

[NOM]	IMAGE	SOURCE (SUM)	PF
1	2	3	4
1010042	X(40)		
1010083	X(10)		
2030044	9(4)	SOURCE	
2030046	X(15)	DATUM	
2030062	X(10)	MATK	
2030073	X(12)	NOM	
2030087	Z(4).22	PREMOM	
3035044	9(4)	MCCP	
3035046	X(15)	SOMPIE	
3035042	X(10)	MATA	
3035073	X(12)	NOM	
3035087	Z(4).22	PREMOM	
4045085	Z(4).22	MCCP	
5060085 CPTB	Z(6).22	SOMPIE	
5070085 CPTL	Z(6).22	SOMPIE	
5080085	Z(6).22	SOMPIE	
5080085	Z(6).22	CPTB	
		CPTL	

2. Les options UPON et RESET que l'on peut associer à un compteur SUM ont été placées sur le bordereau 4.

Cette séparation, outre des raisons de présentation, est justifiée par le fait que ces options sont assez rarement utilisées.

Le compteur SUM auquel elles sont associées est repéré grâce au code de groupe, à la référence de ligne et au numéro de colonne.

Exemple :

On a associé au compteur SUM édité en colonne 85 de la ligne 5060 l'option UPON BAGENT pour effectuer le cumul sélectif des sommes payées par banque ou CCP.

Ce tri manuel peut sembler au premier abord très contraignant pour l'utilisateur. On aurait pu l'éviter en ajoutant des codes sur les différentes cartes afin de permettre un tri automatique, mais cette méthode, outre le travail supplémentaire qu'elle demande à l'utilisateur, ne nous a pas paru se justifier car de simples dispositions organisationnelles permettent de rendre ce tri presque inexistant.

En effet si :

- au moment de l'écriture des bordereaux, on respecte l'ordre des codes de groupe et des références de ligne sur le bordereau 2 et l'ordre des numéros de colonne sur les bordereaux 3 et 4,
- au moment de la perforation du bordereau 2, on perfore la seconde partie de la ligne immédiatement après la première,

alors il ne reste plus qu'à intercaler les cartes du bordereau 1 dans celles du bordereau 2 et les cartes du bordereau 4 (peu utilisé) dans celles du bordereau 3.

3.3. Le générateur de description :

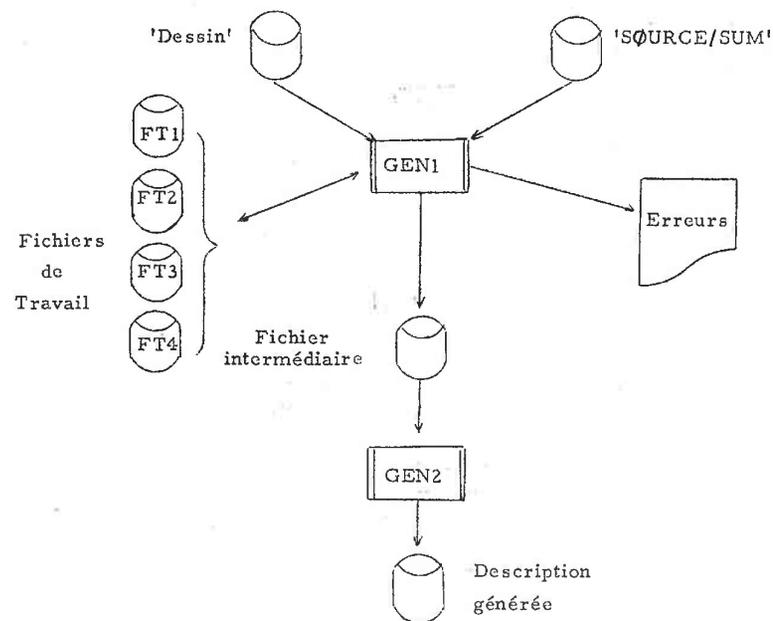
3.3.1. Schéma général :

Le générateur utilise les deux fichiers 'dessin' et 'SOURCE/SUM' que nous avons définis précédemment ; il est composé de deux programmes indissociables (la décomposition en deux programmes a pour seul but d'éviter des programmes trop volumineux) :

- le premier programme (GEN1) contrôle les deux fichiers d'entrée pour détecter les erreurs qui rendraient impossible la génération puis effectue une codification du dessin de l'état (traduction des symboles \$CÔMME\$ et \$JUSQUE\$ en particulier) en créant un fichier intermédiaire ;
- le second programme, utilisant le fichier précédent et le fichier 'SOURCE/SUM' génère la description de l'état syntaxiquement admise par le compilateur Cobol.

.../...

Les étapes de ce travail peuvent être schématisées de la façon suivante :

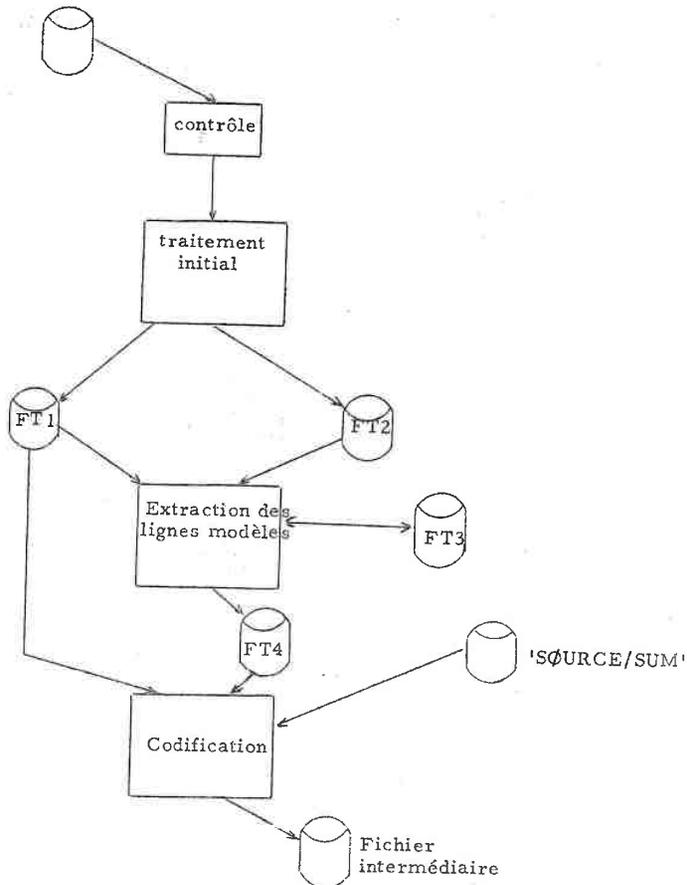


.../...

3.3.2. Le programme GEN 1

Il est constitué de quatre parties s'enchaînant selon l'organigramme suivant :

Fichier
'dessin'



.../...

3.3.2.1. Contrôle

Il s'agit de vérifier que la structure du dessin de l'état est correcte c'est-à-dire en particulier :

- qu'un groupe commence par une carte GRP01,
- qu'une ligne est composée de deux cartes portant la même référence de ligne.

Ce contrôle tend à détecter les erreurs qui rendraient impossible ou sans objet la génération de la description ; lorsqu'une erreur est détectée, l'examen des fichiers se poursuit après impression d'un message mais le second programme n'est pas exécuté.

Aucune vérification de la syntaxe Cobol n'est effectuée (celle-ci aurait de toute façon été incomplète) ; elle le sera normalement à la compilation du programme d'édition.

3.3.2.2. Traitement initial :

- initialisation du fichier de travail (FT1) à partir des cartes de code R (niveau RD).

Pour chaque groupe de l'état, dans l'ordre des codes de groupe :

- écriture, sur le fichier (FT1), d'un enregistrement 'en-tête de groupe' (cartes GRP01 et GRP02 si elle existe).

Pour chaque ligne du groupe, dans l'ordre des références de ligne :

- écriture, sur le fichier (FT1) d'un enregistrement formé des deux cartes constituant la ligne.
- si la ligne est définie par le symbole \$CÔMME\$, on note sur le fichier de travail (FT2) le code de groupe et la référence de la ligne modèle.

3.3.2.3. Extraction des lignes modèles :

Il s'agit ici de construire un fichier contenant le dessin de différentes lignes modèles afin que ceux-ci puissent venir remplacer les symboles \$CÔMME\$.

.../...

Ce travail se déroulera en deux étapes :

1. Le fichier (FT2) est trié puis exploré pour assurer la création du fichier (FT3) des références de ligne à conserver. Il y a, dans (FT3), un seul enregistrement pour un code de groupe et une référence de ligne donnés alors qu'il peut exister plusieurs enregistrements identiques dans le fichier (FT2) si plusieurs lignes du dessin se réfèrent à la même ligne modèle.

2. Pour chaque enregistrement du fichier (FT3), on recherche la ligne correspondante dans le fichier (FT1) et l'on écrit le dessin de cette ligne sur le fichier (FT4) ; ce fichier est un fichier indexé sur le code de groupe et la référence de la ligne modèle.

3.3.2.4. Codification :

Pour chaque enregistrement du fichier de travail (FT1) :

1. s'il s'agit d'un enregistrement de code R (niveau RD) ou d'un en-tête de groupe, on effectue un simple transfert sur le fichier intermédiaire.

2. s'il s'agit d'une ligne :

- si la ligne est définie par un symbole \$CØMME\$, le code du groupe et la référence de ligne placés après ce symbole permettent de trouver le dessin de la ligne modèle dans le fichier de travail (FT4) ; ce dessin vient remplacer le symbole \$CØMME\$.

- Pour chaque enregistrement du fichier 'SØURCE/SUM' se rapportant à cette ligne :

On place un caractère \$ dans la colonne où doit apparaître la donnée émettrice ou à totaliser.

- Si la ligne comporte des symboles \$JUSQUE\$, on les traduit en plaçant, après le premier caractère à répéter, le repère \$ n n n \$ où n n n est le nombre d'occurrences de ce caractère sur la ligne ; le reste de l'intervalle de répétition est mis à blanc.

- La ligne ainsi codifiée et complétée par un caractère \$ après le dernier caractère non blanc pour marquer la fin de la ligne, est écrite sur le fichier intermédiaire.

.../...

3.3.2.5. Exemple

A titre d'exemple, nous donnons ci-dessous l'état du fichier intermédiaire correspondant à l'édition de l'état bancaire.

Ce fichier est très voisin du dessin de l'état lui-même fourni par le bordereau 2 ; il comprend en plus un certain nombre de repères facilitant le travail de génération ultérieur :

- sur la ligne 1005, le symbole \$JUSQUE\$ qui demandait la répétition du caractère * de la colonne 40 à la colonne 94 (donc 55 fois) a été codifié par *\$055\$;

- la ligne 1015 qui était définie par \$CØMME\$ 1005 porte les mêmes libellés que la ligne modèle ;

- sur la ligne 2030, chacune des colonnes où doit débiter une donnée définie par SØURCE (matricule, nom, prénom, numéro de compte et somme payée) est garnie d'un \$;

- toutes les lignes se terminent par \$.

.../...

Fichier intermédiaire de l'état bancaire :

RRD1RD ETAT	PAGE 060 LINES HEADING 001 FIRST DETAIL 001	Niveau RD
RR2LAST DETAIL 060 FOOTING 060		
RR3CCONTROL FINAL		
1GRPO1	TYPE CH CDDP	
1025+03		**055\$
1010+01		\$
1015+01		**055\$
1020+01		*MATR
1025+01		**053\$
2GRPO1 LABENT	TYPE DE	\$
2030+01		** 1\$
3GRPO1 BAGENT	TYPE DE	\$
3035+01		** 1\$
4GRPO1	TYPE CF CDDP	\$
4040+01		**053\$
4045+01		\$
4050+01		**055\$
5GRPO1	TYPE CF FINAL	\$
5055+03		**055\$
5060+01		* CUMUL DES SOMMES PAYEES PAR BANQUE OU CCP \$
5065+01		**053\$
5070+01		* CUMUL DES SOMMES PAYEES EN LIQUIDE \$
5075+01		**053\$
5080+01		* CUMUL TOTAL DES SOMMES PAYEES \$
5085+01		**055\$

En tête des groupes

3.3.3. Le programme GEN2 :

A partir du fichier intermédiaire et du fichier 'SOURCE/SUM', ce programme génère des lignes de description d'état en langage Cobol :

Pour chaque enregistrement du fichier intermédiaire :

1. s'il s'agit d'un enregistrement de code R, génération d'une ligne de la phrase RD.
2. s'il s'agit d'une en-tête de groupe, création du niveau 01 correspondant (nom de groupe et type) ; le code du groupe est noté en colonne 73.
3. Pour chaque ligne d'un groupe, on génère d'abord une phrase de niveau 02 précisant l'espacement de lignes (la référence de ligne est placée en colonne 73) puis on explore la ligne de gauche à droite en effectuant les actions suivantes :

Pour chaque zone de la ligne repérée grâce aux caractères \$:

- création d'une phrase de niveau 03 portant le numéro de colonne (clause COLUMNS) et l'image (clause PICTURE) ;
- génération, selon le cas, d'une clause VALUE simple suivie d'un libellé ou d'une clause VALUE ALL 'caractère' traduisant les symboles \$JUSQUE\$.

Pour chaque caractère \$ trouvé sur la ligne :

- recherche de l'enregistrement (ou des enregistrements) correspondant dans le fichier 'SOURCE/SUM' (c'est-à-dire portant le même code de groupe, la même référence de ligne et le même numéro de colonne) ;
- si l'enregistrement existe :
 - * génération d'une phrase de niveau 03 portant les clauses COLUMNS et PICTURE,
 - * génération d'une proposition SOURCE ou SUM,
 - * s'il y a lieu, génération d'une ligne portant les options GROUP INDICATE, BLANK WHEN ZERO et/ ou JUST RIGHT et d'une ligne donnant les options UPON et/ou RESET.
- sinon, c'est la marque de fin de ligne et l'exploration est terminée.

.../...

Exemple :

Voyons, dans le cas de la génération de l'état bancaire, le traitement effectué par le programme GEN2 sur une ligne particulière, 4045 par exemple

1. L'espacement +01 provoque la génération d'une phrase

```
02 LINE PLUS 01 .
```

2. La première partie de la ligne commencée en colonne 40 par le caractère * et se termine au caractère non blanc précédent \$; il s'agit d'un libellé simple, on génère donc :

- d'une part une ligne

```
03      CØLUMN 040 PIC X(045)
```

- ensuite la clause VALUE

```
VALUE  '*          SØMME TØTALE A PAYER:.'
```

3. Le caractère \$ placé en colonne 85 correspondant à un enregistrement SUM du fichier 'SØURCE/SUM' :

- on écrira d'abord le niveau 03

```
03      CØLUMN 085 PIC Z(6):ZZ
```

- puis la ligne de la proposition SUM

```
SUM SØMPIE
```

4. La seconde zone de la ligne est constituée d'un unique caractère * placé en colonne 94 ce qui provoque la génération des lignes :

```
03      CØLUMN 094 PIC X(001)
```

```
VALUE  '*.
```

5. Le dernier \$ marque la fin de la ligne.

3.4. Utilisation de la description générée dans un programme d'édition :

Une fois la description générée, il reste à l'inclure dans le programme d'édition qui va permettre l'impression effective.

Une première solution consisterait à commander par programme la perforation des cartes de la description sur un perforateur puis à inclure manuellement celles-ci dans le programme d'édition.

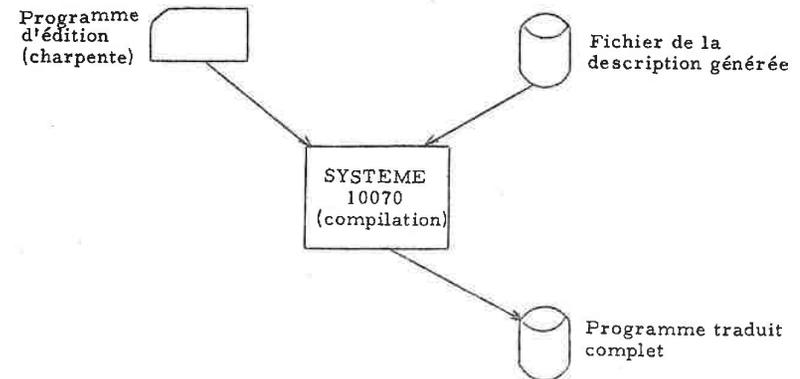
.../...

Une seconde solution, plus rapide et plus sûre, consiste à utiliser une possibilité de Cobol : l'ordre CØPY. Il permet d'inclure, à la compilation, dans la division des données d'un programme une suite de lignes Cobol se trouvant sur un fichier qui n'est connu que par son nom ; en particulier, pour insérer la description générée dans la section édition du programme, il suffira d'écrire :

```
RD nom - d'état [ WITH CØDE nom - mnémorique]
                  CØPY nom - de - fichier .
```

où nom - de - fichier est le nom du fichier sur lequel se trouve la description qui a été générée.

La phrase RD est remplacée, au moment de la compilation par les lignes source en langage Cobol constituant le fichier ; notons que les informations qui précèdent CØPY supplantent en cas de conflit les informations correspondantes du fichier (en particulier le nom d'état):



Le programme traduit, qui est alors complet, pourra ensuite être utilisé comme un programme compilé quelconque.

Remarque : Il est possible d'incorporer plusieurs descriptions dans un même programme par plusieurs ordres CØPY de ce type.

.../...

Exemple du programme d'éditior de l'état bancaire

```

IDENTIFICATION DIVISION.
PROGRAM-ID. EDITION PAR MODE DE PAIEMENT.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE=COMPUTER. CII=10070.
OBJECT=COMPUTER. CII=10070.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT ENTREE ASSIGN TO DISC.
    SELECT LISTE ASSIGN TO PRINTER.
    SELECT F-CART ASSIGN TO DISC.
DATA DIVISION.
FILE SECTION.
FD ENTREE LABEL RECORD STANDARD DATA RECORD ENR.
01 ENR.
    02 MATR PIC 9(4).
    02 NOM PIC X(15).
    02 PRENOM PIC X(10).
    02 FILLER PIC X(36).
    02 CDDP PIC X.
    02 FILLER PIC X(10).
    02 N0CP PIC X(12).
    02 FILLER PIC X(199).
    02 SOMPIE PIC 9(4)V99.
    02 FILLER PIC X(20).
    02 DATUM PIC X(10).
FD LISTE LABEL RECORD OMITTED REPORT ETAT.
FD F-CART LABEL RECORD STANDARD DATA RECORD ENR1.
01 ENR1.
    02 C0DB PIC X.
    02 N0MB PIC X(40).
    02 FILLER PIC X(39).
WORKING-STORAGE SECTION.
77 SOML PIC 9(6)V99 VALUE 0.
77 SOMBC PIC 9(6)V99 VALUE 0.
77 IND COMP VALUE 0.
77 INDM COMP.
77 ANCC PIC X VALUE SPACE.
77 N0MBW PIC X(40).
01 TABLE-B.
    02 TC0D PIC X OCCURS 20.
    02 TN0M PIC X(40) OCCURS 20.
REPORT SECTION.
RD ETAT COPY DETAT.
RD ETAT PAGE 060 LINES HEADING 001 FIRST DETAIL 001
LAST DETAIL 060 FOOTING 060
CONTROL FINAL CDDP.
01 TYPE CH CDDP
02 LINE PLUS 03
03 C0LUMN 040 PIC X(055)
VALUE ALL '*'.
02 LINE PLUS 01
03 C0LUMN 040 PIC X(001)
VALUE '*'.
03 C0LUMN 042 PIC X(40)
SOURCE N0MBW
03 C0LUMN 083 PIC X(10)
SOURCE DATUM
03 C0LUMN 094 PIC X(001)
VALUE '*'.
02 LINE PLUS 01
    
```

La description générée se trouvant sur un fichier nommé DETAT est incluse dans le programme par l'ordre COPY

groupe n°1
1005
1010
ligne 010 du groupe 1

1015

```

03 C0LUMN 040 PIC X(055)
VALUE ALL '*'.
02 LINE PLUS 01
03 C0LUMN 040 PIC X(055)
VALUE '*MATR NOM I PRENOM IN0 DE COMPTEIIS0M
MES *'.
02 LINE PLUS 01
03 C0LUMN 040 PIC X(001)
VALUE '*'.
03 C0LUMN 041 PIC X(053)
VALUE ALL '*'.
03 C0LUMN 094 PIC X(001)
VALUE '*'.
01 LAGENT TYPE DE
02 LINE PLUS 01
03 C0LUMN 040 PIC X(001)
VALUE '*'.
03 C0LUMN 041 PIC 9(4)
SOURCE MATR
03 C0LUMN 045 PIC X(001)
VALUE '||'.
03 C0LUMN 046 PIC X(15)
SOURCE NOM
03 C0LUMN 061 PIC X(001)
VALUE '||'.
03 C0LUMN 062 PIC X(10)
SOURCE PRENOM
03 C0LUMN 072 PIC X(001)
VALUE '||'.
03 C0LUMN 073 PIC X(12)
SOURCE N0CP
03 C0LUMN 085 PIC X(002)
VALUE '|||'.
03 C0LUMN 087 PIC Z(4).ZZ
BLANK ZERO
SOURCE SOMPIE
03 C0LUMN 094 PIC X(001)
VALUE '*'.
01 BAGENT TYPE DE
02 LINE PLUS 01
03 C0LUMN 040 PIC X(001)
VALUE '*'.
03 C0LUMN 041 PIC 9(4)
SOURCE MATR
03 C0LUMN 045 PIC X(001)
VALUE '||'.
03 C0LUMN 046 PIC X(15)
SOURCE NOM
03 C0LUMN 061 PIC X(001)
VALUE '||'.
03 C0LUMN 062 PIC X(10)
SOURCE PRENOM
03 C0LUMN 072 PIC X(001)
VALUE '||'.
03 C0LUMN 073 PIC X(12)
SOURCE N0CP
03 C0LUMN 085 PIC X(002)
VALUE '|||'.
03 C0LUMN 087 PIC Z(4).ZZ
BLANK ZERO
SOURCE SOMPIE
03 C0LUMN 094 PIC X(001)
VALUE '*'.
01 TYPE CF CDDP
02 LINE PLUS 01
03 C0LUMN 040 PIC X(001)
    
```

1020

1025

2
2030

3
3035

4
4040

VALUE	'*1'	
03	COLUMN 041 PIC X(053)	
VALUE ALL	'*1'	
03	COLUMN 094 PIC X(001)	
VALUE	'*1'	
02 LINE PLUS 01		4045
03	COLUMN 040 PIC X(045)	
VALUE	'*'	SOMME TOTALE A PAYER !!
03	COLUMN 085 PIC Z(6).ZZ	
SUM SAMPLE		
03	COLUMN 094 PIC X(001)	
VALUE	'*1'	
02 LINE PLUS 01		4050
03	COLUMN 040 PIC X(055)	
VALUE ALL	'*1'	
01	TYPE CF FINAL	5
02 LINE PLUS 03		5055
03	COLUMN 040 PIC X(055)	
VALUE ALL	'*1'	
02 LINE PLUS 01		5060
03	COLUMN 040 PIC X(044)	
VALUE	'*'	CUMUL DES SOMMES PAYEES PAR BANQUE OU CCP!
03 CPTB	COLUMN 085 PIC Z(6).ZZ	
SUM SAMPLE		
UPON BAGENT		
03	COLUMN 094 PIC X(001)	
VALUE	'*1'	
02 LINE PLUS 01		5065
03	COLUMN 040 PIC X(001)	
VALUE	'*1'	
03	COLUMN 041 PIC X(053)	
VALUE ALL	'*1'	
03	COLUMN 094 PIC X(001)	
VALUE	'*1'	
02 LINE PLUS 01		5070
03	COLUMN 040 PIC X(037)	
VALUE	'*'	CUMUL DES SOMMES PAYEES EN LIQUIDE!
03 CPTL	COLUMN 085 PIC Z(6).ZZ	
SUM SAMPLE		
UPON LAGENT		
03	COLUMN 094 PIC X(001)	
VALUE	'*1'	
02 LINE PLUS 01		5075
03	COLUMN 040 PIC X(001)	
VALUE	'*1'	
03	COLUMN 041 PIC X(053)	
VALUE ALL	'*1'	
03	COLUMN 094 PIC X(001)	
VALUE	'*1'	
02 LINE PLUS 01		5080
03	COLUMN 040 PIC X(031)	
VALUE	'*'	CUMUL TOTAL DES SOMMES PAYEES!
03	COLUMN 085 PIC Z(6).ZZ	
SUM CPTB		
CPTL		
03	COLUMN 094 PIC X(001)	
VALUE	'*1'	
02 LINE PLUS 01		5085
03	COLUMN 040 PIC X(055)	
VALUE ALL	'*1'	

PROCEDURE DIVISION.
DEBUT. OPEN INPUT F-CART ENTREE OUTPUT LISTE. MOVE 1 TO IND.
LE. READ F-CART AT END GO TO ET1.
MOVE C0DB TO TC0D (IND) MOVE N0MB TO TN0M (IND)
ADD 1 IND GO TO LE.
ET1. SUBTRACT 1 FROM IND GIVING INDM.

INITIATE ETAT.
LEC. READ ENTREE AT END GO TO FLEC.
IF MATR = 9999 GO TO FLEC.
IF C0DP NOT = ANCC GO TO ET2.
ET. IF C0DP = 'L' GENERATE LAGENT ELSE GENERATE BAGENT.
GO TO LEC.
ET2. MOVE 1 TO IND.
RE. IF TC0D (IND) = C0DP GO TO ET3.
IF IND = INDM GO TO ET3.
ADD 1 IND GO TO RE.
ET3. MOVE TN0M (IND) TO N0MBW MOVE C0DP TO ANCC GO TO ET.
FLEC. TERMINATE ETAT CLOSE F-CART ENTREE LISTE STOP RUN.

4. Les programmes de service :

Ils complètent l'aide à l'analyse et à la programmation assurée par le descripteur d'état et le générateur associé

Ils sont au nombre de trois :

- le premier assure l'édition de la description qui a été générée sous une forme propre à son insertion dans un dossier d'analyse ;
- le second est l'éditeur de la maquette d'état ;
- le troisième vise à faciliter la maintenance des programmes en permettant une mise à jour automatique de la description générée.

4.1. Programme d'édition de la description générée :

A la fin de la génération, l'utilisateur ne dispose d'aucun listing de la description obtenue qui lui permettrait de vérifier celle-ci avant de l'utiliser dans le programme d'édition.

Ce programme comble cette lacune en donnant à l'utilisateur une édition de la description générée.

La présentation de cette édition sous forme de "pavés" apparaissant nettement a été conçue pour qu'on puisse l'inclure dans un dossier de programmation.

Ce découpage facilite en outre la localisation des anomalies détectées au moment de l'utilisation de la maquette d'état.

Nous donnons ci-après un exemple d'édition de ce type ; il s'agit du début de la description générée de l'état bancaire :

.../...

		DESCRIPTION DE L ETAT ETAT	
RD ETAT	PAGE 060	LINES HEADING 001	FIRST DETAIL 001
		LAST DETAIL 060	FOOTING 060
		CONTRL FINAL	CBDP

		DESCRIPTION DU GROUPE NUMERO 1	
01	TYPE CH	CBDP	
LIGNE NUMERO 1005			
02	LINE PLUS	03	
03	VALUE	COLUMN 040	PIC X(055)
		VALUE ALL ' '.	
LIGNE NUMERO 1010			
02	LINE PLUS	01	
03	VALUE	COLUMN 040	PIC X(001)
		VALUE ' '.	
03	SOURCE	COLUMN 042	PIC X(40)
		SOURCE NMBW	
03	SOURCE	COLUMN 083	PIC X(10)
		SOURCE DATUM	
03	VALUE	COLUMN 094	PIC X(001)
		VALUE ' '.	
LIGNE NUMERO 1015			
02	LINE PLUS	01	
03	VALUE	COLUMN 040	PIC X(055)
		VALUE ALL ' '.	
LIGNE NUMERO 1020			
02	LINE PLUS	01	
03	VALUE	COLUMN 140	PIC X(055)
		TRMTR	NBM
		PRENSY INB DE COMPTE(19M	
		MES ' '.	
LIGNE NUMERO 1025			
02	LINE PLUS	01	
03	VALUE	COLUMN 040	PIC X(001)
		VALUE ' '.	
03	VALUE	COLUMN 041	PIC X(053)
		VALUE ALL ' '.	
03	VALUE	COLUMN 094	PIC X(001)
		VALUE ' '.	

		DESCRIPTION DU GROUPE NUMERO 2	
01	LAGENT	TYPE DE	LAGENT
LIGNE NUMERO 2030			
02	LINE PLUS	01	
03	VALUE	COLUMN 040	PIC X(001)
		VALUE ' '.	
03	VALUE	COLUMN 041	PIC 9(4)

4.2. L'éditeur de maquette :

4.2.1. Présentation générale :

Ainsi que nous l'avons signalé dans le cas de la paie étudié dans la première partie, il est très difficile d'obtenir une définition exacte des documents à éditer avant leur impression réelle par l'ordinateur, le dessin manuscrit sur une grille d'impression n'en donnant pas une image suffisamment représentative et conforme.

En programmation habituelle, l'impression réelle des documents demande l'écriture du programme d'édition et la constitution de jeux d'essais qui, pendant la mise au point de l'état, doivent être modifiés en même temps que la description de celui-ci.

Le but de cette maquette est de donner au gestionnaire une image exacte des sorties réelles avant l'écriture d'un programme d'édition complet et la création de fichiers d'essais, à partir de la description générée, pour lui permettre d'affiner la solution beaucoup plus tôt qu'il n'aurait pu le faire normalement et avec un moindre coût.

Le programme génère des instructions Cobol d'impression qui, incluses à la compilation en même temps que la description générée dans une "charpente" pré-établie, permettent de composer automatiquement un programme d'édition qui simule l'édition réelle.

Mais l'absence de jeu d'essais a deux conséquences :

- il sera impossible de prendre en compte les propositions SOURCE et SUM ; elles sont éliminées et remplacées par une proposition permettant l'impression d'un caractère \$ à l'emplacement où doit s'éditer la donnée.
- les ruptures de contrôle autres que les ruptures du niveau FINAL, étant commandées par le contenu de zones des fichiers d'essais, ne pourront être provoquées.

.../...

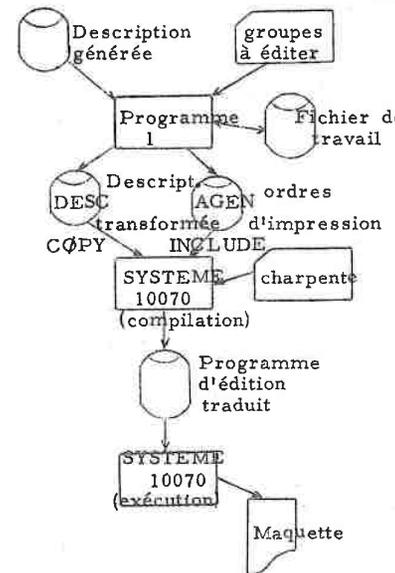
L'impression des groupes de haut et bas de contrôle correspondants, n'étant plus automatique, devra être commandée explicitement par l'utilisateur.

En fait, pour que la génération soit possible, l'utilisateur devra fournir une liste des groupes qu'il veut imprimer, dans l'ordre de leur impression.

Tous les groupes dont l'impression n'est pas commandée automatiquement devront être cités (groupes DETAIL, haut et bas de contrôle autres que FINAL); au contraire les groupes du type haut ou bas de page, les groupes de débordement, de haut et bas de contrôle du niveau FINAL ... sont imprimés de façon automatique donc n'ont pas à être mentionnés.

4.2.2. Schéma de l'éditeur :

A partir de la description générée de l'état et de la liste des groupes à imprimer, l'éditeur va générer un programme d'édition qui assurera l'impression de la maquette selon l'organigramme suivant :



Le premier programme transforme la description de l'état et génère un paragraphe contenant les ordres d'écriture à exécuter (instructions GENERATE)

A la compilation, la 'charpente' du programme d'édition est complétée par la description transformée grâce à un ordre COPY et par les ordres d'impression grâce à l'ordre INCLUDE

Le programme compilé permet d'éditer la maquette de l'état.

.../...

4.2.3. Le premier programme :

Il comprend deux parties :

1. La transformation de la description de l'état imposée par les raisons que nous avons citées.
2. La génération des ordres d'impression qui permettront de simuler l'édition réelle.

4.2.3.1. Transformation de la description de l'état :

- au niveau RD : on élimine la clause CØNTROL donnant les différentes zones de contrôle (en conservant CØNTROL FINAL s'il existe).
- au niveau 01 : les groupes de haut et de bas de contrôle ne se rapportant pas au niveau FINAL sont transformés en groupes DETAIL afin de pouvoir commander leur impression par une instruction GENERATE ; le nom de groupe qui leur est affecté est l'identificateur ZZ i où i est le code de groupe utilisé lors de la génération.

De plus, si cet en-tête de groupe spécifie un groupe de type DETAIL ou transformé en groupe DETAIL, on conserve le code et le nom du groupe dans un fichier de travail qui est en fait une table de correspondance entre codes et noms de groupes.

- aux autres niveaux : les propositions SØURCE et SUM ainsi que les options associées sont supprimées puisque leur prise en compte n'est plus possible et remplacées par une clause VALUE '\$' qui fera apparaître, au moment de l'impression de la maquette, ce caractère dans la colonne où devrait être éditée la donnée.

La description ainsi transformée est placée sur le fichier DESC.

4.2.3.2. Génération des ordres d'impression :

Elle est effectuée à partir de la liste des groupes à éditer donnée par l'utilisateur et mise sur une ou plusieurs cartes ; les groupes sont simplement repérés par leur code, l'ordre des colonnes déterminant l'ordre des impressions.

.../...

La génération est alors extrêmement simple ; il suffit en effet d'explorer les cartes colonne par colonne ; à chaque groupe rencontré et repéré par son code, le programme génère une instruction GENERATE suivie du nom du groupe considéré (le nom du groupe est retrouvé grâce à la table de correspondance construite précédemment).

Chaque instruction générée est placée dans le fichier AGEN.

4.2.4. La 'charpente' de programme d'édition :

Il s'agit simplement d'un programme général d'édition incomplet qui, étant donné sa brièveté, est retranscrit intégralement ci-dessous :

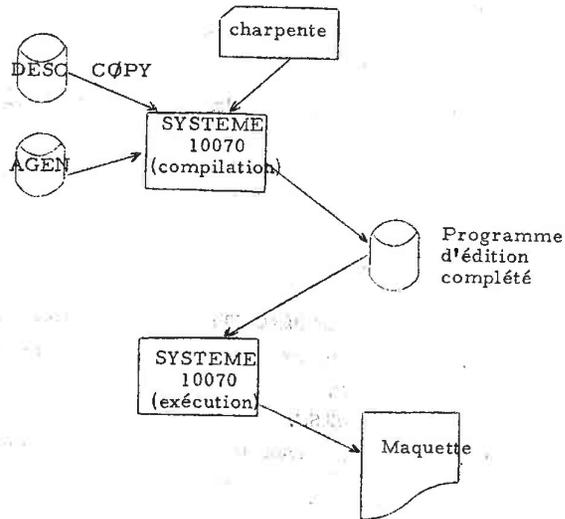
```
DATA DIVISION.
FILE SECTION.
FD IMPR LABEL RECORD STANDARD REPORT ETAT.
NOTE. c'est la déclaration du fichier d'impression de la maquette.
REPORT SECTION.
RD ETAT COPY DESC.
NOTE. Cet ordre, suivant le principe vu précédemment, permet d'incorporer la description transformée (stockée sur le fichier DESC) dans le texte Cobol.
PROCEDURE DIVISION.
DEB. OPEN OUTPUT IMPR INITIATE ETAT.
DEB1. INCLUDE AGEN.
NOTE. le verbe INCLUDE, analogue à COPY, permet d'incorporer les instructions GENERATE se trouvant sur le fichier AGEN dans le texte Cobol. A la compilation, les ordres d'impression viendront se substituer à cette phrase.
DEB2. TERMINATE ETAT CLOSE IMPR STOP RUN.
```

4.2.5. L'édition de la maquette :

Le programme charpente complété à la compilation par l'insertion du contenu des deux fichiers DESC et AGEN est un programme objet normal

.../...

dont l'exécution commande l'impression de la maquette d'état.



A titre d'exemple, nous donnons ci-après le listing partiel du programme d'édition complété et la maquette de l'état bancaire :

.../...

Le programme d'édition :

IDENTIFICATION DIVISION.
 PROGRAM-ID. EDITION GENERALE D'ETAT.
 ENVIRONMENT DIVISION.
 CONFIGURATION SECTION.
 SOURCE-COMPUTER. CII-10070.
 OBJECT-COMPUTER. CII-10070.
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.

Insertion de la description transformée

```

  SELECT IMPR ASSIGN TO PRINTER.
  DATA DIVISION.
  FILE SECTION.
  FD IMPR LABEL RECORD OMITTED REPORT ETAT.
  REPORT SECTION.
  RD ETAT COPY DESC.
  RD ETAT PAGE 060 LINES HEADING 001 FIRST DETAIL 001
  LAST DETAIL 060 FOOTING 060
  CNTRLR FINAL.
  01 ZZ1 TYPE DE
  02 LINE PLUS 03
  03 CBLUMN 040 PIC X(055)
  VALUE ALL '*'.
  02 LINE PLUS 01
  03 CBLUMN 040 PIC X(001)
  VALUE '*'.
  03 CBLUMN 042 PIC X
  VALUE '*'.
  03 CBLUMN 083 PIC X
  VALUE '*'.
  03 CBLUMN 094 PIC X(001)
  VALUE '*'.
  02 LINE PLUS 01
  02 LINE PLUS 01
  03 CBLUMN 040 PIC X(055)
  VALUE ALL '*'.
  PROCEDURE DIVISION.
  DEB. OPEN OUTPUT IMPR INITIATE ETAT.
  DEB1. INCLUDE AGEN.
  PARAG.
  GENERATE ZZ1
  GENERATE LAGENT
  GENERATE LAGENT
  GENERATE SAGENT
  GENERATE ZZ4
  DEB2. TERMINATE ETAT CLOSE IMPR STOP RUN.
  
```

Les zones de contrôle ont été éliminées (sauf FINAL)

le groupe de haut de contrôle relatif au code de paiement a été transformé en groupe DETAIL

La proposition SOURCE a été éliminée, et remplacée par une clause VALUE '*'

La carte d'entrée donnant le code des groupes à éditer était la suivante :

1222222222224133333334

La maquette :

```

*****
*
*****
*MATR      NOM      | PRENOM  INO DE COMPTE | SOMMES *
-----|-----|-----|-----|-----
*$ |$          |$        |$         |$         |$
*$ |$          |$        |$         |$         |$
*$ |$          |$        |$         |$         |$
*$ |$          |$        |$         |$         |$
*$ |$          |$        |$         |$         |$
*$ |$          |$        |$         |$         |$
*$ |$          |$        |$         |$         |$
*$ |$          |$        |$         |$         |$
*$ |$          |$        |$         |$         |$
*$ |$          |$        |$         |$         |$
*$ |$          |$        |$         |$         |$
-----|-----|-----|-----|-----
*
*                               SOMME TOTALE A PAYER :$
*
*****
*
*****
*MATR      NOM      | PRENOM  INO DE COMPTE | SOMMES *
-----|-----|-----|-----|-----
*$ |$          |$        |$         |$         |$
*$ |$          |$        |$         |$         |$
*$ |$          |$        |$         |$         |$
*$ |$          |$        |$         |$         |$
*$ |$          |$        |$         |$         |$
*$ |$          |$        |$         |$         |$
*$ |$          |$        |$         |$         |$
*$ |$          |$        |$         |$         |$
-----|-----|-----|-----|-----
*
*                               SOMME TOTALE A PAYER :$
*
*****
*
*****
* CUMUL DES SOMMES PAYEES PAR BANQUE OU CCP $
*****
* CUMUL DES SOMMES PAYEES EN LIQUIDE $
*****
* CUMUL TOTAL DES SOMMES PAYEES $
*****

```

4.3. Modification de description :

4.3.1. Présentation :

La maintenance des programmes d'édition est toujours un problème délicat, accru ici par le fait que l'affinage du dessin à l'aide de la maquette amène normalement des mises à jour .

L'éditeur Cobol a de grosses lacunes dans ce sens, certaines modifications pouvant demander la révision complète de toute la description (un changement de numéro de colonne par exemple).

Ces lacunes ont été en partie comblées par l'utilisation du descripteur : il est toujours possible de corriger les bordereaux constituant le descripteur (ce qui est en général plus rapide et plus simple que de modifier la description) puis de retaper les cartes pour recommencer la génération.

Mais elles n'ont pas été supprimées ; par exemple un changement de numéro de colonne peut demander la modification du dessin mais aussi celle des bordereaux 3 et 4.

Il nous a donc semblé indispensable d'assurer une mise à jour automatique des descriptions générées, peu coûteuse et fiable. C'est pourquoi, afin d'augmenter l'intérêt et la souplesse du système, nous avons mis au point un ensemble de programmes qui transforment la description obtenue par génération en utilisant des cartes de mise à jour dont nous étudierons le format plus loin et qui donnent, sous une forme concise, les modifications à apporter à la description. La description mise à jour sera alors utilisée comme la description initiale.

Etant donné le nombre et la diversité des changements que l'on peut vouloir apporter à un état, nous avons dû restreindre le nombre des modifications possibles en les classant en quatre types qui doivent permettre, en les combinant, d'effectuer tous les changements désirés :

1. les suppressions de groupes, de lignes ou de propositions SOURCE ou SUM ;

.../...

2. les modifications ou adjonctions de lignes du niveau RD, d'en-têtes de groupes, de lignes d'un groupe ;
3. les modifications ou adjonctions de propositions SOURCE ou SUM ;
4. les modifications portant sur le numéro d'une colonne ou le caractère figurant dans une colonne au niveau de l'état, d'un groupe ou d'une seule ligne ; ou bien les modifications d'espacement de lignes.

4.3.2. Schéma général de la mise à jour :

Il nous a semblé difficile de modifier directement la description générée sans restreindre considérablement les possibilités de mise à jour à moins de demander un nombre prohibitif de spécifications.

Nous avons alors choisi de recréer un fichier intermédiaire 'FINT' et un fichier 'SOURCE/SUM' analogues aux fichiers utilisés lors de la génération puis de mettre à jour ces deux fichiers.

Ceci permet de concevoir les modifications à partir du dessin à deux dimensions et non plus sur la structure arborescente de la description Cobol :

- Le fichier 'FINT' est en effet très proche du dessin de l'état ; en conséquence il est possible de 'visualiser' des modifications telles que changer un caractère ou déplacer une colonne ; de plus, contrairement à l'éditeur, ces modifications pourront être effectuées automatiquement à partir de données limitées (par exemple, pour déplacer une colonne dans un groupe, il suffira de spécifier le code du groupe, le numéro de colonne initial et le numéro final).
- Dans le fichier 'SOURCE/SUM', on pourra très facilement accéder à une proposition particulière en en donnant simplement les références (code de groupe, référence de ligne et numéro de colonne).

Bien entendu, les modifications seront effectuées simultanément sur les deux fichiers sans que l'utilisateur ait à le spécifier : la suppression d'une ligne entraînera non seulement la suppression de l'enregistrement correspondant du fichier 'FINT' mais aussi celle des propositions SOURCE et SUM relatives à cette ligne.

.../...

Pour limiter le nombre des documents d'entrée nous avons, dans la mesure du possible, conservé les bordereaux du descripteur. Ils sont toutefois complétés par des cartes dont nous étudierons la structure plus loin.

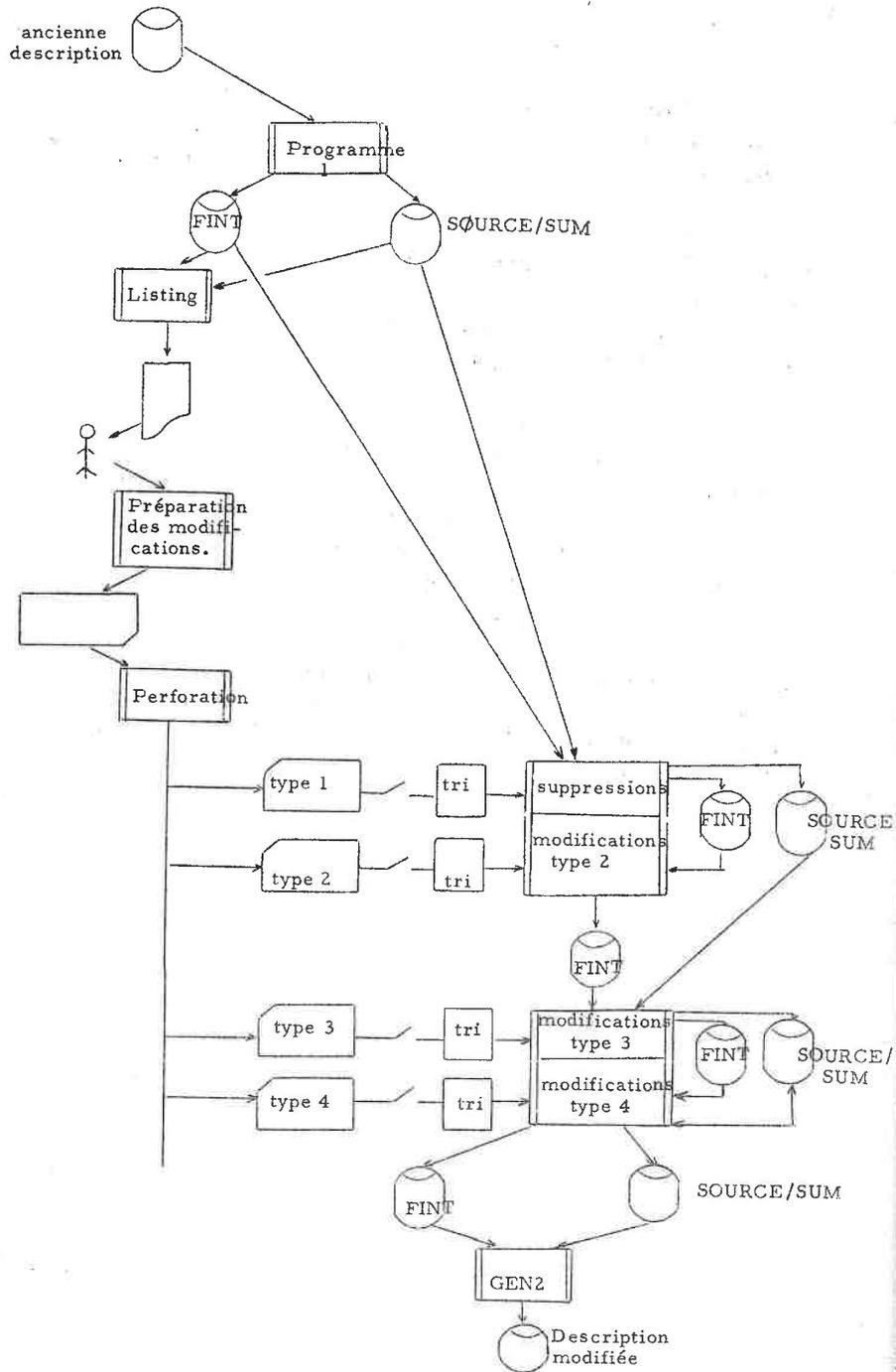
Cette méthode présente cependant un inconvénient pour l'utilisateur : le fichier intermédiaire 'FINT' comprend le dessin de l'état représenté avec certaines codifications ; au moment où l'utilisateur élabore les modifications à partir du listing des deux fichiers, il doit tenir compte de la codification adoptée comme nous le verrons plus loin.

Les modifications de chaque type, transcrites sur cartes perforées, sont rassemblées dans un fichier ; ces fichiers sont pris en compte dans l'ordre des types. Comme il n'existe pas obligatoirement de modification de chaque type donc de fichier correspondant, une clé est associée à chacun des fichiers ; elle est positionnée seulement si le fichier existe (une clé sera représentée par le symbole $_ /$ sur les schémas).

Un premier programme prend en compte les modifications de type 1 et 2, un second les modifications de type 3 et 4, l'utilisateur n'en demandant l'exécution que si des modifications de l'un des types cités existent. Les deux fichiers mis à jour sont utilisés par le second programme (GEN2) du générateur qui crée la description modifiée.

Le déroulement schématique des étapes est ainsi :

.../...



Nous détaillerons, dans les paragraphes suivants, outre la création du fichier intermédiaire et du fichier 'SOURCE/SUM', les procédures de mise à jour et les spécifications correspondantes données par l'utilisateur.

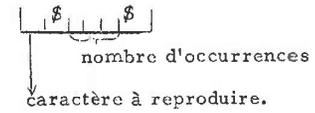
Nous insisterons surtout sur les données utilisées par les programmes pour effectuer les mises à jour ; étant donné leur faible intérêt, les programmes eux-mêmes ne seront pas détaillés, nous nous contenterons de donner quelques précisions sur leurs effets au niveau des fichiers dont la connaissance peut faciliter l'élaboration des données.

Nous nous référerons, tout au long de l'exposé à l'exemple de l'édition de l'état bancaire et terminerons ce chapitre par l'examen d'un groupe complet de modifications passées sur cet état mettant en évidence les possibilités globales du système de mise à jour.

4.3.3. Création du fichier intermédiaire et du fichier 'SOURCE/SUM' (programme 1) :

1. Le fichier 'FINT' est identique au fichier intermédiaire constitué lors de la génération.

Rappelons simplement que les symboles \$JUSQUE\$ de répétition de caractères sont codifiés de la façon suivante :



Ce groupe de caractères est placé à partir de la colonne où débute la répétition.

C'est la seule codification dont l'utilisateur doit tenir compte dans l'élaboration de ces modifications : cet ensemble de caractères doit être considéré par l'utilisateur comme un libellé quelconque qui peut donc être affecté par les modifications portant sur les colonnes où il se trouve ; on peut donc le modifier comme un libellé habituel mais si, au cours d'une modification envisagée par l'utilisateur, un des caractères de ce groupe est modifié en rendant la codification incorrecte, celui-ci doit le restituer ensuite par une autre modification.

... / ...

FICHIER SØURCE/SUM

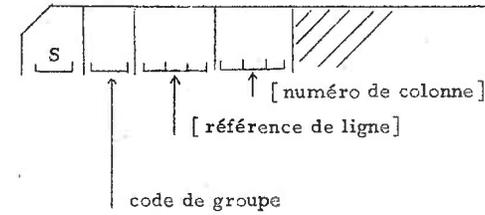
1010042	X(40)	SØURCENØMBØ
1010083	X(10)	SØURCEDATUM
2030041	9(4)	SØURCEMATR
2030046	X(15)	SØURCENØM
2030062	X(10)	SØURCEPRENØM
2030073	X(12)	SØURCENØCP
2030087	Z(4)·ZZ	SØURCESØMPIE
3035041	9(4)	SØURCEMATR
3035046	X(15)	SØURCENØM
3035062	X(10)	SØURCEPRENØM
3035073	X(12)	SØURCENØCP
3035087	Z(4)·ZZ	SØURCESØMPIE
4045085	Z(6)·ZZ	SUM SØMPIE
5060085CPTB	Z(6)·ZZ	SUM SØMPIE
5060085UPØN BAGENT		
5070085CPTL	Z(6)·ZZ	SUM SØMPIE
5070085UPØN LAGENT		
5080085	Z(6)·ZZ	SUM CPTB
5080085		CPTL

1

4.3.4. Les procédures de mise à jour :

4.3.4.1. Les suppressions (type 1) :

a) les données : l'utilisateur a la possibilité de supprimer un groupe, une ligne ou une proposition SØURCE ou SUM ; le format des cartes est le suivant



		Ø	Ø	Ø
Pour la suppression d'un groupe d'une ligne d'une proposition SØURCE/SUM				
Porter	le code du groupe	1	1	1
	la référence de ligne		2	2
	le numéro de colonne			3

Le programme supposant que les cartes sont triées sur les huit premiers caractères par ordre croissant, l'utilisateur devra s'occuper de provoquer ce tri.

b) la mise à jour : elle affecte aussi bien le fichier 'FINT' que le fichier 'SØURCE/SUM' :

actions	Pour une suppression de groupe	de ligne	de proposition SØURCE/SUM
suppression dans 'FINT'	X	X	
suppression dans 'SØURCE/SUM'	X	X	X
ôter le caractère Ø de la colonne correspondante dans le fichier 'FINT'			X
			.../...

4.3.4.2. Les modifications ou adjonctions de lignes du niveau RD, d'en-têtes de groupes ou de lignes d'un groupe (type 2) :

a) les spécifications données par l'utilisateur :

Elles sont portées sur les bordereaux 1 et 2 du descripteur :

- Pour modifier la phrase RD ou bien pour modifier ou ajouter un en-tête de groupe, on utilise le bordereau 1 rempli de la manière habituelle ; les renseignements demandés étant peu nombreux, nous avons considéré que dans les deux cas, l'utilisateur pouvait les reporter intégralement.

- Pour ajouter une ligne, on la dessine entièrement sur le bordereau 2 (\$CØMME\$ n'est pas permis) avec une référence de ligne correspondant à sa place dans son groupe.

- Le bordereau 2 est encore utilisé lorsqu'on veut modifier une ligne repérée par son code de groupe et sa référence.

Dans la zone 'dessin' du bordereau, les seuls caractères pris en compte par le programme de mise à jour sont les caractères non blancs qui se substituent aux anciens caractères.

Il faut bien sûr prévoir la mise à blanc d'une colonne, le caractère utilisé est la barre de division / ; cette utilisation interdit son insertion par une modification de ce type ; elle sera toutefois possible grâce aux modifications de type 4 (code D) que nous verrons plus loin.

De même l'espacement de ligne n'est modifié que s'il n'est pas blanc.

Les cartes perforées à partir de ces bordereaux devront être classées comme pour la génération avant leur saisie par le programme de mise à jour.

b) la mise à jour du fichier 'FINT' :

1. s'il existe des cartes de code R, elles viennent remplacer les enregistrements de même code dans le fichier 'FINT'.

.../...

Pour chaque carte 'GRP01' (ou couple de cartes GRP01, GRP02) :

2. Recherche du groupe de même code dans le fichier 'FINT'.

3. si le groupe existe, il s'agit d'une modification ; l'enregistrement 'en-tête' est effacé et remplacé par les cartes de mise à jour.

4. sinon, c'est une adjonction qui entraîne l'insertion d'un enregistrement 'en-tête' constitué des cartes précédentes.

Pour chaque couple de cartes représentant une ligne :

5. Recherche de la ligne correspondante dans le fichier 'FINT'.

6. Si la ligne existe, c'est une modification ; la ligne représentée par les cartes est explorée de gauche à droite :

- l'espacement de lignes est modifié s'il n'est pas à blanc sur les cartes,

- tout caractère non blanc rencontré dans une colonne de carte vient se placer dans la colonne correspondante de la ligne du fichier intermédiaire à l'exception du caractère / qui provoque la mise à blanc de cette colonne.

7. Sinon la ligne est ajoutée au fichier 'FINT'.

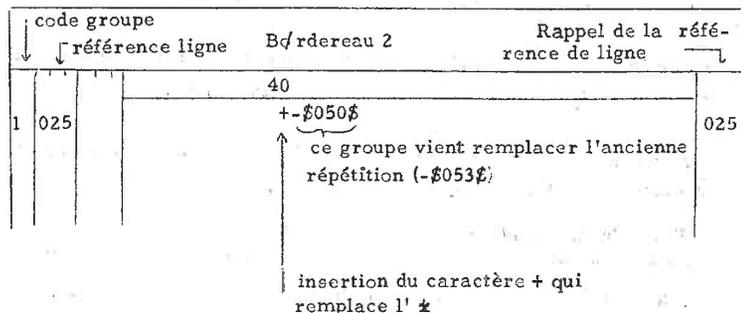
8. Dans les deux cas, la ligne est écrite sur le fichier 'FINT' après codification des symboles \$JUSQUE\$.

c) exemple :

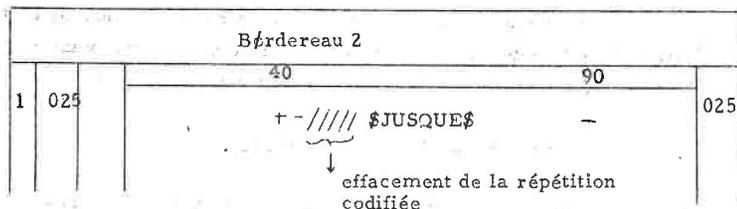
Sur la ligne 1025, on veut placer le caractère + en colonne 40 et répéter le tirtet 50 fois au lieu de 53 ; deux méthodes peuvent être envisagées :

.../...

- modifier la répétition sous forme codifiée :



- effacer la répétition codifiée et placer un symbole \$JUSQUE\$



Le symbole \$JUSQUE\$ marque la répétition du tiret de la colonne 41 à la colonne 90 (soit 50 fois).

4.3.4.3. : Les modifications ou adjonctions de propositions SOURCE ou SUM (type 3) :

a) les données : Elles sont portées sur les bordereaux 3 et 4.

Pour toute donnée émettrice ou tout compteur SUM, repéré par le code de groupe, la référence de ligne et le numéro de colonne, qu'il s'agisse d'une modification ou d'une adjonction, tous les renseignements doivent être reportés comme on le faisait pour la génération.

.../...

Le programme de mise à jour exige que les cartes perforées à partir de ces bordereaux soient classées dans le même ordre que celui qui est requis pour la génération.

b) les modifications : Le fait que l'on ait demandé à l'utilisateur de redonner toutes les caractéristiques d'une proposition rend la mise à jour très simple :

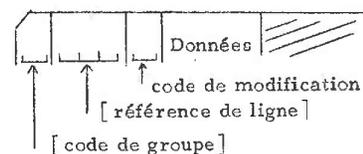
Pour chaque groupe de cartes représentant une proposition SOURCE ou SUM (un groupe peut être réduit à une carte) :

L'enregistrement correspondant existe au fichier 'SOURCE/SUM'	Ø	N
remplacer l'enregistrement par les cartes de mise à jour	1	
création d'un nouvel enregistrement du fichier 'SOURCE/SUM'		1
recherche de l'enregistrement ligne correspondant du fichier 'FINT' et insertion d'un caractère \$ dans la colonne indiquée.		2

4.3.4.4. Les modifications portant sur un espacement de lignes, le numéro d'une colonne ou le caractère figurant dans une colonne (type 4) :

a) Les cartes de données :

Elles ont toutes le même format :



.../...

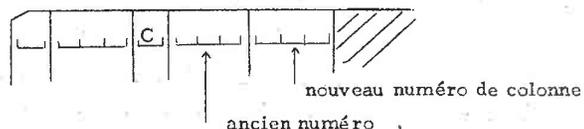
Les codes de modification sont :

- C : déplacement d'une colonne
- D : modification du caractère figurant dans une colonne
- E : changement d'espacement de lignes

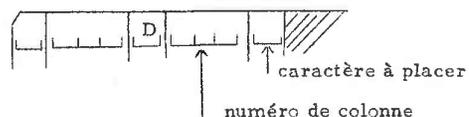
si la modification porte actions	sur tout l'état	sur un groupe	sur une ligne
on indique le code du groupe		1	1
la référence de ligne			2

La zone de donnée a, suivant les cas, la structure suivante :

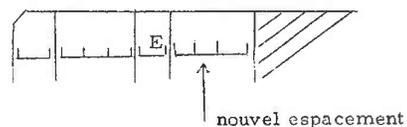
- pour un déplacement de colonne :



- pour la modification d'un caractère :



- pour le changement d'espacement :



Les cartes doivent être triées, automatiquement ou non, sur les cinq premiers caractères ; on a donc après le tri :

- les modifications de code C puis de code D intervenant sur tout l'état

.../...

Pour chaque groupe de l'état, dans l'ordre des codes de groupe :

- les modifications de code C puis de code D portant sur tout le groupe

Pour chaque ligne du groupe, dans l'ordre des références de ligne :

- les modifications de code C, D puis E.

b) Le déroulement de la mise à jour :

1. Mise en mémoire des cartes portant les modifications relatives à tout l'état.

Pour chaque groupe figurant au fichier 'FINT', dans l'ordre des codes de groupe :

2. Mise en mémoire des modifications portant sur tout le groupe.

Pour chaque ligne du fichier 'FINT' appartenant au groupe, dans l'ordre des références de ligne.

3. Mise en mémoire des modifications relatives à cette ligne uniquement.

4. Prise en compte des modifications dans l'ordre suivant :

modifications portant sur	tout l'état	tout le groupe	la ligne
Déplacement de colonne	1	3	5
Modification de caractère	2	4	6
Changement d'espacement			7

Déplacement d'une colonne : le caractère situé sur l'ancienne colonne est transféré dans la nouvelle colonne, l'ancienne colonne est mise à blanc ; si le caractère déplacé est un caractère \$ correspondant à une proposition SØURCE ou SUM, on modifie également le numéro de colonne dans l'enregistrement du fichier 'SØURCE/SUM'.

On peut donc changer ainsi l'origine d'une donnée éditée.

Modification du caractère figurant dans une colonne :

On place le caractère figurant sur la carte dans la colonne indiquée ; ce type de modification peut donc permettre de constituer des 'colonnades' d'un même caractère.

.../...

Modification d'espacement de lignes le nouvel espacement se substitue à l'ancien.

L'utilisateur doit bien entendu tenir compte de l'ordre dans lequel sont effectuées les modifications puisque celles-ci peuvent, dans certains cas, interférer l'une sur l'autre.

c) Exemples :

- si l'on veut éditer la date apparaissant sur la ligne 1010 en colonne 84 au lieu de 83, il suffira de construire la carte :

1	010	C	083	084
---	-----	---	-----	-----

le caractère \$ figurant sur l'enregistrement du fichier 'FINT' en colonne 83 sera déplacé ; de plus le numéro de colonne sera modifié dans l'enregistrement de références 1.010.083 du fichier 'SOURCE/SUM' (les références deviendront 1.010.084)

- On veut remplacer les caractères * par des signes + dans tout le groupe 5 ; une solution peut être la suivante :

On effectue tout d'abord des modifications du caractère figurant dans les colonnes 40 et 94 en les faisant porter sur tout le groupe ; on aura donc les deux cartes :

5			D	040	+
5			D	094	+

mais un caractère + sera apparu sur les lignes 5055 et 5085 en colonne 94 qu'il va falloir supprimer par deux autres modifications de ce type (on place un caractère espace dans la colonne 94 de ces lignes) :

5	05	5	D	094	
5	08	5	D	094	

.../...

- Ce type de modification permet de modifier une répétition de caractères : si l'on veut répéter 57 fois le caractère * figurant sur la ligne 1005 au lieu de 55, il suffira de modifier le libellé marquant cette répétition (on a, en colonne 40, * \$055\$) donc de remplacer le second chiffre 5 par le chiffre 7 :

1	005	D	044	7
---	-----	---	-----	---

4.3.5. : La génération de la description modifiée :

A partir du fichier intermédiaire et du fichier 'SOURCE/SUM' modifiés comme nous venons de l'indiquer, la seconde phase de la génération, c'est-à-dire l'exécution du programme GEN2 du générateur, est identique au cas normal.

Remarquons cependant qu'il a été nécessaire d'intégrer au préalable un tri du fichier 'SOURCE/SUM' ; les déplacements de colonne ont pu en effet rendre incorrect l'ordre des enregistrements de ce fichier.

La description ainsi générée pourra alors être utilisée comme la description initiale, pour l'éditer, pour imprimer la maquette de l'état ou par un ordre COPY à la compilation du programme d'édition.

4.3.6. Un exemple complet d'utilisation dans le cas de l'état bancaire :

Supposons que l'on veuille apporter à l'état bancaire défini au chapitre précédent les modifications suivantes :

a) Ajouter un groupe de début d'état imprimé en haut de page constitué d'une ligne portant le libellé "ETAT BANCAIRE" et la date du jour :

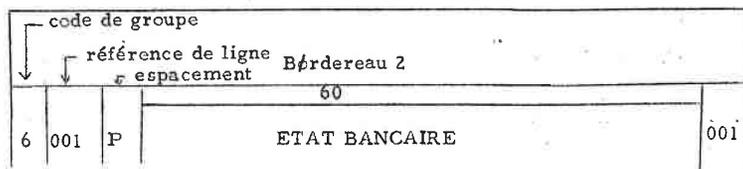
on placera donc :

- l'en-tête du groupe sur le bordereau 1 (modification de type 2) :

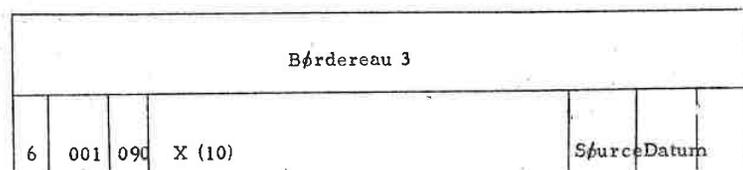
↓ code de groupe			Bordereau 1		
6	GRP	01	TYPE	RH	

.../...

- la ligne constituant le groupe sur le bordereau 2 (modification de type 2) :

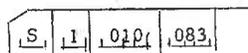


- la proposition SØURCE permettant d'éditer la date (colonne 90) (modification de type 3) :

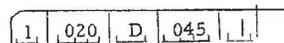


b) supprimer, dans le groupe 1, la date et placer un caractère | après le libellé MATR (on a constaté sur la maquette qu'il y manquait) :

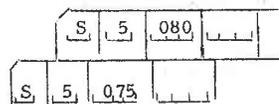
- la date figurant en colonne 83 de la ligne 1010 est supprimée par une modification de type 1



- puis on utilise une modification de type 4 code D



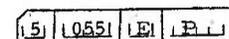
c) Dans le groupe des cumuls, supprimer le cumul total des sommes payées et la ligne qui le précède (modification de type 1) :



On veut également éditer ce groupe en haut de page : il suffit de modifier

.../...

l'espacement de la ligne 5055 (type 4, code E) :



d) Editer les sommes payées sur 7 chiffres (au lieu de 6) et les cumuls sur 9 chiffres (au lieu de 8) :

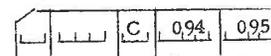
on modifiera donc les propositions SØURCE et SUM correspondantes (modifications de type 3) en portant les nouvelles spécifications sur les bordereaux 3 et 4.

BØRDEREAU 3					
2	030	087	Z(5).ZZ	SØURCE	SØMPIE
3	035	087	Z(5).ZZ	SØURCE	SØMPIE
4	045	085	Z(7).ZZ	SUM	SØMPIE
5	060	085	Z(7).ZZ	SUM	SØMPIE
5	070	085	Z(7).ZZ	SUM	SØMPIE

BØRDEREAU 4					
5	060	085	UPØN	BAGENT	
5	070	085	UPØN	LAGENT	

Mais ces modifications imposent d'adapter le cadre de caractères *

- il faut tout d'abord déplacer la colonne 94 en colonne 95 dans tout l'état (type 4, code C)



- ensuite, il faut modifier les lignes comportant une répétition de caractère ; c'est le cas des lignes 1005, 1015, 1025, 4040, 4050, 5055, 5065 et 5085 (les lignes 5075 et 5080 ont été supprimées) :

.../...

par exemple, le caractère * figurant sur la ligne 1005 doit être répété 56 fois ; il suffit donc de modifier le second chiffre 5 (colonne 44) par une modification de type 4, code D (changement du caractère figurant dans une colonne).

On peut adopter une méthode analogue pour toutes les autres lignes, d'où les cartes :

8	5	085	D	044	6
7	5	065	D	045	4
6	5	055	D	044	6
5	4	050	D	044	6
4	4	040	D	045	4
3	1	025	D	045	4
2	1	015	D	044	6
1	1	005	D	044	6

Les cartes ainsi élaborées devront ensuite être classées par type pour constituer des fichiers triés dans l'ordre que nous avons indiqué.

Les modifications effectuées sont visibles sur la maquette de l'état et le fichier 'SØURCE/SUM' que nous donnons ci-après :

.../...

Maquette

ETAT BANCAIRE

```

*****
* $
*****
*MATRI  NØM      |  PRENØM  |  INØ DE CØMPTÈ  |  SØMMES  *
-----|-----|-----|-----|
*$ | $          | | $        | | $        | | $        *
*$ | $          | | $        | | $        | | $        *
*$ | $          | | $        | | $        | | $        *
*$ | $          | | $        | | $        | | $        *
*$ | $          | | $        | | $        | | $        *
*$ | $          | | $        | | $        | | $        *
*$ | $          | | $        | | $        | | $        *
*$ | $          | | $        | | $        | | $        *
*$ | $          | | $        | | $        | | $        *
*$ | $          | | $        | | $        | | $        *
*$ | $          | | $        | | $        | | $        *
*$ | $          | | $        | | $        | | $        *
*$ | $          | | $        | | $        | | $        *
*$ | $          | | $        | | $        | | $        *
*$ | $          | | $        | | $        | | $        *
-----|-----|-----|-----|
*                               SØMME TØTALE A PAYER : $
*****
    
```

```

*****
* $
*****
*MATRI  NØM      |  PRENØM  |  INØ DE CØMPTÈ  |  SØMMES  *
-----|-----|-----|-----|
*$ | $          | | $        | | $        | | $        *
*$ | $          | | $        | | $        | | $        *
*$ | $          | | $        | | $        | | $        *
*$ | $          | | $        | | $        | | $        *
*$ | $          | | $        | | $        | | $        *
*$ | $          | | $        | | $        | | $        *
*$ | $          | | $        | | $        | | $        *
*$ | $          | | $        | | $        | | $        *
*$ | $          | | $        | | $        | | $        *
*$ | $          | | $        | | $        | | $        *
*$ | $          | | $        | | $        | | $        *
-----|-----|-----|-----|
*                               SØMME TØTALE A PAYER : $
*****
    
```

```

*****
* CUMUL DES SØMMES PAYEES PAR BANQUE ØU CCP $
-----|-----|-----|-----|
* CUMUL DES SØMMES PAYEES EN LIQUIDE      $
*****
    
```

Fichier Source/Sum

1010042	X(40)	SOURCENØMBW
2030041	9(4)	SOURCEMATR
2030046	X(15)	SOURCENØM
2030062	X(10)	SOURCEPRENØM
2030073	X(12)	SOURCENØCP
2030087	Z(5)*ZZ	SOURCEØMPIE
3035041	9(4)	SOURCEMATR
3035046	X(15)	SOURCENØM
3035062	X(10)	SOURCEPRENØM
3035073	X(12)	SOURCENØCP
3035087	Z(5)*ZZ	SOURCEØMPIE
4045085	Z(7)*ZZ	SUM ØMPIE
5060085	Z(7)*ZZ	SUM ØMPIE
5060085UPØN	BAGENT	
5070085	Z(7)*ZZ	SUM ØMPIE
5070085UPØN	LAGENT	
6001090	X(10)	SOURCEDATUM

5. Conclusion

Notre souhait a été de mettre en place un système complet d'aide à l'analyse et à la programmation de la phase d'édition qui s'étende au delà de la programmation jusqu'à la maintenance des programmes :

- Au niveau de l'analyse, le descripteur de documents simplifie la tâche du concepteur et les problèmes d'interface entre utilisateur et informaticien, limite le volume des informations à transmettre grâce aux pré-impressions donc le risque des erreurs de transcription et permet ainsi un gain de temps appréciable.

- Au niveau de la programmation, le générateur dispense intégralement le programmeur de son travail de codification du dessin et de traduction en langage évolué.

Les avantages sont indéniables : gain de temps énorme, suppression des erreurs de syntaxe, fiabilité totale, d'où un coût inférieur en heures homme et machine et même au niveau des cartes perforées puisque nous avons constaté que le nombre de cartes était réduit d'environ moitié ; ainsi dans la chaîne de traitement de la paie :

Exemples	Cartes d'éditeur	Cartes de descripteur
Bulletin de salaire	330	140
Journal des salaires	340	150
Etats récapitulatifs	400	230
Etat bancaire	63	35

- Au niveau de la mise au point, la maquette d'état répond au besoin du gestionnaire en lui proposant un exemplaire des éditions simulant exactement la réalité, sur lequel il peut réfléchir et décider des modifications à apporter, beaucoup plus tôt que par les moyens classiques, à un coût moindre. Le gain de temps est ici encore indéniable.

.../...

- Enfin, au niveau de la mise au point comme de la maintenance des programmes d'édition, les programmes de mise à jour permettent, à partir d'informations limitées, une mise à jour automatique donc avec moins de risques d'erreur et plus rapide.

Dans le même ordre d'idée, le programme d'édition de la description générée permet d'avoir en permanence des dossiers nets et précis en concordance avec les programmes.

Ce système présente cependant quelques inconvénients :

- Tout d'abord, il exige un compilateur muni d'un éditeur Cobol.
- Son emploi demande de la part de l'utilisateur qu'il connaisse, au moins approximativement, la syntaxe du langage Cobol ; il semble également qu'une certaine habitude soit souhaitable pour en utiliser efficacement les possibilités.

C H A P I T R E V

LA SØUS-CØMPILATIØN

1. Introduction :

Les nombreuses modifications que nous avons dû apporter aux programmes de la chaîne de paie étudiée dans la première partie nous ont amenés à rechercher des techniques de programmation facilitant l'écriture et la maintenance des programmes.

Une solution qui nous a semblé adaptée consiste à décomposer les programmes en modules indépendants.

Certaines possibilités du langage Cobol vont dans ce sens :

- le verbe PERFORM qui permet de décomposer un programme en 'pavés' donc de lui donner une structure plus claire et plus compréhensible ;
- les verbes COPY et INCLUDE dont nous avons parlé précédemment qui provoquent l'inclusion de parties de programmes écrites séparément dans une 'charpente' principale.

Cependant, malgré leur grand intérêt, elles n'assurent en aucun cas l'indépendance des différents modules puisqu'étiquettes et identificateurs sont valables pour tout le programme complet ; les différentes parties ne pourront, de ce fait, être rédigées indépendamment. De plus les zones de mémoire référencées dans un paragraphe utilisé par INCLUDE ou PERFORM doivent être déclarées dans le programme 'charpente' ; en conséquence, si l'on veut utiliser une zone supplémentaire dans un de ces paragraphes, il est indispensable d'inclure sa déclaration dans la 'charpente' principale en lui affectant un nom qui n'a pas été employé.

Il nous a donc paru intéressant d'utiliser conjointement une possibilité offerte par le langage Cobol ; s'il s'agit de la sous-compilation :

un programme 'logique' est divisé en deux ou plusieurs programmes distincts, compilés séparément puis recombines ensuite en un seul module exécutable ; un des programmes constitue le programme principal, les autres sont nommés sous-programmes.

L'indépendance des différentes subdivisions est alors presque totale, permettant ainsi d'atteindre le but que nous nous étions fixés :

- les identificateurs déclarés dans une subdivision et les étiquettes utilisées ne sont valables qu'à l'intérieur de celle-ci ;
- chaque subdivision possède ses propres mémoires de travail ce qui permet d'ajouter ou de supprimer une zone de ce type sans influencer le reste du programme.

De plus cette technique fait apparaître la notion de sous-programme Cobol, voisine de celle qui existe dans d'autres langages tels que Fortran ou Algol ; elle en apporte donc les avantages habituels sous certaines limites que nous précisons plus loin.

Remarquons toutefois que la sous-compilation est une extension du langage Cobol - 65 normalisé et, de ce fait, n'est pas utilisable sur tous les matériels.

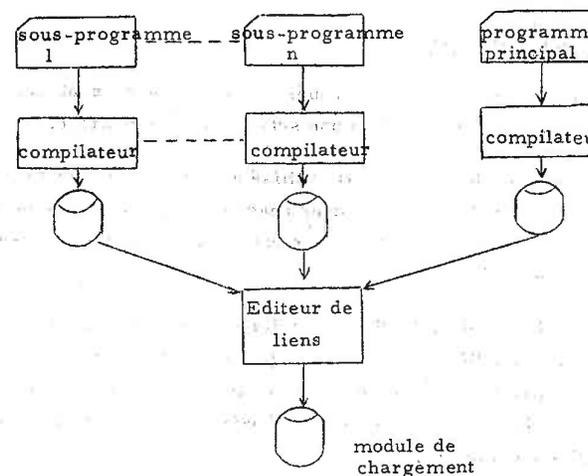
2. Les mécanismes de la sous-compilation :

Programme principal et sous-programmes ont le format d'un programme Cobol complet habituel. Ils sont compilés séparément comme des programmes quelconques ; on indique simplement s'il s'agit du programme principal ou d'un sous-programme (respectivement ! CØBØL MAIN et ! CØBØL SUB).

Cette séparation assure une indépendance totale entre les différentes parties en ce qui concerne les étiquettes et identificateurs utilisés.

C'est à l'édition de liens que les différentes parties sont combinées pour former un seul module de chargement que l'on pourra exécuter comme un programme normal :

.../...



L'exécution de ce programme va bien sûr soulever les deux problèmes habituels : le transfert du contrôle d'un programme à l'autre et la transmission des données (ou passage des paramètres) donc en fait l'allocation de l'espace mémoire.

2.1. Transfert de contrôle :

Il est extrêmement simple ; il se fait au moyen des instructions habituelles de branchement (GØ TØ, PERFORM et ALTER) ; aucun appel spécial n'est requis dans le programme appelant.

Par contre, tout point d'entrée dans une subdivision doit être déclaré comme 'définition externe' de façon à être accessible à d'autres subdivisions ; on utilise pour cela le verbe ENTER CØBØL suivi du nom de paragraphe ou de section utilisé comme point d'entrée.

2.2. La transmission des données :

L'allocation de l'espace mémoire est effectuée différemment selon les sections de la division des données :

.../...

- la section des fichiers :

Les différents fichiers, repérés par leur nom, sont attachés au programme complet et non à une subdivision particulière.

Un même fichier peut être utilisé dans des subdivisions différentes à condition d'être décrit dans chacune d'elles ; la zone enregistrement est alors commune donc accessible à tous les programmes où le fichier est déclaré.

On dispose donc là d'une première manière de transmettre des valeurs entre les différentes parties ; par exemple un enregistrement peut être lu par un premier programme, les valeurs se trouvant alors dans la zone enregistrement peuvent être utilisées par un autre programme si le fichier y est décrit.

- la section édition :

Les zones d'édition d'un même état, comme les zones enregistrements, sont communes à tous les programmes où l'état est décrit.

- la section des mémoires de travail :

Chaque programme a ses propres mémoires de travail ; les données qui figurent dans cette section ne sont pas partagées mais particulières à la subdivision dans laquelle elles sont définies.

La sous-compilation introduit une section supplémentaire, nommée section de liaison ; c'est une zone commune à toutes les subdivisions qui permet de transmettre des valeurs entre les différentes parties par la même technique que celle qui est utilisée dans le langage FORTRAN avec la zone COMMON. Les sections de liaison des différents programmes sont identiques et l'espace mémoire correspondant est partagé entre ceux-ci.

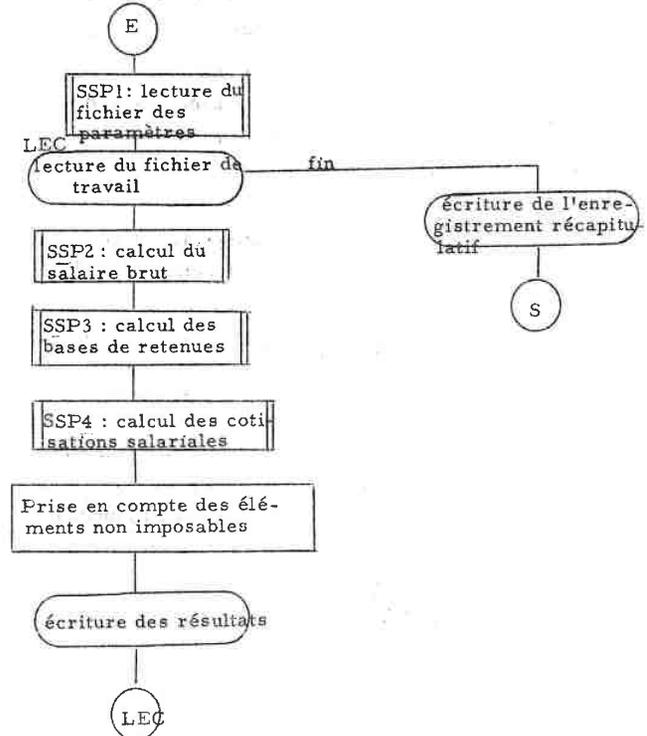
.../...

3. Exemples d'utilisation :

Nous avons employé la technique de la sous-compilation dans deux programmes de la chaîne de paie dont nous avons donné l'analyse dans la première partie :

a) Dans le programme de calcul des salaires en raison de son volume et des modifications nombreuses qui ont dû y être apportées.

Le programme est constitué d'un programme principal faisant appel à quatre sous-programmes (notés SSP i, i = 1 à 4) :



.../...

La section de liaison est incorporée, à la compilation, dans chacun des programmes grâce à un ordre COPY :

COMMON - STORAGE SECTION.

01 COMMUNE COPY COMMON.

où COMMON est le nom du fichier où se trouve la description des zones de cette section. Elle comprend en particulier :

- les paramètres de paie lus par le premier sous-programme et utilisés par le quatrième,
- les bases de retenues calculées par S S P 3 puis utilisées pour le calcul des cotisations salariales par S S P 4,
- des zones de cumul des bases et des cotisations ; les cumuls sont effectués, pour chaque agent, dans les programmes S S P 3 et S S P 4 puis écrits par le programme principal dans le fichier de la paie calculée (enregistrement récapitulatif).

Chaque sous-programme est appelé par un ordre PERFORM, par exemple, le programme (S S P 1) de lecture des paramètres est appelé par l'ordre : PERFORM LECPARAM ;

ce point d'entrée doit donc être déclaré dans le sous-programme :

PROCEDURE DIVISION.

BIDON SECTION.

DEB. ENTER CØBØL LECPARAM.

LECPARAM SECTION.

NOTE. Section permettant la lecture du fichier des paramètres ; ils sont placés dans les zones de la section de liaison.

FIN. EXIT.

Le fichier des paramètres qui n'est utilisé que par le programme S S P 1 est décrit uniquement dans celui-ci ; au contraire le fichier de travail est utilisé dans le programme principal et dans les sous-programmes S S P 2, S S P 3 et S S P 4 qui effectuent leurs calculs à partir des valeurs trouvées dans la zone enregistrement de ce fichier ; il est donc décrit

.../...

dans chacune de ces subdivisions de la manière habituelle (phrase SELECT et description en section des fichiers).

b) Dans le programme d'édition des bulletins de salaire et des récapitulatifs des retenues sociales.

Dans ce programme, on doit lire les paramètres de paie afin de connaître les taux des cotisations totales ; nous avons alors pu faire appel au sous-programme (S S P 1) utilisé dans le programme de calcul.

Il a bien entendu fallu inclure dans le programme principal la description de la section de liaison (par le même ordre COPY que précédemment). D'autre part, le sous-programme comportant un seul point d'entrée LECPARAM marquant un en-tête de section, son appel est obligatoirement PERFORM LECPARAM.

4. Possibilités et limites de la sous-compilation :

Ainsi que nous le voyons dans les exemples cités ci-dessus, les avantages de la sous-compilation sont de deux types :

a) les avantages de la modularité obtenue grâce à l'indépendance des différentes parties :

- Au niveau de la programmation, les différentes subdivisions peuvent être écrites indépendamment ce qui permet de diviser et de localiser les difficultés, les programmes sont d'une taille raisonnable donc d'une rédaction plus facile et plus sûre.

- Une modification apportée à une subdivision particulière n'a aucune influence sur les autres parties ; la maintenance des programmes en est considérablement facilitée.

La modularité obtenue n'est toutefois pas parfaite, le lien existant entre les différentes parties étant relativement rigide ; il n'a pas la souplesse obtenue grâce aux paramètres placés dans une séquence d'appel

.../...

qui assurent une véritable indépendance du sous-programme et permettent son utilisation en connaissant simplement la séquence d'appel tout en ignorant complètement son contenu.

Il semble au contraire qu'ici une connaissance précise de la section de liaison soit indispensable ; le lien est encore plus rigide si l'on utilise des fichiers et des états dans le sous-programme et le programme appelant puisque les zones enregistrement et les zones d'édition sont alors communes donc décrites de la même manière dans les deux programmes (bien que pouvant être découpées différemment).

b) Les avantages provenant de l'introduction de la notion de sous-programme :

Il est possible d'utiliser les sous-programmes comme on le fait dans d'autres langages tels que FORTRAN :

- dans plusieurs programmes d'une même chaîne de traitement,
- plusieurs fois dans un même programme bien que l'absence de paramètres dans la séquence d'appel rende cette utilisation plus délicate.

La sous-compilation présente toutefois l'inconvénient d'allonger assez sensiblement l'écriture des programmes ; en effet :

- chaque programme a le format d'un programme Cobol complet ;
- si un même fichier est utilisé dans plusieurs subdivisions, il doit être déclaré dans chacune d'elles ; il en est de même pour les états ;
- la zone commune (section de liaison) doit être décrite dans chaque subdivision.

Mais il est possible d'éviter la plus grande partie de ces réécritures par l'utilisation de l'ordre COPY : les descriptions des fichiers, des états et de la zone commune sont placées sur des fichiers puis incorporées dans les différentes parties au moment de la compilation par des ordres COPY se référant à ces différents fichiers.

C H A P I T R E VI

LA TABLE DES SALAIRES

1. Introduction

Le calcul des salaires des agents d'une entreprise est une 'fonction' classique mais qui, dans ses détails, varie d'une entreprise à l'autre ; ces différences ne justifient cependant pas que l'on recommence chaque fois l'étude à son début, en se heurtant aux mêmes difficultés depuis l'étude de l'existant jusqu'à la programmation sans tenir compte de l'expérience déjà acquise dans le domaine.

Comme nous l'avons signalé dans la conclusion de la première partie de ce travail, les difficultés sont le plus souvent une conséquence de la connaissance incomplète de la procédure existante ; d'autres études menées sur des problèmes de même nature ont également montré qu'il est en fait très difficile d'obtenir des responsables une documentation exacte et complète sur les règles de calcul utilisées, en particulier sur les cas d'exception.

Notre but a donc été de définir, dans une première étape, un descripteur de traitement adapté à la fonction 'paie' que nous nommons 'table des salaires' qui facilite l'appréhension des paramètres et permette un recensement complet de tous les détails de la technique de paie utilisée dans l'entreprise concernée ; il doit également servir de 'langage commun' entre le gestionnaire et l'analyste responsable du projet.

Il est élaboré par le concepteur au moment des enquêtes menées dans le service de paie puis soumis au responsable de ce service qui peut y apporter les modifications et compléments nécessaires et ensuite donner son accord, supportant ainsi une partie des responsabilités de l'étude de l'existant.

La table des salaires est alors un document de référence pour l'analyse et la programmation.

Ce document doit contenir toutes les informations permettant le calcul des salaires de tous les agents de l'entreprise ; il semble donc possible de supprimer la phase de programmation de ce calcul en générant automatiquement les ordres correspondants à partir des éléments figurant sur

.../...

VI - 102

le descripteur.

Cependant le document d'analyse doit revêtir une forme qui le rende compréhensible par un non - spécialiste donc faire appel autant que possible au langage naturel ; cette exigence rend impossible la génération à partir de ce seul document dans l'état actuel de nos possibilités.

C'est pour cette raison que la procédure définie sur la table doit être codifiée en une écriture plus stricte et plus proche des langages de programmation ; nous obtenons ce que nous pouvons appeler la 'méta-instruction' associée à la table des salaires.

Le passage du document d'analyse à la méta - instruction est effectué par un informaticien après l'analyse de la chaîne de traitement. Il est toutefois évident qu'il doit être suffisamment aisé si l'on veut que la génération ne perde pas tout son intérêt ; c'est dans ce but que nous avons donné une structure semblable à la table des salaires et à la méta-instruction associée, portée sur un document facilitant également ce passage.

La génération demande aussi de compléter la table par des documents annexes précisant les éléments qui y sont utilisés (en particulier la description des zones de mémoire).

Enfin, pour rendre l'usage de cette méta - instruction indépendant du matériel utilisé, nous générons le programme de calcul en langage Cobol ; le générateur lui-même est écrit dans ce langage.

2. Le document d'analyse :

Le descripteur de la fonction 'paie' est établi pour chaque entreprise selon les règles d'analyse que nous allons développer.

2.1. Structure de la table :

Il s'agit d'une table à deux entrées dont la structure rappelle celle du bulletin de salaire :

.../...

- sur les lignes apparaissent les actions à entreprendre c'est-à-dire, dans le cas d'une paie, les éléments que l'on doit calculer ; ces éléments sont soit une rubrique de paie (intervenant directement dans le salaire) soit un élément utilisé dans le calcul d'une de ces rubriques.

chaque ligne de salaire ainsi définie est précédée de la désignation en clair de cet élément.

- les entrées dans les colonnes sont de deux types :

1. La condition qui doit être satisfaite pour que le traitement placé sur la ligne de salaire soit entrepris. Il s'agit ici d'une condition en langage naturel qui doit être suffisamment explicite pour être comprise de tous ('agent horaire', 'cadre supérieur' ...)

2. Le traitement à entreprendre.

Cette organisation rappelle, dans ces grandes lignes, celle qui est utilisée dans la méthode CØRIG puisqu'à chaque tâche est associé un conditionnement.

Cependant le calcul d'un salaire présente une première particularité qu'il est intéressant de traduire dans la table : la structure arborescente de ce calcul reflétée par celle de la fiche de paie. En effet les rubriques non élémentaires sont obtenues par la somme algébrique d'autres rubriques déjà calculées ; par exemple un salaire brut pourra être :

salaire brut = salaire de base + heures supplémentaires + primes
+ rappels - retenues pour absence.

Nous avons alors choisi de traduire cette structure dans la table en utilisant la technique habituelle des nombres niveau qui donnent le lien vertical, l'ordre des lignes donnant le lien horizontal ; l'ordre des calculs à entreprendre est alors exactement défini.

Bien entendu, il aurait été possible de représenter les calculs par une simple liste mais, dans ce cas, les sommes algébriques permettant le calcul des rubriques non élémentaires auraient dû être explicitées alors qu'il suffit, avec cette structure, d'affecter à chaque élément un . signe

.../...

+ ou - indiquant s'il s'agit d'un gain ou d'une retenue à ajouter ou à soustraire à l'élément de niveau supérieur qui précède.

Si l'on reprend l'exemple du salaire brut cité plus haut, nous aurons :

Désignation	Condition	Traitement
Salaire brut		
+ salaire de base		
+ heures supplémentaires		
+ heures à 25 %		
+ heures à 50 %		
+ primes		
+ rappels		
- retenues pour absence		

(en fait l'indication + / - n'est pas portée à cet endroit dans le document définitif mais son sens sera le même).

Les nombres niveau ne sont pas notés explicitement, en décalage graphique suffisant à les représenter.

Cette structure présente deux autres possibilités :

- une condition portée sur une ligne d'un certain niveau est valable pour les lignes des niveaux inférieurs qui en dépendent sans qu'il soit utile de la reproduire (ce que l'on aurait dû faire sur une simple liste) ;
- le calcul d'une rubrique peut être décomposé autant que nécessaire par des lignes de niveau inférieur donnant, par exemple, le calcul d'un taux, d'une base ... ; il suffit d'indiquer que ces éléments n'interviennent pas directement dans le salaire (en omettant l'indication + ou -).

Dans l'exemple précédent, si l'on veut préciser que les heures supplémentaires n'existent que pour les agents payés à l'heure, que les heures à 25 % et 50 % sont calculées par un taux et un nombre obtenus sur des lignes de niveau inférieur, on pourra indiquer :

.../...

Désignation	Condition	Traitement
Salaire brut		
+ salaire de base		
+ heures supplémentaires	agent horaire	
+ heures à 25 %		
taux		
nombre		
+ heures à 50 %		
taux		
nombre		

* La condition 'agent horaire' est valable pour toutes les lignes qui suivent ;

* taux et nombre sont des quantités n'intervenant pas directement dans le salaire mais seulement dans le calcul du montant des heures à 25 % et 50 %.

2.2. Représentation des actions à entreprendre :

La description des traitements est spécifiquement adaptée au problème particulier du calcul d'un salaire.

Précisons tout d'abord que tous les éléments intervenant dans les calculs décrits sont repérés par un nom arbitraire choisi par l'utilisateur ; c'est ce nom qui est utilisé lorsque l'on veut se référer à un élément sur une ligne de salaire ; il est également utile pour une meilleure visualisation des calculs entrepris comme nous le verrons plus loin. Il sera donc indispensable de joindre à la table un 'lexique' donnant la désignation en clair des éléments ainsi codifiés.

Sur chaque ligne de salaire, la zone 'traitement' est composée de deux parties :

- le calcul proprement dit,
- le nom de l'élément calculé précédé d'un signe + ou - indiquant s'il s'agit d'un gain ou d'une retenue, d'un espace si l'élément n'intervient pas directement dans le salaire ; ce couple est placé dans la zone 'résultat'.

.../...

2.2.1. La zone de résultat :

Pour des raisons de clarté de présentation, une zone de ce type a été prévue pour chaque niveau.

Si l'élément est précédé du signe + (respectivement -) la valeur calculée est ajoutée (respectivement retranchée) à l'élément de niveau supérieur qui précède sans que cette action ait à figurer sur la ligne de ce dernier.

Sinon, l'élément est calculé mais la valeur obtenue n'est utilisée dans le calcul d'autres éléments que si l'on s'y réfère explicitement (par le nom de cet élément).

Cette notation n'est plus valable au niveau 1, elle n'est donc pas utilisée ; on suppose que tout élément de ce niveau est obtenu en ajoutant ou en retranchant les valeurs des rubriques qui le composent (selon que ces rubriques sont des gains ou des retenues) à la valeur de l'élément de niveau 1 qui précède (à zéro pour le premier) ; par exemple, pour obtenir le salaire net, on ôte du salaire brut les cotisations salariales payées par l'agent.

Sur l'exemple cité précédemment, nous aurons :

.../...

1	Désignation				condition	calcul	résultat			
	2	3	4	5			1	2	3	4
	Salaire brut						SALB			
	salaire de base							+PNHN		
	heures supplémentaires				agent horaire			+HSUP		
	heures à 25 %								+PH25	
	taux									TX25
	nombre									NH25
	heures à 50 %								+PH50	
	taux									TX50
	nombre									NH50
	(montant des heures)								HSUP	
	primes							+Primes		
	rappels							+Rapp.		
	diminution de salaire							-Dimin		
	(montant du salaire brut)							SALB		
	Salaire net						Salnet			

Remarque : Il est possible de placer dans la table des lignes de commentaires : le libellé porté dans la colonne 'désignation' est alors entouré de parenthèses. Ils permettent d'indiquer par exemple les traitements entrepris sur les lignes suivantes ou, comme sur l'avant-dernière ligne de l'exemple, de représenter la somme algébrique donnant le salaire brut sous une forme claire et imagée.

2.2.2. Le calcul proprement dit :

Si la ligne de salaire concerne une rubrique de paie non élémentaire, aucune indication supplémentaire n'est à porter ; en effet la somme algébrique calculant l'élément est entièrement précisée puisque l'on connaît les termes de la somme ainsi que leur signe. Dans l'exemple précédent, HSUP est calculé par la formule : $HSUP = PH25 + PH50$.

Pour les autres cas, la structure de la zone de calcul a été définie en remarquant que la plupart des éléments intervenant dans les calculs de ce type sont soit obtenus par le produit de deux termes (nombre et taux ou base et taux) soit ont une valeur donnée (fixe ou pour chaque agent).

.../...

Aussi la zone de calcul comporte -t-elle trois colonnes :

Désignation	condition	taux		nombre/base		Valeur donnée	résultat
		nom	calc.	nom	calc.		

a) si l'élément est obtenu par un produit :

On porte les deux termes de l'opération dans les colonnes correspondantes soit sous forme de valeurs numériques (valeurs fixes pour tous les agents) soit en les désignant par leur nom ; dans ce dernier cas, ils peuvent être obtenus de trois manières :

- des valeurs données (pour chaque agent) ;
- calculés par les traitements placés sur des lignes de niveau inférieur (le nom apparaît sur une de ces lignes dans la colonne de résultat précédé d'un espace) ;
- obtenus par une séquence de calcul décrite hors de la table et repérée par un code (sous-colonne 'calcul') lorsque la procédure ne peut être explicitée par les moyens précédents.

Exemple :

On veut calculer le montant des heures supplémentaires ; on trouve :

- des heures de route payées au taux normal TXH fixé pour chaque agent ; leur nombre NHRØUT est calculé par un bouclage (séquence R1) ;
- des heures à 25 % (taux TX25 * nombre NH25)
où TX25 = 1.25 * TXH, NH25 obtenu par la séquence R2 ;
- des heures à 50 % (taux TX50 * nombre NH50)
où TX50 = 1.50 * TXH, NH50 calculé par la même séquence R2.

On aura donc :

.../...

Désignation	condition	Taux		Nomb/base		Valeur donnée	Résultat
		nom.	calc.	nom.	calc.		
H. supplém.	horaire						+HSUP
H. de route		TXH		NHRØUT	R1		+PHRØUT
H. à 25 %		TX25		NH25	R2		+PH25
taux		1.25		TXH			TX25
H. à 50 %		TX50		NH50	R2		+PH50
taux		1.50		TXH			TX50
(montant total)							HSUP

b) si l'élément a une valeur donnée :

Cette valeur peut être un nombre ou repérée par un nom qui peuvent être placés

- soit directement dans la colonne 'résultat',
- soit dans la colonne 'valeur donnée' (l'élément placé dans la colonne 'résultat' prendra alors cette valeur),

selon la logique du calcul.

Exemple : La cotisation de retraite complémentaire est le produit de la base BRETC (séquence R4) par le taux TRC ; ce taux est différent selon la catégorie de l'agent (ouvrier, ETAM, cadre) ; les apprentis ne cotisent pas :

Désignation	condition	Taux	Nomb/base	Valeur	Résultat
cotizat. retraite	non apprenti	TRC	BRETC	R4	-RETCØMP
taux retraite	ouvrier			1.68	TRC
"	ETAM			2.10	TRC
"	CADRE			4	TRC

Toutefois, certains éléments sont obtenus par des calculs plus complexes que l'on ne peut pas représenter dans la table (en particulier les bouclages) ; aussi, afin de ne pas être restrictif, l'élément peut être obtenu par une séquence décrite à l'extérieur de la table ; dans ce cas, le nom de l'élément placé dans la colonne de résultat est accompagné du code de cette séquence placé dans la sous colonne 'calcul'.

.../...

Exemple : Calcul des rappels ; ils sont constitués :

- de rappels en somme donnés pour chaque agent y ayant droit (RAPPEL) ;
 - de rappels en heures pour les agents horaires ayant des cartes 'rappels' ;
- le montant de ces rappels, d'un calcul complexe, est obtenu par la séquence : RP :

Désignation	Taux	Nomb/ base	Valeur	Résultats			
				nom	code	nom	code
Rappels				+rappels			
Rappels en somme						+rappel	
Rappels en heures	horaire ayant des cartes 'rappels'					+rapph	RP
(montant total)						rappels	

2.3. Les documents annexes :

Le premier de ces documents introduit précédemment est un 'lexique' qui donne la signification en clair de tous les noms utilisés dans la table.

Le deuxième type de document sert à détailler les séquences de calcul référencées dans la table d'après les indications du service de paie ; un document est établi pour chacune d'elles. Il comprend la description en clair de la procédure à entreprendre ; cette description sera utilisée par l'informaticien qui programmera ces séquences pour les incorporer dans le reste du programme de calcul comme nous le verrons lors de la génération.

Nous donnons par exemple la séquence de calcul permettant d'obtenir le nombre des heures supplémentaires à 25 % et 50 % :

.../...

Règle	[C]	étiqu.	Ordre ou commentaire
R1	C		Pour chaque semaine du mois :
	C		- si le nombre d'heures est \leq 40, pas d'heure supplémentaire ;
	C		- sinon les huit premières heures supplémentaires sont payées au taux majoré de 25 %, les suivantes au taux majoré de 50 %

Chaque ligne est précédée du code 'C' marquant qu'il s'agit de commentaires ; la suite du document sera composé des lignes Cobol traduisant la procédure ainsi définie.

2.4. Un exemple d'utilisation :

Nous donnons ci-dessous la table des salaires adaptée au cas de la paie présentée dans la première partie. Elle est la traduction exacte de la procédure décrite dans le chapitre I.

Nous joignons à cette table le répertoire alphabétique des noms utilisés et une liste des séquences de calcul avec leur description simplifiée.

Séquences de calcul

Code	objet
R1	Calcul du nombre des heures ôtées, à 25 %, à 50 % par ventilation des heures de chaque semaine du mois.
R2	Calcul du nombre d'heures de route et de nuit ou dimanche.
R3	Calcul des taux horaires (heures à 25 %, 50 % et 100 %).
R4	Calcul de la prime de panier.
R5	Calcul du montant des rappels en heures.
R8	Calcul de la base S. S. totale.
R9	Calcul de la base retraite cadre ou prévoyance non cadre.

3. Le passage à une méta - instruction :

Il s'agit en fait d'une programmation très simple de la table des salaires que nous venons de décrire ; elle a pour but, comme nous l'avons dit, de permettre une génération ultérieure mais également de satisfaire les exigences dûes au passage sur ordinateur apparues après l'analyse organique de la chaîne ; de ce fait, elle doit être effectuée par un informaticien après l'analyse donc, en particulier, lorsque la structure des différents fichiers utilisés a été définie.

Le document résultant décrit une procédure complexe qui est ensuite traduite, par un générateur, en une suite d'instructions assimilables par l'ordinateur ; son contenu a donc été appelé 'méta - instruction'.

3.1. La programmation :

Le document permettant de composer cette instruction a une structure tout à fait semblable à celle du bordereau précédemment utilisé :

Code	Niv	Désignation	Condition	+/-	taux		Nbre/base		Valeur donnée	Résultat		
					nom	code	nom	calc		nom	code	ARR

La programmation comporte tout d'abord une retranscription ligne par ligne de la table aux modifications suivantes près :

- A chaque ligne de salaire est affecté un code quelconque arbitraire utilisé lors de la génération ; les codes doivent être croissants.
- Les niveaux des différentes lignes sont marqués explicitement par un nombre niveau.
- Les conditions sont traduites en langage Cobol ; par exemple, la condition 'agent horaire' se traduira par H - M = 'H' où H - M est le nom de la zone de mémoire contenant le code 'H' pour un agent horaire ou le code 'M' pour un agent payé au mois.

.../...

- Il n'existe plus qu'une seule zone de résultat, le décalage des zones de résultat avait pour seul but une meilleure visualisation des calculs entrepris ; de plus, l'indication +, - ou espace, précisant si l'élément est un gain, une retenue ou un élément n'intervenant pas directement dans le salaire, est placée dans une colonne spéciale.

- Dans le cas où l'élément est obtenu par le produit de deux termes, on précise également si le résultat doit subir l'arrondi habituel ('A' dans la sous-colonne ARR de la colonne de résultat).

Il peut ensuite arriver que l'analyse de la chaîne amène à certaines modifications mineures de la table :

- Certaines valeurs données sous forme de nombres dans la table peuvent apparaître maintenant sous forme de noms de zone ; c'est le cas des différents taux et plafonds lorsqu'ils sont considérés comme des paramètres.

- Une même séquence de calcul peut calculer plusieurs valeurs mais le nom de la séquence ne doit apparaître que sur la ligne du premier élément à calculer alors qu'il pouvait figurer plusieurs fois sur la table. Sur la table donnée en exemple, la séquence R3 calcule le taux des heures à 25 %, 50 % et 100 % ; dans la méta - instruction, le code R3 apparaîtra seulement sur la ligne de désignation 'heures à 25 %'.

- Certaines actions qui n'interviennent pas dans la logique du calcul elle-même mais sont simplement des conséquences du passage sur ordinateur n'apparaissent pas sur la table mais doivent figurer dans la méta - instruction ; l'informaticien aura donc à les expliciter. Par exemple, dans le cas où une rubrique a une valeur donnée pour chaque agent, il faut en général indiquer un transfert de cette valeur depuis le fichier d'entrée vers la zone de résultat ; il suffit pour cela de compléter la table en plaçant le nom de la zone d'origine dans la colonne 'valeur donnée'.

Dans le cas de la paie citée en exemple, on a sur la table :

Désignation		condition	taux	nombre/base	valeur	résultat	
1	2					1	2
	salaires de base						+PNHN

.../...

C'est une valeur donnée trouvée dans la zone SALM du fichier d'entrée ; la ligne est donc complétée de la façon suivante :

code	Niv	Désignation	condition	+/-	taux	nomb/base	valeur	résultat
005	02	salaires de base		+			SALM	PNHN

3.2. Exemple :

Nous présentons ci-dessous la méta - instruction programmée à partir de la table des salaires donnée au paragraphe précédent concernant le cas de paie de la première partie :

.../...

TABLE DES DONNEES

CODE	DESIGNATION	COMPOSITION	TAUX		MONTANT/BASE		RESULTAT	
			ident.	calc.	ident.	calc.	ident.	calc.
00001	SALAIRE NET	H * N = 'M'	+					SALO
00002	SALAIRE BRUT	H * N = 'M'	+					BRUT
00003	SALAIRE NET	H * N = 'M'	+					NET
00004	SALAIRE NET	H * N = 'M'	+					NET
00005	SALAIRE NET	H * N = 'M'	+					NET
00006	SALAIRE NET	H * N = 'M'	+					NET
00007	SALAIRE NET	H * N = 'M'	+					NET
00008	SALAIRE NET	H * N = 'M'	+					NET
00009	SALAIRE NET	H * N = 'M'	+					NET
00010	SALAIRE NET	H * N = 'M'	+					NET
00011	SALAIRE NET	H * N = 'M'	+					NET
00012	SALAIRE NET	H * N = 'M'	+					NET
00013	SALAIRE NET	H * N = 'M'	+					NET
00014	SALAIRE NET	H * N = 'M'	+					NET
00015	SALAIRE NET	H * N = 'M'	+					NET
00016	SALAIRE NET	H * N = 'M'	+					NET
00017	SALAIRE NET	H * N = 'M'	+					NET
00018	SALAIRE NET	H * N = 'M'	+					NET
00019	SALAIRE NET	H * N = 'M'	+					NET
00020	SALAIRE NET	H * N = 'M'	+					NET
00021	SALAIRE NET	H * N = 'M'	+					NET
00022	SALAIRE NET	H * N = 'M'	+					NET
00023	SALAIRE NET	H * N = 'M'	+					NET
00024	SALAIRE NET	H * N = 'M'	+					NET
00025	SALAIRE NET	H * N = 'M'	+					NET
00026	SALAIRE NET	H * N = 'M'	+					NET
00027	SALAIRE NET	H * N = 'M'	+					NET
00028	SALAIRE NET	H * N = 'M'	+					NET
00029	SALAIRE NET	H * N = 'M'	+					NET
00030	SALAIRE NET	H * N = 'M'	+					NET
00031	SALAIRE NET	H * N = 'M'	+					NET
00032	SALAIRE NET	H * N = 'M'	+					NET
00033	SALAIRE NET	H * N = 'M'	+					NET
00034	SALAIRE NET	H * N = 'M'	+					NET
00035	SALAIRE NET	H * N = 'M'	+					NET
00036	SALAIRE NET	H * N = 'M'	+					NET
00037	SALAIRE NET	H * N = 'M'	+					NET
00038	SALAIRE NET	H * N = 'M'	+					NET
00039	SALAIRE NET	H * N = 'M'	+					NET
00040	SALAIRE NET	H * N = 'M'	+					NET
00041	SALAIRE NET	H * N = 'M'	+					NET
00042	SALAIRE NET	H * N = 'M'	+					NET
00043	SALAIRE NET	H * N = 'M'	+					NET
00044	SALAIRE NET	H * N = 'M'	+					NET
00045	SALAIRE NET	H * N = 'M'	+					NET
00046	SALAIRE NET	H * N = 'M'	+					NET
00047	SALAIRE NET	H * N = 'M'	+					NET
00048	SALAIRE NET	H * N = 'M'	+					NET
00049	SALAIRE NET	H * N = 'M'	+					NET
00050	SALAIRE NET	H * N = 'M'	+					NET
00051	SALAIRE NET	H * N = 'M'	+					NET
00052	SALAIRE NET	H * N = 'M'	+					NET
00053	SALAIRE NET	H * N = 'M'	+					NET
00054	SALAIRE NET	H * N = 'M'	+					NET
00055	SALAIRE NET	H * N = 'M'	+					NET
00056	SALAIRE NET	H * N = 'M'	+					NET
00057	SALAIRE NET	H * N = 'M'	+					NET
00058	SALAIRE NET	H * N = 'M'	+					NET
00059	SALAIRE NET	H * N = 'M'	+					NET
00060	SALAIRE NET	H * N = 'M'	+					NET
00061	SALAIRE NET	H * N = 'M'	+					NET
00062	SALAIRE NET	H * N = 'M'	+					NET
00063	SALAIRE NET	H * N = 'M'	+					NET
00064	SALAIRE NET	H * N = 'M'	+					NET
00065	SALAIRE NET	H * N = 'M'	+					NET
00066	SALAIRE NET	H * N = 'M'	+					NET
00067	SALAIRE NET	H * N = 'M'	+					NET
00068	SALAIRE NET	H * N = 'M'	+					NET
00069	SALAIRE NET	H * N = 'M'	+					NET
00070	SALAIRE NET	H * N = 'M'	+					NET
00071	SALAIRE NET	H * N = 'M'	+					NET
00072	SALAIRE NET	H * N = 'M'	+					NET
00073	SALAIRE NET	H * N = 'M'	+					NET
00074	SALAIRE NET	H * N = 'M'	+					NET
00075	SALAIRE NET	H * N = 'M'	+					NET
00076	SALAIRE NET	H * N = 'M'	+					NET
00077	SALAIRE NET	H * N = 'M'	+					NET
00078	SALAIRE NET	H * N = 'M'	+					NET
00079	SALAIRE NET	H * N = 'M'	+					NET
00080	SALAIRE NET	H * N = 'M'	+					NET
00081	SALAIRE NET	H * N = 'M'	+					NET
00082	SALAIRE NET	H * N = 'M'	+					NET
00083	SALAIRE NET	H * N = 'M'	+					NET
00084	SALAIRE NET	H * N = 'M'	+					NET
00085	SALAIRE NET	H * N = 'M'	+					NET
00086	SALAIRE NET	H * N = 'M'	+					NET
00087	SALAIRE NET	H * N = 'M'	+					NET
00088	SALAIRE NET	H * N = 'M'	+					NET
00089	SALAIRE NET	H * N = 'M'	+					NET
00090	SALAIRE NET	H * N = 'M'	+					NET
00091	SALAIRE NET	H * N = 'M'	+					NET
00092	SALAIRE NET	H * N = 'M'	+					NET
00093	SALAIRE NET	H * N = 'M'	+					NET
00094	SALAIRE NET	H * N = 'M'	+					NET
00095	SALAIRE NET	H * N = 'M'	+					NET
00096	SALAIRE NET	H * N = 'M'	+					NET
00097	SALAIRE NET	H * N = 'M'	+					NET
00098	SALAIRE NET	H * N = 'M'	+					NET
00099	SALAIRE NET	H * N = 'M'	+					NET
00100	SALAIRE NET	H * N = 'M'	+					NET

4. La génération des instructions de calcul :

La table des salaires telle que nous l'avons conçue et , en conséquence, la méta - instruction correspondante, détaillent le calcul de la rémunération due à chaque agent ; le générateur aura donc pour but de traduire en langage Cobol la procédure ainsi définie.

Cependant le programme de paie doit comprendre, outre ce calcul, des initialisations, la cinématique des fichiers et d'autres traitements tels que les cumuls ; ces éléments ne figurent pas sur notre descripteur, leur programmation doit donc encore être effectuée manuellement.

C'est dans ce programme que le texte Cobol généré doit ensuite être inclu. La solution que nous avons retenue consiste à générer un sous-programme réalisant le calcul de la paie d'un agent appelé, pour chaque agent, par le programme charpente. Les deux textes source sont rassemblés en un seul module exécutable en utilisant la technique de la sous - compilation étudiée au chapitre précédent.

Cette méthode présente l'avantage d'assurer l'indépendance des programmes, l'utilisateur pouvant écrire et modifier la charpente principale sans se préoccuper des identificateurs et étiquettes utilisés dans la partie générée.

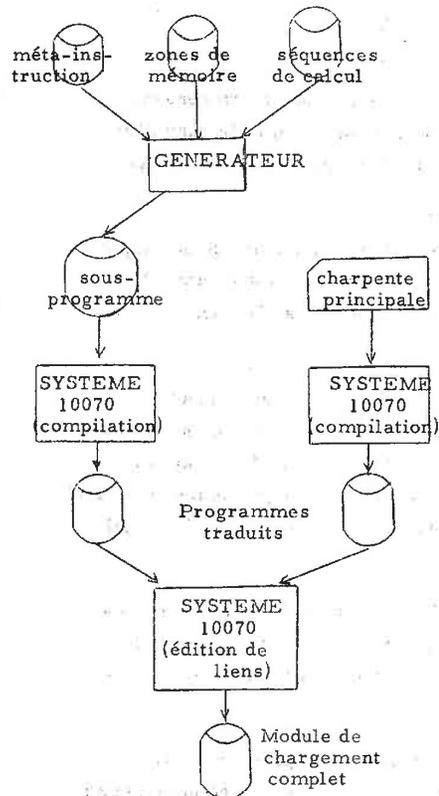
Mais le sous-programme généré a le format d'un programme Cobol complet ; il est donc indispensable de compléter la méta - instruction par :

- le texte Cobol des séquences de calcul extérieures,
- la description des zones de mémoire référencées dans l'instruction ou les séquences précédentes.

C'est le but des documents annexes que nous décrivons ci-après.

Le schéma général de la génération est alors le suivant :

.../...



4.1. Les documents annexes :

4.1.1. Les séquences de calcul :

Comme nous l'avons vu précédemment, l'informaticien dispose, pour chaque séquence référencée dans la table des salaires (donc dans la méta - instruction associée), d'un bordereau donnant la procédure à entreprendre rédigée en langage usuel.

Il complète alors chaque document par la traduction en langage Cobol de la procédure définie avant, chaque séquence a le format d'une section

.../...

de programme Cobol ; la première ligne doit porter le code de la séquence utilisé dans la table qui constituera le nom de la section dans le programme généré.

4.1.2. Les zones de mémoire :

Il s'agit de décrire chaque zone de mémoire référencée soit dans la méta - instruction soit dans les séquences de calcul annexes.

Un premier document est établi pour chaque fichier utilisé comprenant

- la phrase FD,
- la description de chaque enregistrement de ce fichier.

Un second document semblable au premier (pas de phrase FD) donne les zones de travail ou de la section de liaison.

Il s'agit en fait de descriptions en langage Cobol mais présentées sous une forme plus condensée utilisant la structure de tableau ; le générateur se charge de les compléter pour les rendre compatibles avec la syntaxe du langage. Les lignes apparaissent dans l'ordre logique des descriptions.

Les documents portent également la désignation en clair du contenu de chaque zone afin de les rendre plus lisibles et utilisables dans un dossier d'analyse ou de programmation.

Le format exact de ces documents annexes apparaît sur les exemples que nous donnons ci-dessous ; ce sont :

- les descriptions des zones utilisées dans la méta-instruction donnée au paragraphe précédent,
- une séquence de calcul programmée donnant la 'ventilation' des heures :

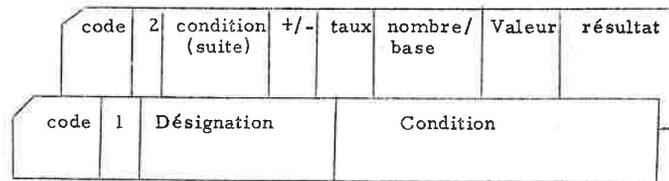
no	identifi- cateur	Re. définis copy	IMAGE	Usage	no pt	Libellé explicatif du contenu
52	IND					Section mémoires de travail
77	TXS		9(5)	COMP		indica semaine
77	TXS0		9(5)			Taux de l'heure à 25 %
77	TX400		9(5)			à 50 %
77	TAC		V9(4)			à 400 %
77	TAC49		V9(4)			Taux retraite complémentaire
77	TXM		9(5)			Taux cadre ou privilège
77	TXS5		9(5)			Taux de l'heure de travail
77	TXS0		9(5)			à 25 %
77	TX400		9(5)			à 50 %
77	TRAV		9(6)			à 400 %
09	DTX		9(7)			
01	FILLEA	DTX				États de calcul des taux horaires
01	DTX1		9(5)			
02	DTX2		99			
53						
77	NBS		9			Section de liaison
01	CTS		V9(4)			042 Taux cotisations sociales
01	PLF		9(6)			001 Plafonds

Régle	(C)	Étiquette	ORDRE ou COMMENTAIRES
R1	C		Calcul des nombres d'heures:
	C		Pour chaque semaine du mois:
	C		- si le nombre d'heures est < 40, on dit les heures non travaillées
	C		(le nombre d'heures).
	C		- sinon les 8 premières heures supplémentaires sont des heures
	C		à 25 %, les suivantes des heures à 50 %.
	C		- on prend en compte la rubrique 'heures à 0%'
R1	C		NOIE à TP IND.
		DT	IF HEUR (IND) < 40
		DT4	COMPUTE NHOT = NHOT + 40 - HEUR (IND).
			IF HEUR (IND) > 40 AND < 48
			COMPUTE NHIS = NHIS + HEUR (IND) - 40.
			IF HEUR (IND) > 48
			ADD 8 NHIS
			COMPUTE NH50 = NH50 + HEUR (IND) - 48.
			ADD 8 NH50 (IND) NHOT
			IF IND < NOS ADD 1 IND 04 TA ET1.
			EXIT.
		DL	

4.2. La génération du sous-programme de calcul :

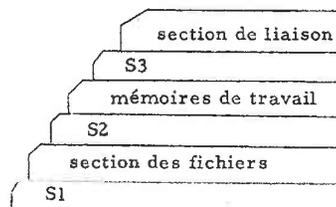
4.2.1. Perforation des documents en vue de la génération :

- Chaque ligne de la méta - instruction est transcrite sur deux cartes ; le code qui précède chacune d'elles est perforé au début de chaque carte suivi du numéro de séquence (1 et 2) :



Ce code permet d'associer les deux cartes représentant une ligne de salaire et d'en vérifier la présence ; de plus il rend possible un tri automatique des cartes (les deux cartes représentant une ligne doivent se suivre).

- Les documents portant les descriptions des zones utilisées sont perforés à raison d'une carte par ligne ; il est toutefois indispensable de préciser si l'élément décrit appartient à la section des fichiers, à celle des mémoires de travail ou bien à la section de liaison ; aussi les cartes représentant ces sections seront-elles précédées de cartes de séparation portant respectivement les codes S1, S2 et S3 sur les deux premières colonnes :



- Chaque ordre Cobol figurant sur les documents donnant les séquences de calcul est perforé sur une carte dans l'ordre du bordereau (les lignes de commentaire (codé 'C') ne sont pas perforées).

Les cartes ainsi obtenues sont réparties dans trois fichiers, le premier fichier 'TABLE' contient les cartes représentant la méta -instruction, le

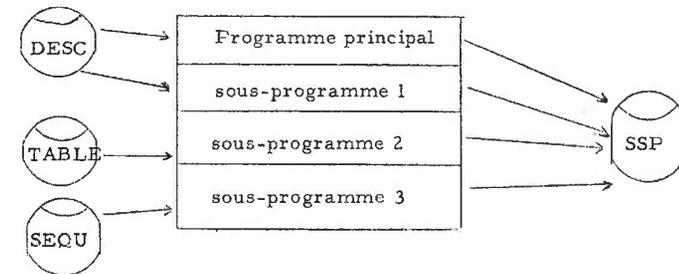
.../...

second 'DESC' est constitué des descriptions qui formeront la division des données du sous-programme généré et enfin le troisième 'SEQU' porte les ordres Cobol des séquences de calcul extérieures.

4.2.2. Le générateur :

Le générateur utilise les trois fichiers précédents pour générer le texte Cobol du programme de calcul écrit sur un fichier de sortie 'SSP'.

Il est constitué d'un programme principal et de trois sous-programmes :



4.2.2.1. Le programme principal :

1. Initialisation du fichier de sortie par le texte du début du sous-programme généré (division identification, section configuration).

Pour chaque description de fichier rencontrée dans le fichier 'DESC' :

2. Ecriture de la phrase SELECT correspondante.
3. A la rencontre d'une carte de code S1, S2 ou S3 du fichier 'DESC', écriture de l'en-tête de la section (fichiers, mémoires de travail ou liaison selon le cas) et appel du premier sous-programme qui en génère la description.
4. Appel du second sous-programme générant, à partir du fichier 'TABLE', les ordres de la division des traitements.
5. La division des traitements est complétée par les sections de calcul figurant au fichier 'SEQU' (sous-programme 3).

.../...

4.2.2.2. Les sous-programmes :

Les sous-programmes de génération de la division des données et des séquences de calcul, n'effectuant que de simples transcodifications des fichiers d'entrée 'DESC' et 'SECU' ne seront pas décrits ici.

Nous nous contenterons d'exposer le principe du sous-programme de traduction de la méta - instruction dont la compréhension peut aider à l'élaboration de celle-ci.

.../...

niveau 1

Pour chaque couple de cartes représentant une ligne et après vérification de la concordance des codes :

1. Si la ligne lue est de niveau plus élevé que la ligne précédente ou égal, génération des ordres Cobol correspondant aux lignes du niveau le plus bas jusqu'au niveau de la ligne qui vient d'être lue.
2. Si la ligne lue est de niveau 01, génération d'un ordre de transfert pour initialiser la zone de résultat par la valeur du niveau 01 qui précède ou par zéro pour la première.
3. Si un conditionnement est indiqué, génération d'une instruction conditionnelle.
4. Sauvegarde des rubriques + / - , taux, nombre/base, valeur et résultat en mémoire centrale.

.../...

niveau 2

3. Génération d'une instruction conditionnelle. :

3.1. s'il s'agit d'une ligne de niveau élémentaire (dont ne dépend aucune autre ligne de salaire), génération d'un ordre IF suivi de la condition donnée et rangement du niveau de la ligne.

3.2. sinon, génération d'un ordre IF NOT (condition) puis d'un ordre de branchement (GO TO) suivi d'une étiquette de format Z Z k où k est un entier de trois chiffres. Rangement du niveau de la ligne et de l'étiquette générée.

Remarque : ceci a pour but de permettre l'emploi des conditions imbriquées qu'il est impossible de traduire par des ordres IF simples.

.../...

niveau 2

1. Génération des ordres Cobol correspondant à une ligne de salaire :

Pour chaque sous-colonne 'calcul' :

- 1.1. génération, s'il y a lieu, d'un ordre PERFORM suivi du nom de la séquence de calcul.
- 1.2. si l'élément est calculé par le produit d'un taux par un nombre ou une base, génération d'une instruction MULTIPLY avec l'option GIVING ; la colonne 'arrondi' commande la génération de l'option ROUNDED.
- 1.3. si un nom figure dans la colonne 'valeur donnée', génération d'un ordre MOVE permettant le transfert de la valeur vers la zone de résultat.
- 1.4. si la rubrique + ou - n'est pas vide, génération d'un ordre d'addition ou de soustraction à l'élément de niveau plus élevé qui précède.
- 1.5. si un ordre IF simple a été généré pour cette ligne (rappelons que l'on a conservé le niveau au moment de la génération de l'instruction), génération d'une ligne portant un point de ponctuation.
- 1.6. s'il s'agissait d'un ordre IF NOT, génération en plus d'une ligne portant l'étiquette suivant l'ordre GO TO placé après l'instruction IF NOT.

4.2.3. Exemple :

Nous présentons ici le texte du sous-programme généré à partir de la méta - instruction et des documents annexes adaptés au cas de la paie de la première partie.

A titre d'illustration, nous donnons le principe de la génération de quelques lignes de ce sous-programme :

.../...

Niv	Désignation		+/-	taux	Nombre/ base	Valeur donnée	Résultat
02	Cadre ou prévoyance	A-T='T'	-	TRCAD	BRETCAD R9	RETCAD	A
03	taux prévoyance ouvrier	ØEN='Ø'				CTS(10) TRCAD	
03	" " ETAM	ØEN='E'				CTS(11) TRCAD	
03	taux retraite pet. cadres	NPS='P'				CTS(6) TRCAD	
03	" cadres sup	NPS='S'				CTS(7) TRCAD	
02	chômage						

- lecture de la ligne 'cadre ou prévoyance' :

* ce n'est pas une ligne élémentaire ; on génère donc une instruction :
 IF NOT (A-T = 'T')
 GØ TØ ZZ004.

* la ligne est rangée en mémoire

- lecture de la ligne 'taux prévoyance ouvrier' :

* c'est une ligne élémentaire donc on génère un ordre
 IF (ØEN = 'Ø')

* la ligne est rangée en mémoire.

- lecture de la ligne 'taux prévoyance ETAM' :

* génération des ordres correspondant à la ligne de niveau 03
 qui précède : MØVE CTS (10) TØ TRCAD

* une instruction IF de niveau 03 a été écrite donc on place sur la ligne suivante un point de ponctuation.

- Pour les lignes suivantes, la génération sera identique :

IF (ØEN = 'E')
 MØVE CTS (11) TØ TRCAD

IF (NPS = 'P')
 MØVE CTS (6) TØ TRCAD

IF (NPS = 'S')
 MØVE CTS (7) TØ TRCAD

.../...

- A la lecture de la ligne 'chômage', on génère également les ordres correspondant à la première ligne, c'est à dire :

- (1) PERFORM R9
- (2) MULTIPLY TRCAD BY BRETCAD GIVING RETCAD RØUNDED
- (3) SUBTRACT RETCAD FROM SALNET
- (4) ZZ004.

(1) la base est calculée par la séquence de calcul R9.

(2) la cotisation est obtenue par le produit de deux termes ; un arrondi est demandé.

(3) la cotisation est une retenue ; on ôte cette valeur de l'élément de niveau supérieur précédent, ici SALNET.

(4) un ordre IF NOT est relatif au niveau 02 ; on génère donc l'étiquette qui était placée après l'ordre GØ TØ.

IDENTIFICATION DIVISION.

PROGRAM-ID. CALCUL.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. C11-10070.
OBJECT-COMPUTER. C11-10070.
INPUT-OUTPUT SECTION.

FILE-CONTROL.

SELECT FREN5 ASSIGN TO DISC.
SELECT FPAIE ASSIGN TO DISC.

DATA DIVISION.

FILE SECTION.

FD FREN5 LABEL RECORD STANDARD DATA RECORD

ZFREN5

01 ZFREN5

02 FILLER PIC X(29)
02 PLSS PIC X
02 PLCAD PIC X
02 FILLER PIC X(28)
02 E-S PIC X
02 H-M PIC X
02 NPS PIC X
02 A-T PIC X
02 FILLER PIC XXX
02 TXH PIC 9(4)
02 SALM PIC 9(6)
02 FILLER PIC X(12)
02 6EN PIC X
02 FILLER PIC X

OCURS 005.

03 HEUR PIC 99V99
03 NUIT PIC 99V99
03 R0UT PIC 99V99
03 ENLY PIC 99V99
03 PANIER PIC 9(6)
02 RAPPEL PIC 9(6)
02 PRIMES PIC 9(6)
02 INDTE PIC 9(6)
02 LBY PIC 9(6)
02 0PP0S PIC 9(6)
02 AVAN PIC 9(6)
02 RETDIV PIC 9(6)
02 DIMSAL PIC 9(6)
02 FILLER PIC 9(6)

OCURS 006.

03 RNHN PIC 99V99
03 RNH25 PIC 99V99
03 RNH50 PIC 99V99
03 RNH100 PIC 99V99
03 RTH0R PIC 9(4)
02 N0RAP PIC 9

FD FPAIE LABEL RECORD STANDARD DATA RECORD

ZFPAIE

01 ZFPAIE

02 FILLER PIC X(93)
02 FILLER PIC X(93)
03 NHN PIC 99V99
03 PNHN PIC 9(6)
03 NHST PIC 99V99
03 PNHST PIC 9(6)

03 NHR0UT PIC 99V99
03 PNHR0UT PIC 9(6)
03 NH25 PIC 99V99
03 PH25 PIC 9(6)
03 NH50 PIC 99V99
03 PH50 PIC 9(6)
03 NH100 PIC 99V99
03 PH100 PIC 9(6)
02 PPAIER PIC 9(6)
02 PPRIMES PIC 9(6)
02 RAPPELS PIC 9(6)
02 DIMIN PIC 9(6)
02 SALB PIC 9(6)
02 FILLER PIC 9(6)
03 BSST PIC 9(6)
03 SSST PIC 9(5)
03 BSSPLF PIC 9(6)
03 MALAD PIC 9(5)
03 VIETL PIC 9(5)
03 BRETCPL PIC 9(6)
03 RETCOMPL PIC 9(5)
03 BRETCAD PIC 9(6)
03 RETCAD PIC 9(5)
03 CHOM PIC 9(5)
02 SALNET PIC 9(6)
02 INDEMN PIC 9(6)
02 SALT0T PIC 9(6)
02 LBYER PIC 9(6)
02 0PP0SI PIC 9(6)
02 AVANC PIC 9(6)
02 RETD PIC 9(6)
02 00MPIE PIC 9(6)
02 FILLER PIC X(30)

WORKING-STORAGE SECTION.

77 IND

C0MP

77 TX25 PIC 9(5)
77 TX50 PIC 9(5)
77 TX100 PIC 9(5)
77 TRC PIC V9(4)
77 TRCAD PIC V9(4)
77 RTXH PIC 9(5)
77 RTX25 PIC 9(5)
77 RTX50 PIC 9(5)
77 RTX100 PIC 9(5)
77 TRAV PIC 9(6)
01 DTX PIC 9(7)

01 FILLER REDEFINES DTX

02 DTX1 PIC 9(5)
02 DTX2 PIC 99

COMMON-STORAGE SECTION.

77 N0S PIC 0
01 CTS PIC V9(4)

OCURS 012.

01 PLF PIC 9(6)

OCURS 003.

PROCEDURE DIVISION.

BID0N SECTION.

DEB. ENTER C0B0L CALCUL.

CALCUL SECTION.

ZZ0.

MOVE 0 TO SALB
MOVE SALM TO PNHN
ADD PNHN SALB
MOVE DIMSAL TO DIMIN
SUBTRACT DIMIN FROM SALB

```

IF NOT (H=M = 'H')
GO TO ZZ001.
PERFORM R1
MULTIPLY TXH BY NH0T GIVING PNH0T ROUNDED
SUBTRACT PNH0T FROM SALB
PERFORM R2
MULTIPLY TXH BY NHR0UT GIVING PNHR0UT ROUNDED
ADD PNHR0UT SALB
PERFORM R3
MULTIPLY TX25 BY NH25 GIVING PH25 ROUNDED
ADD PH25 SALB
MULTIPLY TX50 BY NH50 GIVING PH25 ROUNDED
ADD PH25 SALB
MULTIPLY TX100 BY NH100 GIVING PH100 ROUNDED
ADD PH100 SALB

```

ZZ001.

```

IF (H=M = 'H')
PERFORM R4
ADD PPAINTER SALB
.
MOVE PRIMES TO PPRIMES
ADD PPRIMES SALB
ADD RAPPEL RAPPELS
IF (H=M = 'H' AND NBRAP NOT = 0)
PERFORM R5
ADD RAPH RAPPELS
.
ADD RAPPELS SALB
MOVE SALB TO SALNET
PERFORM R8
MULTIPLY CTS (1) BY BSST GIVING SSBT ROUNDED
SUBTRACT SSBT FROM SALNET
MOVE BSST TO BSSPLF
IF (PLSS = '1' AND BSST > PLF (1))
MOVE PLF (1) TO BSSPLF
.
MULTIPLY CTS (2) BY BSSPLF GIVING MALAD ROUNDED
SUBTRACT MALAD FROM SALNET
MULTIPLY CTS (3) BY BSSPLF GIVING VIEIL ROUNDED
SUBTRACT VIEIL FROM SALNET
IF NOT (A=T = 'T')
GO TO ZZ002.
IF (GEN = '0')
MOVE CTS (4) TO TRC
.
IF (GEN = 'E')
MOVE CTS (12) TO TRC
.
IF (NPS NOT = 'N')
MOVE CTS (5) TO TRC
.
IF (NPS = 'N' AND GEN = 'N')
MOVE 0 TO TRC
.
IF (NPS = 'N')
MOVE SALB TO BRET CPL
.
IF NOT (NPS NOT = 'N')
GO TO ZZ003.
MOVE SALB TO BRET CPL
IF (SALB > PLF (2))
MOVE PLF (2) TO BRET CPL

```

GO TO ZZ003.
ZZ003.

```

MULTIPLY TRC BY BRET CPL GIVING RETC0MPL ROUNDED
SUBTRACT RETC0MPL FROM SALNET

```

ZZ002.

```

IF NOT (A=T = 'T')
GO TO ZZ004.
IF (GEN = '0')
MOVE CTS (10) TO TRCAD
.
IF (GEN = 'E')
MOVE CTS (11) TO TRCAD
.
IF (NPS = 'P')
MOVE CTS (6) TO TRCAD
.
IF (NPS = 'S')
MOVE CTS (7) TO TRCAD
.
PERFORM R9
MULTIPLY TRCAD BY BRET CAD GIVING RETCAD ROUNDED
SUBTRACT RETCAD FROM SALNET

```

ZZ004.

```

IF (A=T = 'T')
MULTIPLY CTS (8) BY BSST GIVING CH0M ROUNDED
SUBTRACT CH0M FROM SALNET
.
MOVE SALNET TO SALT0T
MOVE INDETE TO INDEMN
ADD INDEMN SALT0T
MOVE SALT0T TO S0MPIE
MOVE L0Y TO L0YER
SUBTRACT L0YER FROM S0MPIE
MOVE SPP0S TO SPP0SI
SUBTRACT SPP0SI FROM S0MPIE
MOVE AVAN TO AVANC
SUBTRACT AVANC FROM S0MPIE
MOVE RETDIV TO RETD
SUBTRACT RETD FROM S0MPIE

```

ZZF. EXIT.

```

R1 SECTION:
D1. MOVE 1 TO IND.
ET1. IF HEUR (IND) < 40
      COMPUTE NH0T = NH0T + 40 * HEUR (IND).
      IF HEUR (IND) > 40 AND < 48
        COMPUTE NH25 = NH25 + HEUR (IND) - 40.
      IF HEUR (IND) > 48 ADD 8 NH25
        COMPUTE NH50 = NH50 + HEUR (IND) - 48.
      ADD ENLV (IND) NH0T.
      IF IND < NBS ADD 1 IND GO TO ET1.
D2. EXIT.
R2 SECTION:
D3. MOVE 1 TO IND.
ET2. ADD R0UT (IND) NHR0UT ADD NUIT (IND) NH100
      IF IND < NBS ADD 1 IND GO TO ET2.
D4. EXIT.
D3 SECTION:
D5. MULTIPLY 125 BY TXH GIVING DTX
      IF DTX2 = 0 MOVE DTX1 TO TX25 ELSE ADD 1 DTX1 GIVING
TX25. MULTIPLY 150 BY TXH GIVING DTX
      IF DTX2 = 0 MOVE DTX1 TO TX50 ELSE ADD 1 DTX1 GIVING
TX50. MULTIPLY 2 BY TXH GIVING TX100.
EXIT.
D6 SECTION:
D7. PERFORM ADP VARYING IND FROM 1 BY 1 UNTIL IND > NBS

```

```
ADR.   GO TO D8.
DB:    ADD PANIER (IND) PPAIER.
R5     EXIT.
R5     SECTION.
D9.    PERFORM CAL1 VARYING IND FROM 1 BY 1 UNTIL
      IND > NBRAP GO TO D10.
CAL1:  MOVE RTMOR (IND) TO RTXH.
      MULTIPLY RTXH BY RNHN (IND) GIVING TRAV ROUNDED
      ADD TRAV RAPH MULTIPLY 125 BY RTXH GIVING DTX
      IF DTX2 = 0 MOVE DTX1 TO RTX25 ELSE ADD 1 DTX1 GIVING
      RTX25. MULTIPLY RTX25 BY RNH25 (IND) GIVING TRAV
      ROUNDED ADD TRAV RAPH MULTIPLY 150 BY RTXH GIVING DTX
      IF DTX2 = 0 MOVE DTX1 TO RTX50 ELSE ADD 1 DTX1 GIVING
      RTX50. MULTIPLY RTX50 BY RNH50 (IND) GIVING TRAV
      ROUNDED ADD TRAV RAPH MULTIPLY 2 BY RTXH GIVING RTX100
      MULTIPLY RTX100 BY RNH100 (IND) GIVING TRAV ROUNDED
      ADD TRAV RAPH.
D10.   EXIT.
R8     SECTION.
D11.   MOVE SALB TO BSST
      IF E=S = 'S' MULTIPLY CTSL (9) BY SALB GIVING TRAV
      ROUNDED SUBTRACT TRAV FROM SALB GIVING BSST.
D12.   EXIT.
R9     SECTION.
D13.   IF BEN = '0' BR = 'E' MOVE SALB TO BRETCAD GO TO D14.
      SUBTRACT BRETCPL FROM SALB GIVING BRETCAD
      IF BRETCAD > PLF (3) AND PLCAD = '1' MOVE PLF (3) TO
      BRETCAD.
D14.   EXIT.
```

4.2.4. Appel du sous-programme généré :

Un seul point d'entrée a été prévu dans le sous-programme généré ; il est déclaré dans ce dernier par une instruction ENTER COBOL CALCUL.

Lors de l'écriture du programme principal, l'utilisateur doit prévoir l'appel du programme généré par un ordre PERFORM CALCUL.

5. Conclusion :

Dans leur forme actuelle, la table des salaires et le système de génération associé (méta-instruction et générateur) ne constituent qu'une première ébauche d'un système plus complet d'aide à l'analyse et à la programmation d'une chaîne de traitement de paie.

La table que nous avons définie permet cependant de remédier aux principales difficultés que nous avons rencontrées au moment de l'étude de la procédure existante :

- en donnant une liste exhaustive des éléments à calculer ;
- en présentant les règles de calcul utilisées dans les différents cas sous une forme claire et explicite sans en donner toutefois tous les détails ;
- en obligeant, par sa formalisation, à une analyse plus stricte donc plus complète.

On peut également penser que la structure arborescente donnée à l'ensemble des couples (conditionnement, action), si elle est particulièrement bien adaptée au cas particulier d'un calcul de paie, est peut-être utilisable pour d'autres problèmes de gestion.

La méta - instruction et le générateur associés permettent, quant à eux, une rédaction plus rapide du programme de calcul ; la modification des règles de calcul est également facilitée puisqu'il suffit de modifier la table puis conjointement la méta - instruction.

.../...

Un système plus complet permettrait sans doute une analyse et une programmation plus rapide et plus sûre ; on peut par exemple imaginer un schéma général fixé de chaîne de traitement adapté à chaque cas particulier.

Cette adaptation serait obtenue, comme pour la partie 'calcul' :

- par la définition de descripteurs et de méta - instructions associés ,
- par la génération des programmes correspondants à partir de ces méta - instructions.

Une extension de ce type a déjà été réalisée pour la phase d'édition grâce au système présenté au chapitre IV ; on peut cependant remarquer que, dans ce cas, la simplicité du problème avait permis de placer la méta - instruction sur le descripteur lui-même.

CONCLUSION

Les outils que nous venons de développer interviennent à deux niveaux dans l'élaboration d'une chaîne de traitement :

1. Au niveau de l'analyse par l'utilisation des descripteurs qui visent à faciliter l'appréhension des éléments d'entrée et à permettre une analyse stricte et complète en fournissant à l'utilisateur des documents clairs et formalisés qui cependant restent compréhensibles par un non - spécialiste.

2. Au niveau de la programmation et de la maintenance par la génération de programmes en langage évolué (ici le langage Cobol).

Cette phase en général se déroule en deux étapes, l'une manuelle, l'autre automatique :

a) Construction d'une méta - instruction à partir du descripteur utilisé lors de l'analyse par traduction du contenu de celui-ci en un langage plus 'informatique' tenant compte des exigences dues au passage sur ordinateur.

b) Génération du texte Cobol correspondant à cette méta - instruction, incorporé ensuite dans la 'charpente' écrite par l'utilisateur.

Dans les cas simples, la première de ces deux étapes peut être inexistante si la méta - instruction est suffisamment lisible (le même document fait alors fonction de descripteur et de méta - instruction). Dans les autres cas, cette codification manuelle semble encore indispensable.

C2

Nous avons utilisé ce principe dans le système d'édition et dans la génération du programme de calcul de paie (table des salaires).

Il est sans doute possible de construire, en suivant la même idée, d'autres systèmes d'aide à l'analyse et à la programmation adaptés à d'autres parties de la chaîne de paie, comme nous l'avons déjà suggéré ou même à d'autres fonctions de gestion.

Cependant la solution que nous avons choisie, si elle est adaptée aux cas que nous avons traités, n'est pas la seule que l'on puisse envisager ; on peut citer :

- Pour des fonctions particulières telles que la fonction 'paie', une solution couramment employée est la construction d'un programme général ou package incorporé dans chaque chaîne de traitement.
- Une seconde technique, surtout utilisée en calcul scientifique, consiste à construire une bibliothèque de programmes adaptés à des fonctions élémentaires assez générales et utilisables dans toute une classe de problèmes.

C'est la solution qui a été retenue par exemple pour les mises à jour de fichiers (programme général de mise à jour CLAMS utilisé dans la chaîne de paie de la première partie).

Cette technique paraît cependant difficile à étendre à des fonctions plus complexes.

- On peut également construire un langage permettant de décrire les éléments d'entrée, un compilateur adapté traduisant celui-ci pour le rendre assimilable par l'ordinateur (langages d'édition paramétrés par exemple).

Bien entendu, le choix entre ces différentes solutions est fonction de la nature du problème et du but qui est recherché.

A N N E X E S

D E L A P R E M I E R E

P A R T I E

ANNEXE I

Répertoire des documents d'entrée :

Code	Désignation	Création	Utilisation
E1	Fiche d'embauche.	imprimé	CLAMS
E2	Fiche de modification du fichier personnel	P2	CLAMS
E3	Fiche de modification des paramètres de paie	imprimé	P6
E4	Fiche de modification de la table des modes de paiement	imprimé	manuelle
E5	Bordereau de perforation des cartes 'activité'	P3	P5
E6	Bordereau de perforation des cartes 'supplémentaires'	P4	P5
E7	Bordereau de perforation des cartes 'rappels'	imprimé	P5
E8	Fiche de modification de paie calculée	imprimé	P12

- Fiche d'embauche -

E1

TYPE : imprimé

Première carte 1 8

Matricule A P 1 0

Nom 9 23

Prénom 24 33

Plafonné S. S. oui : 1 ; non : 0 34

Plafonné pour retraite cadre oui : 1 ; non : 0 35

Qualification 36 38

Numéro S. S. 39 51

Deuxième carte 1 8

Matricule A P 1 1

Date de naissance mois 9 jour 11 année 13

Date d'entrée mois 15 jour 17 année 19

Employé : E - Salarié S 21

Horaire : H - Mensuel : M 22

Non cadre : N - petit cadre : P - Cadrer supérieur : S 23

Apprenti : A - non apprenti : T 24

Cornité de prévention oui : P ; non : N 25

Chômage intempéries oui : C ; non : N 26

Code de paiement 27

Taux horaire (en centimes) 28 31

Salairé mensuel (en centimes) 32 37

Numéro de compte en banque ou CCP 38 49

Prévoyance : ouvrier : O - ETAM : E - non cotisant : N 50

E2

Fiche de modification du fichier personnel

TYPE IMPRIME PAR ORDINATEUR

* MATRICULEY 0890 NDMI FRANCOISE *
*112 516 819 119 128 1381391 *
*MI0890IP01 4 2 3 *
*112 516 819 112 125 131 *
*MI0890IP02 4 2 3 4 *
*112 516 819 11011112113114115116 *
*MI0890IP03 4 1 2 13 14 15 16 17 1 9 *
*112 516 819 113 119 *
*MI0890IP04 4 2 131 *
*MI0890IP *

- P01 : Pour une modification de matricule porter la nouvelle valeur dans la zone 1 nom 2 prénom 3
P02 qualification 1 numéro de S. S. 2 date de naissance 3 date d'entrée 4
P03 employé - salarié 1 horaire - mensuel 2 non cadre - petit cadre - cadre supérieur 3 apprenti - non apprenti 4 inscription au comité de prévention 5 inscription au chômage - intempéries 6 code de paiement 7 numéro de compte en banque ou CCP 9
P04 taux horaire (agents horaires) 1 salaire mensuel (agents payés au mois) 2 ouvrier - Etam 3

Départ d'un agent : porter la mention 'parti le ' sur la ligne 'S'

Paramètres de paye

Libellés							
1 ← Colonnes des cartes →	49	50	51	52			57
Plafonds							
Plafond S. S. -----	1	1					
Plafond retraite -----	1	2					
Plafond cadres -----	1	3					
Taux des cotisations salariales							
S. S. sur base totale -----	2	1					
S. S. maladie -----	2	2					
S. S. vieillesse -----	2	3					
Retraite complémentaire ouvriers--	2	4					
Retraite complémentaire cadre ---	2	5					
Retraite petits cadres -----	2	6					
Retraite cadres supérieurs -----	2	7					
Chômage -----	2	8					
Abattement pour les salariés -----	2	9					
Prévoyance ouvriers -----	2	A					
Prévoyance ETAM -----	2	B					
Retraite complémentaire ETAM ---	2	C					
Taux des cotisations totales							
S. S. sur base totale -----	3	1					
S. S. maladie -----	3	2					
S. S. vieillesse -----	3	3					
Allocations familiales -----	3	4					
Accidents du travail -----	3	5					
Chômage -----	3	6					
Congés payés -----	3	7					
Retraite complément. non cadres --	3	8					
Décès non cadres -----	3	9					
Retraite complémentaire cadres ---	3	A					
Prévoyance cadres -----	3	B					
Décès cadres -----	3	C					
Retraite petits cadres -----	3	D					
Retraite cadres supérieurs -----	3	E					
Prévoyance ouvrier -----	3	F					
Prévoyance ETAM -----	3	G					
Comité de prévention -----	3	H					

Fiche de modification de la table des modes de paiement

Code de Paiement	Libellé du mode de paiement
A	SOCIETE NANCIEENNE DE CREDIT
B	SOCIETE GENERALE NANCY
C	CREDIT LYONNAIS NANCY
D	BANQUE NATIONALE DE PARIS NANCY
E	CHEQUES POSTAUX NANCY
F	CAISSE MUTUELLE DEPOTS PRETS NANCY
G	BANQUE POPULAIRE DE LORRAINE NANCY
H	CAISSE REGIONALE CREDIT AGRICOLE NANCY
I	SOCIETE GENERALE POMPEY
J	CAISSE REGIONALE CREDIT AGRICOLE VANDOEUVRE
K	CAISSE REGIONALE CREDIT AGRICOLE FROUARD
L	LIQUIDE
Z	DIVERS.

Modification paie calculée

E8

TYPE : imprimé

Carte 1 : matricule 1 Nom 6

Prénom 21 Plafond SS 31 Plafond cadre 32

Qualification 33 N° SS 36

Date de naissance 49 Date d'entrée 55

E-S 61 H-M 62 N-P-S 63 A-T 64 P-N 65 C-N 66

Code de paiement 67 Taux horaire 68

Salaire mensuel 72

Carte 2 : N° de compte en banque 1 2

Ø - E - N 14

Heures normales :	nombre	<input type="text"/> 15	Montant	<input type="text"/> 20
Heures otées :	nombre	<input type="text"/> 26	Montant	<input type="text"/> 31
Heures de route :	nombre	<input type="text"/> 37	Montant	<input type="text"/> 42
Heures à 25 % :	nombre	<input type="text"/> 48	Montant	<input type="text"/> 53
Heures à 50 % :	nombre	<input type="text"/> 59	Montant	<input type="text"/> 64
Heures à 100 % :	nombre	<input type="text"/> 70	Montant	<input type="text"/> 75

Carte 3

Panier 1 3 Primes 8

Rappels 14 Diminution de salaire 20

Salaire brut 26

Base SS 32 1 % maladie 38

Base plafonnée 43 maladie 49 Vieillesse 54

Base retraite complémentaire 59 cotisation 65

Carte 4

Base retraite cadre ou prévoyance 1 4 cotisation 8

Base chômage 13 cotisation 19

Salaire net 24

Indemnités 30 salaire total 36

Loyer 42 oppositions 48

Avances 54 retenues diverses 60

somme payée 66

Carte 5

Nombre d'heures de rappels : normales 5

à 25 % 7

à 50 % 12

à 100 % 17

Date de paie (mois, jour, année) 22

ANNEXE 2

Répertoire des fichiers utilisés

Code	Désignation	Création	Utilisation
F1	Fichier personnel du mois précédent	mois précédent	CLAMS
F1.1	Fichier personnel modifié	CLAMS	P1, P3, P4, P5
F2	Fichier des enregistrements modifiés du fichier personnel	CLAMS	P1
F3	Enregistrements modifiés corrects	P1	P2
F4	Fichier de travail servant au calcul de la paie	P5	P7
F5	Fichier des paramètres	P6	P7, P11
F6	Fichier de la paie calculée	P7	P8, P9, P10, P11
F7	Fichier des salaires bruts (salariés)	P7	P10
F8	Fichier des salaires bruts (employés)	P7	P10
F9	Table des modes de paiement	cartes	P9

Fichier : Personnel et enregistrements modifiés				Code :	
Organisation séquentielle				F1	
Longueur enregistrement : 93				F2	
				F3	
				F1.1	
Enr	Niv	n° caract.	Désignation	image	Occ # 1
I	2	1	Matricule de l'agent	9(4)	
		5	libre	X	
		6	Nom de l'agent	X(15)	
		21	Prénom	X(10)	
		31	Plafonnement S. S. (0:non, 1:oui)	X	
		32	Plafonnement cadre (0:non, 1:oui)	X	
		33	Libre	X(3)	
		36	Qualification	X(3)	
		39	Numéro de S. S.	X(13)	
		52	Mois de naissance	99	
		54	Jour de naissance	99	
		56	Année de naissance	99	
		58	Mois d'entrée	99	
		60	Jour d'entrée	99	
		62	Année d'entrée	99	
		64	Employé E, salarié S	X	
		65	Horaire H, mensuel M	X	
		66	Non cadre N, petit cadre P, Cadre supérieur S	X	
		67	Apprenti A, non apprenti T	X	
		68	Comité de prévention (inscrit P, non N)	X	
		69	Chômage intempéries (inscrit C, non N)	X	
		70	Code de paiement (cf table des codes)	X	
		71	Taux horaire (centimes)	9(4)	
		75	Salaire mensuel (centimes)	9(6)	
		81	N° de compte en banque ou CCP	X(12)	
		93	ouvrier Ø, Etam E, sinon N	X	

Fichier : Fichier de travail servant au calcul de la paie				Code :	
Organisation séquentielle				F4	
Longueur enregistrement : 368					
Enr	Niv	n° caract.	Désignation	image	Occ # 1
I	2	1	Matricule de l'agent	9(4)	
	2	5	Nom	X(15)	
	2	20	Prénom	X(10)	
	2	30	Plafonnement S. S. (0:non, 1: oui)	X	
	2	31	Plafonnement cadre (0:non, 1: oui)	X	
	2	32	Qualification	X(3)	
	2	35	Numéro de S. S.	X(13)	
	2	48	Date de naissance(mois, jour, année)	9(6)	
	2	54	Date d'entrée (mois, jour, année)	9(6)	
	2	60	Employé E, salarié S	X	
	2	61	Horaire H, mensuel M	X	
	2	62	Non cadre N, Petit cadre P, Cadre Supérieur S	X	
	2	63	Apprenti A, non apprenti T	X	
	2	64	Comité de prévention (inscrit P, non N)	X	
	2	65	Chômage intempéries (inscrit C, non N)	X	
	2	66	Code de paiement	X	
	2	67	Taux horaire (centimes)	9(4)	
	2	71	Salaire mensuel (centimes)	9(6)	
	2	77	N° de compte en banque ou CCP	X(12)	
	2	89	Ouvrier Ø, Etam E, sinon N	X	
	2	90	Heures et primes de panier de chaque semaine		5
	3		Nombre d'heures de jour (1/100 d'heure)	9(4)	
	3		Nombre d'heures à 100 %	9(4)	
	3		Nombre d'heures de route	9(4)	
	3		Nombre d'heures à oter	9(4)	
	3		Montant de la prime de panier (centimes)	9(6)	

.../...

Enr	Niv	n° caract.	Désignation	image	Øcc ≠ 1
2		200	Rappels	9(6)	
2		206	Primes diverses	9(6)	
2		212	Indemnités	9(6)	
2		218	Loyer à payer	9(6)	
2		224	Oppositions	9(6)	
2		230	Avances	9(6)	
2		236	Retenues diverses	9(6)	
2		242	Diminution de salaire (montant)	9(6)	
2		248	Rappels en heures		6
3			Nombre d'heures normales	9(4)	
3			Nombre d'heures à 25 %	9(4)	
3			Nombre d'heures à 50 %	9(4)	
3			Nombre d'heures à 100 %	9(4)	
3			Taux de l'heure pour ce rappel	9(4)	
2		368	Nombre de Rappels en heures	9	

Fichier : Paramètres					Code : F5	
Organisation : Séquentielle						
Longueur enregistrement : 80						
Enr	Niv	n° caract.	Désignation	image	Øcc ≠ 1	
I	2	1	Libellé du paramètre	X(49)		
	2	50	Code du type de paramètres (= 1)	9		
	2	51	Code du paramètre	9		
	2	52	Valeur du paramètre	9(6)		
	2	58	Libre	X(23)		
II	2	1	Libellé du paramètre	X(49)		
	2	50	Code du type de paramètre (= 2 ou 3)	9		
	2	51	Code du paramètre	9		
	2	52	Valeur du paramètre	9(4)		
	2	56	Libre	X(25)		

Fichier des paramètres

Plafond S. S.	11	183000	Plafonds
Plafond Retraite	12	183000	
Plafond Cadres	13	568000	
S. S. sur base totale	21	0100	Taux des cotisations salariales
S. S. maladie	22	0250	
S. S. vieillesse	23	0300	
Retraite complémentaire ouvrier	24	0168	
Retraite complémentaire cadres	25	0400	
Retraite petits cadres	26	0400	
Retraite cadres supérieurs	27	0600	
Chômage	28	0008	
Abattement pour les salariés	29	1000	
Prévoyance ouvrier	2A	0040	
Prévoyance Etam	2B	0060	
Retraite complémentaire Etam	2C	0210	
S. S. sur base totale	31	0300	
S. S. maladie	32	1295	
S. S. vieillesse	33	0875	
Allocations familiales	34	1050	
Accidents du travail	35	0547	
Chômage	36	0040	
Congés payés	37	1580	
Retraite complémentaire non cadres	38	0420	
Décès non cadres	39	0100	
Retraite complémentaire cadres	3A	0800	
Prévoyance cadres	3B	0150	
Décès cadres	3C	0280	
Retraite petits cadres	3D	0800	
Retraite cadres supérieurs	3E	1600	
Prévoyance ouvrier	3F	0175	
Prévoyance Etam	3G	0130	
Comité de prévention	3H	0010	
Désignation	Codes	Valeurs	

Fichier : Paie calculée				Code : F6	
Organisation : séquentielle					
Longueur enregistrement : 323					
Enr	Niv	n° caract	Désignation	image	Øcc # 1
I	2	1	Matricule de l'agent	9(4)	
	2	5	Nom	X(15)	
	2	20	Prénom	X(10)	
	2	30	Plafonnement S. S. (oui:1, non:0)	X	
	2	31	Plafonnement Cadre(oui:1, non:0)	X	
	2	32	Qualification	X(3)	
	2	35	Numéro de S. S.	X(13)	
	2	48	Date de naissance (mois, jour, année)	9(6)	
	2	54	Date d'entrée (mois, jour, année)	9(6)	
	2	60	Employé E, salarié S	X	
	2	61	Horaire H, mensuel M	X	
	2	62	Non cadre N, petit cadre P, cadre supérieur S	X	
	2	63	Apprenti A, non apprenti T	X	
	2	64	Comité de prévention (inscrit P, non N)	X	
	2	65	Chômage intempéries (inscrit C, non N)	X	
	2	66	Code de paiement (cf table des codes)	X	
	2	67	Taux horaire (centimes)	9(4)	
	2	71	Salaire mensuel	9(6)	
	2	77	N° de compte en banque ou CCP	X(12)	
	2	89	Ouvrier Ø, Etam E, non N	X	
	2	90	Nbre d'heures normales (non mensuralisés)	9(5)	
	2	95	Salaire de base	9(6)	
	2	101	Nombre d'heures ôtées	9(5)	
	2	106	Montant	9(6)	
	2	112	Nbre d'heures de route	9(5)	

.../...

Enr	Niv	n° caract	Désignation	image	Øcc # 1
2		117	Montant	9(6)	
2		123	Nbre d'heures à 25 %	9(5)	
2		128	Montant	9(6)	
2		134	Nbre d'heures à 50 %	9(6)	
2		139	Montant	9(6)	
2		145	Nbre d'heures à 100 %	9(5)	
2		150	Montant	9(6)	
2		156	Prime de panier	9(6)	
2		162	Primes diverses	9(6)	
2		168	Rappels	9(6)	
2		174	Diminution de salaire	9(6)	
2		180	Salaire brut	9(6)	
2		186	Base S. S. totale	9(6)	
2		192	Cotisation maladie sur base totale	9(5)	
2		197	Base S. S. plafonnée	9(6)	
2		203	Maladie	9(5)	
2		208	Vieillesse	9(5)	
2		213	Base retraite complémentaire	9(6)	
2		219	Cotisation	9(5)	
2		224	Base cadre ou base prévoyance	9(6)	
2		230	Cotisation cadre ou cotisation prévoyance	9(5)	
2		235	Base chômage	9(6)	
2		241	Cotisation chômage	9(5)	
2		246	Salaire net	9(6)	
2		252	Indemnités	9(6)	
2		258	Salaire total	9(6)	
2		264	Loyer	9(6)	
2		270	Oppositions	9(6)	
2		276	Avances	9(6)	
2		282	Retenues diverses	9(6)	
2		288	Somme payée	9(6)	

.../...

Enr	Niv	n° caract	Désignation	image	Øcc #
	2		Heures de rappels (nombres)		
	3	294	Heures normales	9(5)	
	3	299	Heures à 25 %	9(5)	
	3	304	Heures à 50 %	9(5)	
	3	309	Heures à 100 %	9(5)	
	2	314	Date de la paie	X(10)	
II	2		Matricule (= 9999)	9(4)	
	2		Cumuls des bases de cotisations salariales	9(8)	13
	2		Cumuls des cotisations salariales	9(8)	13
	2		Date de paie	X(10)	

Fichier : Salaires bruts employés/salariés Organisation : Séquentielle Longueur enregistrement : 38				Code : F7 F8	
Enr	Niv	N° caract	Désignation	image	Øcc ≠ 1
I	2	1	Matricule de l'agent	9(4)	
	2	5	Nom	X(15)	
	2	20	Prénom	X(10)	
	2	30	Qualification	X(3)	
	2	33	Salaire brut (centimes)	9(6)	

Fichier : Table des modes de paiement Organisation : Séquentielle Longueur enregistrement : 80				Code F9	
Enr	Niv	N° caract.	Désignation	image	Øcc ≠ 1
I	2	1	Code de paiement	X	
	2	2	Libellé du mode de paiement	X(40)	
	2	42	Libre	X(39)	

ANNEXE 3

Répertoire des documents de sortie :

Code	Désignation	Edition
S1	Bulletin de salaire	P11
S2	Récapitulatifs des retenues sociales	P11
S3	Journal des salaires	P8
S4	Liste des salaires bruts employés/salariés	P10
S5	Etat bancaire	P9

SOCIETE [REDACTED]

NANCY 54

ORCSXAF NANCY
INSEE [REDACTED]

BULLETIN DE SALAIRE MOIS DE AVRIL

0373 JEAN-MARIE P3

ELEMENTS DE SALAIRE	TAUX	NOMBRE	GAINS	BASES	RETEN
SALAIRE DE BASE	6.90		1196.00		
HEURES OTEES	6.90				
HEURES DE ROUTE	6.90	6.50	44.85		
HEURES A 25 %	8.63	30.00	258.90		
HEURES A 50 %	10.35				
HEURES A 100 %	13.80				
PRIMES DE PANIER			154.00		
AUTRES PRIMES			23.46		
RAPPELS					
DIMINUTION DE SALAIRE					
SALAIRE BRUT			1677.21		
S.S. SUR BASE TOTALE	0.0100		1509.49	15.09	
S.S. MALADIE	0.0250		1509.49	37.74	
S.S. VIEILLESSE	0.0300		1509.49	45.28	
RETRAITE COMPLEMENTAIRE	0.0168		1677.21	28.18	
PREVYANCE	0.0040		1677.21	6.71	
CHOMAGE	0.0008		1509.49	1.21	
TOTAL DES COTISATIONS				134.21	
SALAIRE NET			1543.00		
INDEMNITES			148.83		
SALAIRE TOTAL			1691.83		
LOYER					
OPPOSITIONS					
AVANCES					
RETENUES DIVERSES					
SOMME PAYEE			1691.83		

* RECAPITULATIFS MOIS DE AVRIL *

RECAPITULATIF S.S. MOIS DE AVRIL

	BASE	BASE	PART	PART	COT.
	TOTALE	PLAFON.	SALARIES	PATRON.	TOTALE
MALADIE SUR BASE TOTALE	8576367		85764	171527	257291
MALADIE		7070617	176770	738875	915645
VIEILLESSE		7070617	212118	406561	618679
ALLOCATIONS FAMILIALES		7070617		742415	742415
ACCIDENTS DU TRAVAIL		7070617		386763	386763

SALARIES RECAPITULATIF S.S. MOIS DE AVRIL

	BASE	BASE	PART	PART	COT.
	TOTALE	PLAFON.	SALARIES	PATRON.	TOTALE
MALADIE SUR BASE TOTALE	4805866		48058	96118	144176
MALADIE		4701616	117543	491316	608259
VIEILLESSE		4701616	141047	270344	411391
ALLOCATIONS FAMILIALES		4701616		493670	493670
ACCIDENTS DU TRAVAIL		4701616		257175	257175

EMPLOYES RECAPITULATIF S.S. MOIS DE AVRIL

	BASE	BASE	PART	PART	COT.
	TOTALE	PLAFON.	SALARIES	PATRON.	TOTALE
MALADIE SUR BASE TOTALE	3770501		37706	75409	113115
MALADIE		2369001	59227	247559	306786
VIEILLESSE		2369001	71071	136217	207288
ALLOCATIONS FAMILIALES		59227		248745	248745
ACCIDENTS DU TRAVAIL		59227		129584	129584

```

*****
* RETRAITE COMPLEMENTAIRE MOIS DE AVRIL *
*****
*      * BASE * PART * PART * COT. *
*      * SALAR. * PATRON. * TOTALE *
*****
* RETRAITE OUVRIER * 4077985* 68510* 102765* 171275*
*      *      *      *      *      *
* RETRAITE ETAM * 2623624* 53099* 55099* 110192*
*      *      *      *      *      *
* DECES ETAM * 2623624* * 26236* 26236*
*      *      *      *      *      *
* PREVOYANCE ETAM * 2623624* 15741* 18366* 34107*
*      *      *      *      *      *
* PREVOYANCE OUVRIER * 4077985* 15741* 55055* 71365*
*      *      *      *      *      *
* RETRAITE CADRE * 732000* 29280* 29280* 58560*
*      *      *      *      *      *
* DECES CADRE * 939500* * 26306* 26306*
*      *      *      *      *      *
* PREVOYANCE CADRE * 732000* * 10980* 10980*
*****
    
```

```

*****
* RETRAITE CADRE MOIS DE AVRIL *
*****
*      * BASE * PART * PART * COTIS. *
*      * SALAR. * PATRON. * TOTALE *
*****
* RETRAITE PETITS CADRES * 371500* 14860* 14860* 29720*
*      *      *      *      *      *
* RETRAITE CADRES SUPERIEURS* 568000* 34080* 56800* 90880*
*****
    
```

```

*****
* ASSEDIC MOIS DE AVRIL *
*****
*      * BASE * PART * PART * COTIS. *
*      * SALAR. * PATRON. * TOTALE *
*****
* CHOMAGE EMPLOY ET SALARIES* 8080948* 6469* 171275* 32324*
*      *      *      *      *      *
* CHOMAGE SALARIES * 4510447* 3612* 14430* 18042*
*      *      *      *      *      *
* CHOMAGE EMPLOYES * 3570501* 2857* 11425* 14282*
*****
    
```

```

*****
* CONGES PAYES MOIS DE AVRIL *
*****
*      * BASE * PART * PART * COTIS. *
*      * SALAR. * PATRON. * TOTALE *
*****
* CONGES PAYES * 8927352* * * 1410522* 1410522*
*      *      *      *      *      *
* COMITE DE PREVENTION * 1491415* * 1491* 1491*
*****
* CONGES PAYES SALARIES * 5156851* * 814782* 814782*
*      *      *      *      *      *
* COMITE DE PREV. SALARIES * 1217915* * 1218* 1218*
*      *      *      *      *      *
* CONGES PAYES EMPLOYES * 3770501* * 595739* 595739*
*      *      *      *      *      *
* COMITE DE PREV. EMPLOYES * 273500* * 274* 274*
*****
    
```

```

*****
*
* BASE CHOMAGE=INTEMPERIES * 154970*
*
*****
    
```

JOURNAL DES SALAIRES

NBM	HEUR	SAL.	BASE	BASE	BASE	BASE	S.S.	S.S.	RET.
	BRUT	S.S.	C.P.	COMPL	CADRE	TOT.	PLAF.	COMPL	
	19000	143400	129060	143400	143400		1291	7099	2409
		200000	200000	200000			2000		
		173553	173553	173553	173553		1736	9546	3645

MOIS DE AVRIL

PREV.	RET.	CHGM.	TOTAL	SAL.	INDEMN.	RET.	SOMME
CADRE	RETEN.	NET			DIVERS	PAYEE	
574	103	11476	131924				131924
		2000	198000				198000
1041	139	16107	157446	12447	104891		65002

CUMULS RECAPITULATIFS MOIS DE AVRIL

LIBELLES	SOMMES	BASES	RETENUES
SALAIRES BRUTS	9110352		
BASES SECURITE SOCIALE		8576367	
BASES CONGES PAYES		8927352	
BASES RETRAITE COMPLEMENTAIRE		7433609	
BASES RETRAITE CADRES		939500	
RETENUES S.S. SUR BASE TOTALE			85764
RETENUES S.S. SUR BASE PLAFONNEE			388888
RETENUES RETRAITE COMPLEMENTAIRE			152889
RETENUES PREVOYANCE NON CADRES			32051
RETENUES RETRAITE CADRES			48940
RETENUES CHOMAGE			6469
TOTAL DES RETENUES SOCIALES			715001
SALAIRES NETS	8395351		
INDEMNITES NON IMPOSABLES	138073		
RETENUES NON IMPOSABLES			1070917
SOMMES PAYEES	7462507		

Liste des salaires bruts

RECAPITULATIF SALAIRES

MATRICI	NBM	PRENOM	QUAL.	SAL. BRUT
0104		JEAN	HQ	143400
0145		HARALD	CAD	260500
0880		YANNICK	MD	108800
CUMUL DES SALAIRES BRUTS :				5339851

RECAPITULATIF EMPLOYES

MATRICI	NBM	PRENOM	QUAL.	SAL. BRUT
0135		PIERRE		200000
0136		GILBERT		173553
0890		FRANÇOISE	ETA	150000
CUMUL DES SALAIRES BRUTS :				3770501

Répertoire des programmes utilisés

```

*****
* CREDIT LYONNAIS NCV APR 30 72 *
*****
*MATRI NOM | PRENOM IND DE COMPTE | SOMMES *
*****
*04831| | | | |
|ROGER 13500 004 J | 111750.69*
*0743| | | | |
|ANDRE 146 00 028 W | 766.22*
*****
* SOMME TOTALE A PAYER : 2516.91*
*****

```

```

*****
* BANQUE NATIONALE DE PARIS NCV APR 30 72 *
*****
*MATRI NOM | PRENOM IND DE COMPTE | SOMMES *
*****
*0373| | | | |
|JEAN-MARIE| 31 592 | 111691.83*
*0842| | | | |
|FRANCOIS 1086-593 | 111147.19*
*****
* SOMME TOTALE A PAYER : 2839.02*
*****

```

```

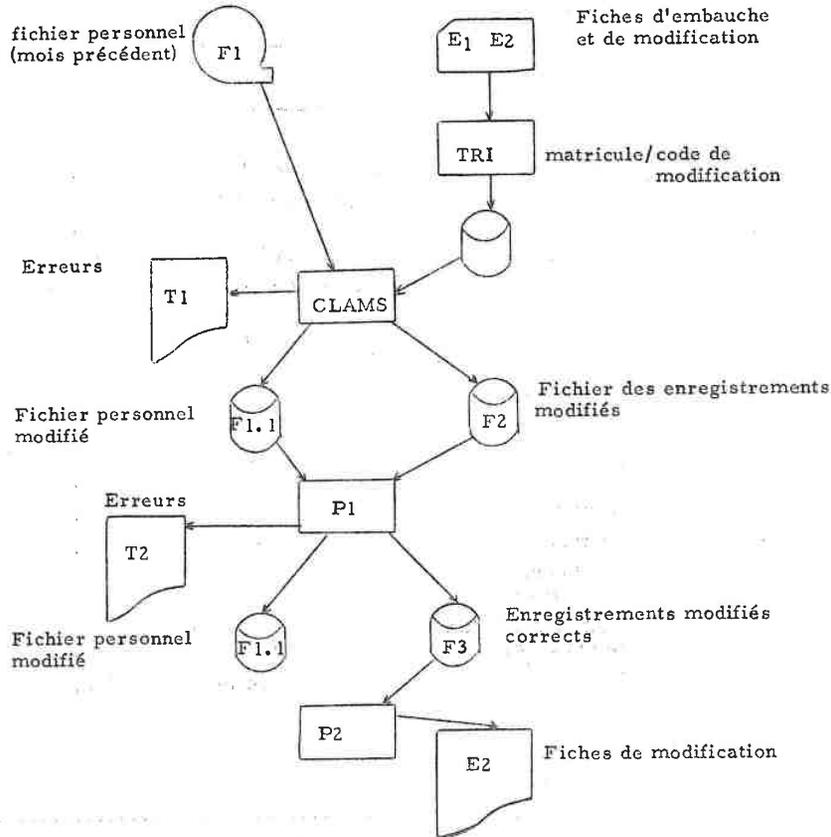
*****
* CUMUL DES SOMMES PAYEES PAR BANQUE OU CCP 55535.01*
*****
* CUMUL DES SOMMES PAYEES EN LIQUIDE 19090.06*
*****

```

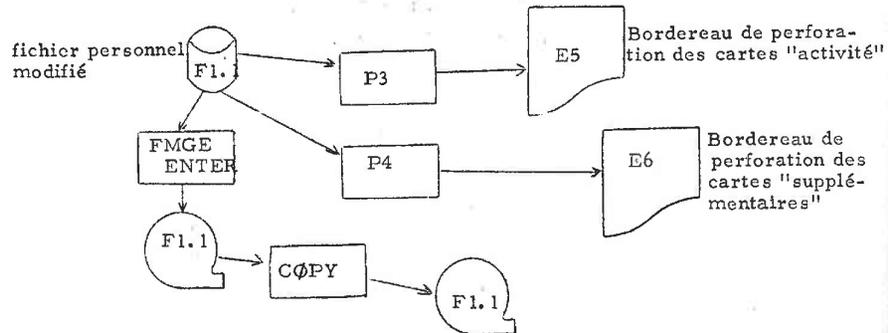
Code	Objet du programme	Modules
CLAMS *	Programme général de mise à jour de fichier	2.2
P1	Vérification des enregistrements modifiés du fichier du personnel	2.3
P2	Edition des fiches de modification (E2)	2.4
P3	Edition du bordereau de perforation (E5) des cartes 'activité'	2.6
P4	Edition du bordereau de perforation (E6) des cartes 'supplémentaires'	2.7
P5	Vérification et prise en compte des éléments variables de la paie	5.2 à 5.7
P6	Vérification du fichier des paramètres	3.2
P7	Calcul de la paie	6.1 à 6.10
P8	Edition du journal des salaires	7.4
P9	Edition de l'état bancaire des sommes versées	7.5
P10	Edition de la liste des salaires bruts par employés et salariés	7.6
P11	Edition des bulletins de salaire et des récapitulatifs des retenues sociales	7.2 à 7.4
P12	Mise à jour du fichier de la paie calculée	8.2 et 8.3
*	programme écrit et mis au point à l'IUCA de NANCY - est actuellement opérationnel pour 10070 C. I. L. sous système BPM.	

Train n° 1

module 2



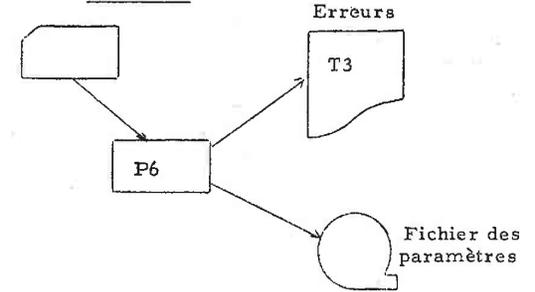
Lorsqu'aucune erreur n'est détectée



Train n° 2

Fichier des paramètres modifié sur cartes

module 3

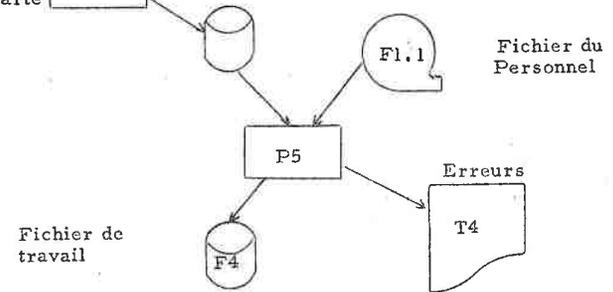


Train n° 3

Eléments variables E5 E6 E7

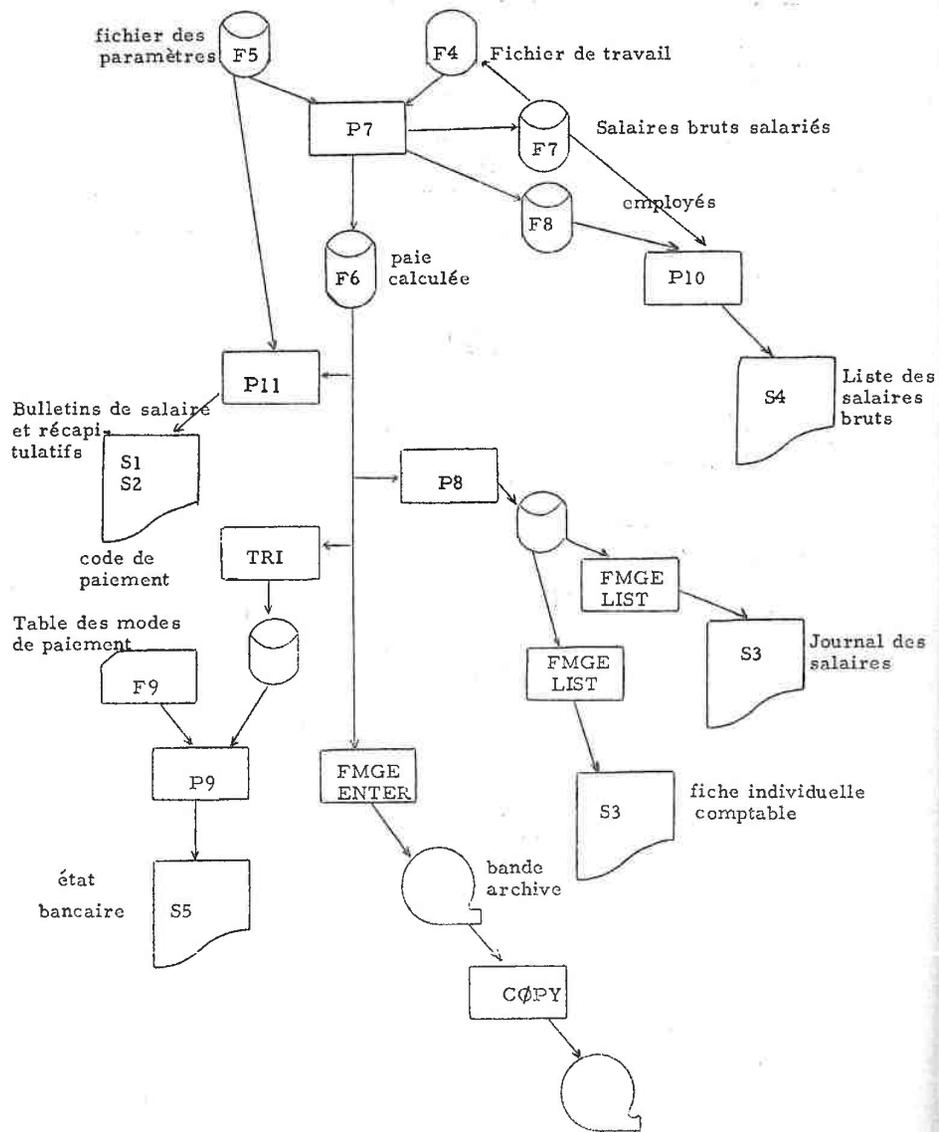
matricule/code carte TRI

module 5



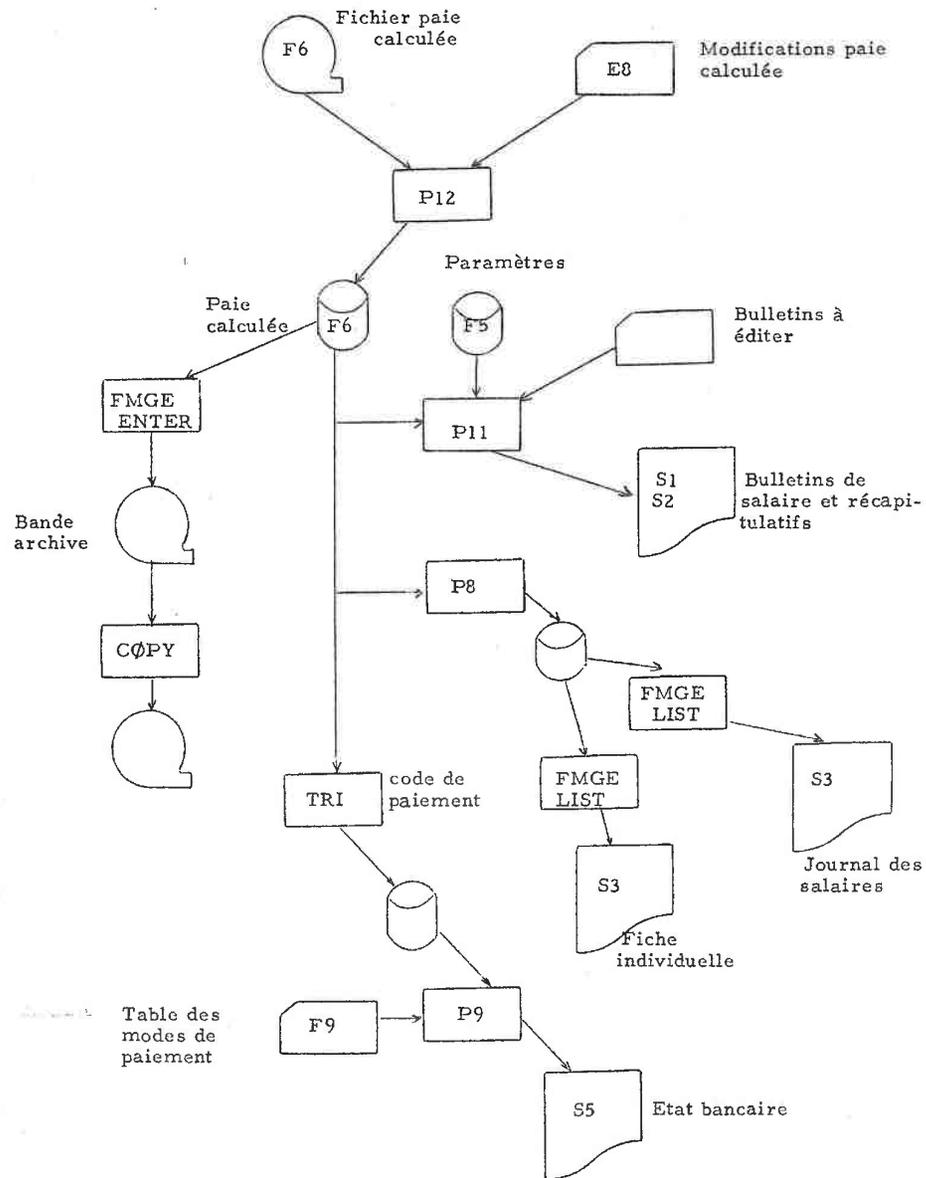
Train n° 4

modules 6 et 7



Train n° 5

module 8



BIBLIOGRAPHIE

Analyse :

- BAUVIN L'informatique de gestion
Hommes et Techniques
- REIX L'analyse en informatique de gestion (2 tomes)
Dunod 1971
- MALLET La méthode informatique
Hermann
- WARNIER et FLANAGAN Entraînement à la programmation ;
constructions de programmes (2 tomes)
Ed. d'Organisation
- CHENIQUE Analyse fonctionnelle et organique
Dunod Economie
- Les outils de l'analyse en informatique de gestion
Séminaires de Grenoble 18 et 19 Octobre 1971

Méthodes d'analyse et de programmation

- CØRIG C. G. I.
- ARIANE Gamma groupe Bossard
- MINØS CEGØS informatique
- CANTØR I. C. L.

Packages

- | | | |
|-------------|--------|------------|
| <u>Paie</u> | PAPA | I. S. S. |
| | TØPAZ | SØGECIM |
| | GAIN | INFØRAMA |
| | CRESUS | INFØRMATIS |

Editions

- SCRIPT Langage d'édition paramétré
Méthodes et techniques de l'informatique
- AGAPE Générateur de programmes d'édition
C. G. I.
- PIST Edition de tableaux
SERTI



NOM DE L'ETUDIANT : ORY Daniel

Nature de la thèse : Doctorat de Spécialité en Mathématiques Appliquées

Vu, Approuvé

et permis d'imprimer

NANCY, le 29.09.1972

Le Président du Conseil de l'Université de NANCY I



J.R. HELLUY