



## THESIS / THÈSE

### MASTER EN SCIENCES INFORMATIQUES

#### Conception d'une banque de données pour le service du corporate planning d'une société aérienne

Van Doorne, Jean-Louis

*Award date:*  
1984

*Awarding institution:*  
Universite de Namur

[Link to publication](#)

#### General rights

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

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

#### Take down policy

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



CONCEPTION D'UNE BANQUE DE  
DONNEES POUR LE SERVICE DU  
CORPORATE PLANNING D'UNE  
SOCIETE AERIENNE.

Mémoire présenté en vue de l'obtention  
du grade de Licencié et Maître en  
Informatique, sous la direction  
de Monsieur le Professeur J.L.HAINAUT

par

Jean-Louis Van Doorne

Année Académique 1983-1984

Nous voudrions remercier Monsieur le Professeur Hainaut d'avoir accepté de diriger ce mémoire. Nous tenons également à remercier les membres du personnel de la société et tout particulièrement Monsieur F. Vanderhoeven, Directeur Economique, Monsieur M. Houze, Chef de Division responsable du service, ainsi que ses collaborateurs Madame Grendel et Monsieur Van Reppelen, qui n'ont jamais refusé de se soustraire à leurs tâches pour répondre à nos questions.

## PREMIERE PARTIE

- I. Présentation de la société
- II. Situation actuelle et objectifs
  - 2.1. Situation actuelle
  - 2.2. Objectif du service
- III. Objet du mémoire
- IV. Méthode

## DEUXIEME PARTIE

- V. Etude de l'existant
  - 5.1. Définitions
    - 5.1.1. Problème
    - 5.1.2. Nouvelles définitions
  - 5.2. Grandes étapes
    - 5.2.1. Les quotas
    - 5.2.2. Constitution des axes
    - 5.2.3. Harmonisation
    - 5.2.4. Standards
    - 5.2.5. Rentabilité
    - 5.2.6. Equipages
    - 5.2.7. Budget intégré
    - 5.2.8. Périodicité
  - 5.3. Détail des étapes
    - 5.3.1. Les quotas
      - 5.3.1.1. Passager
      - 5.3.1.2. Commissions de vente
      - 5.3.1.3. Autres recettes

- 5.3.2. Constitution des axes
  - 5.3.2.1. Axes
  - 5.3.2.2. Parcours
  - 5.3.2.3. Contraintes d'escales
- 5.3.3. Harmonisation
- 5.3.4. Standards
  - 5.3.4.1. Coûts extérieures
  - 5.3.4.2. Centres producteurs
  - 5.3.4.3. Index B
- 5.3.5. Rentabilité
- 5.3.6. Equipages
  - 5.3.6.1. Slip Crew
  - 5.3.6.2. Calcul d'équipage
  - 5.3.6.3. Equipages année civile
  - 5.3.6.4. Coût équipage
- 5.3.7. Budget intégré
  - 5.3.7.1. Frais de vente
  - 5.3.7.2. Autres frais
- 5.4. Inventaire des informations

## VI. Nouveau Système d'information

- 6.1. Objectif
  - 6.1.1. Objectif général
  - 6.1.2. Objectifs opérationnels
  - 6.1.3. Choix

## 6.2. Processus nécessaires

### 6.2.1. Types de processus

6.2.1.1. Processus de communication

6.2.1.2. Processus de décision bien ou semi-structuré

6.2.1.3. Processus de décision mal-structuré

6.2.1.4. Production de rapports

## 6.3. Système proposé

6.3.1. Architecture

6.3.2. Processus de communication

6.3.3. Processus de décision bien ou semi-structuré

6.3.4. Processus de décision mal structuré

6.3.5. Production de rapports

## 6.4. Avantages et inconvénients

### 6.4.1. Avantages

6.4.1.1. Gain de temps pour l'introduction des données

6.4.1.2. Gain de temps pour les calculs

6.4.1.3. Information plus pertinente

6.4.1.4. Extensions possibles

### 6.4.2. Inconvénient

## 6.5. Conditions de réalisation du système

6.5.1. Sensibilisation

6.5.2. Les problèmes de compétences

6.5.3. Motivation des utilisateurs

## 6.6. Environnement logiciel

### 6.6.1. Traitements

6.6.1.1. Tâches

6.6.1.2. utilisation

6.6.2. Données

6.6.3. Hiérarchie

6.7. Conclusion

TROISIEME PARTIE - DEVELOPPEMENT DE LA BANQUE DE DONNEES

VII Schéma conceptuel

7.1. Introduction

7.2. Sous-schéma produits et moyens

7.3. Sous-schéma quota

7.4. Sous-schéma facteurs de production

7.5. Sous-schéma coûts unitaires

7.6. Sous-schéma budget

7.7. Sous-schéma équipages

7.8. Quantifications

VIII Schéma des accès possibles

8.1. Sous-schéma produits et moyens

8.2. Sous-schéma quotas et facteurs de production

8.3. Sous-schéma coûts unitaires

8.4. Sous-schéma budget

8.5. Sous-schéma équipages

8.6. Schéma des items

IX Schéma des accès nécessaires

9.1. Introduction

9.2. Constitution des axes

- 9.3. Quotas
- 9.4. Harmonisation
- 9.5. Standards
- 9.6. Rentabilité
- 9.7. Equipages
- 9.8. Métaalgorithmes
- 9.9. Sous-schémas
  - 9.9.1. Sous-schéma produits et moyens
  - 9.9.2. Sous-schéma quotas et facteurs de production
  - 9.9.3. Sous-schéma coûts unitaires
  - 9.9.4. Sous-schéma budget
  - 9.9.5. Sous-schéma équipages

#### X. Schéma IMAGE/H.P.

- 10.1. Description de IMAGE
- 10.2. Sous-schémas IMAGE
  - 10.2.1. Produits et moyens
  - 10.2.2. Quota et facteurs de productions
  - 10.2.3. Coûts unitaires
  - 10.2.4. Budget
  - 10.2.5. Equipages
- 10.3. Algorithmes ADL/IMAGE
  - 10.3.1. Constitution des axes
  - 10.3.2. Quotas
  - 10.3.3. Harmonisation
  - 10.3.4. Standards
  - 10.3.5. Rentabilité
  - 10.3.6. Equipages
  - 10.3.7. Métaalgorithmes

10.4. Transformation du schéma de base

10.5. Schéma interne

10.5.1. Data base

10.5.2. Items

10.5.3. Algorithmes PSEUDO-FORTRAN/IMAGE

10.5.3.1. Introduction

10.5.3.2. Constitution des axes

10.5.3.3. Quotas

10.5.3.4. Harmonisation

10.5.3.5. Standards

10.5.3.6. Rentabilité

10.5.3.7. Equipages

10.5.3.8. Métaalgorithmes

10.5.3.9. Modules d'accès

10.5.4. Place mémoire

XI Conclusion

PREMIERE PARTIE

INTRODUCTION

## I - PRESENTATION DE LA SOCIETE.

La société où nous nous sommes rendu a comme activité principale la réalisation de transport aérien. A cette fin, elle dispose d'une vingtaine d'avions desservant 120 escales réparties dans quatre continents.

La société est divisée en plusieurs directions:

- la Direction technique chargée de l'entretien des avions
- la Direction des Opérations chargée de l'exécution des vols
- la Direction Commerciale
- la Direction Administrative
- la Direction financière
- la Direction Economique.

La Direction Economique est divisée en quatre départements:

- Département Informatique
- Département des Investissements
- Département des Statistiques
- Département des Carburants
- Département du Budget.

Le département du budget est divisé en deux services:

- Service du Corporate Planning
- Service du Budget Court-terme.

Le service du Corporate Planning, désormais désigné par "le service", a pour fonction d'élaborer les budgets à long terme. et moyen terme.

Ces prévisions se font sur base de trois types d'informations:

- des prévisions de ventes (part de marché, prix)
- des prévisions de transport (vols, avions)
- des prévisions de coûts (exploitation, administration, et coûts financiers).

Le service est dirigé par un responsable , secondé par six collaborateurs et une secrétaire. De plus, dans chacune des autres directions de la société, le département du budget dispose de correspondants budget, désormais appelés "correspondants".

## II - SITUATION ACTUELLE ET OBJECTIFS

### 2.1. Situation actuelle.

Les prévisions faites par le service sont établies par différentes procédures qui sont soit automatisées, semi-automatisées ou encore parfois manuelles.

Nonante pour cent de l'information traitée provient d'autres départements, et est communiquée manuellement par courrier interne, ensuite cette information est répartie dans plusieurs fichiers soit automatisés, soit également manuels. Afin de mener à bien ces tâches automatisées, le service dispose d'un HP.1000.

## 2.2. Objectif du service.

L'objectif que s'est fixé le service est d'automatiser l'élaboration des prévisions et de pouvoir par la suite effectuer certaines simulations.

Afin de réaliser cet objectif, ce service désirerait d'abord que l'information de base dont il dispose réponde plus adéquatement à ses besoins, et ce tant sur le contenu de l'information que sur la manière de l'obtenir. De plus, le service souhaiterait également réduire les redondances existant dans les fichiers actuels.

## III - OBJET DU MEMOIRE.

L'objet du présent mémoire est d'étudier et de proposer un système d'information répondant aux objectifs du service. Afin d'atteindre cet objectif, nous allons établir un système de fichiers qui devra être au moins plus performant que le système actuel et ce tant sur le plan d'espace mémoire nécessaire que des temps d'accès.

Comme contraintes nous retiendrons:

- la possibilité d'extension à de nouvelles applications
- la comptabilité avec les systèmes d'information des autres services de la société et principalement l'informatique centrale.

#### IV - METHODE

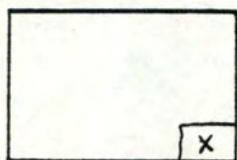
Nous nous proposons dans la deuxième partie de premièrement décrire le système d'information actuel, ensuite en fonction des objectifs et contraintes, nous proposerons un nouveau système.

Dans la troisième partie, nous développerons la banque de données du système proposé. A cette fin, nous établirons le schéma conceptuel à l'aide du modèle Entité Association [1], ensuite nous suivrons la méthodologie proposée par le Professeur J.L.Hainaut dans "Cadre de référence pour la conception de base de données" [3], c'est-à-dire qu'à partir du schéma conceptuel, nous établirons le schéma des accès possibles. Puis, à l'aide des algorithmes prédicatifs et effectifs, nous développerons le schéma des accès nécessaires. Ensuite nous traduirons ce schéma en schéma IMAGE/HP disponible dans le service.

Le service ne disposant que de peu de documentation, c'est exclusivement au cours de nombreuses entrevues avec les membres du service et d'autres départements que nous avons récolté les informations, que nous avons pu établir le diagramme des flux de l'existant et que nous avons validé le schéma conceptuel.

DEUXIEME PARTIE

ANALYSE FONCTIONNELLE

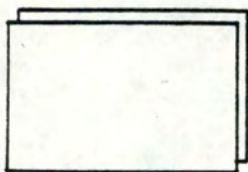
CONVENTIONS

Traitement

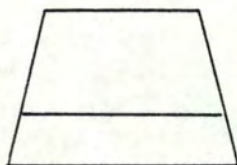
X : A = Automatique

M = Manuel

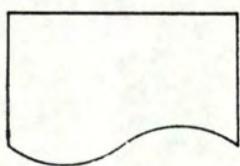
I = Interactif



Fiches tenues manuellement



Entrées de données dans l'ordinateur



Listing



Fichier informatique

Frontière entre le service  
et la Direction technique

---

TECHNIQUE

## V. ETUDE DE L'EXISTANT

### 5.1. Définitions

#### 5.1.1. Problèmes

Dès le début de nos entretiens avec le service, nous nous sommes rendu compte que les définitions des objets conceptuels manipulés n'étaient pas rigoureuses. C'est pourquoi, avant d'aborder l'étude de l'existant, nous nous proposons de choisir pour chaque concept important une définition rigoureuse.

Les définitions retenues sont celles qui d'une part sont à nos yeux correctes et qui d'autre part ont le plus de chance d'être utilisées par un maximum d'utilisateurs au sein de la société.

Nous avons pu identifier plusieurs sources d'ambiguïtés dans les définitions.

- Premièrement, le catalogue officiel des statistiques de la société ne reprend pas la définition de tous les concepts.
- Deuxièmement, les définitions reprises dans ce catalogue sont souvent incomplètes et ambiguës.
- Troisièmement, il est fréquent que d'un service à l'autre, un concept soit couvert par plusieurs dénominations différentes, et qu'une dénomination couvre plusieurs concepts totalement différents.
- Quatrièmement, les identifiants utilisés sont souvent différents d'un service à l'autre, même d'une application à l'autre dans un même service.

Par conséquent pour chaque concept nous allons choisir une dénomination et retenir une définition. Les identifiants quant à eux seront traités dans le modèle conceptuel.

La démarche suivie sera la suivante: parmi les définitions et dénominations existantes, nous en choisirons une que nous améliorerons si nécessaire. Notre préférence ira d'abord aux définitions et dénominations provenant du catalogue statistique, ensuite à celles les plus utilisées dans la société.

#### 5.1.2. Nouvelles définitions.

Vol: un vol est un déplacement d'un avion entre deux points préétablis, points qui peuvent être soit confondus, soit distincts, avec ou sans escale intermédiaire.

Vol commercial: vol effectué en exécution du programme d'exploitation ou à la suite d'une demande des services commerciaux. Il doit être générateur de recettes.

Vol non commercial: vol effectué en-dehors du programme d'exploitation et pour satisfaire des besoins non commerciaux. En aucun cas il n'est générateur de recettes.

Ligne: tout vol identifié par un numéro dont la fréquence, l'itinéraire et les heures de départ et d'arrivée sont prévues aux horaires.

Nous ajouterons qu'une ligne est un vol commercial, et que Bruxelles ne peut être l'escale de départ et d'arrivée.

P.K.T. : Passagers kilomètres transportés

S.K.O. : Sièges kilomètres offerts

T.K.T. : Tonnes kilomètres transportés

T.K.O. : Tonnes kilomètres offerts

Tronçon: partie de vol comprise entre deux escales successives.

Nous définissons:

Ligne aller: toute ligne ayant Bruxelles comme première escale de départ.

Ligne retour: toute ligne ayant Bruxelles comme dernière escale d'arrivée.

Ligne intermédiaire: toute ligne consécutive à une ligne d'aller qui n'est pas une ligne retour.

Lignes consécutives: les lignes A et B sont consécutives si la dernière escale d'arrivée de A est la première escale de départ de B.

Axe: ensemble des lignes aller, intermédiaires et retour consécutives, dont l'itinéraire et les escales des lignes aller, intermédiaires et retour sont identiques.

Réseau: ensemble d'axes desservant une vaste partie du monde correspondant généralement à un continent.

Parcours: partie de vol commercialisable constitué par un ou plusieurs tronçons d'un même axe.

Commercialisable signifie que la société a obtenu les droits de trafic pour transporter du fret ou des passagers. Ces droits de trafic accordés par les autorités aéronautiques des escales d'arrivée et de départ du parcours.

Soit l'axe A : Bruxelles - Istanbul - Bruxelles contient :

les lignes aller: Bruxelles - Istanbul

les lignes retour: Istanbul - Bruxelles

Les lignes aller contiennent:

les tronçons : Bruxelles - Athènes

Athènes - Istanbul

les parcours : Bruxelles - Athènes

Bruxelles - Istanbul

Athènes - Istanbul

Les lignes retour contiennent:

les tronçons : Istanbul - Bruxelles

le parcours : Istanbul - Bruxelles

Relation: trajet constitué par un ou plusieurs parcours indépendamment de l'axe. Une relation ne peut avoir un changement d'Axe à Bruxelles.

Saison hiver x - y: saison débutant le 1er novembre année x et se terminant le 31 mars Y.

$y = X + 1$ .

Saison été x - x: saison débutant le 1er avril x et se terminant le 31 octobre x.

La période couverte par ces saisons est uniforme pour l'ensemble des compagnies aériennes membres de l'IATA (International Air Transportation Association). C'est pourquoi on les appelle aussi saison IATA.

Mission: ensemble de bureaux de vente.

Bureau de vente: point de vente ayant le droit d'émettre un ticket de transport de la société.

Quota: objectifs de ventes d'une mission.

Service producteur: service nécessaire à l'exécution d'un vol; ce service est produit par la société.

Service externe: service nécessaire à l'exécution d'un vol, service rendu par une tierce personne à la société.

Standard: coût unitaire d'un service externe ou producteur, lié directement à l'exécution d'un vol.

Programme: ensemble des vols commerciaux prévus pour une saison IATA.

## 5.2. Grandes étapes

### 5.2.1. Les quotas

Le service recueille les objectifs de vente par saison IATA. Ces informations proviennent du service commercial, et ne se rapportent généralement qu'aux deux saisons IATA à venir.

Les quotas relatifs au long terme sont quant à eux créés par le service.

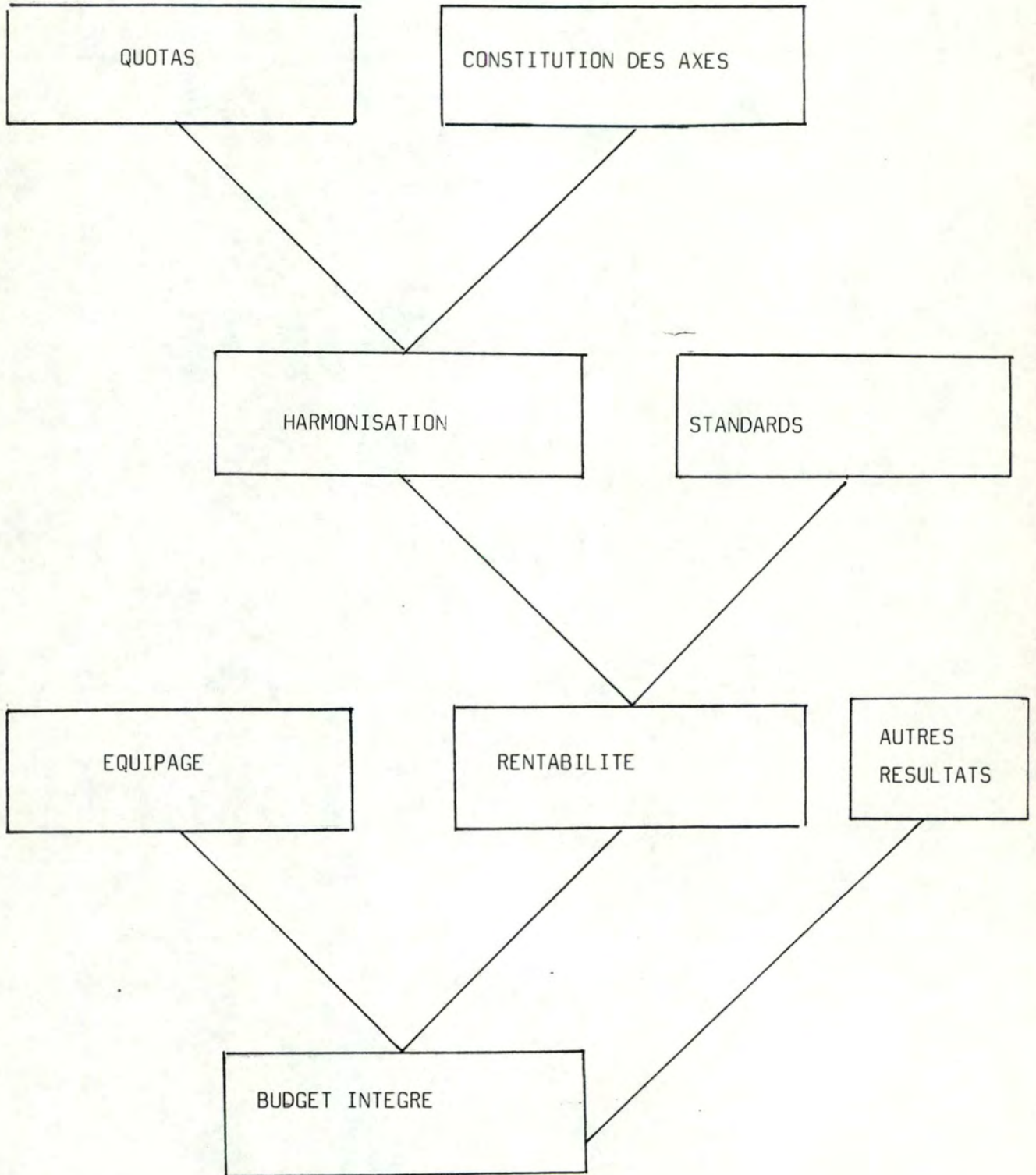
### 5.2.2. Constitution des axes

Le but de cette application va être de déterminer les produits offerts par la société pour une saison donnée. Il s'agit en fait de construire un programme reprenant tous les axes qui seront exploités. Ce programme devra couvrir le plus de relations reprises dans les quotas.

Les critères retenus pour choisir un programme sont:

- taux d'utilisation des avions
- homogénéité du type d'avion . au sein d'un réseau
- horaires "vendables"
- des connexions possibles

5.2. GRANDES ETAPES



Les contraintes sont:

- le matériel volant disponible
- les droits de survol
- les droits d'atterrissage
- la disponibilité des équipages.

En ce qui concerne le long terme, les contraintes sont de même type que celles à moyen terme. Toutefois, elles seront nettement moins restrictives.

### 5.2.3. Harmonisation

L'harmonisation a pour but de répartir le trafic par relation sur les différents parcours réalisables.

L'harmonisation tente de répartir le trafic de telle manière que le coefficient de chargement soit le plus élevé, et que le nombre de refus soit minimal.

Exemple: - Soit l'axe n°1: Bruxelles - Athènes - Dubaï -  
Bombay - Dubaï - Athènes - Bruxelles  
- Soit l'axe n°2: Bruxelles - Athènes - Bruxelles.

Il est absurde d'avoir sur le vol n°1 un coefficient de chargement de 100% entre Bruxelles - Athènes, de 5% sur le reste du vol et de refuser des passagers sur les parcours:

Bruxelles - Dubaï

Bruxelles - Bombay.

Par conséquent, l'harmonisation essaie d'accepter le plus de passagers sur les longs parcours d'un même

AXE ), quitte à refuser des passagers sur le parcours Bruxelles - Athènes de l'axe n°1, mais à déverser les refusés sur l'axe n°2. Ainsi, les coefficients de chargement de l'axe n°1 sont élevés le plus longtemps possible, et le nombre de refus est minimal.

La répartition optimale n'est pas atteinte au premier essai. Par conséquent, il y a lieu de redéfinir de nouveaux axes et parcours.

A l'issue de cette application, le programme retenu n'est que provisoire. Ce n'est qu'à l'issue des deux étapes suivantes que le programme définitif est fixé.

#### 5.2.4. Standarts

Le service va calculer l'ensemble des coûts unitaires d'exploitation, appelé standards. Les standards sont la contrepartie financière d'un service fourni. Ce service peut être soit issu d'un centre producteur, soit d'un centre externe. En ce qui concerne les standards issus d'un centre producteur, le service va éclater le montant selon 16 natures comptables. A ce stade-ci, aucune optimisation n'est effectuée s'il s'agit essentiellement de récolter et de traiter des données.

#### 5.2.5. Rentabilité

Le but du calcul de la rentabilité est de dégager par axe les différentes marges d'exploitation pour une année civile. Le calcul de la rentabilité d'un axe ne nécessite pas quant à lui un processus d'optimisation. Par contre, à l'issue du calcul, le service peut être amené à opérer certaines modifications au niveau des applications de constitution des axes et de l'harmonisation.

#### 5.2.6. Equipages

Le but de cette application est de calculer le nombre d'équipages strictement nécessaire pour une année civile. A cette

fin, le service détermine les différents types de routing des équipages par saison IATA. Lors de ce calcul, le service tâche de minimiser les temps d'inactivités des équipages tout en tenant compte des conventions collectives. Une fois cette phase d'optimisation finie, le service calcule le coût réel des équipages pour une saison.

#### 5.2.7. Budget intégré

Le service récolte les données relatives au calcul des coûts d'administration générale, des charges financières, des amortissements et des recettes et des dépenses, tant annexes qu'industrielles. Le service élabore ensuite le bilan de la société, ainsi que le bilan consolidé des filiales.

#### 5.2.8. Périodicité des traitements

1. les quotas: 2 saisies par saison.  
5 - 10 créés par saison.
2. constitution des axes: création de 4 plans par saison.  
Chaque plan nécessite une vingtaine d'essais.
3. harmonisation : 5 - 15 par saison.
4. standarts : 1 à 2 calculs complets par plan, plus une vingtaine de calculs partiels.
5. rentabilité: minimum 4 calculs complets par saison, plus une quarantaine de calculs partiels par saison.
6. équipage: 5 à 10 par saison.
7. budget intégré: 1 par an.

### 5.3. Détail des étapes.

#### 5.3.1. Quotas.

##### 5.3.1.1. Passager et fret.

a) Chaque année, le service commercial communique ses quotas de ventes.

Les quotas donnent pour chacune des relations aller-retour confondus (environ 300) et par mission (60):

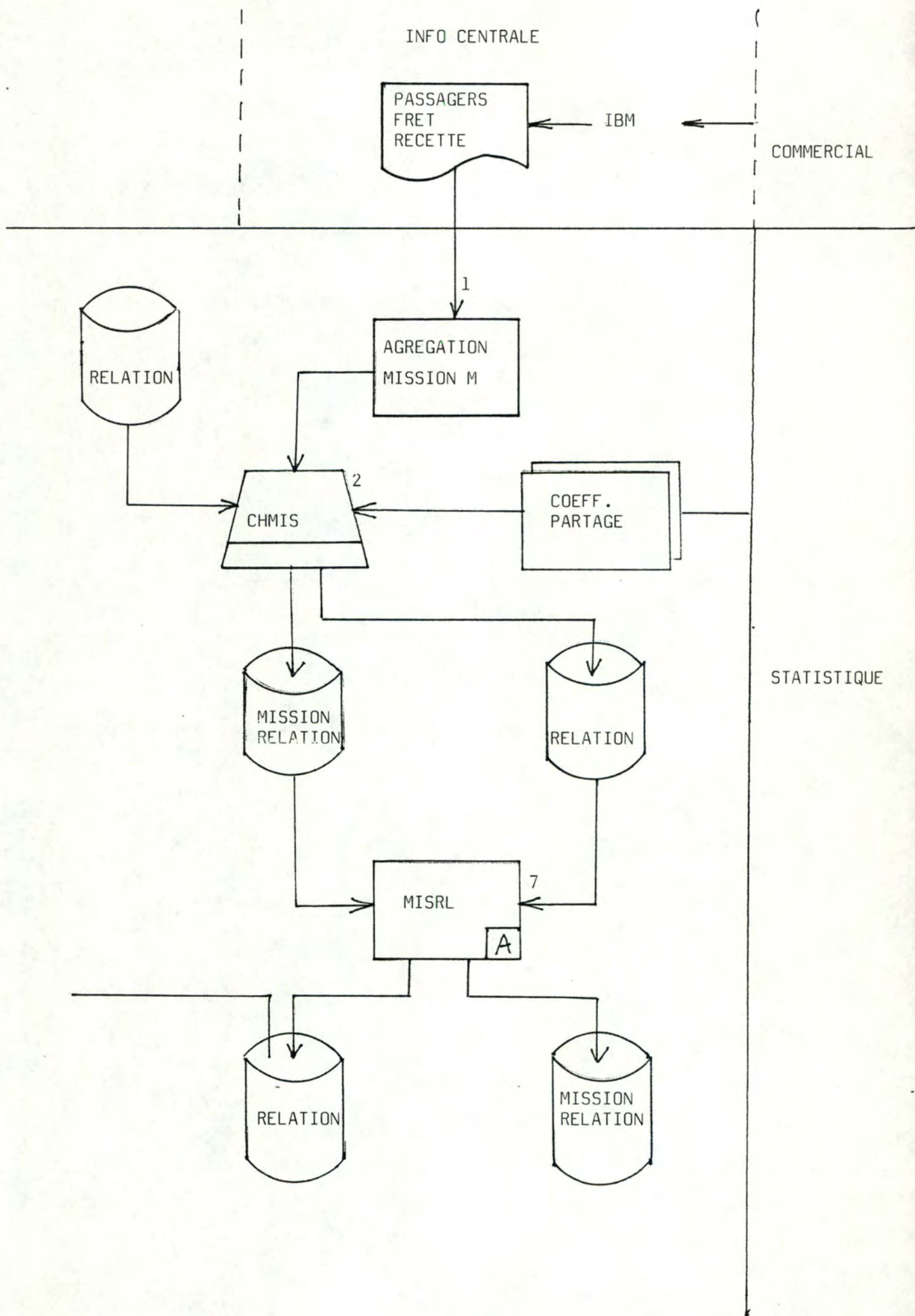
- le nombre de passagers (aller, retour),
- la recette moyenne unitaire passager (classes et types de prix confondus),
- le nombre de tonnes fret pour le sens de la relation,
- la recette moyenne unitaire fret.

b) Pour chaque relation, le service va agréger en une mission divers, les missions ayant un chiffre d'affaires passager faible pour la relation considérée. En fait, lors de ce traitement (numéro 1 dans le flow chart), le service va, pour une relation donnée, sélectionner les 15 premières missions réalisant 80% du chiffre d'affaires passagers, et regrouper les autres dans une mission divers.

Ces quotas de ventes pour chaque relation et agrégés en 16 (15 + 1) missions sont mémorisés dans le fichier mission-relation.

c) Afin de connaître le trafic passager dans le sens de la relation, le service obtient du service des statistiques les coefficients de partage enregistrés les années précé-

5.3.1.1. PASSAGER ET FRET



dentes. Ces coefficients sont chargés dans le fichier relation.

- d) A partir des quotas mémorisés dans le fichier mission-relation, et les coefficients de partage, on va calculer le trafic global pour chaque relation et la recette unitaire moyenne.

Ce traitement (n°7) automatique appelé MISRL, cumule pour chaque relation le trafic passager et fret des 16 missions, ensuite applique le coefficient de partage au trafic passager, et ne retient le trafic passager que dans le sens de la relation. Il calcule ensuite par relation la recette unitaire moyenne tant passager que fret. Les résultats de ces calculs sont mémorisés dans le fichier relation.

#### 5.3.1.2. Commissions de vente et frais spécifiques

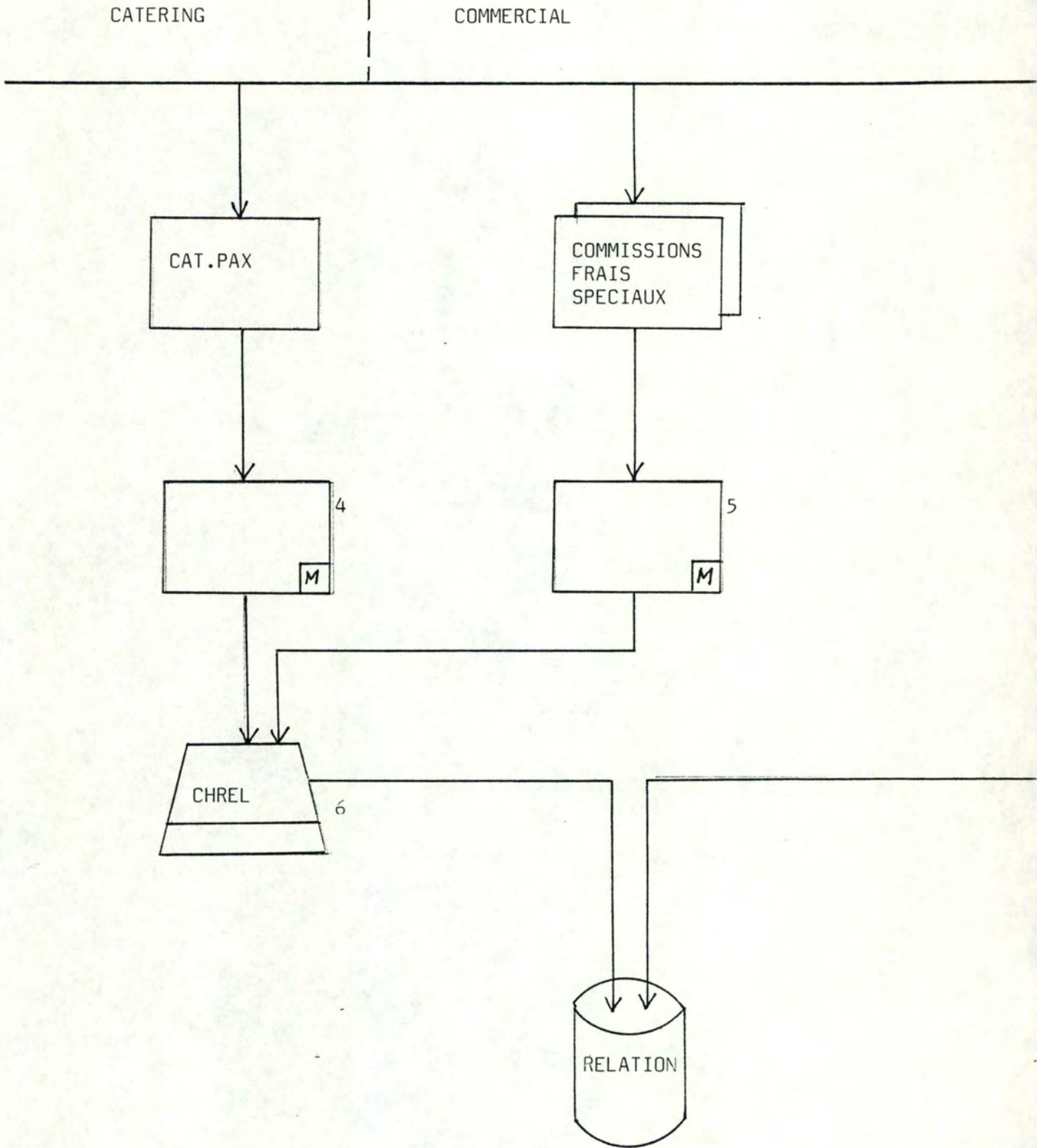
Les commissions de vente, frais ou moyens spécifiques ainsi que les avances sur factures, sont les commissions ristournées aux points de ventes. Le service commercial fournit directement le montant de ces frais par relation. Ces données sont mémorisées dans relation.

#### 5.3.1.3. Autres recettes

Parmi les autres recettes, on retient:

- les recettes du transport de la poste,
- les locations d'espace (recette),
- les correctifs de pool,
- les excédents de bagages (%).

5.3.1.2. COMMISSIONS DE VENTE ET FRAIS SPECIFIQUES



Les locations d'espace sont des contrats liant la Sabena à un autre transporteur aérien. Au terme de ce contrat, la Sabena vend un nombre spécifique de sièges ou d'espace fret spécifique pour une période déterminée. Les correctifs de pool sont quant à eux une correction de recette (soit négative, soit positive), suite aux accords d'exploitation commune d'une ligne par la société et un autre transporteur. Les locations d'espace, les pool et les recettes postes sont fournies par le service commercial.

Les excédents de bagages sont exprimés en pour cent de la recette passager. Ils proviennent du service statistiques qui communique les données précédemment réalisées. C'est le service qui les indexera éventuellement.

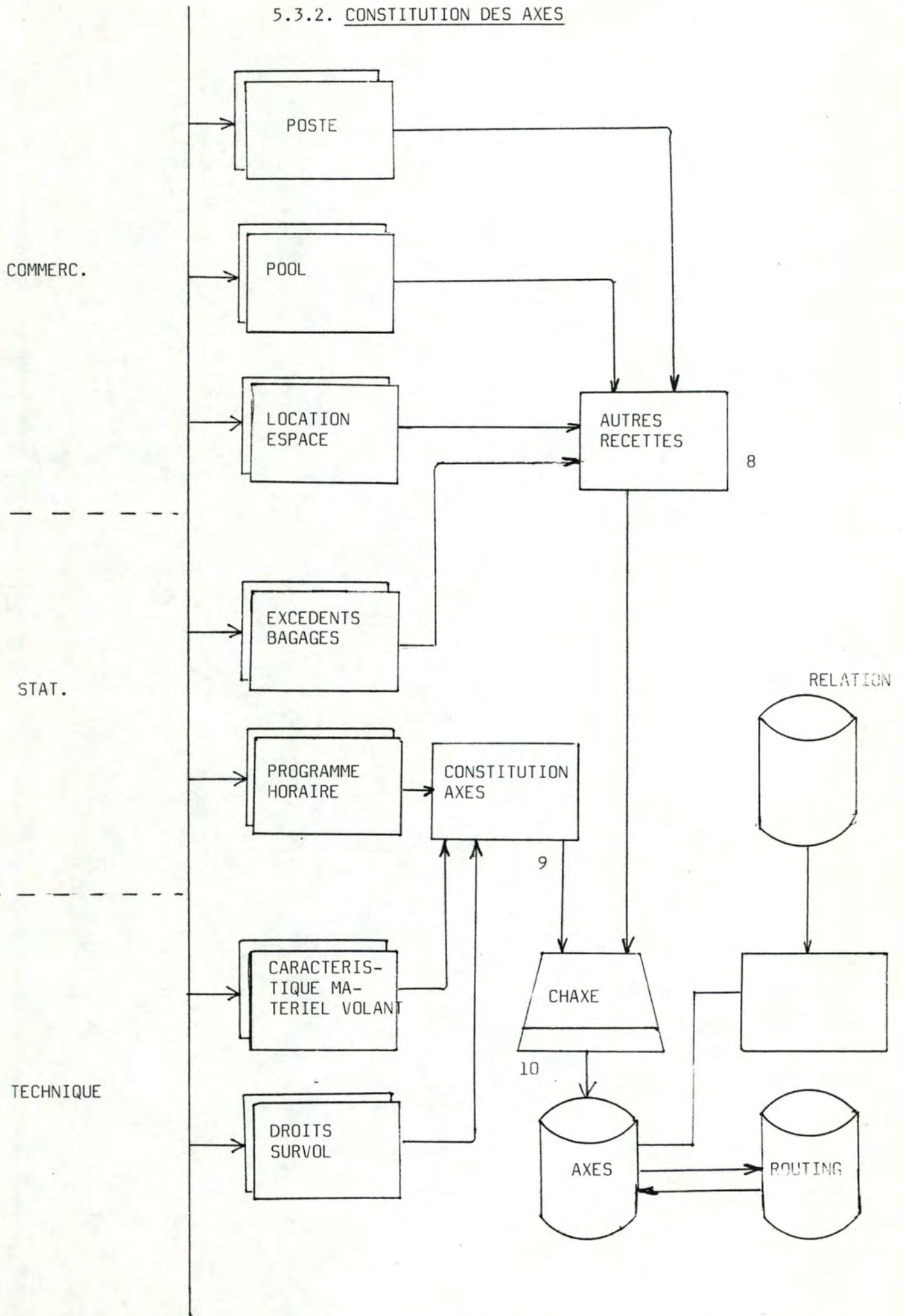
L'ensemble des "autres recettes" est mémorisé dans le fichier axe, c'est-à-dire que ces recettes sont agrégées par axe. L'information reçue par le service est généralement donnée par relation ou parcours. C'est donc le service qui agrège ces données selon les axes choisis (nous verrons ci-dessous la sélection des axes). Traitement (8).

### 5.3.2. Constitution des axes

#### 5.3.2.1. Axes

Le but de ce traitement (n°9) est de déterminer ou même parfois de définir l'itinéraire (en terme d'escales), ainsi que le type d'avion et son type d'aménagement, des différents axes.

5.3.2. CONSTITUTION DES AXES



Une fois ce traitement fini, on disposera de l'ensemble des tronçons et parcours techniquement réalisables, ainsi que de la charge disponible, du nombre de sièges disponibles.

Généralement, le service reprend le routing arrêté des années précédentes, auquel il modifie certains axes ou parfois en crée de nouveaux.

Le programme de base comprend généralement 75 axes, plus une quarantaine de variantes.

Afin de déterminer ces axes exploitables, le service dispose:

- des caractéristiques techniques du matériel volant,
- des programmes d'horaires passés, provenant du service statistiques. Ces programmes contiennent:
  - le nombre d'heures de vol,
  - la distance,
  - le nombre d'heures de mobilisation,
  - le nombre de rotations,
  - les différentes escales.
- des droits de survol.

Une fois que les axes de base et que les variantes sont arrêtés, on disposera par axe:

- du type d'avion et de son aménagement,
- du nombre d'heures de vol,
- du nombre de rotations,
- de la charge offerte totale,
- de la distance.

Ces informations sont mémorisées dans le fichier axe. L'itinéraire est quant à lui mémorisé dans le fichier routing.

#### 5.3.2.2. Parcours

Connaissant les parcours techniquement réalisables (traitement n°6) et les relations commerciales désirées, le service va établir (traitement n°11) les parcours commercialement et techniquement réalisables, appelés les parcours. Pour rappel, les relations tiennent compte des droits de trafic, c'est-à-dire de l'autorisation de transporter et/ou d'embarquer, débarquer des passagers.

Le service modifiera éventuellement les axes, en fonction de ces parcours.

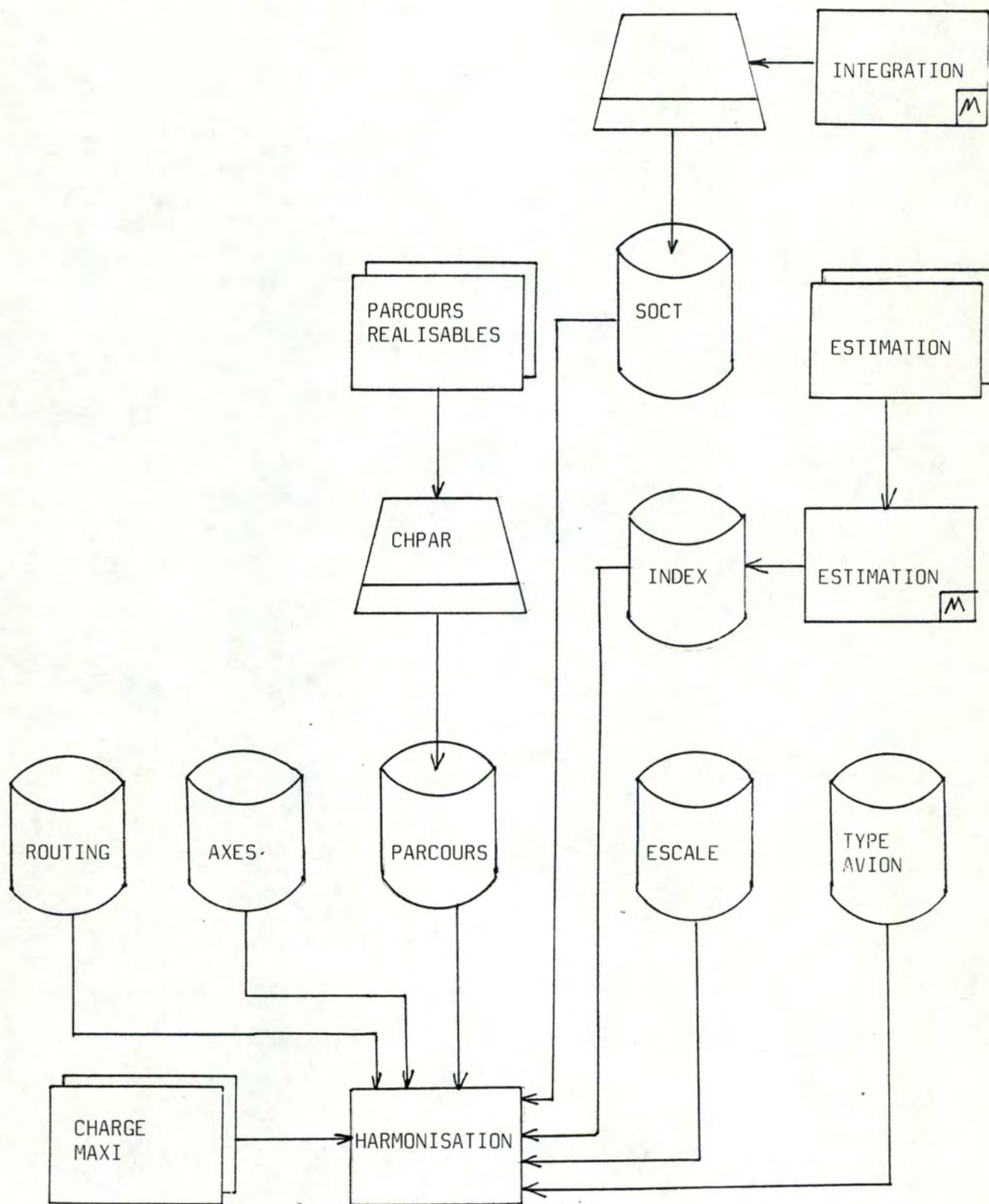
Une fois les parcours arrêtés, le fichier parcours est initialisé, c'est-à-dire qu'on y mémorise les escales de départ et les escales d'arrivée.

#### 5.3.2.3. Contraintes d'escales

Un avion ne peut décoller ou atterrir avec la même charge dans toutes les escales. En effet, la charge maximale dépend de la longueur de la piste, de l'altitude de l'escale et éventuellement d'autres facteurs.

Le service technique communique au service l'ensemble de ces contraintes par type d'avion et type d'aménagement, et ce pour tous les tronçons.

5.3.2.3. HARMONISATION



Le service agrège ces données et retient par type d'avion et type d'aménagement sur un réseau: (TRAITEMENT 12)

- le nombre de sièges offerts,
- la charge offerte totale,
- la charge fret maximale,
- la charge totale maximum.

Ces informations sont mémorisées dans le fichier SOCT.

#### Index A.

Généralement, les données liées à la recette sont des prévisions pour une ou deux saisons (IATA ) futures, parfois même ces données ne sont simplement que les réalisations des années précédentes. Afin de pouvoir élaborer des budgets sur des horizons temporels plus grands, le service établit et dispose de tables d'index permettant d'indexer les données.

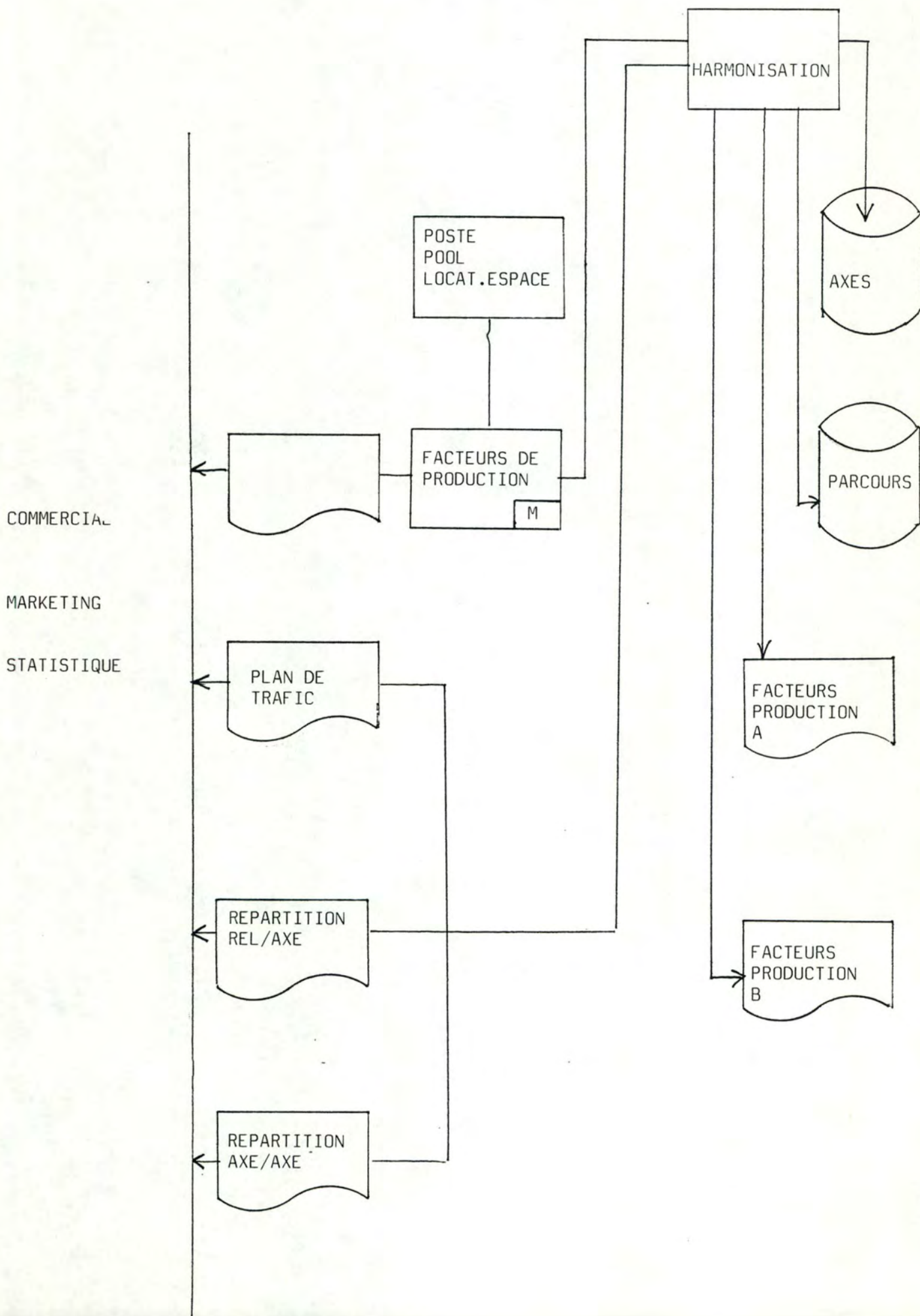
#### 5.3.3. Harmonisation

L'harmonisation va calculer:

- les facteurs de production (TRT, PKT, SKO) -disponibles
  - réalisables
- le plan de trafic
- la répartition des relations sur les axes
- la répartition des axes selon les relations
- les coefficients d'occupation
- les données relatives aux parcours commercialisés - passagers (nombre)
  - fret

Le programme d'harmonisation (traitement n° 15) répartit les trafics des relations sur les axes, dans l'ordre des recettes unitaires nettes décroissantes (recette unitaire

5.3.3. HARMONISATION



moins toutes les commissions) et en fonction de la place disponible par tronçon.

Le chargement par tronçon est déterminé interactivement à partir des chargements maxima par tronçon.

Le programme détermine par parcours:

- le nombre de passagers qui pourront être transportés,
- la recette unitaire passager,
- le fret pouvant être transporté,
- la recette unitaire fret.

par tronçon:

- le chargement passager et les refus,
- le chargement fret et les refus.

par axe:

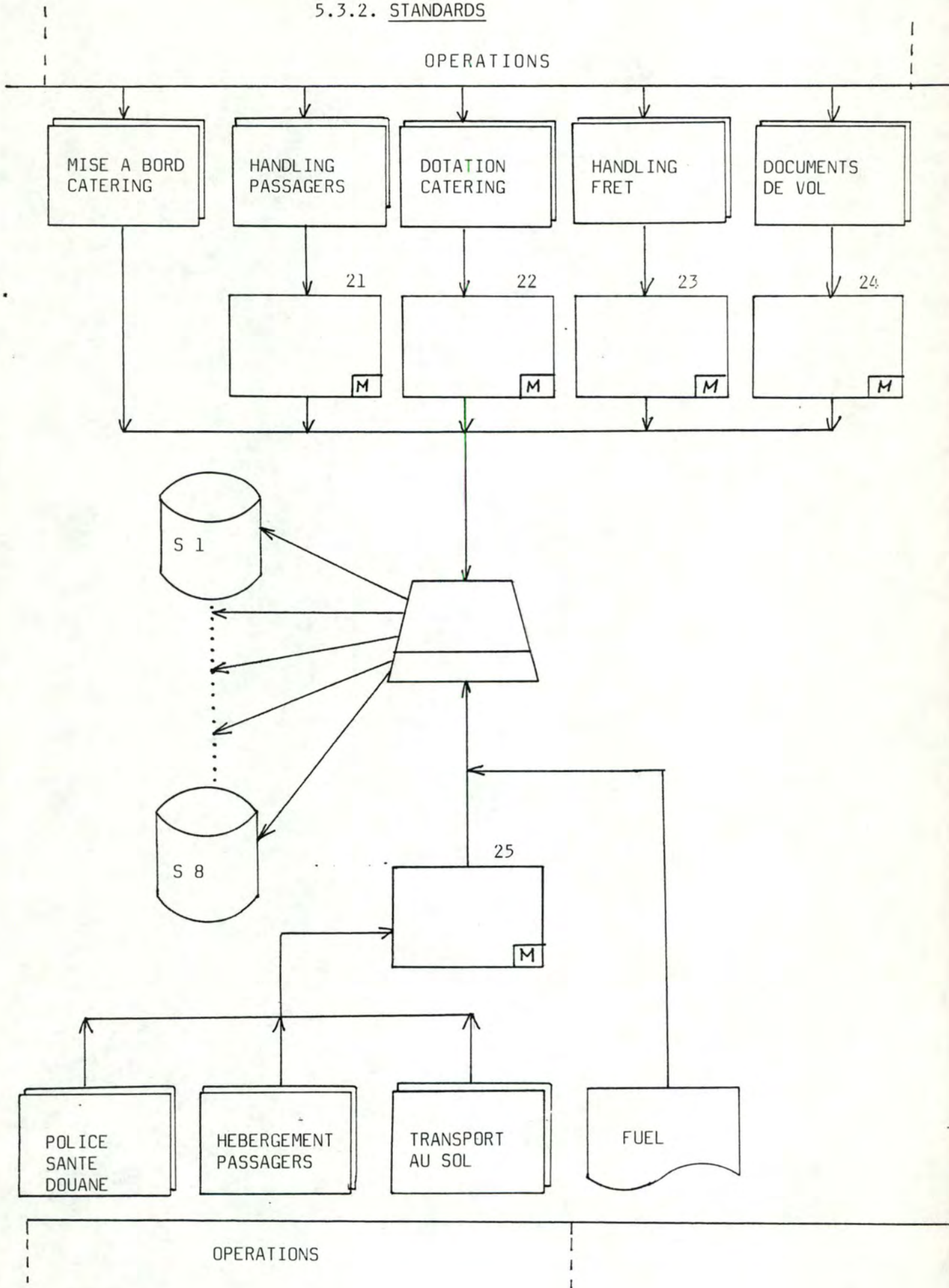
- les chargements passagers et fret, ainsi que les refus,
- les facteurs de production, tels que TRT, PKT, SKO.

Note: les refus représentent le nombre de passagers ou de tonnes fret ne pouvant être transporté sur un tronçon ou un parcours. En effet, soit l'axe Bruxelles-Athènes-Istanbul - Athènes-Bruxelles. Si la société dispose de 1000 sièges et que dans les relations on propose 600 sièges pour Istanbul et 600 pour Athènes et si la recette unitaire nette d'Athènes est inférieure à celle d'Istanbul alors l'harmonisation refusera 200 sièges sur Athènes.

Les informations suivantes:

- le nombre de passagers pouvant être transportés,
- la recette unitaire passager,

5.3.2. STANDARDS



- le fret pouvant être transporté,
  - la recette unitaire fret,
- seront mémorisées dans le fichier parcours. Les autres résultats seront publiés sous forme de listing, soit:
- le plan de trafic,
  - la répartition des axes selon les relations et les coefficients d'occupation,
  - la répartition des relations sur les axes,
  - les facteurs de production, non compris Poste, Pool et locations d'espace qui sont rajoutés manuellement.

#### 5.3.4. Standards

##### 5.3.4.1. Coûts extérieurs

1/ Fuel. Le service des opérations fournit directement le coût moyen à l'heure de vol pour un axe donné. Mémorisé dans S7.

2/ Assurances passager et fret. Le service dispose de ces données par axe.

3/ Frais de réservation safir. Même chose que le 2 ci-dessus.

4/ Douane - Police - Santé. Calculé par escale puis agrégé sur les axes; vu le caractère fastidieux et les faibles montants en jeu, le service indexe souvent les réalisations passées.

Il en est de même pour les points 5 et 6 ci-dessous.

5/ Hébergement passager.

6/ Transport passager au sol.

#### 7/ Taxes d'atterrissage

Le calcul des montants par escale et par vol est entièrement effectué par le service. Pour ce, le service se base sur l'annuaire édité et mis à jour par l'IATA. Ce calcul est fastidieux et prend au moins une semaine de travail à temps plein. En effet, pour chaque aéroport, les types de tarifs et les montants sont différents selon le type d'appareil, les heures, les saisons, ...

#### 8/ Handling tiers

Il s'agit du handling effectué pour compte de la Sabena, à l'étranger.

Pour chaque escale, la Sabena est liée à un fournisseur différent. Le service comptable qui dispose des contrats fournit les informations nécessaires au calcul.

Tout comme au point 7, chaque vol pour chaque escale nécessite un calcul différent, ensuite les résultats sont agrégés à l'axe.

#### 9/ Assistance technique tiers

Même chose qu'au point 8 ci-dessus.

#### 10/ Taxes d'embarquement et de débarquement

Même chose qu'au point 7, à part que les données ne dépendent que de l'escale.

#### 11/ Survol

Taxe payée pour le survol d'un pays non affilié à l'Euro-control.

Calcul fait par le service à partir de l'annuaire de l'IATA.

Prend 2 semaines de travail.

12/ Redevance Eurocontrol

Même chose qu'au point 11 ci-dessus, mais procédures de calcul standardisées.

13/ Déplacement Personnel navigant

En principe, l'information par axe est fournie par le service des opérations. Généralement le service indexe simplement les données précédentes.

14/ Matériel volant

Il s'agit en fait de l'amortissement économique entièrement calculé par le service.

5.3.4.2. Centres producteurs.

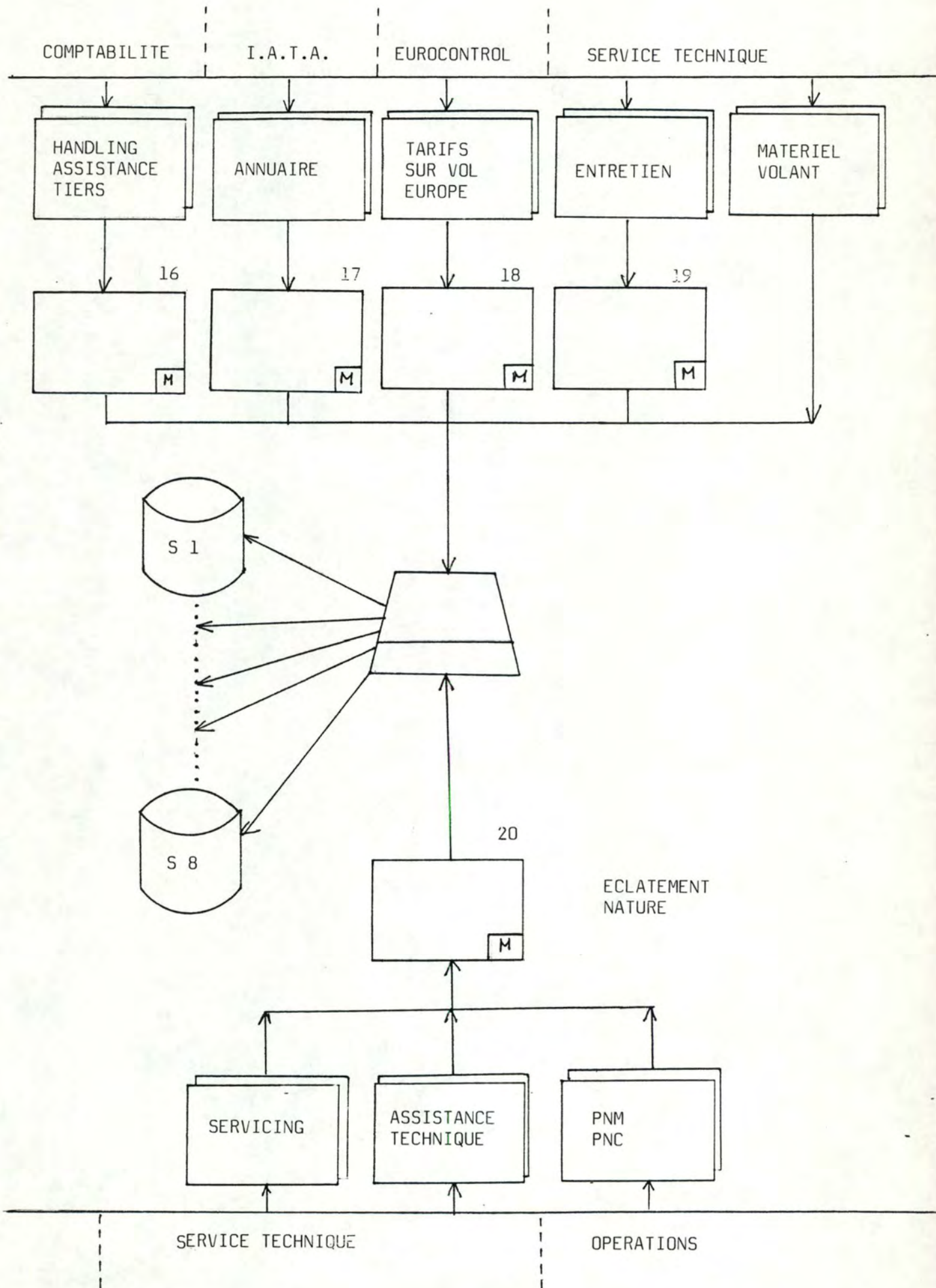
Les coûts des centres producteurs sont éclatés en 16 natures et 2 familles.

Famille des frais variables: -personnel navigant de maîtrise,

- personnel de cabine,
- personnel sous contrat Bruxelles,
- personnel engagé à l'étranger,
- matières
- Atlas (frais révision d'un pool de transporteur),
- catering.

Famille des frais fixes:

- personnel navigant de maîtrise,
- personnel de cabine,
- personnel sous contrat Bruxelles,
- représentations à l'étranger,
- centres auxiliaires (études, médical, ...),
- Atlas,
- bâtiments,
- équipement,
- prestations extérieures (téléphone, ...).



1/ Coût du personnel navigant

Le service reçoit le coût à l'heure de vol, il éclate ce coût selon les 16 natures.

2/ Coût du technique

Il s'agit du coût d'entretien des avions. Le coût à l'heure de vol par type d'appareil est fourni au service. Le détail du coût est donné selon les natures comptables, le service agrège ces natures selon "ses natures".

3/ Servicing

Le service technique fournit le coût pour un type d'avion et son aménagement. Le service dispose d'une clé de répartition de ce coût selon les natures.

4/ Service technique sociétés à l'étranger et self handling à l'étranger

Coût par escale fourni par le service technique et selon les natures comptables, ensuite agrégé par le service et imputé à l'axe.

5/ Handling passager à Bruxelles et enregistrement

Le service des opérations donne le détail complet du temps de chacune des opérations, du coût unitaire selon les natures comptables. Le service calcule selon ses natures et à l'axe.

Le handling passager couvre une dizaine d'opérations.

6/ Mise à bord catering et dotation catering

Le service reçoit le coût par axe et déjà éclaté selon les natures.

#### 7/ Documents de vol

Le service reçoit le coût par axe, il éclate ce coût selon une clé qu'il a créée. L'information provient du service des opérations.

#### 8/ Handling fret

Coût exprimé à la tonne reçu des opérations, le service agrège selon ses natures.

#### 9/ Catering passager

Fourni par le catering et par relation.

L'ensemble de l'information des standards est mémorisée dans 8 fichiers standards. Seul le catering passager est mémorisé dans le fichier relation, les assurances et frais de réservation (SAPHIR) sont stockés sur fiches tenues manuellement.

Pour rappel: les standards sont calculés par saison IATA, tout comme l'harmonisation.

Actuellement, les composants de calculs de la majorité des standards pour l'année courante sont mémorisés par l'application de comptabilité industrielle à l'informatique centrale. Ce service édite ces données, les transmet aux différents centres producteurs et au service. Les centres producteurs mettent à jour ces données et les transmettent par courrier interne au service.

#### 5.3.4.3. Index B

Tout comme pour l'harmonisation, le service ne dispose pas toujours de toutes les prévisions; de ce fait, il utilise toute une série d'index permettant d'évaluer ces données à partir des réalisations.

#### 5.3.5. Rentabilité

Le programme de rentabilité (traitement n°34) a pour fonction de calculer par axe et par saison la rentabilité brute, c'est-à-dire la différence entre la recette et les dépenses des centres (producteurs ou extérieurs) liés directement à la réalisation des vols.

Ce programme utilise l'information des fichiers

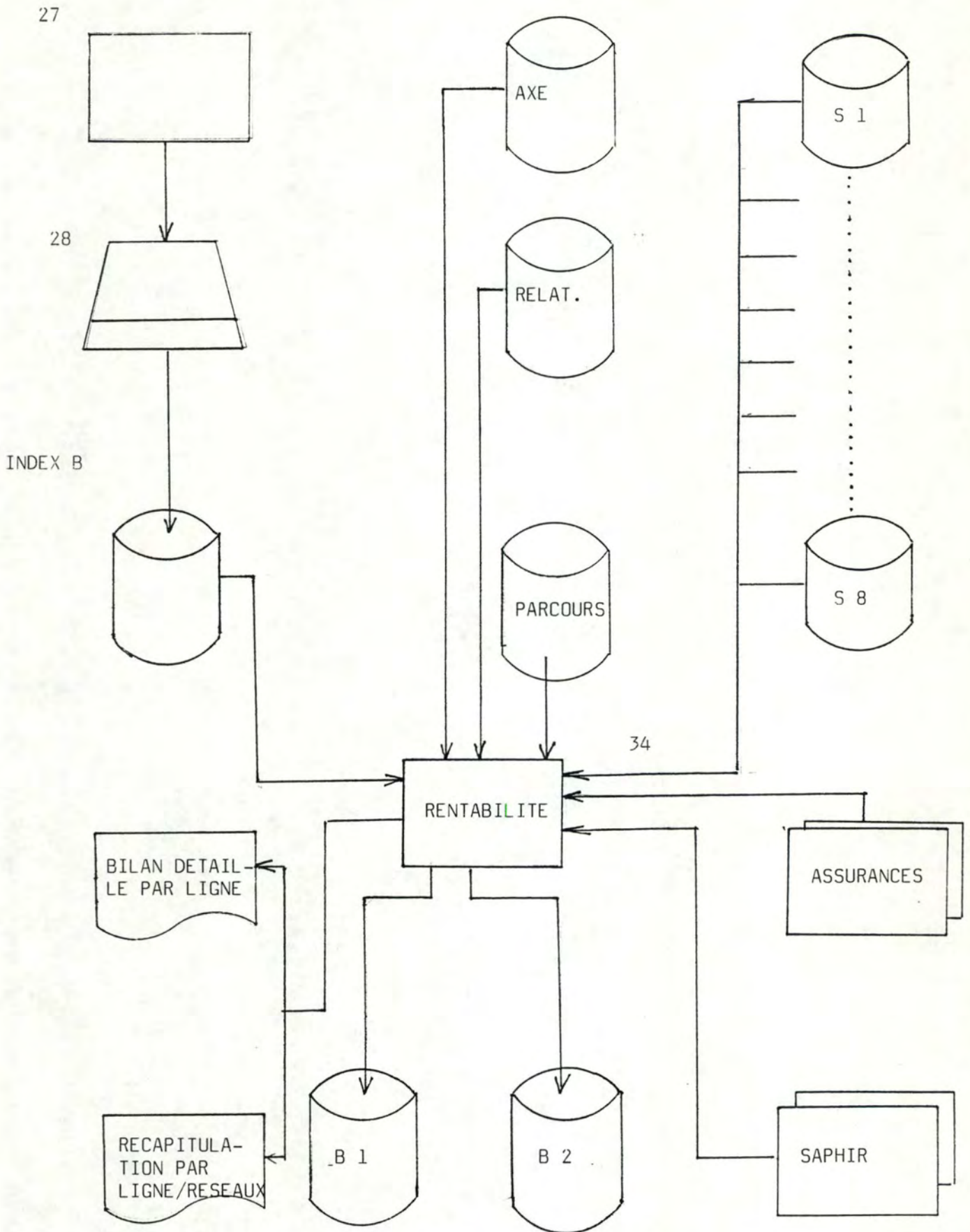
- Axe
- Index B
- Parcours
- Relation
- Standards.

Il calcule toutes les recettes par axe, ainsi que tous les coûts externes et les coûts des centres producteurs. Les coûts des centres producteurs sont éclatés selon les 16 natures vues ci-dessus.

Les résultats de la rentabilité sont mémorisés dans deux fichiers,  $B_1$  et  $B_2$ .

$B_1$  est en fait une matrice à 3 dimensions, reprenant:

- l'axe,
- le coût du centre producteur,
- la nature du coût.



$B_2$  est une matrice à 2 dimensions, reprenant:

- l'axe,
- le coût du centre extérieur et les recettes.

A partir de ces deux matrices, il est possible d'éditer plusieurs types de Récapitulatifs:

- par axe
- par réseau
- par appareil
- par saison et même par année
- par centre.

#### 5.3.6. Equipage

Le but de ces traitements va être de déterminer par axe et par saison le nombre d'équipages nécessaires.

En ce qui concerne le réseau européen, le service utilise une norme par type d'appareil, sans tenir compte des axes.

Par contre, pour les autres réseaux, le service va déterminer le routing des équipages en tenant compte de certaines contraintes, telles que:

- le nombre d'heures de vol,
- le nombre d'heures de prestation,
- les repas obligatoires,
- la période d'absence de Belgique,
- les escales insalubres.

#### Exemple.

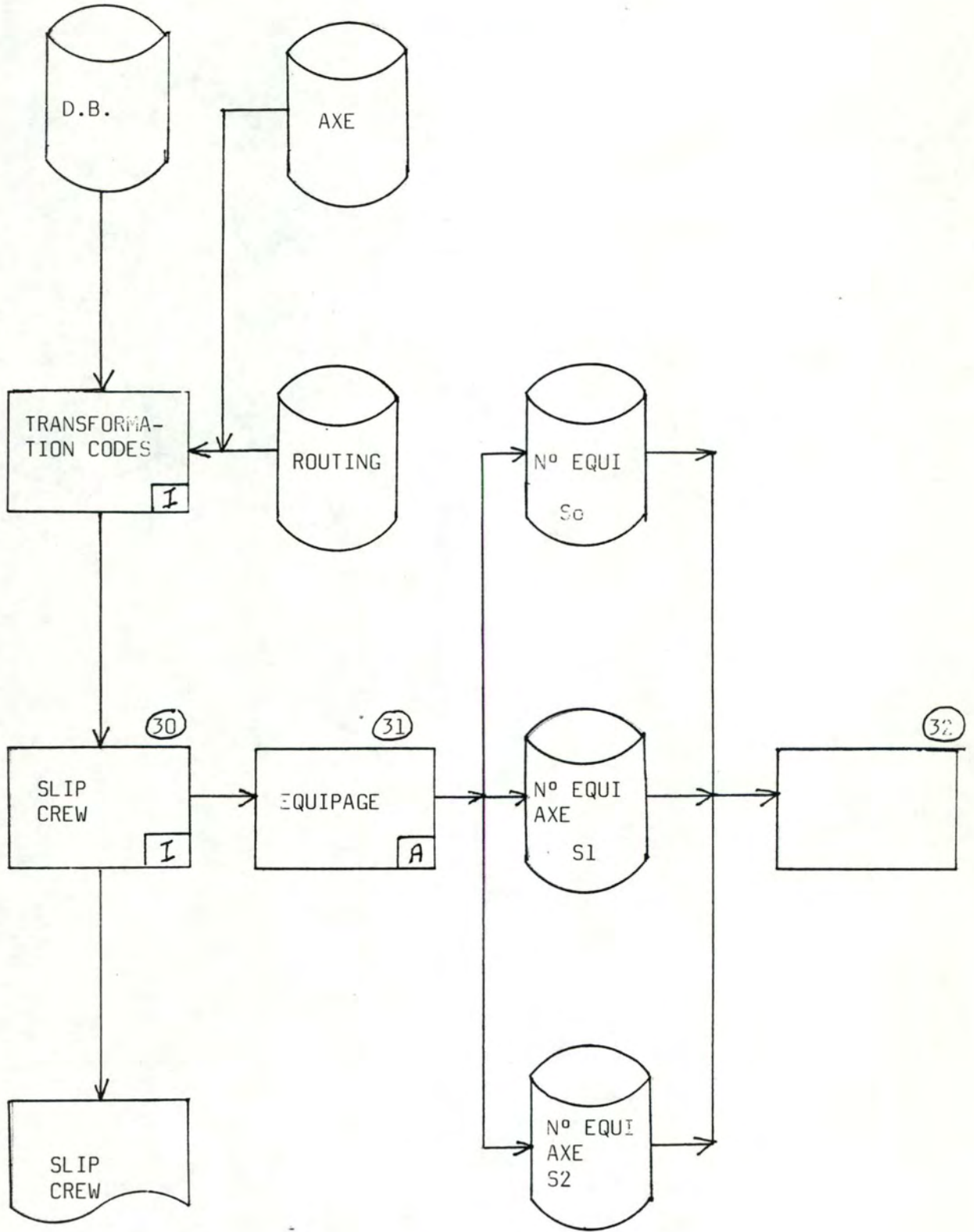
Soit axe 1 : Bruxelles - Athènes - Bruxelles

axe 2 : Bruxelles - Athènes - Dubaï - Bombay -

Dubaï - Athènes - Bruxelles;

Soit Bombay escale insalubre;

5.3.6. EQUIPAGES



on peut imaginer le routing suivant pour un équipage:

Axe 1 : Bruxelles - Athènes

une nuit de repos

Axe 2 Athènes - Dubaï - Bombay - Dubaï

repos

Axe 2 Dubaï - Athènes - Bruxelles

#### 5.3.6.1. Slip\_Crew

A partir de la Banque de données exploitées par le service commercial et contenant tous les programmes et les horaires pour une saison, ou à partir des fichiers axe et routing, le service va déterminer (traitement n°30) de manière interactive l'escale de repos pour chacun des axes. Le traitement est interactif afin d'obtenir un temps de prestation équipage par vol le plus long possible, tout en tenant compte de certaines contraintes non retenues par le programme. Ensuite, la rotation des équipages (ou encore le routing) est établie.

#### 5.3.6.2. Calcul\_d'équipage

Le traitement n°31 établit automatiquement:

- le nombre de jours équipage par rotation d'équipages,
- le nombre d'équipages pour 1 saison et l'ensemble des axes,
- le nombre d'équipages nécessaires par axe pour une saison et le nombre d'heures de vol.

Ensuite on recommence le 5.3.6.1. et le 5.3.6.2. pour les deux saisons IATA restantes, car il ne faut pas oublier qu'une année contient 3 saisons IATA.

### 5.3.6.3. Equipages année civile

Le traitement n°32 établit le nombre d'équipages et d'heures de vol par axe pour une année civile.

### 5.3.6.4. Coût équipage

A partir du fichier des standards et l'output du traitement précédent, le traitement n°33 établit le coût du personnel navigant à l'heure de vol et par axe.

Le nombre d'heures de vol, le nombre d'équipages et le coût à l'heure de vol.

### 5.3.7. Budget intégré

#### 5.3.7.1. Frais de vente

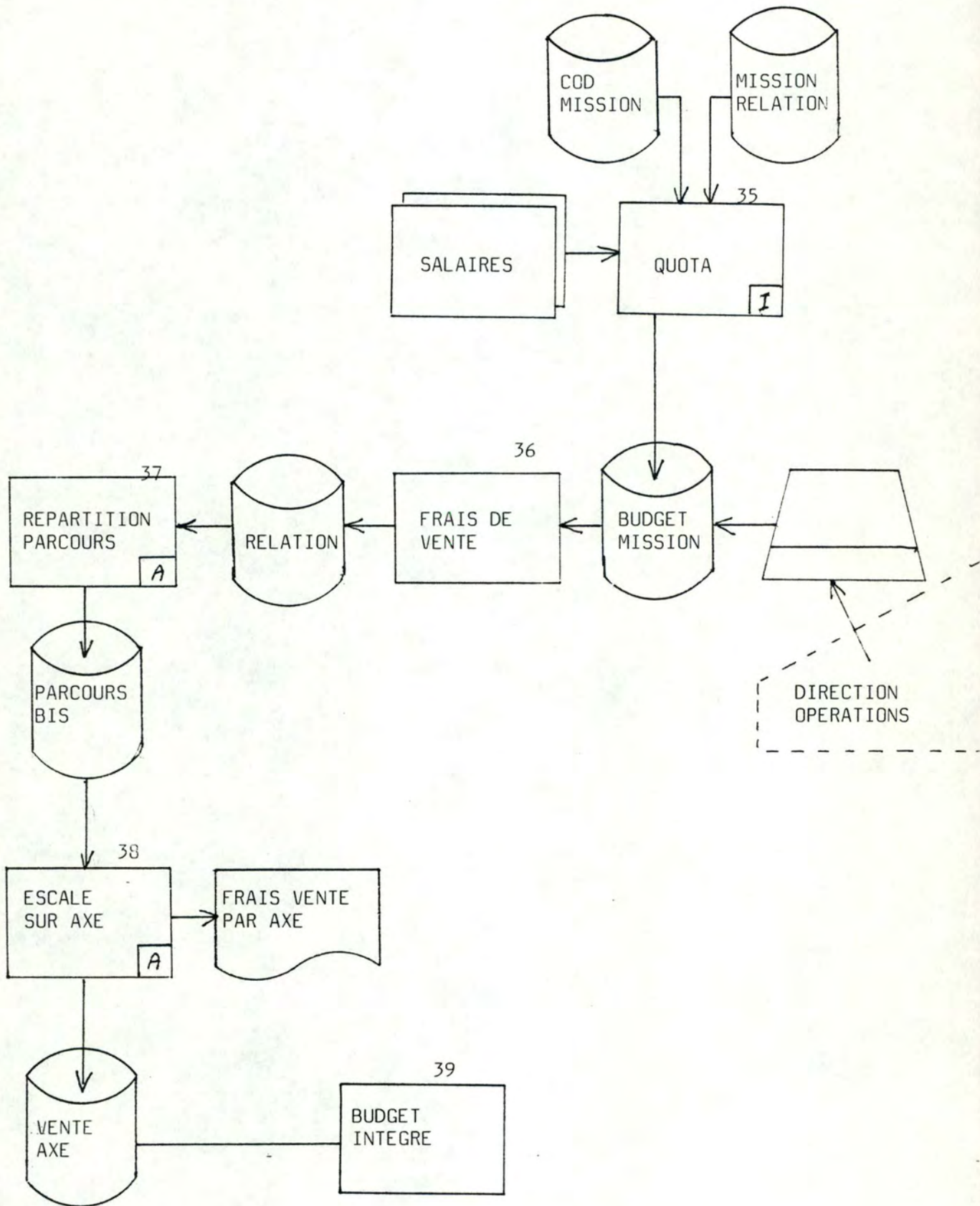
L'objet de ces traitements est de déterminer les coûts des agences Sabena à l'étranger et d'imputer ces coûts sur les axes.

#### 1.1. Budget mission

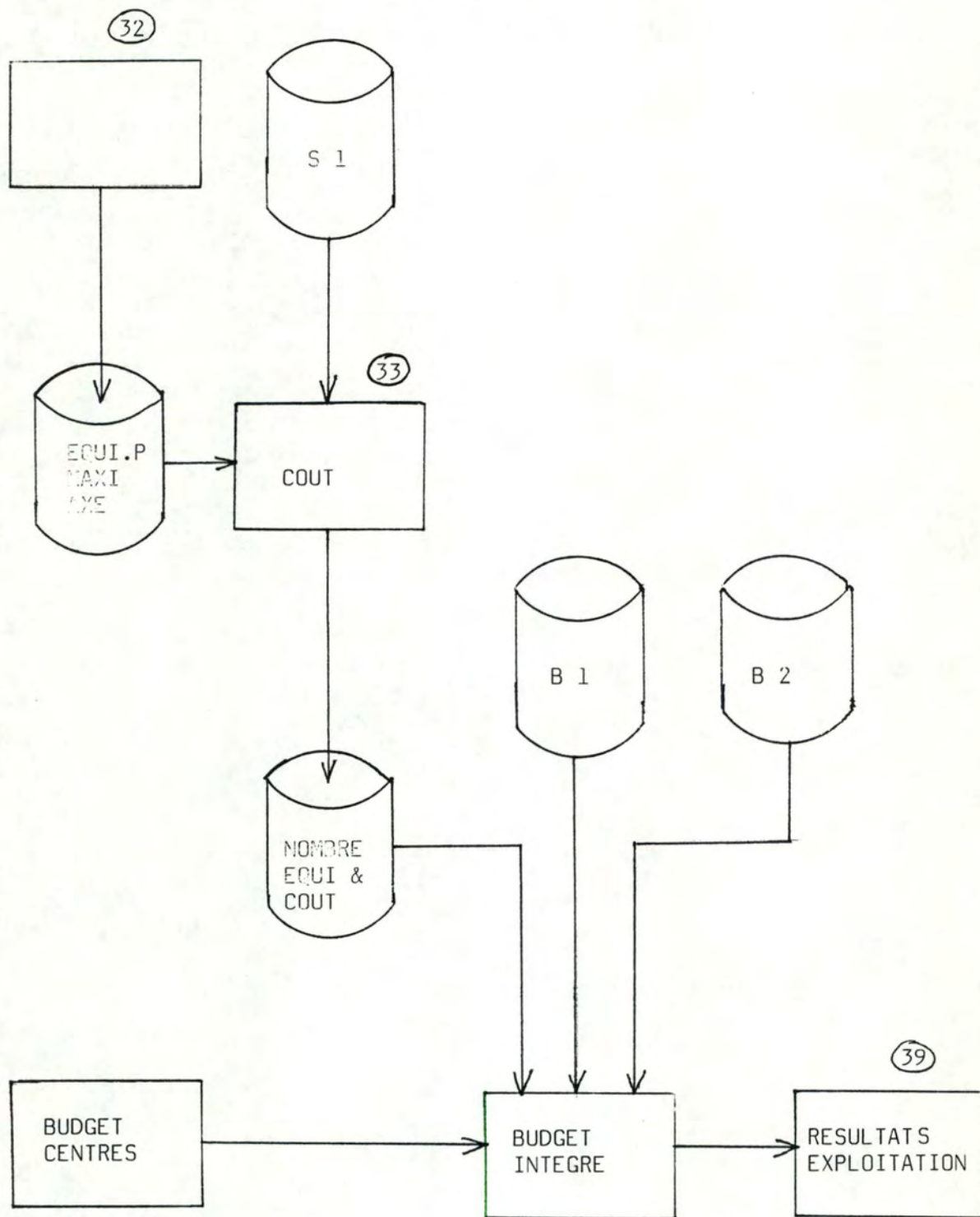
A partir du fichier mission-relation et du fichier cool-mission -contient le trafic et la recette des 2 saisons IATA précédentes, le traitement quota (n°35) calcule par mission le trafic de l'année civile. Ces résultats sont mémorisés dans le fichier Budget mission. Ce fichier contient également par mission:

- les effectifs sous contrat Bruxelles,
- les engagés sur place,
- la devise utilisée,
- la part de l'activité fret et passager.

5.3.7. BUDGET INTEGRE



## 5.3.7. BUDGET INTEGRE (suite)



## 1.2. Frais par axe

A partir des tables de salaires, il sera possible de calculer et d'imputer ces frais à la relation puis au parcours et finalement à l'axe.

### 5.3.7.2. Autres données

Le service estime à partir des données comptables les frais de l'administration, les résultats connexes et industriels, les charges d'amortissement et les charges financières.

## 5.4. Inventaire des informations

Passager, Fret, recette u: document également appelé quota.

Ce document donne par relation l'objectif de trafic de toutes les missions.

Trafic: - nombre de passagers (sens confondus)

- recette unitaire passagers

- nombre de tonnes fret

- recette unitaire fret.

Coeff. Partage: par relation la part passager dans le sens de la relation.

Relation: donne par relation (environ 300):

- le nombre de passager objectifs

- la recette unitaire passagers objectifs

- les commissions de ventes passagers (objectifs)

- le nombre de tonnes fret objectifs

- la recette unitaire fret

- les commissions fret

- les moyens spécifiques passagers

- les moyens spécifiques fret

- le coût du catering passager

- le coefficient de partage (Coeff. Partage)
- les frais de vente passagers
- les frais de vente fret.

Le fichier relation est informatisé.

Mission relation: fichier informatique.

Donne par relation et mission:

- le nombre de passagers objectifs (sens confondus)
- la recette unitaire passager objectif
- le nombre de tonnes fret objectifs
- la recette unitaire fret objectif
- le nombre de passagers objectifs dans le sens de la relation.

Commissions, Frais spéciaux: par relation:

- les commissions de ventes passagers
- les moyens spécifiques passagers
- les commissions de ventes fret
- les moyens spécifiques fret.

Cat Pass: par relation:

- le coût du catering passager.

Excédent bagage: par relation:

- % de la recette passagers pour excédent de bagage.

Poste: recettes poste par relation.

Location d'espace: recettes par relation .

Routing: fichier informatisé. Donne par axe l'itinéraire.

Axes: fichier informatisé. Donne par axe:

- le numéro de réseau
- l'occupation du tronçon le plus chargé
- le nombre d'heures de vol

- le nombre d'heures de mobilisation
- le nombre de rotations
- la charge offerte totale
- les recettes Postes, Pool, Location espaces
- les excédents bagages
- la distance.

Charge - Maxi: donne par type d'avion et aménagement la charge maximale emportable par tronçon (compte tenu de la piste, de l'altitude).

SOCT : fichier informatisé. Donne par type d'avion et aménagement:

- le réseau où il peut être exploitable
- le nombre de sièges disponibles
- la charge offerte totale
- le pourcentage de charge fret
- le pourcentage de charge passagers
- la charge fret maximale
- le poids bagages passagers
- la charge totale maximale
- le coefficient objectif de chargement
- le coefficient objectif d'occupation.

Parcours: fichier informatisé. Donne par parcours commercialisé:

- l'escale de départ
- l'escale d'arrivée
- le nombre de passagers possibles
- la recette unitaire d'un passager
- le nombre de tonnes fret

- les royalties
- la distance.

Escales: fichier informatisé. Donne l'ensemble des escales desservies par la société et leurs codes.

Type d'avion: fichier informatisé. Donne par avion et aménagement les différents codes existants.

Handling & Assistance tiers: par escale les services fournis et les coûts.

Annuaire:

- par escale :
- les taxes d'atterrissage
  - les taxes d'embarquement
  - les taxes de débarquement
- par pays :
- les taxes de survol.

Entretien: par avion: coût à l'heure de vol, éclaté selon les natures comptables.

Matériel volant: par avion: l'amortissement comptable.

Servicing: coût pour un service par type d'avion et son aménagement.

Assistance technique: coût par escale pour un service éclaté selon les natures comptables.

PNM, PNC: coût à l'heure de vol d'un équipage (selon l'appareil):

- d'un commandant
- d'un second
- d'un system engineer
- d'un commis
- des hôtesses.

Mise à bord catering: coût par axe éclaté selon les natures du service.

Dotation catering: coût unitaire par axe éclaté selon les natures du service.

Handling passager: coût unitaire de chacun des services.

Handling fret : coût unitaire à la tonne selon les natures comptables.

Documents de vol: coût unitaire par axe.

Police, douane, santé: tarifs par escale de même que pour:

- hébergements passagers
- transport au sol.

Fuel: coût unitaire moyen à l'heure de vol par axe.

Plan de trafic: itinéraire et horaire.

Récapitulatifs: ensemble des coûts et recettes donnés

- par axe
- par réseau
- par saison ou par année.

B<sub>1</sub>: fichier informatique. Donne selon les axes et les centres et les natures du service les coûts et recettes d'exploitation.

B<sub>2</sub>: fichier informatique. Donne selon les axes et les centres les coûts d'exploitation.

N° équi/axe: fichier informatique. Donne par axe le nombre d'équipages strictement nécessaires pour une saison.

COD - mission: fichier informatique. Par mission:

- le trafic des deux dernières saisons.

Budget mission: par mission:

- la devise utilisée
- les effectifs d'employés sous contrat Bruxelles
- les effectifs engagés sur place
- la part des activités dues au fret
- la part des activités dues aux passagers
- le chiffre d'affaires passager pour l'année.

## VI NOUVEAU SYSTEME D'INFORMATION

Le but de ce chapitre est de décrire le système que nous proposons. Pour ce faire, nous partirons des objectifs assignés au système, ensuite nous identifierons les processus nécessaires pour atteindre ces objectifs. Nous proposerons ensuite un système, nous en décrirons les avantages, les inconvénients et les conditions de réalisation. Nous terminerons par une description de l'environnement logiciel à développer.

### 6.1. Objectifs

#### 6.1.1. Objectif général

L'objectif du système d'information doit être que le service dispose d'un système permettant la réalisation de simulations budgétaires. Par conséquent, il est important que la notion de simulation soit définie et intégrée au nouveau système. De plus, le système ne doit pas couvrir uniquement l'élaboration des résultats d'exploitation, mais également les résultats d'entreprise.

#### 6.1.2. Objectifs opérationnels

- 1) Etablir une meilleure saisie des données
- 2) Améliorer les procédures de calcul
- 3) Permettre au système d'être évolutif.

Parmi les évolutions envisageables, nous pouvons citer:

- calcul de la rentabilité à la ligne et non plus à l'axe.
- passer d'une technique de "full costing" au "direct costing" dans le calcul du budget intégré.
- tenir compte de la part des devises dans les recettes et les dépenses.

### 6.1.3. Choix

De commun accord avec le service, nous avons décidé que le système d'information effectivement développé dans le présent mémoire ne couvrirait que l'élaboration des résultats d'exploitation. Les raisons de ce choix sont que un des objectifs prioritaires du service est justement d'améliorer l'élaboration des résultats d'exploitation.

En ce qui concerne les évolutions, nous ne retiendrons que la première citée (rentabilité par ligne), vu que l'analyse économique des deux autres n'est pas encore faite. Toutefois, à l'issue de ce mémoire, nous montrerons dans quelle mesure le système proposé pourra être étendu.

## 6.2. Processus nécessaires

### 6.2.1. Types de processus

L'élaboration des budgets nécessite quatre types de processus:

- les processus de communication,
- les processus de décision bien ou semi structurés,
- les processus de décision mal structurés,
- la production de rapport.

Concernant la distinction entre processus de décision structuré, semi ou mal structuré, nous avons repris la nomenclature du modèle de H. SIMON, proposée par P.R. KEEN et S. MORTON, cités en [2.]

#### 6.2.1.1. Processus de communication

Il s'agit essentiellement de:

- la saisie: - des quotas mémorisés à l'informatique centrale
  - des composants des standards calculés par les correspondants
  - des données techniques des avions. Ces informations sont calculées par le service technique
  - des horaires, disponibles sur le système du service mais propriété du service commercial
  - des tarifs d'Eurocontrol.
- la transmission des résultats de:
  - l'harmonisation
  - la rentabilité
  - du calcul des équipages.

#### 6.2.1.2. Processus de décision bien ou semi structuré

Bien structuré.

- Agrégation des quotas
- Calcul des standards suivants:
  - coût du personnel navigant
  - coût du technique
  - servicing
  - self handling à l'étranger
  - handling passager à Bruxelles
  - mise à bord catering
  - documents de vol
  - handling fret
  - catering passager
  - Eurocontrol
  - fuel.

- Calcul de la rentabilité.

Seul le calcul de la rentabilité est actuellement automatique.

#### Semi structuré.

- Calcul des standards:
  - survol
  - taxes d'atterrissage
  - douane police santé
  - hébergement passager
- Harmonisation
- Calcul des équipages.

Actuellement seuls l'harmonisation et le calcul des équipages sont automatisés.

#### 6.2.1.3. Processus de décision mal structuré

- Constitution des axes
- Calcul du standard
  - assistance technique tiers.

#### 6.2.1.4. Production de rapports

Actuellement tous les rapports sont dactylographiés par le pool des secrétaires. Les tableaux publiés par le système du service sont collationnés selon les besoins des différents utilisateurs.

### 6.3. Système proposé

#### 6.3.1. Architecture

Nous proposons de relier l'informatique centrale au système du service par un réseau. De plus nous proposons également que des micro-ordinateurs connectés au réseau soient installés chez les correspondants budget dans chacune des autres directions.

Ci-dessous, nous allons étudier l'impact de ce réseau sur les processus à exécuter.

#### 6.3.2. Processus de communication

- Quotas: ils sont directement transmis de l'informatique centrale au système.
- Composants des standards: Les correspondants budget disposent directement des données de la comptabilité industrielle stockée à l'informatique centrale. De plus, ils possèdent quelques fichiers locaux à l'aide desquels ils mettent les composants des standards à jour; ensuite ils les transmettent directement au service via le réseau.
- Les résultats des calculs sont quant à eux transmis à qui de droit.

#### 6.3.3. Processus de décision bien ou semi structuré

##### Bien structuré.

Nous proposons que le système du service effectue toutes ces tâches automatiquement.

##### Semi structuré.

Nous proposons que le service dispose d'un système de type système d'aide à la décision (Visicalc, Lotus 1 2 3, ...) afin d'effectuer ces tâches. De plus nous proposons que les correspondants disposent également de tels systèmes afin de mettre à jour les composants des standards.

#### 6.3.4. Processus de décision mal structuré

Nous pensons qu'un système d'aide à la décision de haut

niveau pourrait être utilisé pour la constitution des axes.

#### 6.3.5. Production de rapports

Nous proposons qu'un système de traitement de texte associé à un système de génération de graphiques soit associé au réseau.

#### 6.4. Avantages et inconvénients

##### Avantages:

- Gain de temps pour l'introduction des données.
- Gain de temps pour les calculs.
- Information plus pertinente.
- Extensions possibles.

##### Inconvénients:

- Automatisation d'une organisation existante.

##### 6.4.1. Avantages

###### 6.4.1.1. Gain de temps pour l'introduction des données

Les gains sont très importants. En effet, les quotas contiennent 18.000 données qui sont pour chaque saison IATA agrégés manuellement en 4.800 données introduites à la main. Pour chaque simulation, le service introduit plus de 4.580 données de standards, si nous tenons compte des composants des standards, le chiffre s'élève à 45.800.

###### 6.4.1.2. Gain de temps pour les calculs

Ce sont les calculs des standards qui sont actuellement les plus longs. En effet, il faut un à deux hommes/semaine pour calculer le survol et les taxes d'atterrissage et ce pour chaque simulation.

#### 6.4.1.3. Information plus pertinente

La qualité de l'information peut être améliorée à deux niveaux:

1) Dans le système actuel, lors de la création d'une simulation, il arrive souvent que le service ne recalcule pas tous les standards, car le temps nécessaire demandé par les correspondants est trop long. Par conséquent, le service opère une simple indexation. Dans certains cas, comme le calcul de fuel, cette simple estimation est difficile, voire hasardeuse, car les montants en jeu sont importants.

2) Vu que les données sont directement saisies dans l'application de comptabilité industrielle, la part des devises, tant dans les recettes que dans les dépenses, peut être connue. De ce fait, le système d'information peut être enrichi, vu que 70% des recettes et dépenses sont en devises.

#### 6.4.1.4. Extensions possibles

L'intérêt d'un pareil système est qu'il permet d'être facilement utilisable par d'autres applications voisines à celle que nous étudions.

Parmi les extensions possibles, nous en voyons principalement à deux niveaux:

1) au sein des services budgets, tant court terme que long terme.

2) le contrôle budgétaire.

1) Les services du budget .

- Une liaison entre le budget court terme et moyen terme peut être établie.
- Le calcul de la rentabilité des investissements et des plans financiers.

2) Le contrôle budgétaire.

- Une des tâches les plus importantes des correspondants est d'effectuer le contrôle de gestion. Ce contrôle se fait à partir du budget arrêté et des données réalisées enregistrées à la comptabilité.

A cette fin, chacun des correspondants dispose actuellement d'un terminal relié directement à l'informatique centrale. Par conséquent, le réseau proposé peut parfaitement se substituer aux lignes actuelles, et les micros utilisés dans les tâches de contrôle.

6.4.2. Inconvénient

6.4.2.1. Automatisation d'une organisation existante

Le système proposé n'est que l'automatisation d'un système existant, alors que peut-être une autre forme d'organisation également automatisée serait plus performante. Ceci dépasse le cadre de ce travail, car il ne nous appartient pas d'étudier l'organisation de l'ensemble des services du budget de la société.

6.5. Conditions de réalisation du système

Si techniquement le nouveau système est séduisant, il ne

faut pas perdre de vue que certaines conditions doivent absolument être réalisées avant qu'un pareil système ne soit implanté. En effet, la combinaison de systèmes d'aide à la décision sur micros reliés entre eux par un réseau constitue une "première" dans l'entreprise.

#### 6.5.1. Sensibilisation

Une des caractéristiques importantes des systèmes d'aide à la décision est son aspect évolutif. Nous pouvons grossièrement résumer ce processus évolutif comme suit. Au début, le système est utilisé pour des tâches bien précises. De par l'utilisation du système, l'utilisateur découvre lui-même les possibilités d'extension du système à d'autres tâches.

Ceci entraîne la motivation d'autres utilisateurs, et nous assistons à un effet de boule de neige. Une fois que les différents utilisateurs maîtrisent le ou les systèmes mis en place, un besoin impératif de réorganisation des tâches nécessaires pour remplir les fonctions de l'organisation se fait sentir. De plus, ce n'est qu'à partir du moment où les tâches sont réorganisées que le pay back de l'investissement apparaît. Nous voyons donc que l'introduction d'une telle technologie est d'une importance considérable, et ce pour les raisons suivantes:

- malgré le gain financier à terme, au début il est très difficile d'estimer le pay back de l'investissement,
- l'impact sur l'organisation de l'entreprise.

Nous voyons donc que l'introduction de tels systèmes

relève des choix stratégiques d'une organisation. C'est pourquoi la littérature s'accorde à dire que le top management de l'organisation doit être associé à cette introduction, c'est-à-dire qu'il doit motiver, favoriser l'implantation de tels systèmes, mais aussi contrôler.

La fonction d'un service de corporate planning est de produire les inputs nécessaires aux choix stratégiques arrêtés par le top management. Par conséquent, un projet pilote de systèmes d'aide à la décision, liés à un réseau, mené par le service aurait comme avantages:

- de sensibiliser, si la nécessité s'en fait sentir, le top management,
- de montrer au top management l'intérêt et l'efficacité de tels systèmes. En effet, par ce nouveau système le top management est mieux informé,
- de maîtriser l'effet boule de neige de par le nombre limité de personnes impliquées dans ce nouveau système.

#### 6.5.2. Les problèmes de compétences

L'introduction de réseaux et de systèmes partagés peuvent selon la littérature entraîner des luttes de compétence entre les responsables de l'informatique centrale et les utilisateurs. Nous pensons que ce risque pourrait être évité vu que dans la société étudiée, le corporate planning, ainsi que le département informatique, dépendent de la même direction, c'est-à-dire de la Direction Economique.

### 6.5.3. Motivation des utilisateurs

Lors de nos entretiens avec les différents membres du service, nous avons enregistré non seulement un intérêt croissant, mais aussi une réelle coopération. De ce fait, cette dernière condition nous semble parfaitement remplie.

## 6.6. Environnement logiciel

### 6.6.1. Traitements

#### 6.6.1.1. Tâches

En plus des programmes existants, nous proposons que des logiciels spécifiques soient rédigés par le service pour les processus de décision bien structurés. Pour les processus de décision semi structurés, nous pensons que des logiciels existants dans le commerce peuvent être utilisés. Quant aux autres processus, une analyse plus détaillée devrait être menée.

#### 6.6.1.2. Utilitaires

Un ensemble de programmes utilitaires doit également être rédigé. Nous citons pour mémoire les principaux:

- Logiciel de création de simulations

Le but d'un pareil programme est d'éviter que lorsque l'utilisateur crée plusieurs simulations, il ne doive introduire plusieurs fois les mêmes données inchangées dans plusieurs simulations.

- Logiciels de transformation

Le but de ces logiciels est de rendre compatibles les données end-users avec celles mémorisées dans le système. En effet, nous devons nous rappeler que d'un service à l'autre, un même concept possède plusieurs identifiants différents.

- Logiciels de transparence

C'est dans ce type de logiciel que les transformations d'imputation des données à l'axe en données imputées à la ligne peuvent être effectuées.

6.6.2. Données

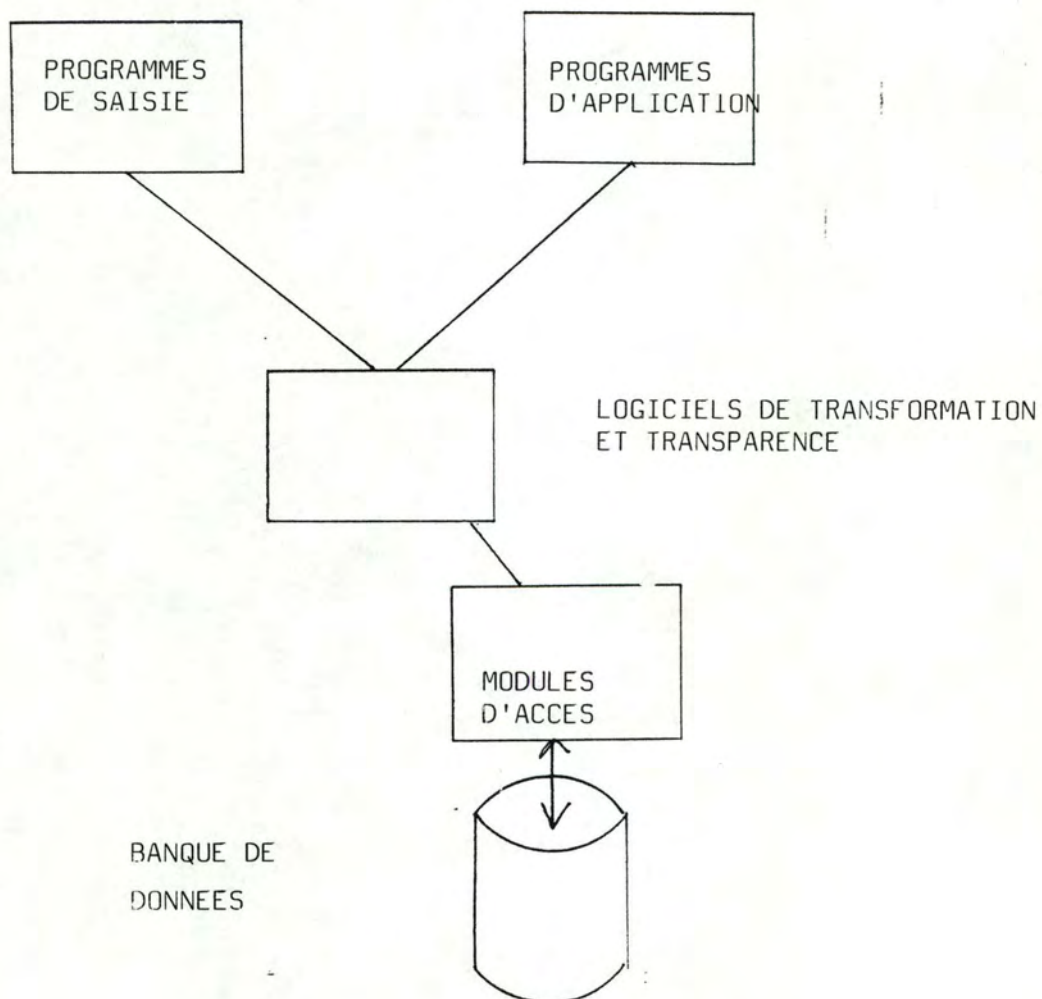
Nous ne pouvons préjuger des décisions quant à l'implantation du système proposé. De toute façon, il ressort de l'étude de l'existant qu'il est nécessaire de développer un nouveau système de fichiers. En effet, nous avons constaté que parmi les différents fichiers, il existait de nombreuses redondances, de plus que les fichiers avaient été construits pour des applications spécifiques, et qu'ils ne permettaient pas de ce fait d'être utilisés pour d'autres applications. De plus, l'espace mémoire nécessaire à ces fichiers est relativement important et devient même critique, vu les capacités disponibles.

Ce problème de capacité entraîne le montage et démontage fréquents de supports. C'est pourquoi nous nous proposons de développer une base de données regroupant les informations nécessaires au service. La base que nous proposons a pour caractéristique d'être immédiatement opérationnelle pour les applications du système actuel, mais qui d'autre part peut, selon quelques modifications, être facilement utilisée dans le cadre d'un autre système. De plus, afin de rendre cette banque de données indépendante par rapport au SGBD (système de gestion de banque de données), disponible actuellement dans le service,

nous proposons que les accès à cette banque se fassent via des modules d'accès ayant pour caractéristique d'être indépendants du SGBD, mais complètement dépendants des stratégies d'accès.

### 6.6.3. Hiérarchie

Nous pouvons établir une hiérarchie des principaux logiciels qu'il faut mettre en oeuvre (nous ne tiendrons pas compte des logiciels de gestion du système ou du réseau).



Par cette hiérarchie, nous pensons qu'il est possible de développer une banque de données répondant aux exigences actuelles, mais aussi futures du service. En effet, des modifications n'entraînent pas le changement complet de tous les logiciels.

Exemples:

- 1) Si une source de données est modifiée, seul le niveau 4 est affecté.
- 2) Possibilité de passer d'un SGBD de famille totale à un SGBD relationnel en ne changeant que le niveau 2.

6.7. Conclusion

Nous proposons ce nouveau système car à nos yeux, il constitue un véritable système d'information permettant de réaliser les objectifs du service. Dans notre proposition, nous montrons ce qui est techniquement réalisable, mais cette proposition n'est qu'une première étape. Avant de réaliser ce projet, il faut réaliser les deux étapes suivantes, la première est au niveau de la gestion, la seconde au niveau informatique.

1) Au plan de la gestion:

- Prendre une décision concernant l'intérêt d'introduire un réseau de micro-ordinateurs.
- Etudier l'impact probable d'un tel réseau sur l'organisation des services concernés.

2) Au plan informatique:

Il est nécessaire d'étudier plus profondément les systèmes d'aide à la décision, ainsi que le matériel tant

logiciel que machines nécessaires. Cette étape est importante car elle permet de fournir une idée des moyens financiers à consentir pour développer cette application pilote.

TROISIEME PARTIE :  
DEVELOPPEMENT DE LA  
BANQUE DES DONNEES.

## VII. SCHEMA CONCEPTUEL

### 7.1. Introduction

Nous proposons de décrire le schéma conceptuel par le modèle entité-association ( 1 ). A chacune des étapes citées dans le chapitre 5.2. de la deuxième partie, nous associons un sous-schéma.

<u>Etape</u>	<u>Sous-schéma</u>
Constitution axes	Produit et moyens
Quotas	Quotas
Harmonisation	Facteurs de production
Standards	Coûts unitaires
Rentabilité	Budget
Equipages	Equipages

Les sous-schémas proposés sont cohérents entre eux. Par conséquent, une fois qu'ils sont établis, ils ne doivent plus être intégrés. Les concepts représentés dans ces sous-schémas sont ceux déjà définis dans la description de l'existant, auxquels nous en ajoutons de nouveaux, tels que celui de simulation.

Simulation : pour un programme fixé et pour une saison IATA déterminée, une simulation fixe un ensemble de coûts et de recettes.

L'introduction du concept de simulation est important, car il signifie qu'un ensemble de mêmes objets conceptuels a des valeurs différentes ou encore des occurrences

différentes selon la simulation.

Exemple:

A nos yeux, la ligne 127 de la simulation 42, c'est-à-dire de la saison été 84 est totalement différente de la ligne 127 des simulations 43 et 44.

Elles couvrent toutes deux la saison hiver 84, mais avec un ensemble de coûts différents.

Nous ajoutons également quelques concepts qui vont permettre de calculer certains standards, tels que Eurocontrol, survol, fuel, documents de vol. Nous ne décrivons pas ces concepts ici car ils sont relativement simples et sont repris comme attributs dans les sous-schémas.

Plusieurs concepts possèdent actuellement, au sein de la firme, plusieurs jeux d'identifiants (il s'agit essentiellement des axes, lignes, avions, escales, réseaux). Nous avons choisi par objet un jeu qui, à nos yeux, avait le plus de chance d'être utilisé tant par le service concerné que par les autres.

<u>Objet</u>	<u>Identifiant</u>
Ligne	Numéro de ligne "horaire"
Axe	Centre de rentabilité
Escale	Code ville
Avion	Numéro des statistiques

Note:

Conventions pour les attributs:

D : Décomposable

R : Répétitif

S : Simple

I : Identifiant

## 7.2.Sous-schéma: Produits et Moyens

### LIGNE

Définition: tout vol commercial identifié par un numéro, dont la fréquence, l'itinéraire et les heures de départ et d'arrivée sont prévus pour une simulation.

Contrainte d'intégrité: Bruxelles ne peut être l'escale de départ et d'arrivée.

Attributs: NUM L & SIM : Numéro de ligne simulation (I)  
 TYPE-LI : Type de ligne (aller, retour, intem) (S)

### AXE

Définition: ensemble des lignes aller, intermédiaires et retour consécutives dont l'itinéraire et les escales des lignes aller, intermédiaires et retour sont identiques pour une simulation.

### Attributs:

- NUM-AXE : numéro de l'axe et du centre de rentabilité
- SIM : simulation
- ROT (1) : nombre de rotations pour une saison (S)
- N-JOUR-E : nombre de jours-équipages nécessaires pour une saison (S)
- SYMT : nom de l'escale constituant le point de symétrie de l'axe (S)

(1) Rotation: nombre de fois qu'un axe est exécuté.

Remarques:

- NUM-AXE : concaténation  
numéro "Axe", Centre de rentabilité
- Identifiant: concaténation: NUM-AXE, SIM

PARCOURS

Définition: partie de vol commercialisable entre deux escales d'un même AXE pour une simulation donnée.

Attributs:TRONCON

Définition: partie de vol comprise entre deux escales successives.

Attributs:

- DIST : distance km (S)
- ESAD : équivalent still air distance (S).

ESCALE

Définition: point d'atterrissage ou de décollage.

Attributs:

- VILLE : code ville (I)
- AEROPORT : nom de l'aéroport (S)
- INSALUBRE : indicateur si la ville est considérée insalubre pour les équipages (S)
- LOGEMENT : D & R
  - HOTEL: coût par simulation du logement (S)  
des équipages (coût global en Fr B)
  - TRANS: coût global par simulation du transport au sol des équipages (S)
  - SIM : simulation.

PAYS

Définition: territoire d'une nation.

Attributs :

- NOM-PAYS : nom du pays (I)
- DEVISE : devise dans laquelle les droits de survol doivent être payés (S)
- VOL:-T.TARIF: type de tarif pour les droits (S) de survol (unitaire, au poids, forfait, pourcentage de la taxe d'atterrissage).
- VAL : valeur du tarif (S)
- POND : pondération \$ donnée par le pays (S)
- SIM : simulation

AVION

Définition: tout type d'appareil pouvant voler, identifié par son type et son aménagement.

Attributs:

- APP-AM : type d'appareil et son aménagement (I)
- BLOCK-FUEL : fonction de consommation de carburant en fonction du temps block
  - $\alpha$  : ordonnée à l'origine (S)
  - $\beta$  : coefficient (S)
- AVI-A : constante de consommation de carburant au sol (S)
- AVI-B : coefficient de consommation de carburant au sol (S)
- RESEAU : réseau où il peut être affecté (S)
- SIEGES : nombre maximum de sièges disponibles (S)
- CHARG.S : poids maximum des passagers (S)
- COEFFICIENT OBJECTIF

- CHARG. F : poids maximum de fret (S)
- CHARG. B : poids maximum de bagages (S)
- COEF. P : coefficient de chargement moyen (S)
- COEF. F : coefficient de chargement fret moyen (S)

#### CONTIENT-P.

Si  $L \underline{E}$  LIGNE \*  
 $P \underline{E}$  PARCOURS } alors  $\langle L, P \rangle$  signifie que:

L contient P, c'est-à-dire que la ligne

L contient le parcours P.

Contrainte: L et P doivent appartenir à la même simulation.

#### COMPOSE-T

Si  $P \underline{E}$  Parcours et  $T \underline{E}$  Tronçon alors  $\langle P, T \rangle$  signifie que P est composé du tronçon T, et ce pour une simulation donnée.

#### Attribut:

SIM : simulation.

#### A.L.

Si  $A \underline{E}$  Axe et  $L \underline{E}$  LIGNE, alors  $\langle A, L \rangle$  signifie que L appartient à A.

#### P-DEP

Si  $P \underline{E}$  Parcours et  $E \underline{E}$  Escale alors  $P \text{-DEP} \langle P, E \rangle$  signifie que P a E comme escale de départ.

#### P-ARR

Si  $P \underline{E}$  Parcours et  $E \underline{E}$  Escale, alors  $P \text{-ARR} \langle P, E \rangle$  signifie que P a E comme escale d'arrivée.

(\*) Note: E = appartient.

T-DEP

Si T E Tronçon et E E Escale, alors T-DEP <T,E>

signifie que le tronçon T a comme escale de départ E.

Attributs:

- HORAIRE : (R & D)
- JOUR : jour de semaine de départ (R)
- HEURE-0 : heure GMT de départ (S)
- HEURE-L : heure locale de départ (S)
- N.L : numéro de la ligne (S)
- SIM : simulation (S)

T-ARR

Similaire à T-DEP mais pour l'arrivée.

SURVOL

Si T E Tronçon et P E Pays alors survol <T,P>

signifie que T survole le pays P.

Attribut:

- DISTANCE-P : distance en km du tronçon survolé dans le pays (S).

LOCALISE

Si P E Pays et E E Escale alors Localise <P,E> signifie

que E est localisé dans le pays P.

AFFECTE

Si L E Ligne et AP E Type Avion alors Affecté <L, AP >

signifie que AP est affecté à la L.

Attributs:

- CHARGE-PAX-A : nombre de sièges offerts (S)
- CHARGE-FRET-A : charge fret offerte (S)
- MOB : heures de mobilisation appareil (S)
- HEURES - VOL : heures de vol (S).

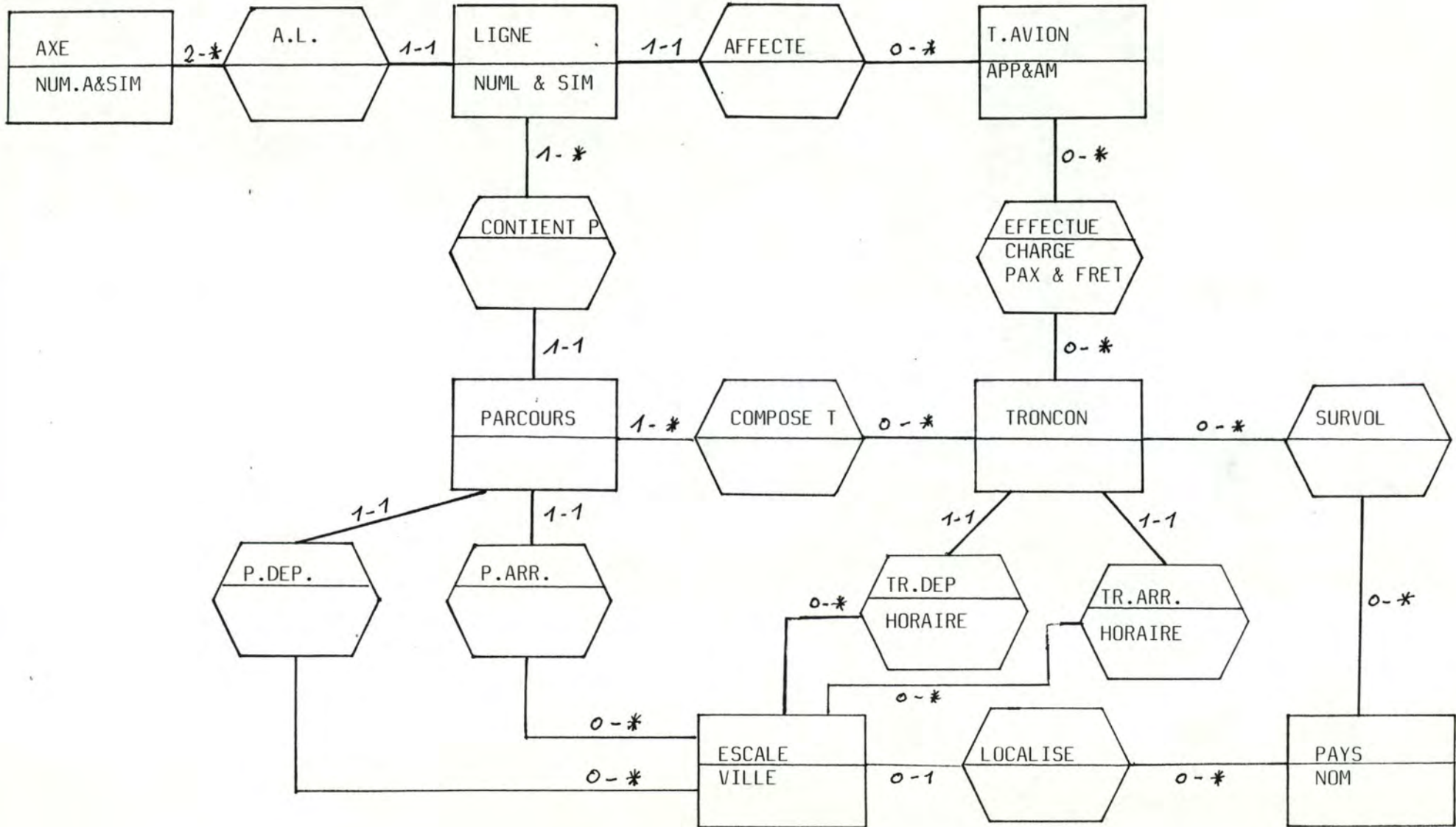
EFFECTUE

Si AP E Type Avion et T E Tronçon alors Effectue <AP,T>  
signifie que l'appareil peut effectuer le tronçon T.

Attributs:

- CHARGE-PAX-T : sièges qui peuvent être offerts (S)
- CHARGE-FRE-T : charge fret disponible (S)
- TEMPS-BLOCK : temps de vol block à block (S)
- HV-T : heures de vol (S)
- HM-T : heures de mobilisation de l'appareil (S)
- HM-P : heures de mobilisation du personnel  
navigant (D)
- COEF-OBJ : coefficient objectif d'occupation passager (S)
- T.O.W. : take off weight (S).

.. SOUS-SCHEMA PRODUIT ET MOYEN



### 7.3. Sous-schéma QUOTA

#### RELATION

Définition: trajet constitué par un ou plusieurs parcours indépendamment de la ligne. Toutefois une relation ne peut avoir une rupture d'axe (changement d'axe pour une même relation).

#### Attribut

#### T. RECETTE

Définition: type de recette associée à une prestation de transport fournie (passager, fret, pool, ...).

#### Attribut:

- NOM R : nom du type de la recette (I).

#### T. MOYEN

Définition: type de moyen commercial associé à la prestation de transport fournie.

#### Attribut:

- NOM M : nom du type de moyen (I).

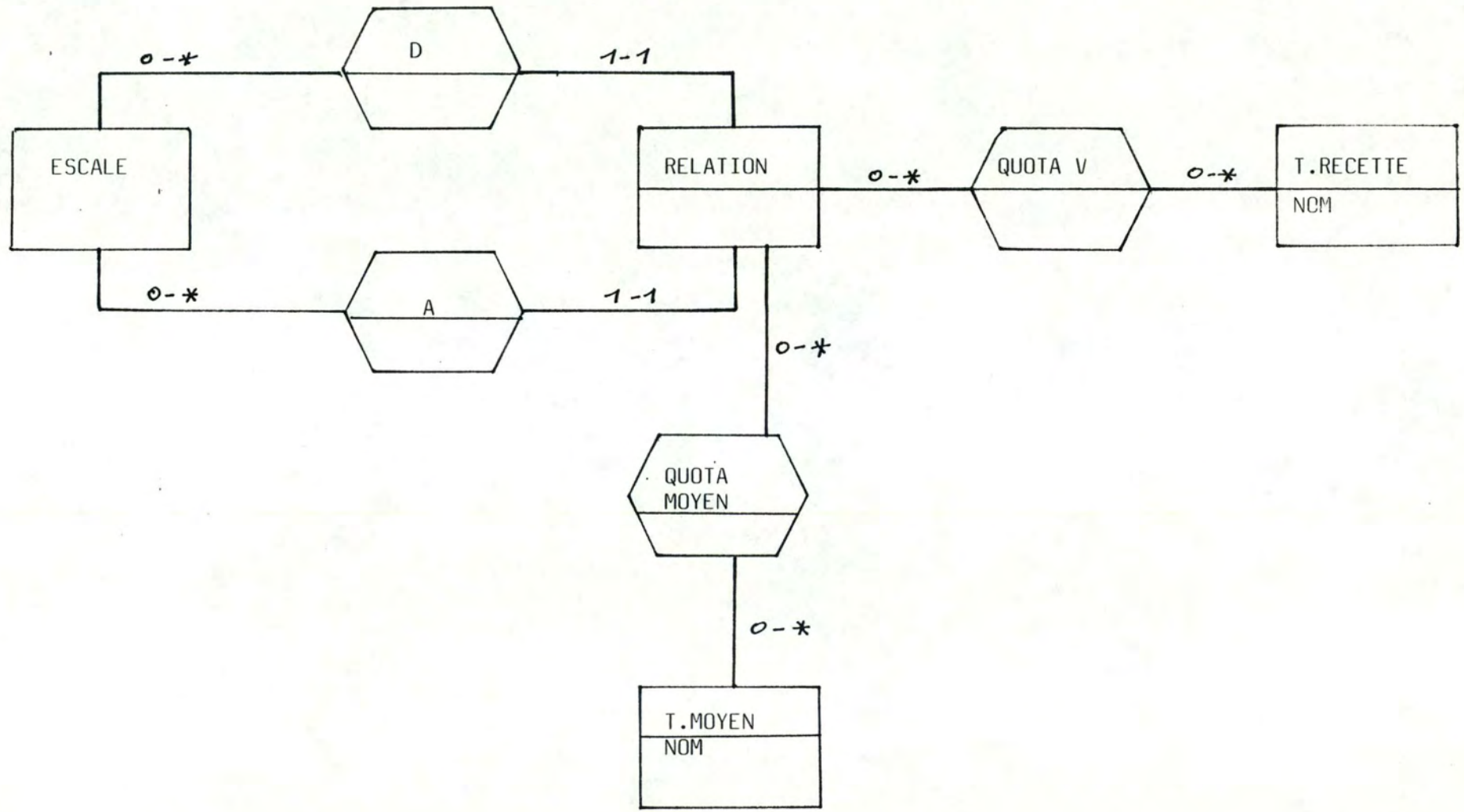
#### QUOTA V

Définition: si  $T.R \in T$ . Recette et si  $R \in R$  Relation  $\langle T.R, R \rangle$  indique l'objectif de vente pour une recette TR pour la relation R.

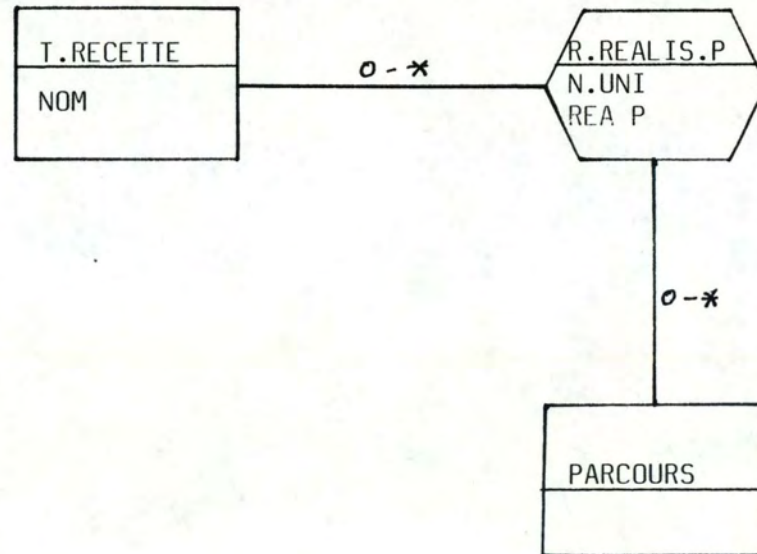
#### Attributs:

- Q.V. (R & D)
  - Recette unitaire (S)
  - Nombre d'unités (S)
  - SIM (S).

SOUS-SCHEMA QUOTAS



SOUS-SCHEMA FACTEUR DE PRODUCTION



QUOTA. M

Définition: si  $T.M \in T.Moyen$  et  $R \in Relation \Rightarrow \langle T.M, R \rangle$   
 indique le quota de moyen commercial TM à consentir par  
 la Relation R.

Attributs:

- Q.M. (R & D)
- NUM (S)
- SIM (S)

D.

Si  $R \in Relation$  et  $E \in Escal e \Rightarrow \langle R, E \rangle$  signifie que R  
 a pour escale de départ E.

A.

Même chose que pour D.

7.4. Sous-schéma Facteurs de productionR. REALISABLE P

Définition: si  $T \in Tronçon$  et  $TR \in T. Recette \Rightarrow \langle T, TR \rangle$   
 indique la recette réalisable pour le tronçon T.

Attributs:

- N.UNI.REA.P : nombre d'unités réalisables (S)
- R. UNI.REA.P : recette unitaire réalisable (S)
- N.UNI.REF.P : nombre d'unités refusées (S).

### 7.5. Sous-schéma Coûts- unitaires

#### SERVICE-PRODUIT

Définition: service nécessaire pour la préparation ou l'exécution d'un vol. Ce service est produit par un département de la société.

#### Attributs:

- NOM-SP : nom du service (I)

#### SERVICE-EXTERNE

Définition: service nécessaire pour la préparation ou l'exécution d'un vol. Ce service est produit par une entité tierce à la société.

#### Attributs:

- NOM-SE : nom du service (I)

#### STANDARD-I

Si L E Ligne et si SP E Service-Produit alors Standard-I  $\langle L, SP \rangle$  signifie que chaque fois qu'il y a une rotation de la ligne, le service SP est presté, dont le coût dépend de la ligne L.

#### Attribut:

- MONTANT-I : coût unitaire du service, (D) décomposé selon les 16 natures comptables.

#### STANDARD-II

Si E E Escale et si SP E Service-Produit alors Standard-II  $\langle E, SP \rangle$  signifie qu'à chaque rotation de la ligne desservant l'escale E pour une simulation, le service SP est presté, dont le coût dépend de l'escale E.

Attributs:

- II (D&R)
  - MONTANT-II 1 coût unitaire du service (S) décom-
    - 
    - posé selon les 16 natures comptables.
    - 
    - 
    - 16
  - SIM : simulation (S)

STANDARD-III

Si AP E Type-Avion et SP E Service-Produit, alors Standard-III  $\langle AP, SP \rangle$  signifie qu'à chaque rotation d'une ligne effectuée par le type d'avion AP pour une simulation, le service SP est presté, dont le coût dépend du type d'avion AP.

Attributs:

- III - (D & R)
  - MONTANT III coût unitaire du service décom-
    - 1
    - sable selon les 16 natures comptables.
    - 
    - 
    - 16
  - SIM : simulation.

STANDARD-IV

Si SE E Service-Externe et E E Escale alors Standard-IV  $\langle SE, E \rangle$  signifie que les rotations d'une ligne desservant l'escale E pendant une simulation déterminée nécessitent la prestation du service SE, dont le coût dépend de E.

Attributs:

- IV - (R&D)
  - MONTANT-IV : coût global (S)
  - SIM : simulation (S).

STANDARD-V

Si L E Ligne et SE E Service Externe alors  $\langle L, SE \rangle$  signifie que les rotations de L pendant toute la simulation nécessitent la prestation du service S.E.

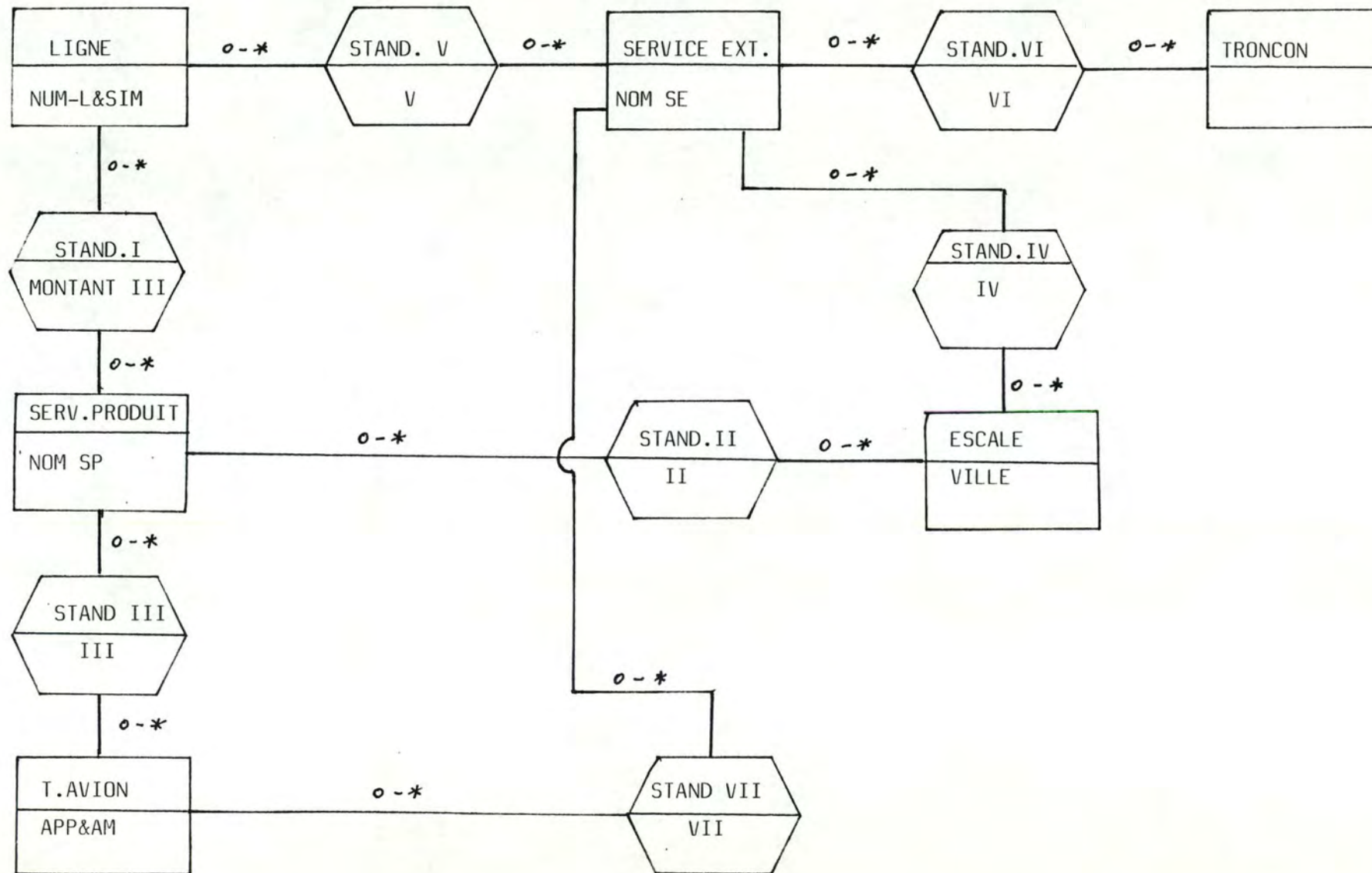
Attributs:

MONTANT V : montant global (S)

STANDARD VI et VII

Même principe que STANDARD IV, mais T.Avion et Tronçon remplacent Escale.

SOUS-SCHEMA COUTS UNITAIRES



BUDGET SP

Si L E LIGNE  
Si SP E Service-produit } alors BUDGET SP <L,SP>

signifie qu'à la ligne L un budget de service SP est alloué.

Attribut:

- Montant (D):

Décomposable selon les 16 natures comptables.

BUDGET SE

Si L E LIGNE  
Si SE E Service Ext. } alors BUDGET SE <L,SE>

signifie qu'à la ligne L un budget de service SE est alloué.

Attribut:

- Montant (S).

BUDGET R

Si L E LIGNE  
Si T E T.Recette } alors BUDGET R <L,T>

signifie que la recette T est allouée à la ligne L.

Attribut:

- Montant (S).

BUDGET MC

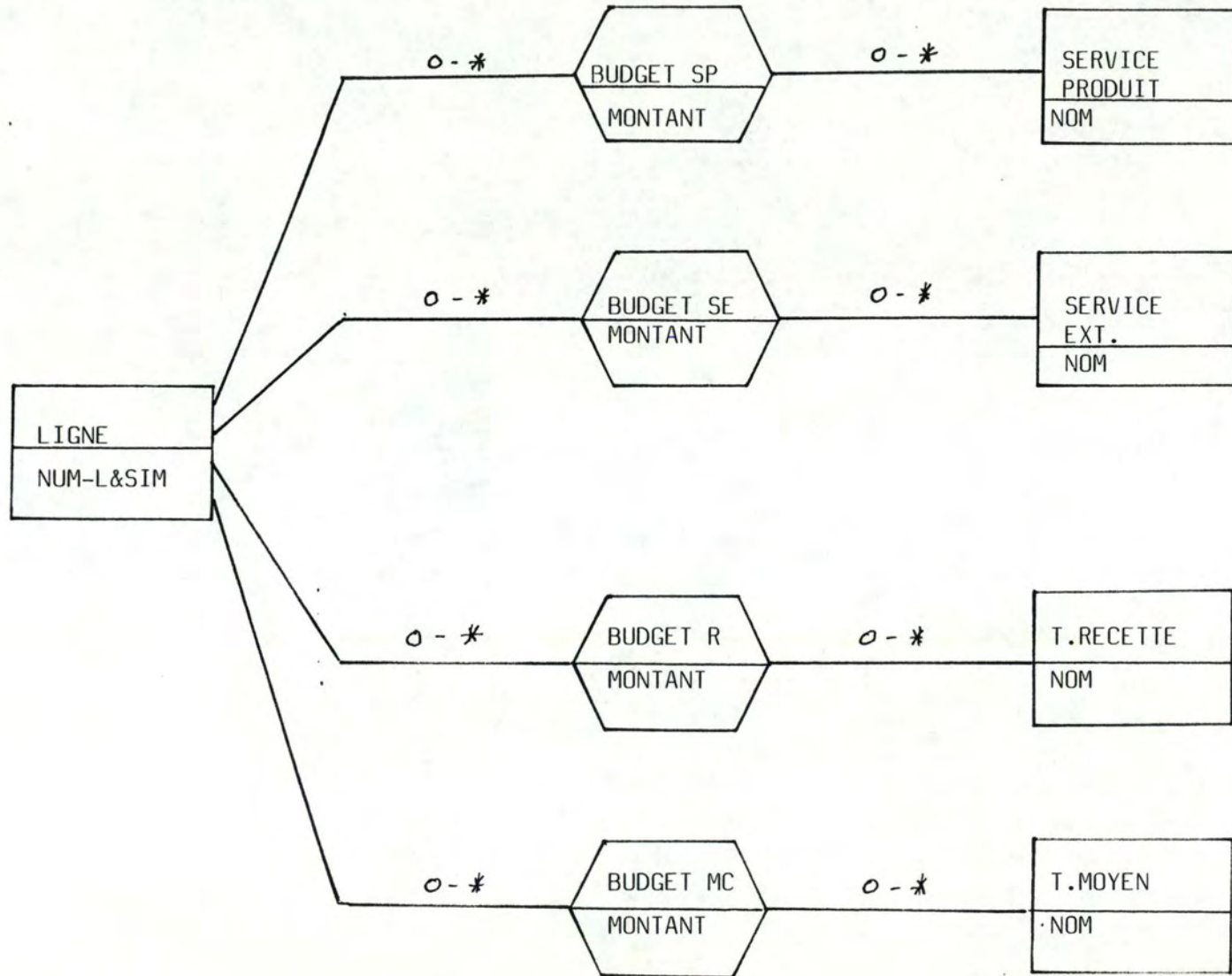
Si T E Type-Moyen-Com }  
Si L E LIGNE } alors BUDGET MC <T,L>

signifie que le type de moyen commercial T est affectable à la ligne L.

Attribut:

- Montant (S).

SOUS-SCHEMA BUDGET



### 7.7. Sous- schéma Equipages

#### T.E.M.

Définition: type de personnel de maîtrise nécessaire pour le vol d'un type d'appareil.

#### Attributs:

- ITEM (D) Composition
  - PILOTE : indique si présence d'un pilote
  - SECOND : indique si présence d'un second
  - FLIGHT ENG: indique si présence d'un flight eng.
  - AUTRE : indique si présence d'un autre  
(navigateur, radio, ...)
- H.PREST (S) : nombre maximum d'heures de prestation pour un vol
- H. MOB (S) : nombre maximum d'heures de mobilisation pour un vol
- REPOS (D ) : repos minimum après un vol

#### Décomposable

- CLASSE : classe de la longueur de vol (S)
- H.REPOS : nombre d'heures de repos en escale associé à la classe (S)
- JOUR-ET : nombre maximum de jours consécutifs passés à l'étranger (S)
- JOUR-BX : nombre minimum de jours consécutifs passés à Bruxelles après une prestation (S)
- COUT-HV : coût à l'heure de vol (S)
- MAX-PRE : maximum d'heures à prester pendant une année (S)
- MIN-PRF : minimum d'heures à prester pendant une année (S)

- Le T.E.M. dépend du type d'appareil et non de son aménagement.
- La composition minimale d'un type d'équipage est constitué d'un pilote et d'un second.

### T.E.C.

Définition: type de personnel de cabine nécessaire pour le vol d'un avion.

#### Attributs:

- ITEC (D)
  - CHEF (S)
  - PREMIER (S)
  - N-STEWARD : nombre de stewards (S)
  - N-HOSTESS : nombre d'hôteses (S)

Les autres attributs sont les mêmes que pour TEMP sauf COMP.

### T. ROUTING PNC

Définition: type de routing qu'un type d'équipage de cabine peut effectuer.

#### Attribut:

- NUM (I)

Remarque: même chose pour équipage de maîtrise, mais pour le type d'entité T. ROUTING.PNM.

### Nécessité - M

Si A E T. Avion }  
 Si TM E TEM } alors Nécessite -M <A, TM >

signifie le type d'équipage de maîtrise TM nécessaire pour l'avion A.

Même type de définition pour Nécessite-C.

ASSOCIE.C

Si R E T.ROUTING PNC }  
 Si T E TEC } alors R,T signifie le

type d'équipage associé au routing R.

Même type de définition pour Associe-M.

CONTIENT - C

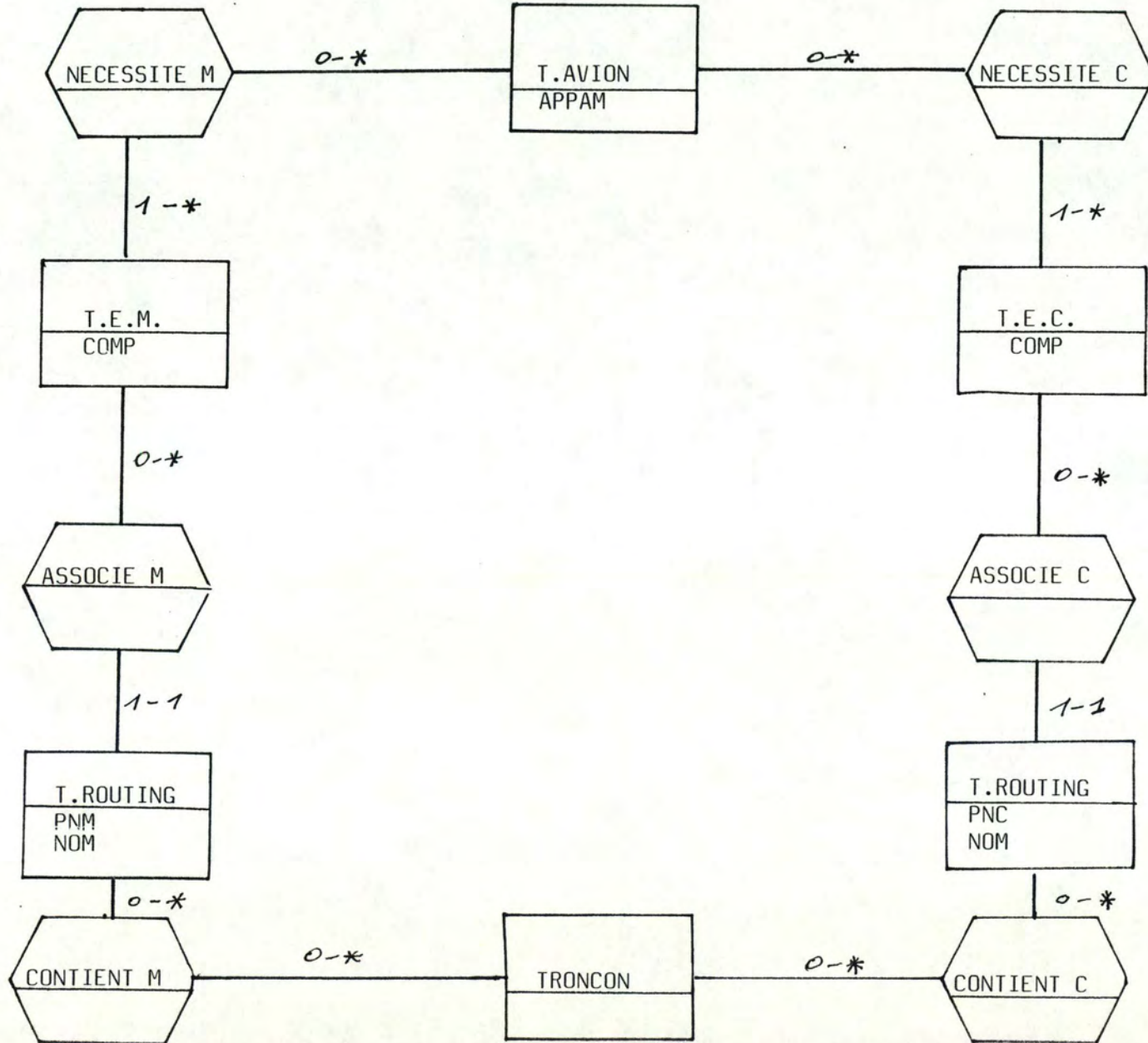
Si R E T.ROUTING PNC }  
 Si T E Tronçon } alors R,T

signifie que le routing R contient le tronçon T.

Attributs:

- Horaire: horaire choisi par le routing pour ce tronçon (D )
- HDL : heure de départ locale (S)
- JD. : jour de départ (S)
- SIM : simulation (S)
- SLIP : Indicateur si à l'issue du tronçon  
il y a un slip crew (S)
- N.C. : indicateur de l'ordre d'exécution des  
tronçons dans un T.Routing (S).

SOUS-SCHEMA DES EQUIPAGES



7.8. Quantifications

X : nombre de simulations

Ligne : 300 . X

Axe : 75 . X

Parcours: 500 . X

Tronçon : 250.

Escale : 120.

Pays : 200.

T.Avion : 40.

Contient P: 1 ligne contient au maximum 6 parcours,  
en moyenne 2.

Compose T : 1 parcours contient au maximum 6,  
en moyenne 2.

A.L. : en moyenne 4 par axe.

Survol : 1000.

Pays : 200.

Effectue : 2000.

Service-produit : 13.

Service-externe : 19.

Standard-I : 900 . X

Standard-II : 240 . X

Standard-III : 320 . X

Standard-IV : 960 . X

Standard-V : 1500 . C

Standard-VI : 500 . X

Relation : 300

T-recette : 5

T. Moyen : 6  
Quota V : 1500 . X  
Quota N : 1800 . X  
R.Réalisable.P: 2500 . X  
Budget-SP : 3900 . X  
Budget-SE : 5700 . X  
Budget-R : 1500 . X  
Budget-M : 1800 . X  
T.E.M. : 40  
T.E.C. : 60  
T.Routing PNC : 5 par T.E.C.  
T.Routing PNM : 5 par T.E.M.  
Necessite-M : 80  
Necessite-C : 120  
Cont-M : 800 . X  
Cont-C : 1120 . X  
42 escales par parcours  
2 lignes par parcours  
30 CONT.M par tronçon  
50 T ROUT.PNM par TEM  
4 CONT.M par T. ROUT.PNM

## VIII- SCHEMA DES ACCES POSSIBLES

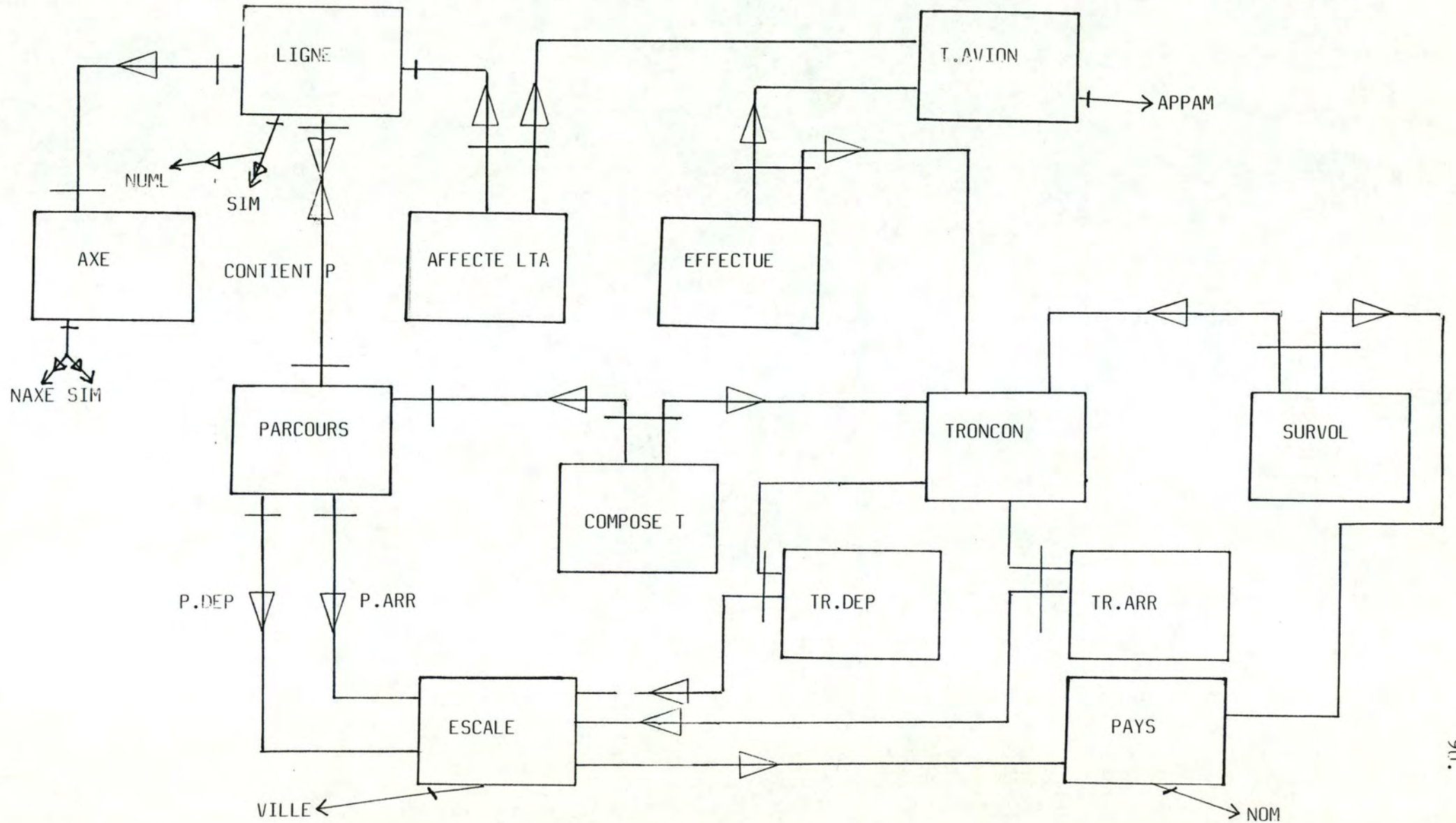
Nous proposons de définir l'implantation des structures de données et des requêtes sur une machine abstraite constituée d'un SGBD du type MAG et d'un processeur de type ADL [3] .

Ce schéma des accès possibles représente une structure de données qui est en fait une traduction directe du schéma conceptuel.

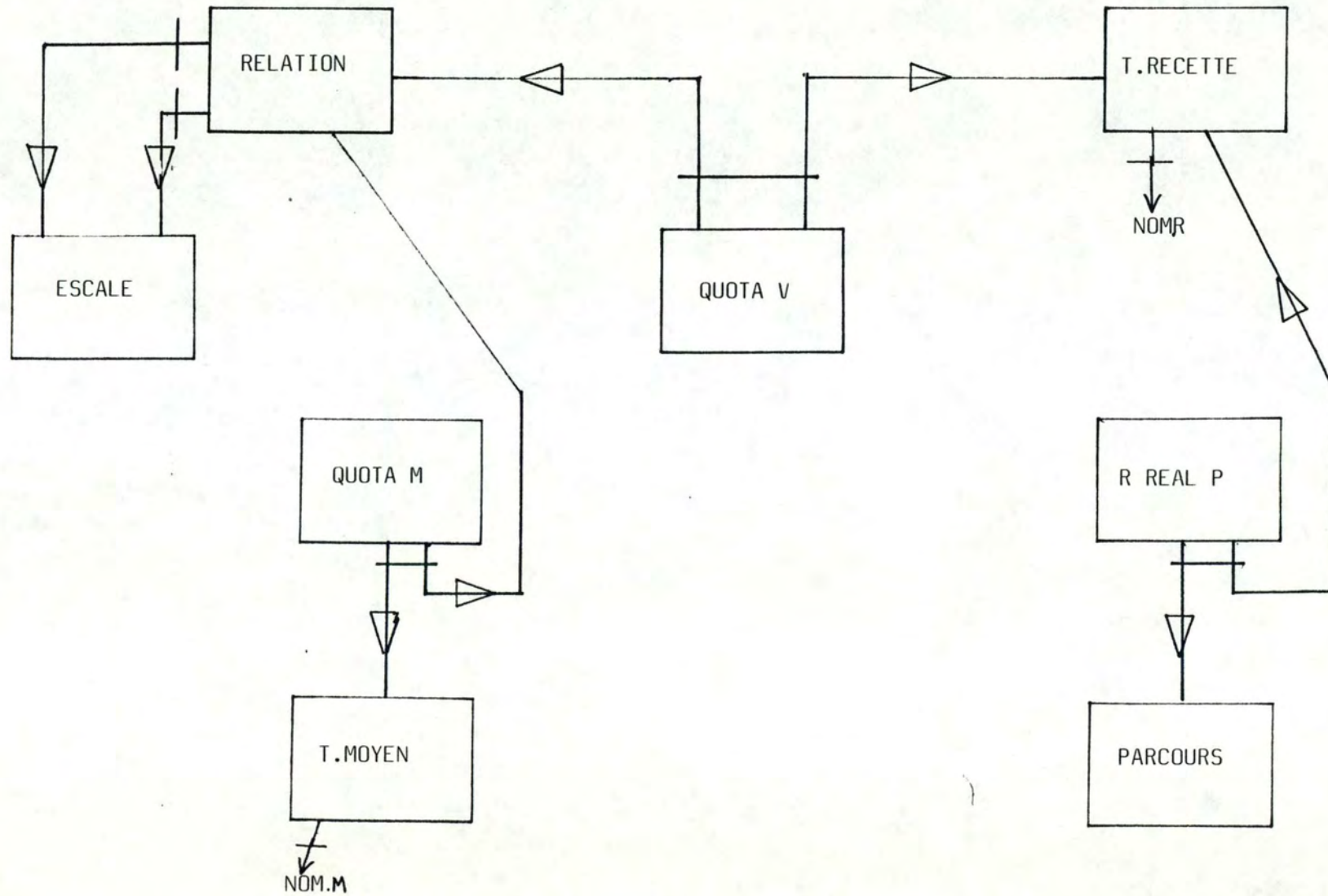
A chacun des sous-schémas du modèle conceptuel correspond un sous-schéma des accès possibles (excepté Quotas et Facteurs de production qui ont été fusionnés).

Afin de ne pas alourdir les sous-schémas, nous ne présentons qu'une version simplifiée, c'est-à-dire que seuls les types d'articles, types de chemin et les items identifiants sont représentés. A la fin du présent chapitre, le lecteur trouvera une liste détaillée de tous les types d'articles et de leurs items.

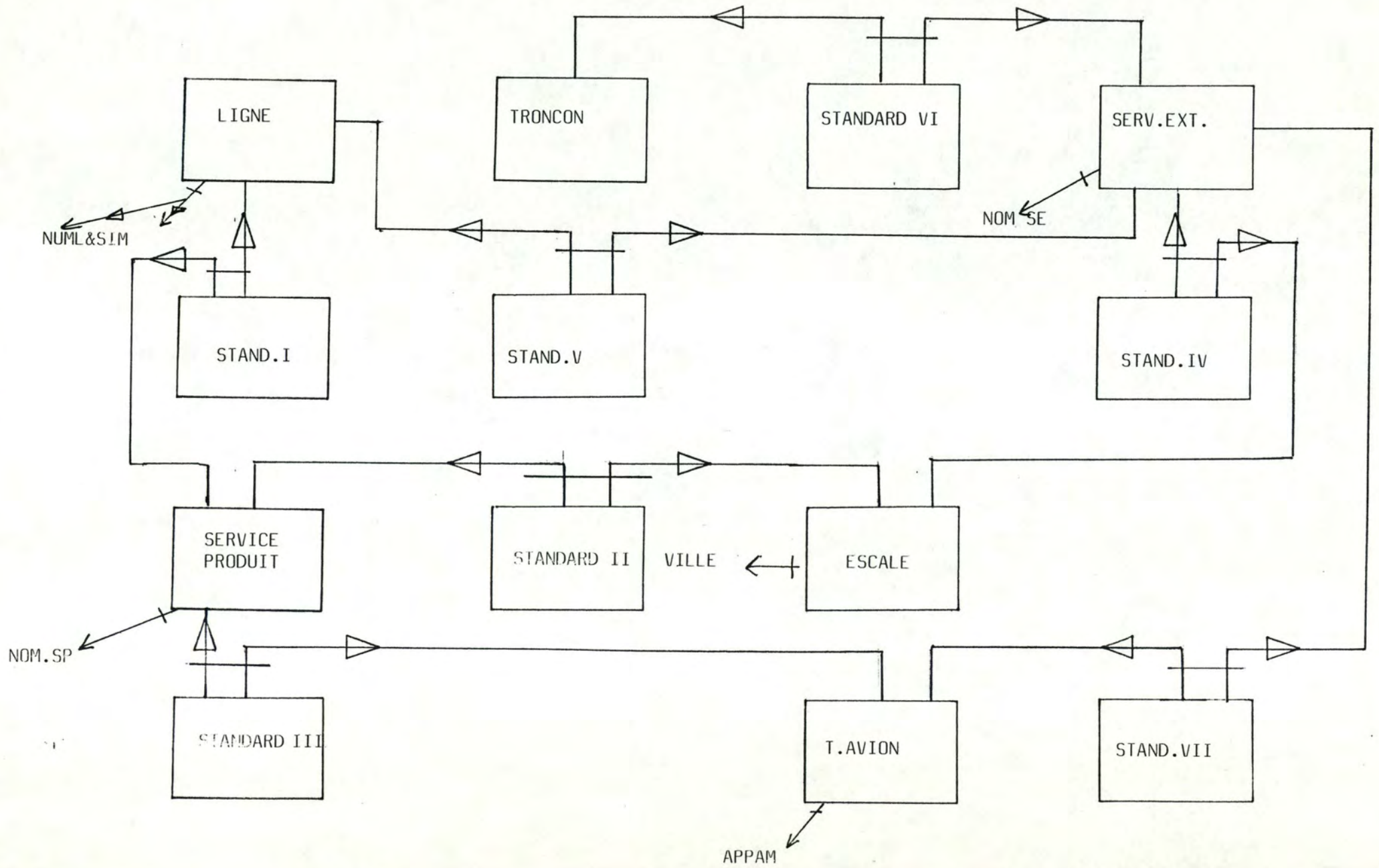
8.1. SOUS-SCHEMA PRODUIT ET MOYEN



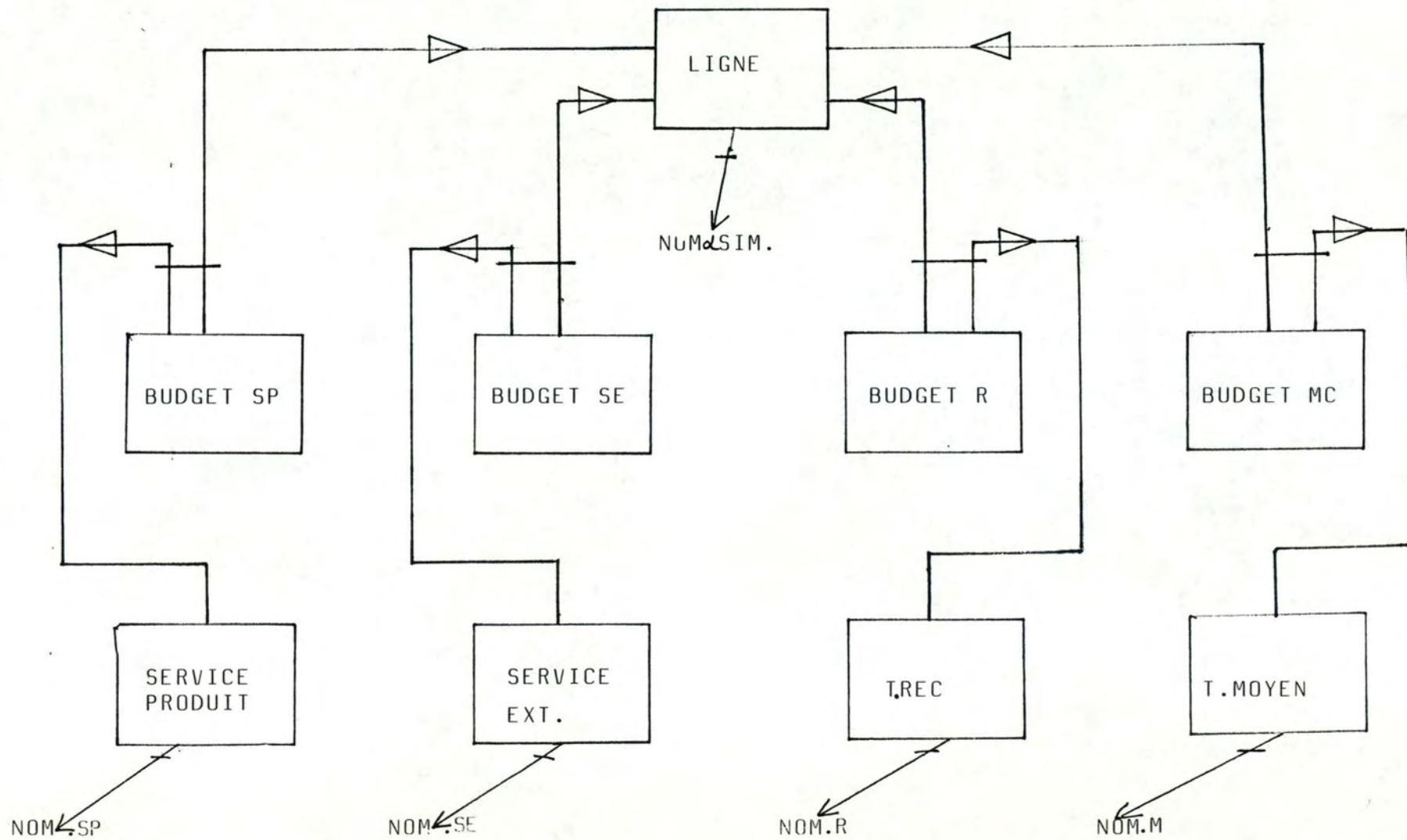
8.2. SOUS-SCHEMA QUOTAS ET SOUS-SCHEMA FACTEURS DE PRODUCTION



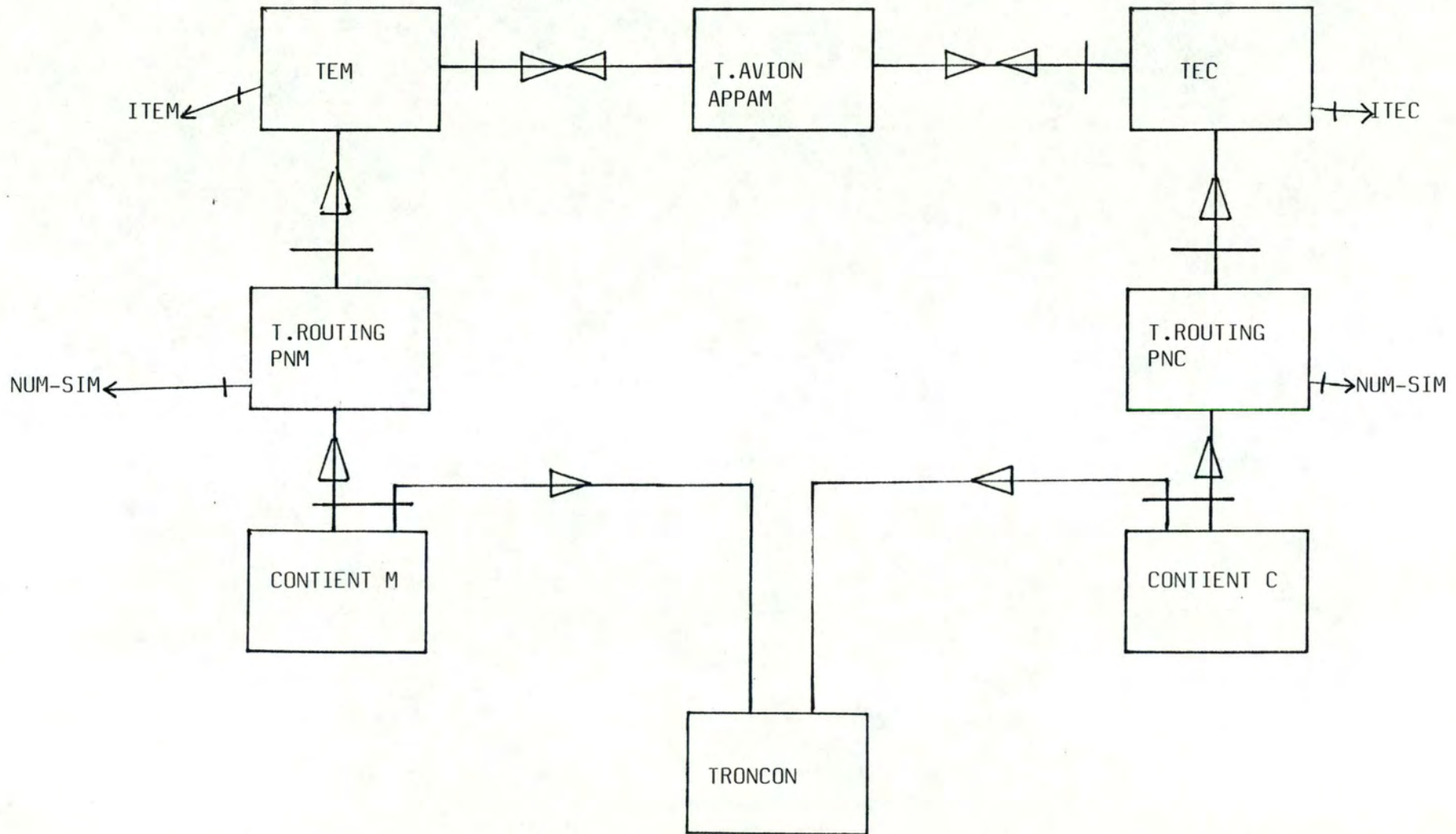
8.3. SOUS-SCHEMA COUTS UNITAIRES

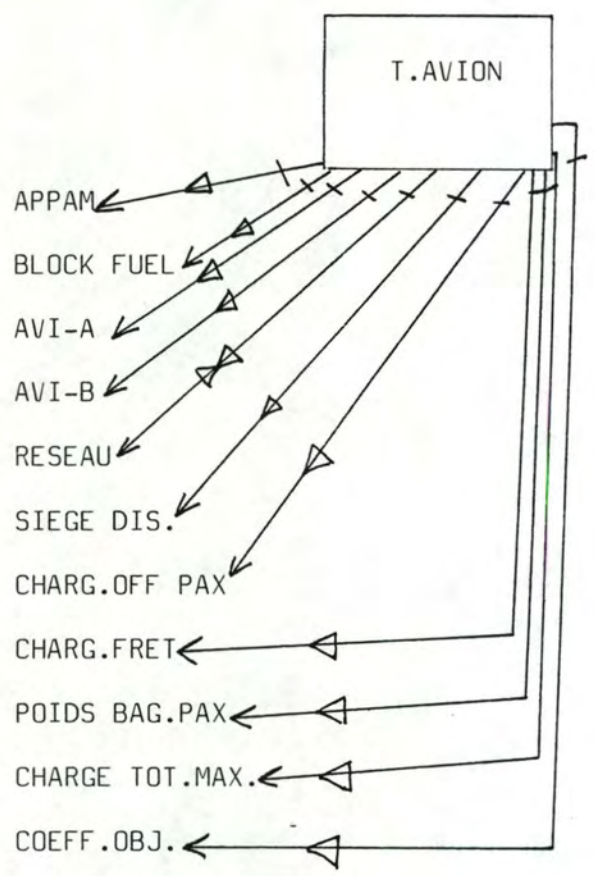
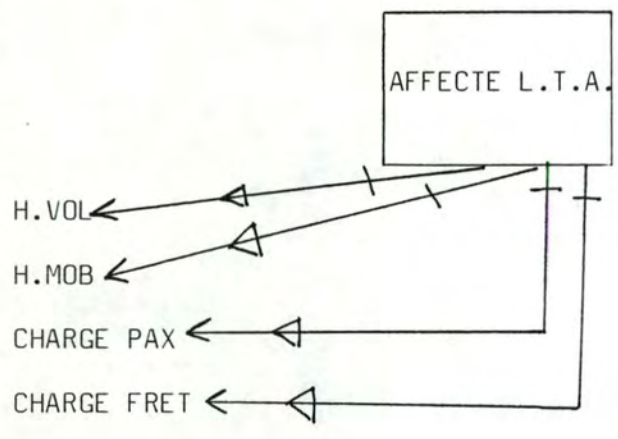
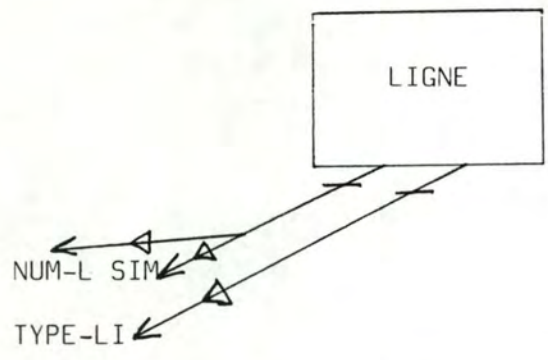


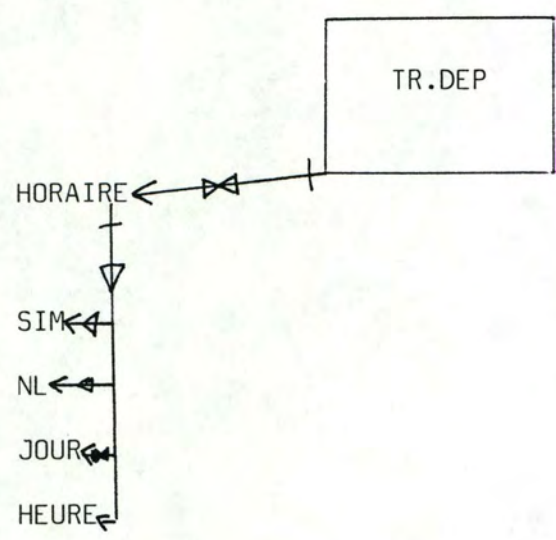
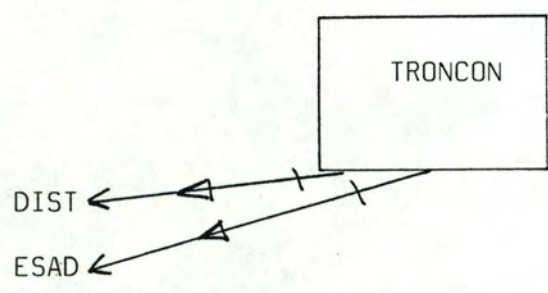
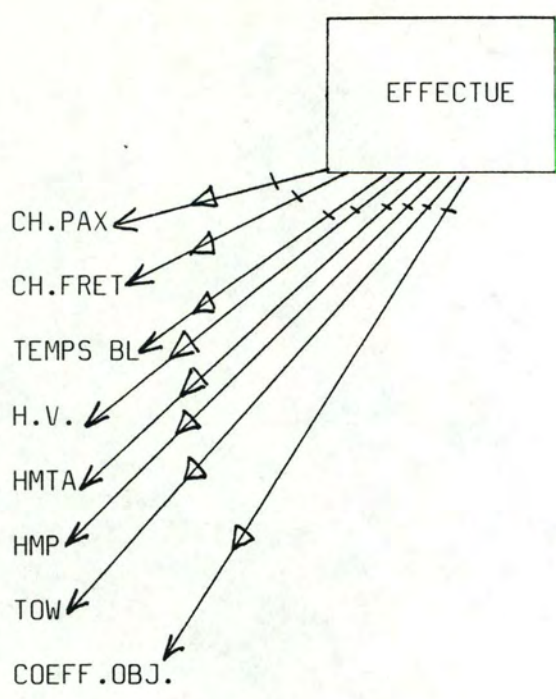
8.4. SOUS-SCHEMA BUDGET

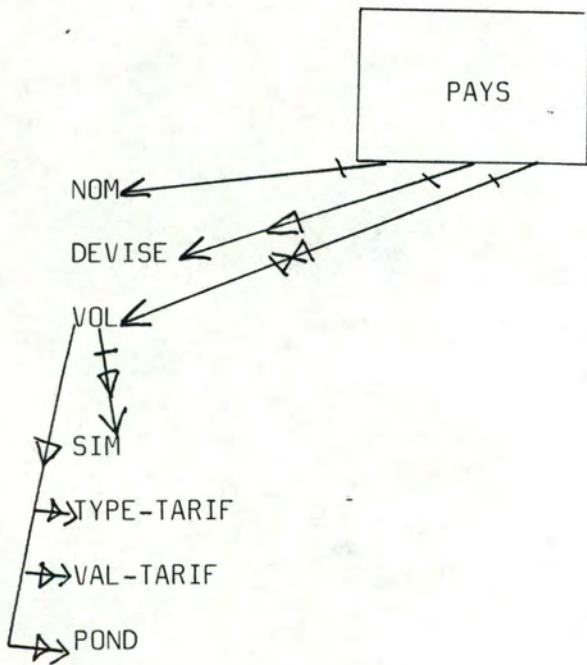
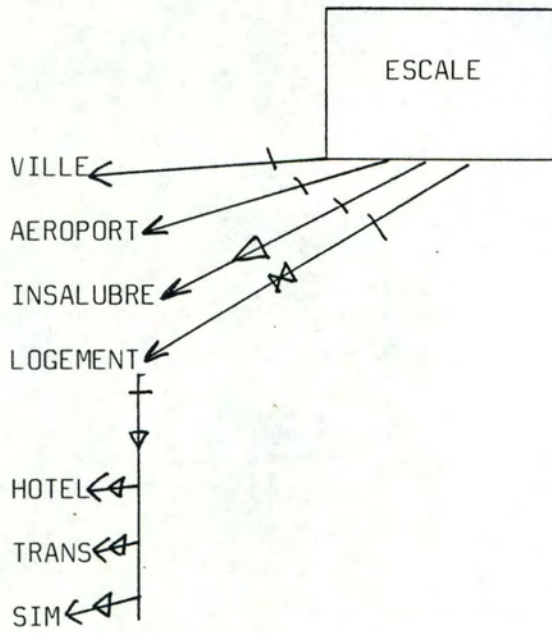
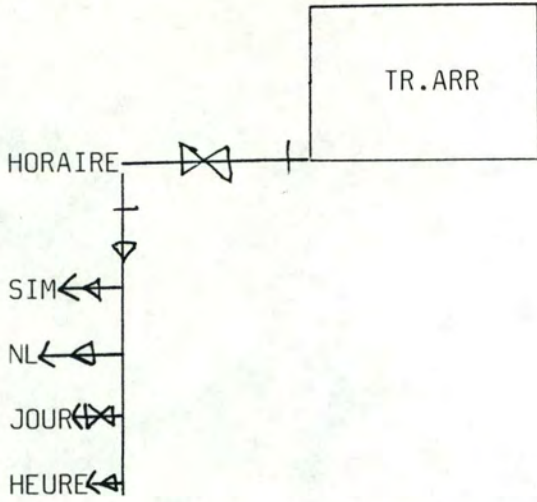


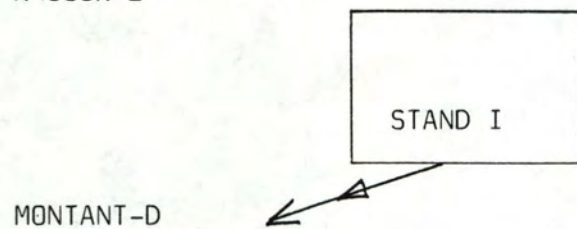
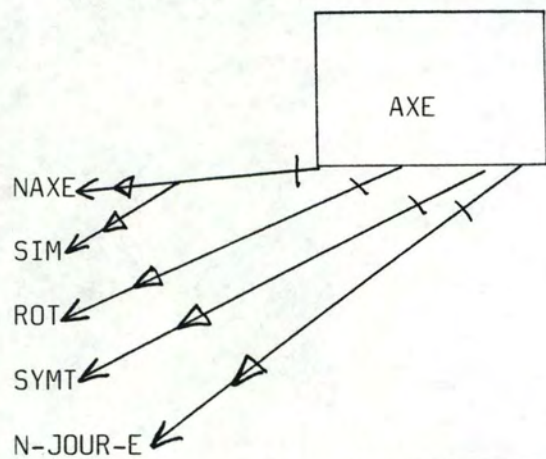
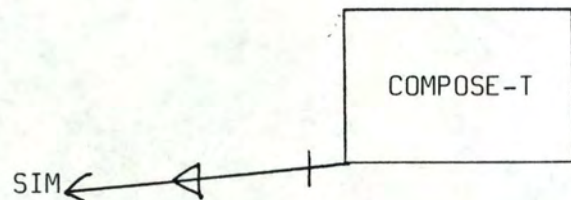
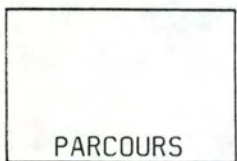
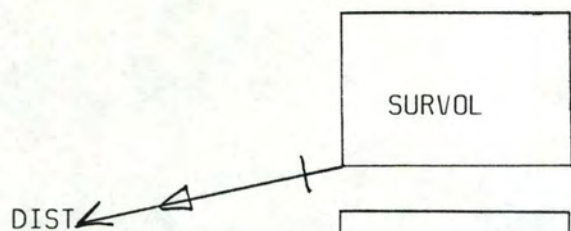
SOUS-SCHEMA EQUIPAGES



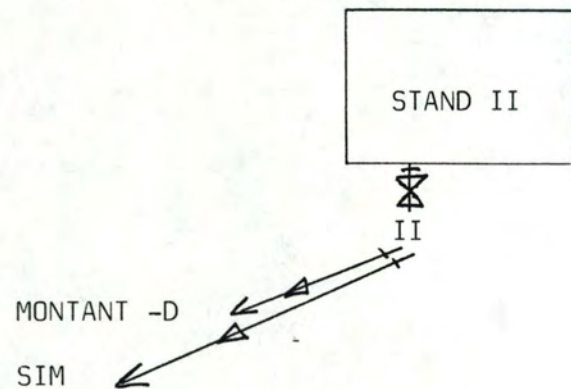


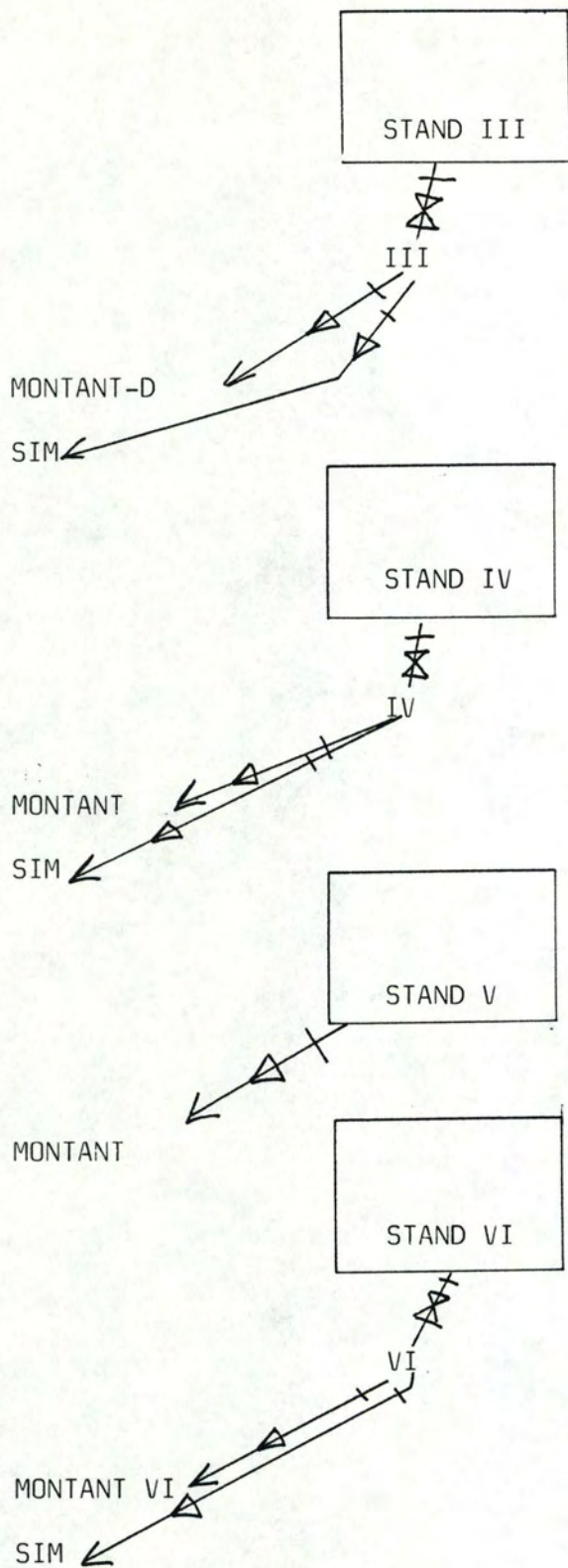


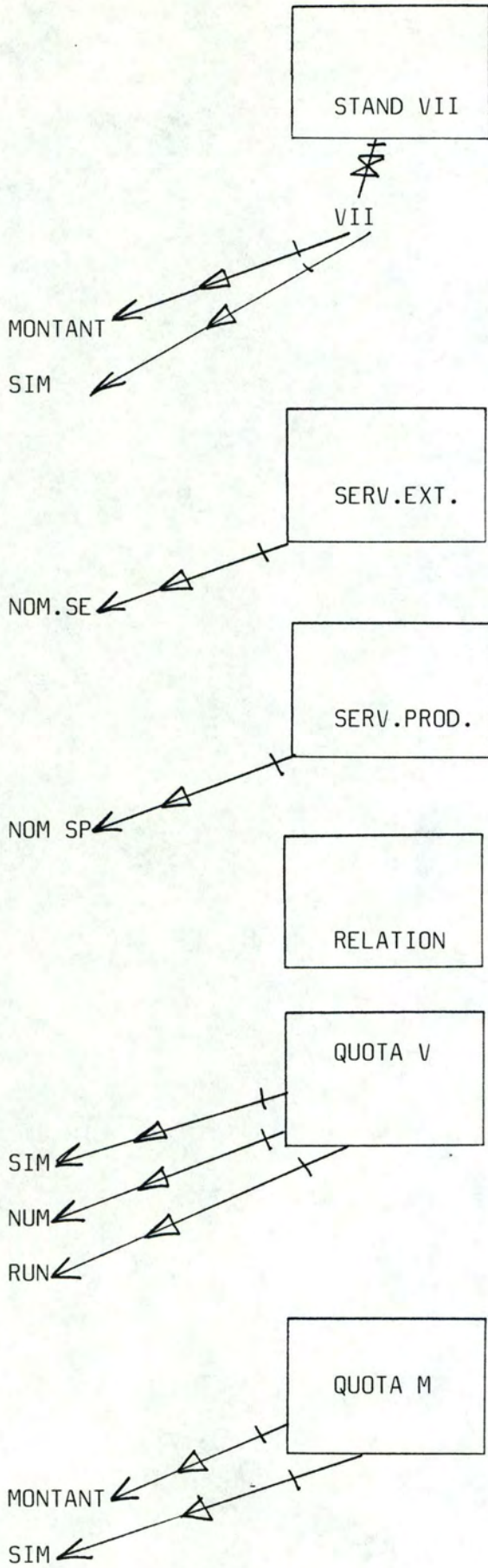


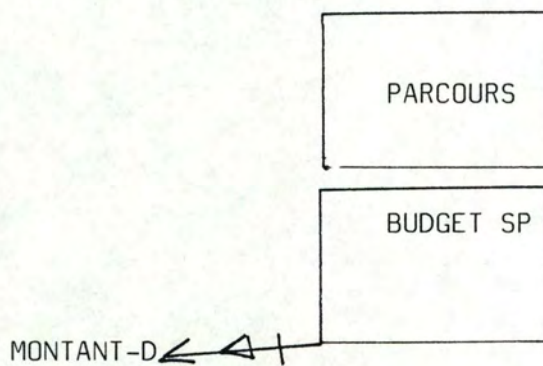
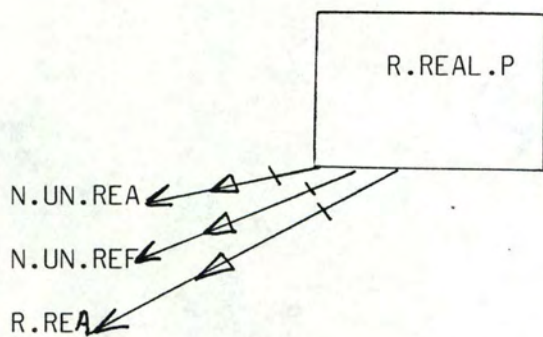
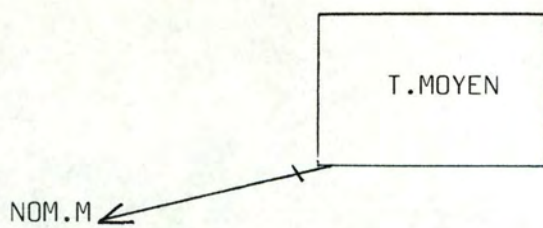


(\* il s'agit des 16 natures comptables\*)

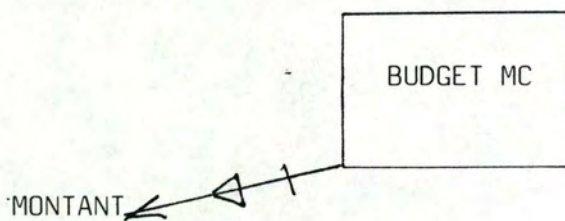
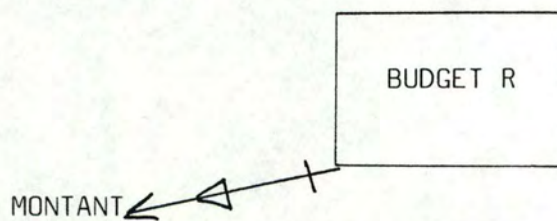
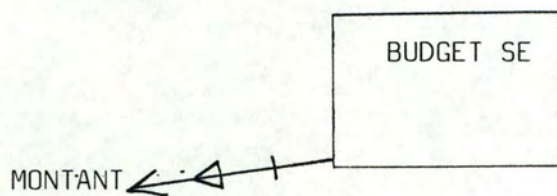


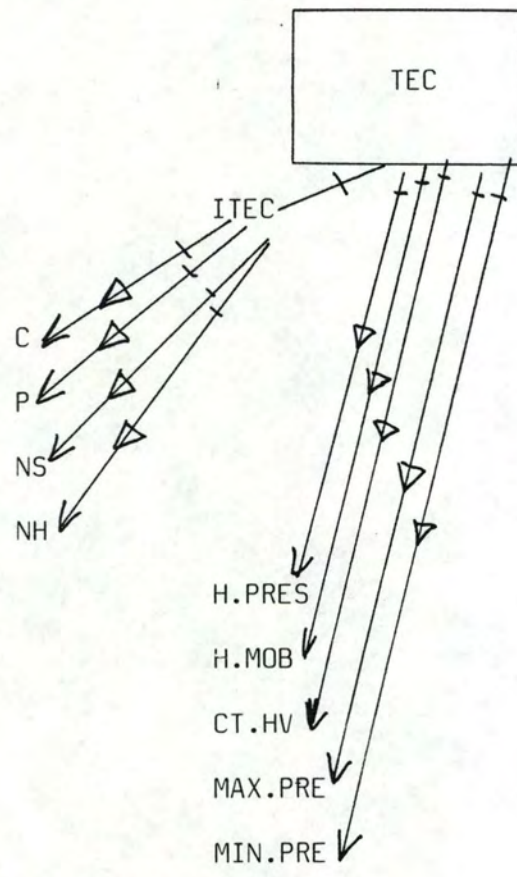
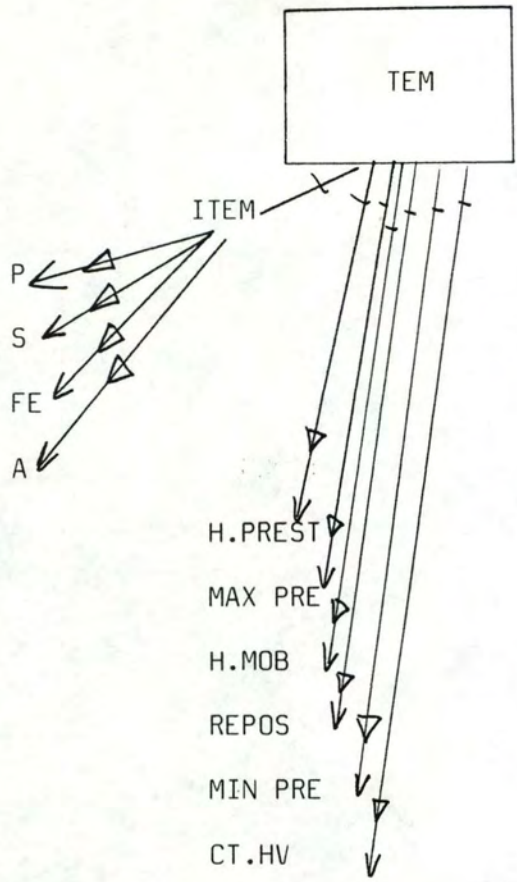


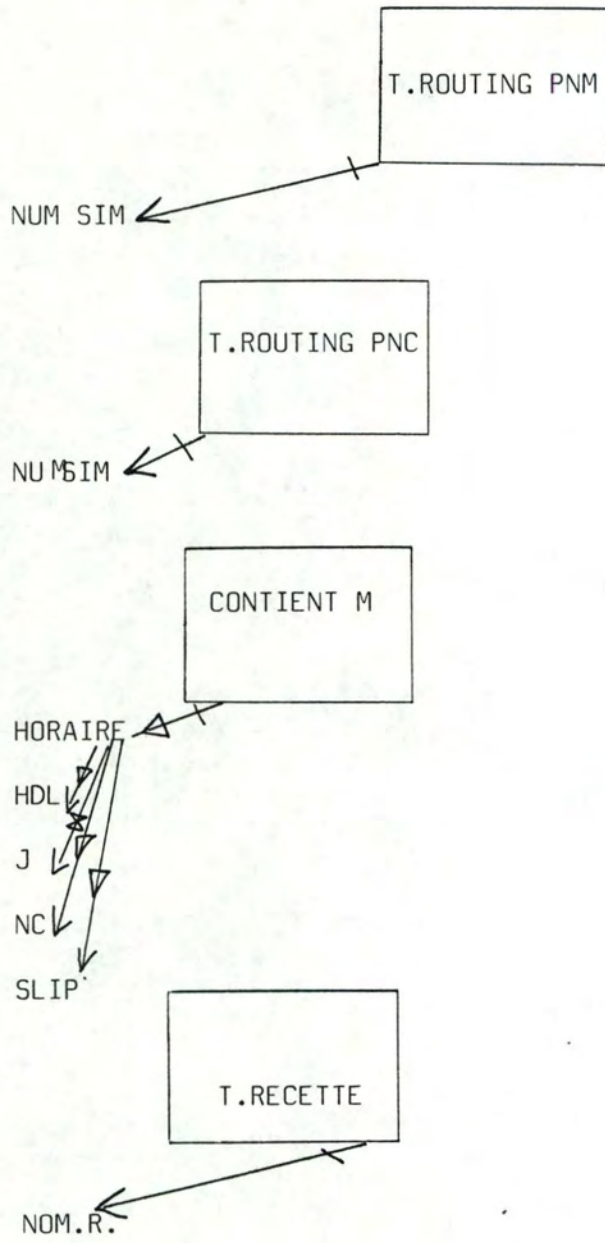




(\*décomposable en 16 natures comptables)







## IX SCHEMA DES ACCES NECESSAIRES

### 9.1. Introduction

Le but de ce chapitre est de déterminer à partir des requêtes des applications les accès nécessaires dans la banque de données. Nous avons regroupé les requêtes par application. Pour chacune des ces requêtes, nous travaillerons en deux étapes. Durant la première étape nous développerons un algorithme prédictif qui décrit l'ensemble des données manipulées, alors que dans l'algorithme effectif, nous tiendrons compte d'une stratégie d'accès.

Nous choisirons la stratégie contenant le moins d'accès logiques à la base. Dans l'exposé qui suit, nous ne reprendrons pas pour chacune des requêtes toutes les stratégies possibles, et ce afin de ne pas alourdir l'exposé. De plus, de par une simple consultation des quantifications, il est souvent aisé de voir que parmi plusieurs chemins, certains ne sont pas performants et qu'il est par conséquent inutile de développer leur algorithme.

De même, afin de ne pas alourdir l'exposé, nous avons regroupé plusieurs requêtes sous deux types de méta-algorithmes que nous exposons au point 9.8.

Afin de développer ces algorithmes nous avons supposé l'existence de quelques procédures que nous définissons ci-dessous:

Input (x,Y) : saisie d'arguments  
 Newlist (x) : création d'une liste  
 Putlist (x,y) : ajouter l'élément y à la liste x  
 Unique list (x): suppression des doubles dans la liste x  
 New-1 x where ((:y= $\alpha$ ) : ajout d'une valeur  
                    $\alpha$  (:z= $\beta$ ))  
                   d'item à l'item répétitif x  
                   décomposable en y et z.

Exemple:



Conventions

for L: = LIGNE ( N= X)\* - L'étoile signifie qu'il s'agit  
d'un accès par clé.

Le calcul des accès se fait sous l'hypothèse de 10 simulations.  
Afin de ne pas alourdir certains algorithmes, nous n'avons pas  
mentionné une procédure de déconcaténation parfois nécessaire  
pour effectuer un test.

Exemple: if SIM(AXE):= SIX

devrait être déconcaténation NAXE&SIM (AXE)

if SIM = SIX, then ...

## 9.2. Constitution des axes

### 1) Créer pour une simulation

- un axe
- les lignes, parcours et tronçons de cet axe
- les avions des lignes

Soit CHAXE

- 1) NAXE & SIM
- 1) NBRE-LIGNES
- 1) LIGNES repeat (NBRE-LIGNES, )
  - 2) NUML & SIM
  - 2) APPAM
- 1) NBRE-PARCOURS
- 1) PCOURS repeat (NBRE-PARCOURS, )
  - 2) ESD
  - 2) ESA
  - 2) NBRE-TRC
  - 2) NUML&SIM
  - 2) TRC repeat NBRE-TRC
    - 3) ESD
    - 3) ESA
  - 2) SIM

### Prédicatif

```
input (CHAXE)

create A := AXE (:NAXE & SIM:=NAXE&SIM (:CHAXE))

for LG := LIGNES (:CHAXE)
  create L:=LIGNE (:NUML&SIM:=NUML&SIM(:LG))
  attach L to A via AL
  create A.L.T.A.:=AFFECTE L.T.A.
  attach L to A.L.T.A.
  for T.A: = T. AVION (:APPAM:=APPAM(:LG))
    attach A.L.T.A. to T.A.
```

```

endfor LG
for PR:=PARCOURS (:CHAXE)
  create P:=PARCOURS
  attach A to P
  for E 1 := ESCALE (:VILLE=ESD (:PR))
    attach P to E1 via P.DEP
  endfor E1
  for E2:= ESCALE (:VILLE=ESA(:PR))
    attach P to E2 via P.ARR.
  endfor E2

for TR:=TRC (:CHAXE)
  create CPT:=COMPOSE T (:SIM=SIM (:PR))
  attach P to CPT
  for T:=TRONCON ((:TR.DEP (:ESCALE (:VILLE=ESD (:TR))))
    α (:TR.ARR(:ESCALE(:VILLE=ESA (:TR))))))
    attach CPT to T
  endfor TR

for L:=LIGNE (:NUML&SIM=NUML&SIM (:PR))
  attach L to P.
endfor

Effectif
input (CHAXE)

create A:=AXE (:NAXE&SIM:=NAXE&SIM (:CHAXE))

for NL:= 1..NBRE-LIGNES
  create L:=LIGNE (NUML&SIM:=NUML&SIM(:LIGNES [NL] (:CHAXE)))
  attach L to A via A.L.
  create A.L.T.A.:= AFECTE L.T.A.
  attach A.L.T.A. to L.
  for T.A.: T.AVION (:APPAM=APPAM(:LIGNE [NL] (:CHAXE)))*

```

```

        attach T.A. to A.L.T.A.
enfor NL

for N.P.: = 1..NBRE-PARCOURS
    create P:=PARCOURS.
    attach A to P.
    for E1:=ESCALE (:VILLE=ESD(:P.COURE [NP] (:CHAXE)))*)
        attach E1 to P. via P.DEP
    endfor EA
    for E2:=ESCALE (:VILLE=ESA (:P.COURE [NP] (:CHAXE)))*)
        attach E2 to P via P.ARR.
    endfor E2.
    for NT:= 1...NBRE-TRC
        create CPT:=COMPOSE.T (:SIM=SIM (:P.COURE [NP] (:CHAXE )))
        attach CPT to P
        for E3:=ESCALE (:VILLE=ESD (:TRC [NT] (:PCOURE [NP] (:CHAXE))))*)
            for TR.D:=TR.DEP (:E3)
                for T:=TRONCON (:TR.D)
                    for TR.A:=TR.ARR (:T)
                        for E4:=ESCALE (:TR.A)
                            if VILLE (:E4=ESA (:TRC [NT] (:PCOURE [NP] (:CHAXE))))
                                then attach T to CPT.
                            endif
                        endfor E4
                    endfor TR.A
                endfor T
            endfor TR.D
        endfor E3
    endfor NT
    for L:=LIGNE (:NUML&SIM=NUML&SIM (:PCOURE [NP] (:CHAXE)))*)
        attach L to P
    endfor L
Endfor.

```

Accès

T.AVION	4
ESCALE	4X( 1+1)+2X(1+3) )
TR.DEP	4 X 3 X 2
TRONCON	4 X 3 X 2
TR.ARR	4 X 3 X 2
LIGNE	4

---

120

2) Pour une ligne d'une simulation créer les horaires  
soit CHHOR

- 1) NBRE-TRC
- 1) TRC repeat (NBRE-TRC, )
  - 2) ESD
  - 2) ESA
  - 2) NUML&SIM
  - 2) HEURE-D
  - 2) JOUR-D (1..7)
  - 2) HEURE-A
  - 2) JOUR-A (1...7)

Prédicatif

input (CHHOR)

for TR:=TRC(:CHHOR)

for TD:=TR.DEP(:ESCALE(:VILLE=ESD(:TR)))

new-1 HD:=HORAIRE(:TD)

where ((:NUML&amp;SIM:=NUML&amp;SIM(:TR))

α(for J:=1.. 7

JOUR (:HD):=JOUR-D [JJ] (:TR))

↙(:HEURE:=HEURE.D (=TR))

end new 1

```
for T.A.: = TR.ARR(:ESCALE(:VILLE=ESA(:TRC)))
```

```
new-1 HA:=HORAIRE(:TH)
```

```
where ((:NUML&SIM:=NUML&SIM(:TR))
```

```
  ( for J:=1..7
```

```
    JOUR:=JOUR.A [J] (:TR))
```

```
  ( :HEURE:=HEURE A (:TR))
```

```
end new 1
```

```
endfor
```

Effectif

```
input (CHHOR)
```

```
for NT:=1.. NBRE-TRC
```

```
for E1:=ESCALE(:VILLE=ESD(:TRC [NT](:CHHOR))))*
```

```
for T.R.D.: = TR.DEP (:E1)
```

```
for T:= TRONCON (:T.R.D.)
```

```
for E2:=ESCALE (:T)
```

```
if VILLE (:E2)=ESA(:TRC [NT](:CHHOR))
```

```
then new-1 HD:=HORAIRE (:T.RD) where
```

```
((:NUML&SIM:=NUML&SIM (:TRC [NT](:CHHOR)))
```

```
(HEURE:=HEURE.D(:TRC [NT](:CHHOR)))
```

```
(for J:=1..7
```

```
  JOUR:=JOUR [J] (:TRC [NT] (:CHHOR))))
```

```
end new 1 HD
```

```
new-1 HA:=HORAIRE (:T.RA) where
```

```
( : NUML&SIM:=NUML&SIM(:TRC [NT](:CHHOR)
```

```
(HEURE:=HEURE.A (:TRC [NT](:CHHOR)))
```

```
(for J:=1..7
```

```
  JOUR:=JOUR A [J] (:TRC [NT](:CHHOR))))
```

```
end new 1 HD
```

```
endif
```

```
endfor
```

ESCALE : (1 + 3 ) X 4  
 TR.DEP : (3) X 4  
 TRONCON : (3) X 4  
 TR.ARR. : (3) X 4

---

52

### 9.3. Quotas

3) Créer pour une relation et une simulation les quota-vente  
 soit CHQUOT

- 1) ESD
- 1) ESA
- 1) NBRE-QVX
- 1) QVX repeat (NBRE-QVX, )
  - 2) SIM
  - 2) NUM
  - 2) RUN
  - 2) NOM.R.

#### Prédicatif

Input (CHQUOT)

```
for R:=RELATION ((RD:ESCALE(:VILLE=ESD(:CHQUOT)))
                 &(RA:ESCALE(:VILLE=ESA(:CHQUOT))))
```

```
for QVZ:=QVX(:CHQUOT)
```

```
create QV:=QUOTA.V((:SIM:=SIM(:QVX))
```

```
                  &(:NUM=NUM(:QVX))
```

```
                  &(:RUN:=RUN(:QVX))
```

```
attach R to QV
```

```
for TR:=T.RECETTE(:NOM.R=NOM.R(QVX))
```

```
attach QV to TR
```

```
endfor
```

Effectif

input (CHQUOT)

for E1:=ESCALE(:VILLE=ESD(:CHQUOT))\*

for R:=RELATION(RD:E1)

for E2:=ESCALE(RA:R)

if VILLE(:E2)=ESA(:CHQUOT)

then for NQ:=1..nbre-QVX

create QV:=QUOTA.V ((:SIM:=SIM(:QVX [NQ] (:CHQUOT)))

&amp; (:NUM:=NUM(:QVX [NQ] (:CHQUOT)))

&amp; (:RUN:=RUN(:QVX [NQ] (:CHQUOT))))

attach QV to R

for TR:=T.RECETTE(:NOM.R=NOM.R (:QVX [NQ] (:CHQUOT)))

attach TR to QV.

endif

endfor

Accès

ESCALE 1+3

RELATION 3

T.RECETTE 5 \_\_\_\_\_

12

4) Pour une relation et une simulation, créer les  
quota moyens.

Soit CHQMO

1) ESD

1) ESA

1) NBRE-QMO

1) QMO-repeat (NBRE-QMO, )

2) SIM

2) NUM

2) NOM.M.

Prédicatif

```

input (CHQMO)

for R:=RELATION ((RD:ESCALE(:VILLE=ESD(:CHQMO)))
                &(RA:ESCLAE(:VILLE:ESA(:CHQMO))))

  for QMZ:=QMO(:CHQMO)
  create QM:=QUOTA.M((:SIM:=SIM(:QMZ))
                    &(:NUM:=NUM(:QMZ)))

  attach R to QM

  for T.M:=T.MOYEN(:NOM=NOM.M(:QMZ))
  attach QM to T.M.

endfor

```

Effectif

```

input (CHQMO)

for E1:=ESCALE(:VILLE=ESD(:CHQMO))*
  for R:=RELATION(RD:E1)
    for E2:=ESCALE(RA:R)
      if VILLE(:E2)=ESA(:CHQMO)
        then for NQ:=1..NBRE-QMO
          create QM:=QUOTA.M((SIM:=SIM(QMO [NQ])(:CHQMO)))
                            &(NUM :=NUM (QMO [NQ])(:CHQMO)))
          attach QM to R
          for T.M:=T.MOYEN(:NOM.M=NOM.M(:QMO [NQ])(:CHQMO))
          attach TM to QM

        endif
      endif
    endif
  endif
endfor

```

Accès

```

ESCALE 1+3
RELATION 3
T.MOYEN 6
13

```

9.4. Harmonisation

5) Pour une relation R donner la Recette Z pour la simulation SIX.

Prédicatif

```
input (ESD, ESA, Z, SIX)
for QV:=QUOTA.V(:RELATION((RD:ESCALE(:VILLE=ESD))
                        &(RA:ESCALE(:VILLE=ESA))))
  if SIM(:QV)=SIX
    then for T.R.=T.RECETTE(:QV)
      if NOM.R (:T.R.)=Z
        then print NUM(:QV)
              RUN(:QV)
        endif
      endif
    endfor
```

Effectif

```
input (ESD, ESA, Z, SIX)
for E1:=ESCALE(:VILLE=ESD)*
  for R:=RELATION (RD:E1)
    for E2:=ESCALE(RA:R)
      if VILLE(:E2)=ESA
        then
          for Q.V:=QUOTA.V(:R)
            if SIM(:QV)=SIX
              then for TR:=T.RECETTE(:QV)
                if NOM.R(:TR)=Z
                  then print NUM (:QV)
                          RUN(:QV)
                  endif
                endif
              endfor
            endif
          endfor
        endif
      endfor
    endfor
  endfor
```

endfor

Accès

ESCALE 1 + 3

RELATION 3

QUOTA.V 50

T.RECETTE 5

---

62

6) Pour une relation d'une simulation donner les  
moyens commerciaux

Prédicatif

Input (ESD, ESA, SIX)

for QM:=QUOTA.M(:RELATION (( RD:ESCALE(:VILLE=ESD))  
&(RA:ESCALE(:VILLE=ESA))))

if SIM(:Q)=SIX then

print N°U(:Q)

NOM(:T.MOYEN(:Q))

endif

endfor

Effectif

Input (ESD, ESA, SIX)

for E1:=ESCALE(:VILLE=ESD)\*

for R :=RELATION(RD:E1)

for E2:=ESCALE(RA:R)

if VILLE(:E2)=ESA then

for QM:=QUOTA.M(:R)

if SIM(:QM)=SIX

then print NUM (:QM)

for TM:=T.MOYEN(QM)

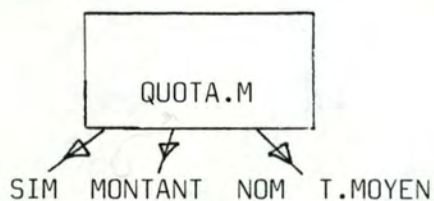
```

print NOM.M(:TM)
endfor
endif
endfor

```

#### Accès

ESCALE	1 + 3
RELATION	3
QUOTA.M	60
T.MOYEN	6
	<hr/>
	73



Si nous introduisons une redondance en mettant le nom du type de moyen dans QUOTA.M, nous gagnons 6 accès. Toutefois nous n'opterons pas pour cette redondance car nous ne trouvons pas que le gain d'accès soit considérable.

7) Pour une simulation donnée, donner tous les axes ayant un parcours reliant les escales x et y.

#### Prédicatif

```

for A:=AXE(AL:LIGNE(CONTIENT P:PARCOURS(P.DEP:ESCALE(:VILLE=X)
&(P.ARR:ESCALE(:VILLE=Y)
if SIM(:A)=SIX then print N°AXE(:A)
endif
end for A

```

Effectif

Input (X,Y,SIX)

Newlist (Alist)

for E1:=ESCALE(VILLE=X)\*

for P:=PARCOURS(P.DEP:E1)

for E2:=ESCALE(P.ARR:P)

if VILLE(:E2)=Y then for L:=LIGNE(CONTIENT:P)

if SIM(:L)=SIX then

for A:=AXE(AL:L)

Putlist (Alist, A)

endfor

endif

endif

endfor

unique (Alist)

for A:=Alist print N°AXE (:A)

endfor

Quantifications

ESCALE 1 + 42

PARCOURS 42 (1 escalepointe en moyenne sur 42 parcours

  sauf si Bruxelles  $\Rightarrow$  =1000 parcours)

LIGNE 10 X 2 X 2 = 40 (ESD-ESA couvre en moyenne 2 parcours

par simulation et 1 parcours pointe sur 2 lignes)

AXE 4

129

Un passage direct de parcours à axe pourrait nous faire gagner des accès.

En effet nous aurions l'algorithme suivant:

117.

```
Input (x,y,six)
for E1:=ESCALE(:VILLE=x)*
  for P:=PARCOURS(P.DEP:E1)
    for E2:=ESCALE(P.ARR:P)
      if VILLE(:E2)=y then for A:=AXE(:P)
        if SIM(:A)=SIX then
          print N°AXE(:A)
        endif
      endif
    endfor
  endfor
```

8) Pour 1 Axe d'une simulation, donner tous les parcours.

```
Input (.AXEX & SIX)
for P:PARCOURS(:AXE(:N°AXE & SIM=AXEX & SIX))
  print VILLE(:ESCALE(P.DEP))
  VILLE(:ESCALE(P.ARR))
endfor
```

Effectif

```
for A:=AXE(:N°AXE&SIM=AXEX&SIX)*
  for P:=PARCOURS(:A)
    for E1:=ESCALE(P.DEP:P)
      forE2: ESCALE(P.ARR:P)
        print VILLE(:E1)
        VILLE(:E2)
      endfor
    endfor
```

AXE = 1

PARCOURS = 4

ESCALE = 8

ESCALE	1 + 42
PARCOURS	42
AXE	<u>20</u>
	105

Cette requête étant souvent formulée, nous allons rajouter un chemin à notre schéma des accès possibles.

9) Pour une ligne d'une simulation, donner les parcours.

#### Prédicatif

Input (NUMX & SIX)

for P:=PARCOURS(CONTIENT P:LIGNE (:NUML & SIM = NUMX & SIX))

    print VILLE(:ESCALE(P.DEP:P))

        VILLE(:ESCALE(P.ARR:P))

endfor

#### Effectif

for L:=LIGNE(:NUML&SIM=NUMX&SIX)\*

    for P:=PARCOURS(CONTIENT P:L)

        for E1:=ESCALE(P.DEP:P)

            print VILLE(:E1)

        endfor E1

        for E2:=ESCALE(P.ARR:P)

            print VILLE(:E2)

        endfor E2

endfor

#### Accès

LIGNE	1
-------	---

PARCOURS	2
----------	---

ESCALE	<u>4</u>
--------	----------

	7
--	---

10) Donner tous les axes d'une simulation ayant ESD comme  
 escale de déoart d'un parcours.

Prédicatif

```
input (ESD, SIX)
for A:= AXE(:PARCOURS(P.DEP:ESCALE(:VILLE:ESD)))
  if SIM(:A)=SIX
    then print NUMX (:A)
  endfor.
```

Effectif

```
input (ESD, SIX)
newlist (LISTA)
for E:=ESCALE(:VILLE=ESD)*
  for P:=PARCOURS(P.DEP:E)
    for A:=AXE(:P)
      if SIM(:A)=SIX
        then putlist (LISTA,A)
      endif
    endfor
  endfor
```

Unique (LISTA)

for A:=LISTA

print NAXE(:A)

endfor

Accès

ESCALE        1

PARCOURS     42

AXE           42

85

11) Pour un axe donné d'une simulation, donner la charge  
 offerte par tronçon de toutes les lignes de cet axe.

#### Prédicatif

```

newlist (Tlist)
input (AXEX&SIX)
for T:=TRONCON (:COMPOSE T(:PARCOURS(CONTIENT P:LIGNE(A L:AXE
  2->(:NAXE&SIM=AXEX&SIX))))))

  putlist (Tlist, T)
for T.A:=T.AVION(:AFFECTE L.T.A.(:LIGNE(AL:AXE(:NAXE&SIM=AXEX&SIX))))
  for TR:=TLIST
    for E:=EFFECTUE(:TA) from TR
      print CH.PAX (:E)
      CH.FRET(:E)
      VILLE(:ESCALE(:TR.DEP(:TR)))
      VILLE(:ESCALE(:TR.ARR(:TR)))
      NUML(:LIGNE(AL:AXE(:NAXESIM=AXEX&SIX)))
    endfor T.A.
  soit TRL
    1) ESD
    1) ESA
    1)NUML
  
```

#### Effectif

```

newlist (Tlist)
input (AXEX&SIX)
for A:=AXE(:NAXE&SIM=AXEX&SIX)*
  for L:=LIGNE(:A)
    for P:=PARCOURS(:L)
      for C T:=COMPOSE T (:P)
    
```

```

for T:=TRONCON (:CT)
  for TD:=TR.DEP(:T)
    for E1:=ESCALE(TD)
      ESD(:TRL):=VILLE(:E1)
    endfor TD
  for TA:=TR.ARR(:T)
    for E2:=ESCALE(:TA)
      ESA(:TRL):=VILLE(:E2)
    endfor TA
  NUML(:TRL)=NUML(:L)
  putlist (TLIST, TRL)
endifor P
uniquelist (TLIST) (*la procédure tient compte de
  ESD&ESA&NUML*)
for A.L.T.::=AFFECTE L.T.A.(:L)
  for T.A:=T.AVION(:A.L.T.)
    for EF:=EFFECTUE(:T.A.)
      for T:=TRONCON(:EF)
        for TRD:=TR.DEP(:T)
          for E3:=ESCALE(:TRD)
            for TRL:=TLIST
              if VILLE(:E3)=ESD(:TRL)
                then for TRA:=TR.ARR(T)
                  for E4:=ESCALE(:TRA)
                    if VILLE(:E4)=ESA(:TRL)
                      then print VILLE (:E4)
                                VILLE (:E3)
                                CH PAX(:EF)
                                CH FRE(:EF)
                  endfor E4
                endfor TRA
              endfor TRL
            endfor E3
          endfor TRD
        endfor T
      endfor EF
    endfor T.A.
  endfor A.L.T.
endifor.

```

Accès

AXE	1
LIGNE	4
PARCOURS	2 X 4
COMPOSE T	2 X 2 X 4
TRONCON	(2 X 2 X 4) + (4 X 150)
ESCALE	(2 X 2 X 4) X 2 + (4 X 150) X 2
TR.ARR	(2 X 2 X 4) + (4 X 150)
TR.DEP	(2 X 2 X 4) + (4 X 150)
AFFECTE L.T.A.	4
T.AVION	4
EFFECTUE	4 X 150

---

3717

12) Donner pour un axe et une simulation le nombre d'heures de vol et mobilisation et la charge offerte par ligne.

Prédicatif

```
input (AXEX&SIX)
for AL:=AFFECTE L.T.A.(:LIGNE (:AXE (:NAXE&SIM=AXEX&SIX)))
  print NUM.L (:AL)
  H.VOL (:AL)
  H.MOB (:AL)
  CHARGE PAX (:AL)
  CHARGE FRET (:AL)
endfor.
```

Effectif

```
input (AXEX&SIX)
for A:=AXE(:N°AXE&SIM=AXEX&SIX)*
  for L:LIGNE(AL:A)
    print NOM.L (:L)
```

```

for AL:=AFFECTE LTA (:L)
print H.VOL (:AL)
      H.MOB (:AL)
      CHARGE PAX (:AL)
      CHARGE FRET (:AL)

endfor.

```

Accès

AXE 1

LIGNE 4

---

5

13) Pour une ligne et 1 simulation donner le routing  
et l'horaire.

Prédicatif

input (NUMX&SIX)

for T:=TRONCON(:COMPOSE T(:PARCOURS(CONTIENT P:LIGNE(:NUML&SIM=NUMX&SIX))))

for HD:=HORAIRE(:TR.DEP(:T))

if SIM&NL(:HD)=SIX&NUMX

then print VILLE(:ESCALE(:TR.DEP(:T)))

print HEURE (:HD)

for J:=JOUR(:HD)

print J.

endfor J

endif

endfor H.D.

for H.A.:HORAIRE(:TR.ARR(:T))

if SIM&NL (:H.A)=SIX&NUMS

then print VILLE(:ESCALE(:TR.ARR(:T)))

print HEURE(:HA)

```

        for J:=JOUR(:HA)
        print J
        endfor J

    endif

endfor H.A.

endif

Effectif

newlist (Tlist)

input (NUMX&SIX)

for L:LIGNE (:NUML&SIM=NUMX&SIX)
    for P:=PARCOURS(CONTIENT P:L)
        for C.T:=COMPOSE T (:P)
            for T:=TRONCON (:CT)
                for TRD:=TR.DEP(:T)
                    for E1:= ESCALE(:TRD)
                        ESD(:TRL):=VILLE(:E1)
                    endfor E2

                    for HD:=HORAIRE(:TRD)
                        if NL&SIM(:HD)=NUMX&SIX
                            then HEURE D(:TRL):HEURE(:HD)
                                for J:=JOUR (:HD)
                                    for NJ:=1..7
                                        JTD[NJ](:TRL)=J
                                    endfor J
                                endif
                            endfor HD
                        endfor TRD
                    endfor T
                endfor C.T
            endfor P
        endfor L
    endfor LIGNE
endfor H.A.

```

```

for H.A:=HORAIRE(:TR.A)
  if NL&SIM (:HA)=NUMX&SIX
    then HEURE A(:TRL):=HEURE (:HA)
      for J:=JOUR(:HA)
        for NJ:=1..7
          JTA [NJ] (:TRL)=5
        endfor J
      endif
    endfor TRA
  putlist (Tlist, TRL)
endfor L.
uniquelist (Tlist)
for T:=Tlist
  print ESD (:T)
  for JD:=JTD(:T)
    print JD
  endfor JD
  print HEURE (:T)
  PRINT ESA (:T)
  for JA:=JTA(:T)
    print JA
  endfor JA

```

Accès

LIGNE	1
PARCOURS	2
COMPOSE T	2 X 2
TRONCON	2 X 2
TR.DEP	2 X 2
ESCALE	(2 X 2) X 2
TR.ARR	<u>2 X 2</u>

TRL

1) ESD
1) ESA
1) JTD [1..7]
1) JTA [1..7]
1) HEURE.A
1) HEURE.D.

14) Pour les parcours d'un axe, créer les recettes réalisables par parcours pour 1 simulation.

Soit CHARM

- 1) AXEX & SIX
- 1) NBRE-PARC
- 1) PARC repeat (NBRE-PARC, )
  - 2) ESD
  - 2) ESA
  - 2) NBRE-REC
  - 2) REC repeat (NBRE-REC, )
    - 3) SIX
    - 3) N.UN.REA
    - 3) N.UN.REF
    - 3) R.REA
    - 3) NOM.R

Prédicatif

input (CHARM)

for PRC:=PARC (:CHARM)

for P:=PARCOURS((P.DEP:ESCALE(:VILLE=ESD(:PRC)))

&(P.ARR:ESCALE(:VILLE=ESA(:PRC))))

if NAXE&SIM (:AXE(:P))=ATEX&SIX(:CHARM)

then for RR:=REC(: PRC)

create R:=R.REAL.P((N.UN.REA:=N.UN.REA(:RR)

&(N.UN.REF:=N.UN.REF(:RR)

&(R.REA:=R.REA(:RR))

attach P to R

for T:=T.RECETTE(:NOM.R=NOM.R(:RR))

attach R to T

endfor PRC

Effectif 1

input (CHARM)

for A:=AXE(:NAXE&amp;SIM=AXEX&amp;SIX(:CHARM))\*

for NP:=1..NBRE-PARC

for P:=PARCOURS(:A)

for E1:=ESCALE(P.DEP:P)

if VILLE(:E1)=ESD(PARC [NP] (:CHARM))

then for E2:=ESCALE (P.ARR:P)

if VILLE(:E2)=ESA(PARC [NP] (:CHARM))

then for NR:1..NBRE-REC

                create R:=R.REAL.P((N.UN.REA:=N.UN.REA(:REC [NR]  
  (:PARC [NP] (:CHARM))))))  &(N.UN.REF:=N.UN.REF(:REC [NR]  
  (:PARC [NP] (:CHARM))))))  &(R.REA:=REA(:REC [NR]  
  (:PARC [NP] (:CHARM))))))

attach R to P

                for T:=T.RECETTE(:NOM.R=NOM.R(:REC [NR]  
  (:PARC [NP] (:CHARM))))\*)

attach T to R.

endif

endif

endifor

endfor

Accès

AXE           1

PARCOURS     4 X 4

ESCALE       4 X (1 + 1) X 4

T.RECETTE    1 X 5

Effectif 2

input (CHARM)

for N.P:=1..NBRE-PARC

for E1:=ESCALE(:VILLE=ESD (:PARC [NP](:CHARM)))\*

for P:PARCOURS(P.DEP:E1)

for E2:=ESCALE(P.ARR:P)

if VILLE(:E2)=ESA(:PARC [NP](:CHARM))

then for A:=AXE(:P)

if NAXE&amp;SIM(:A)=AXE&amp;SIX(:CHARM)

then for N.R:=1..NBRE-REC

create R:=R.REAL.P((N.UN.REA:=N.UN.REA (:REC [NR]  
(:PARC [NP] (:CHARM))))&(N.UN.REF:=N.UN.REF (:REC [NR]  
(:PARC [NP] (:CHARM))))&(R.REA:=R.REA (:REC [NR]  
(:PARC [NP] (:CHARM))))

attach R to P

for T:=T.RECETTE(:NOM.R=NOM.R(:REC [NR]  
(:PARC [N.P] (:CHARM))))\*

attach T to R

endif

endif

endfor

Accès

ESCALE (1 + 42 ) . 4

PARCOURS (42) . 4

AXE (42) . 4

T.RECETTE (5) . 4

528.

Nous choisissons le premier algorithme.

15) Pour un tronçon ayant comme escales, X et Y, et donner les pays survolés (NOM + VOL).

Prédicatif

```
for T:=TRONCON ((:TR DEP (:ESCALE(:VILLE=X)))
                &(:TR ARR (:ESCALE(:VILLE=y))))
  print NOM (:PAYS (:SURVOL (:T)
  for V:=VOL (:PAYS (:SURVOL (:T)
    print V
  endfor V
endfor
```

Effectif

```
for E1= ESCALE (:VILLE=X)*
  for TRD: TR.DEP (E1)
    for T:=TRONCON(:TRD)
      for TR.AR :=TRARR(:T)
        for E2:=ESCALE(:TR.AR)
          if VILLE(:E2)=y then
            for S:=SURVOL(:T)
              for P:=PAYS (:S)
                print NOM(:P)
                DEVISE (:P)
                for V:=VOL (:P) print V
              endfor
            endfor
          endfor
        endfor
      endfor
    endfor
  endfor
```

Accès

ESCALE	1 + 3
TRONCON	3
SURVOL	10
PAYS	<u>10</u>

### 9.5. Standards

16) Pour une ligne et un service produit, créer les stand 1.

Soit (CHSTI)

- 1) NUML & SIM
- 1) NBRE - STAND
- 1) STAND repeat (NBRE-STAND, )
  - 2) MONTANT-D
  - 2) NOM-SP

#### Prédicatif

```
input (CHSTI)
for ST:=STAND(:CHSTI)
  create S:=STAND I (:MONTANT-D:MONTANT-D (:ST))
  attach S to LIGNE (:NUML & SIM = NUML & SIM(:CHSTI))
  attach S to SERVICE PRODUIT (:NUM SP=NOM SP (:ST))
endfor.
```

#### Effectif

```
input (CHSTI)
for NS:=1..NBRE-STAND
  create S:STAND I (:MONTANT-D:=MONTANT-D (:STAND [NS](:CHSTI)))
  for L:=LIGNE (:NUML&SIM=NUML&SIM(:CHSTI))*
    attach S to L
  endfor L
  for SP:=SERVICE PRODUIT (:NOM.SP=NOM.SP (:STAND [NS](:CHSTI)))
endfor
```

#### Accès

```
LIGNE          1
SERVICE PRODUIT 3
```

17) Pour une escale donnée, créer le standard II des services produits.

Soit CHST II

- 1) ES
- 1) NBRE-STAND
- 1) STAND repeat (NBRE-STAND, )
  - 2) MONTANT-D
  - 2) SIM
  - 2) NOM-SP

Prédicatif

input (CHST II)

for ST:=STAND (:CHST II)

for S:=STAND II ((:ESCALE(:VILLE=ESD (:CHST II)))

&(:SERVICE.PRODUIT(:NOM.SP=(:ST)))

new-i A:=II(:S) where (MONTANT(:A):=MONTANT-D (:ST)

&(SIM(:A):=SIM(:ST))

endfor

Effectif

input (CHST II)

for NS:=1..NBRE-STAND

for E:=ESCALE(:VILLE=ES (:CHST II))\*

for S:=STAND II(:E)

for SP:=SERVICE PRODUIT (:S)

if NOM.SP (:SP)=NOM.SP [NS](:STAND (:CHST II))

then new-i A := II

where (MONTANT(:A):=MONTANT-D [NS](:STAND(:CHST II))

&(SIM(:A):=SIM [NS](:STAND (:CHST II)))

endif

endfor

Accès

ESCALE	1
STAND II	2 X 2
SERVICE PRODUIT	2 X 2
	<hr/>
	9

C'est ce type d'algorithme que nous utiliserons aussi pour les autres standards.

9.6. Rentabilisté

18) Pour une ligne donnée et une simulation donnée, donner les recettes passagers réalisables par parcours ainsi que les refus et les escales départs et arrivées.

Prédicatif

```
input (NLX & SIX)
for R.R.:=R. REAL.(:PARCOURS(CONTIENT P:LIGNE (:NUML&SIM=NLX&SIX)))
  print N.UN.REA (:RR)
      N.UN.REF (:RR)
      R.REA (:RR)
      NOM (:T.RECETTE (:RR))
      VILLE (:ESCALE (P.DEP:PARCOURS (:RR)))
      VILLE (:ESCALE (P.ARR:PARCOURS (:RR)))
endfor
```

Effectif

```
input (NLX & SIX)
for L:=LIGNE (:NUM-L&SIM = NLX&SIX)*
  for P:=PARCOURS (CONTIENT P:L)
    for E1:=ESCALE (P.DEP:P)
      print VILLE (:E1)
    endfor E1
```

```

for E2:=ESCALE (P.ARR:P)
    print VILLE (:E2)
endfor E2
for RR:=R.REAL.P (:P)
    for T.R:=T.RECETTE(:RR)
        print NOM(:TR)
            N.UN.REA(:RR)
            N.UN.REF(:RR)
            R.REA (:RR)
    endfor T.R
endfor RR
endfor P
endfor L

```

Accès

LIGNE	1
PARCOURS	2
ESCALE	2 X 2
R.REAL.P	2 X 5
	<hr/>
	17

19) Donner pour 1 ligne STANDARD I et son nom.

Prédicatif

```

input (NUMX&SIX)
for SI:=STAND.I (:LIGNE(:NUML&SIM=NUMX&SIX))
    print MONTANT (:SI)
        NOM-SP (:SERVICE PRODUIT (:SI))
endfor

```

Effectif

```

for L:=LIGNE(:NUML&SIM=NUMX&SIX)*
  for SI:=STANDARD I (:L)
    print MONTANT-D(:SI)
    for SP:=SERVICE PRODUIT(:SI)
      print NOM SP(:SP)
    endfor
  endfor

```

Accès

```

LIGNE          1
STAND.I        3
SERVICE PRODUIT 3
                      
                7

```

20) Donner les standards II d'une simulation pour l'escale ESD.

Prédicatif

```

input (ESD;SIX)
for S:=STAND II (:ESCALE(:VILLE=ESD))
  if SIM (:II(:S))=SIX
    then print MONTANT-DII(:II(:S))
           NOM SP (:SERVICE PRODUIT(:S))
  endfor
endfor

```

Effectif

```

input (ESD, SIX)
for E:=ESCALE (:VILLE=ESD)*
  for S:=STAND II (:E)
    for A:=II (:S)
      if SIM(:A)=SIX
        then print MONTANT-D (:DA)
      endif
    endfor
  endfor
endfor

```

```

endfor A
  for SP:=SERVICE PRODUIT (:S)
    print NOM SP (:SP)
  endfor

```

#### Accès

ESCALE	1
STAND.II	2
SERVICE PRODUIT	2
	<hr/>
	5

Le même type d'algorithme sera utilisé pour les autres standards.

#### 9.7. Equipages

21) Pour une escale, une simulation, un type d'équipage de maîtrise donnés, donner les tronçons ayant cette escale comme départ ainsi que les lignes contenant ces tronçons.

#### Prédicatif

```

input (ESD,SIX, ITEX)
for T.R=T.ROUT.PNM((:SIM=SIX)
  &(:TEM(:ITEM=ITEX))
  &(:CONT.M(:TRONCON(:TR.DEP(:ESCALE(:VILLE=ESD))))))
  for HD:=HORAIRE ((:TR.DEP(:TRONCON(:CONTIENT.M(:T.R))))
    &(:SIM=SIX))
    print H.D
      ESD
    endfor HD
  for H.A:=HORAIRE ((:SIM=SIX)
    &(:TR.HRR(:TRONCON(:CONTIENT.M(:T.R))))
    print H.A.
  endfor H.A.
endfor T.R

```

```

print VILLE (:ESCALE(:TR.ARR(:H.A)))

endfor

Effectif

input (ESD,SIX,IEX)

for E1:=ESCALE(:VILLE=ESD)*
  for TRD:=TR.DEP(:E1)
    for T:=TRONCON (:TRD)
      for CM:=CONTIENT.M (:T)
        for TR:=T.ROUTING.PNM(:CM)
          if SIM (:TR)=SIX
            then for TM:=TEM(:TR)
              if ITEM(:TM)=ITEX
                then for H.D:=HORAIRE (:TRD)
                  if SIM(:HD)=SIX
                    then print ESD
                           H.D
                  endif
                endfor H.D
              for TRA:=TR.ARR(:T)
                for H.A:=HORAIRE(:TRA)
                  if SIM(:HA)=SIX
                    then print H.A
                           for E2:=ESCALE(:TRA)
                               print VILLE (:E2)
                           endfor
                  endif
                endfor
              endif
            endfor
          endif
        endfor
      endfor
    endfor
  endfor
endfor.

Accès

ESCALE      1+ 3
TR.DEP      3
TRONCON     3

```

CONTIENT M 3 X 30  
 T.ROUTING PNM 3 X 30  
 TEM 3 X 3  
 TR.ARR 3 (au maximum)  
 202

Effectif 2

```

input (ESD,SIX,ITEX)
for TM:=TEM(:ITEM=ITEX)*
  for TR:=T.ROUTING.PNM(:TM)
    if SIM(:TR)=SIX
      then for CM:=CONTIENT.M (:TR)
        for T:=TRONCON (:CM)
          for TRD:=TR.DEP (:T)
            for E1:=ESCALE(:TRD)
              if VILLE (:E1)=ESD
                then for HD:=HORAIRE (:TRD)
                  if SIM (:HD)=SIX
                    then print HD
                      ESD
                  endif
                endfor
              for TRA:=TR.ARR (:T)
                for H.A:=HORAIRE (:TRA)
                  if SIM(:HA)=SIX
                    then print H.D
                  endif
                for E2:=ESCALE (:TRA)
                  print VILLE (:E2)
                endfor
              endfor
            endfor
          endfor
        endfor
      endfor
    endfor
  endfor
endfor

```

Accès

TEM	1
T.ROUTING PNM	1 X 50
CONTIENT.M	1 X 5 X 4
TRONCON	1 X 5 X 4
TR.DEP	1 X 5 X 4
ESCALE	1 X 5 X 4 + 3
TR.ARR	1 X 3

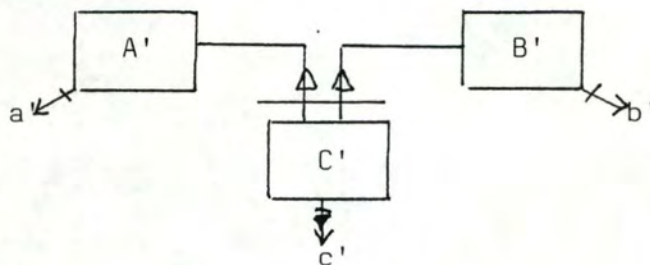
---

 137

Nous choisissons le deuxième algorithme.  
Même type d'algorithme pour le personnel de cabine.

9.8. Métaalgorithmes

Soit A', B', C', 3 types d'article et a', b', c', leurs items respectifs:



1) Premier type

Créer C avec c comme valeur d'item de c' et C occurrence de C'.

soit CHARGE

- 1) a
- 1) NBRE-C
- 1) C repeat (NBRE-C)
  - 2) c
  - 2) b

Prédicatif

```

input (CHARGE)
for CR:=C (:CHARGE)
    create CX:=C'(:c':=c(:CR))
    attach CR to A'(:a'=a(:CHARGE))
    attach CR to B'(:b'=b(:CR))
endfor

```

Effectif

```

input (CHARGE)
for A:=A'(:a'=a(:CHARGE))
    for N:=1..NBRE-C
        create CX:=C'(:c':=c(:C [N](:CHARGE)))
        attach C X to A
        for B:=B'(:b'=b(:C [N] (:CHARGE)))
            attach B to CX
        endfor
    endfor
endfor

```

## 2) Deuxième type

Donner pour une occurrence de A' les occurrences de C' et B'.

Prédicatif

```

input (a)
for C:= C'(:A'(a'=a))
    print c'(:C)
    print b'(:B'(:C))
endfor

```

Effectif

```

input (a)
for A:=A'(:a'=a)

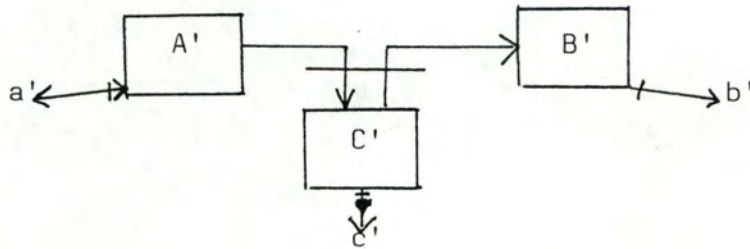
```

```

for C:=C'(:A)
  print c'(:C)
  for B:=B'(:C)
    print b'(:B)
  endfor.

```

Accès nécessaires



Nous retrouvons ces algorithmes pour les données suivantes:

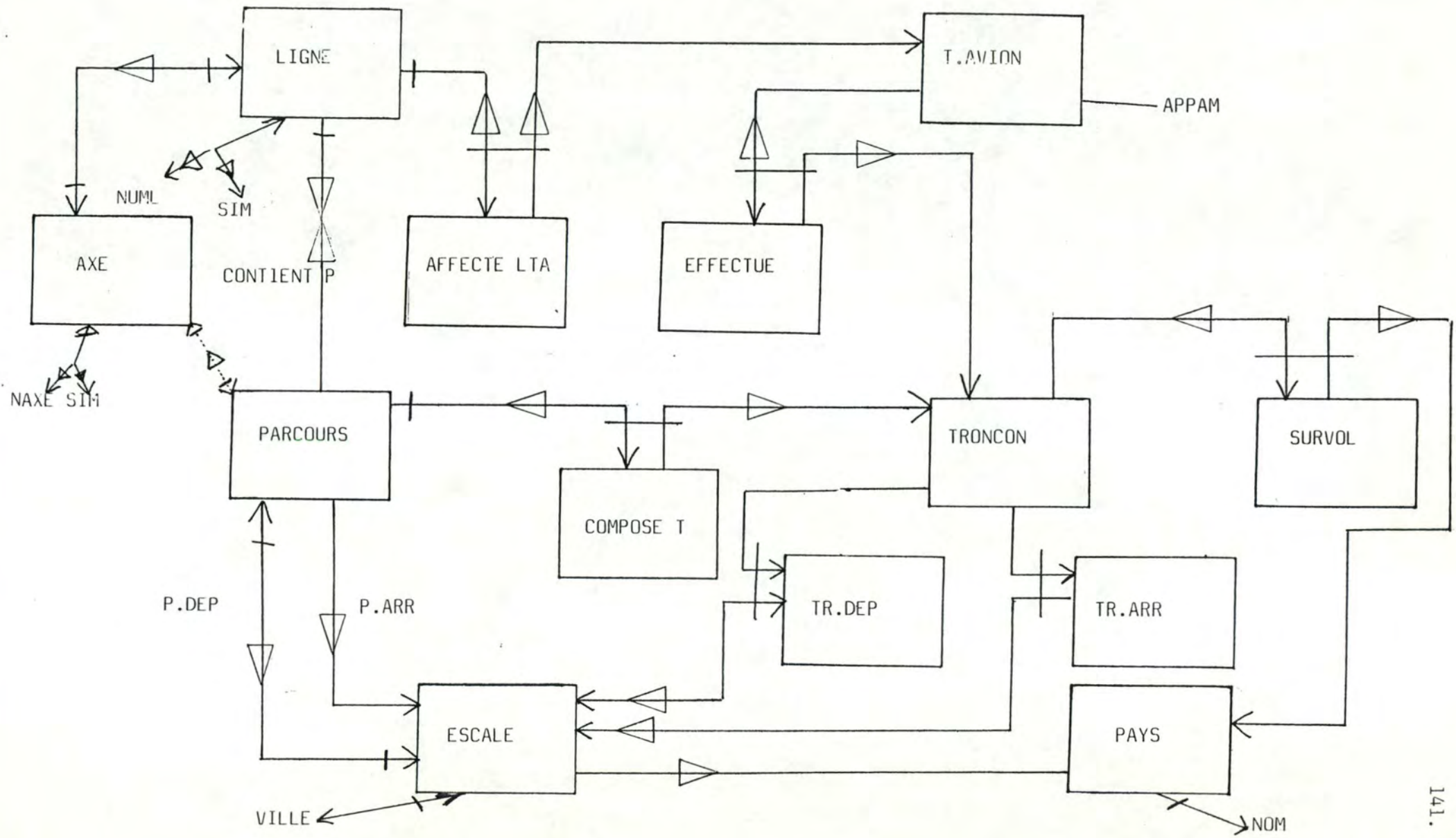
A'	B'	C'
T.AVION	TRONCON	EFFECTUE
TRONCON	PAYS	SURVOL
LIGNE	SERVICE PRODUIT	BUDGET SP
LIGNE	SERVICE EXT	BUDGET SE
LIGNE	T.REC	BUDGET R
LIGNE	T.MOYEN	BUDGET MC
T.ROUTING PNM	TRONCON	CONTIENT M
T.ROUTING PNC	TRONCON	CONTIENT C

## 9.9. Sous-schémas

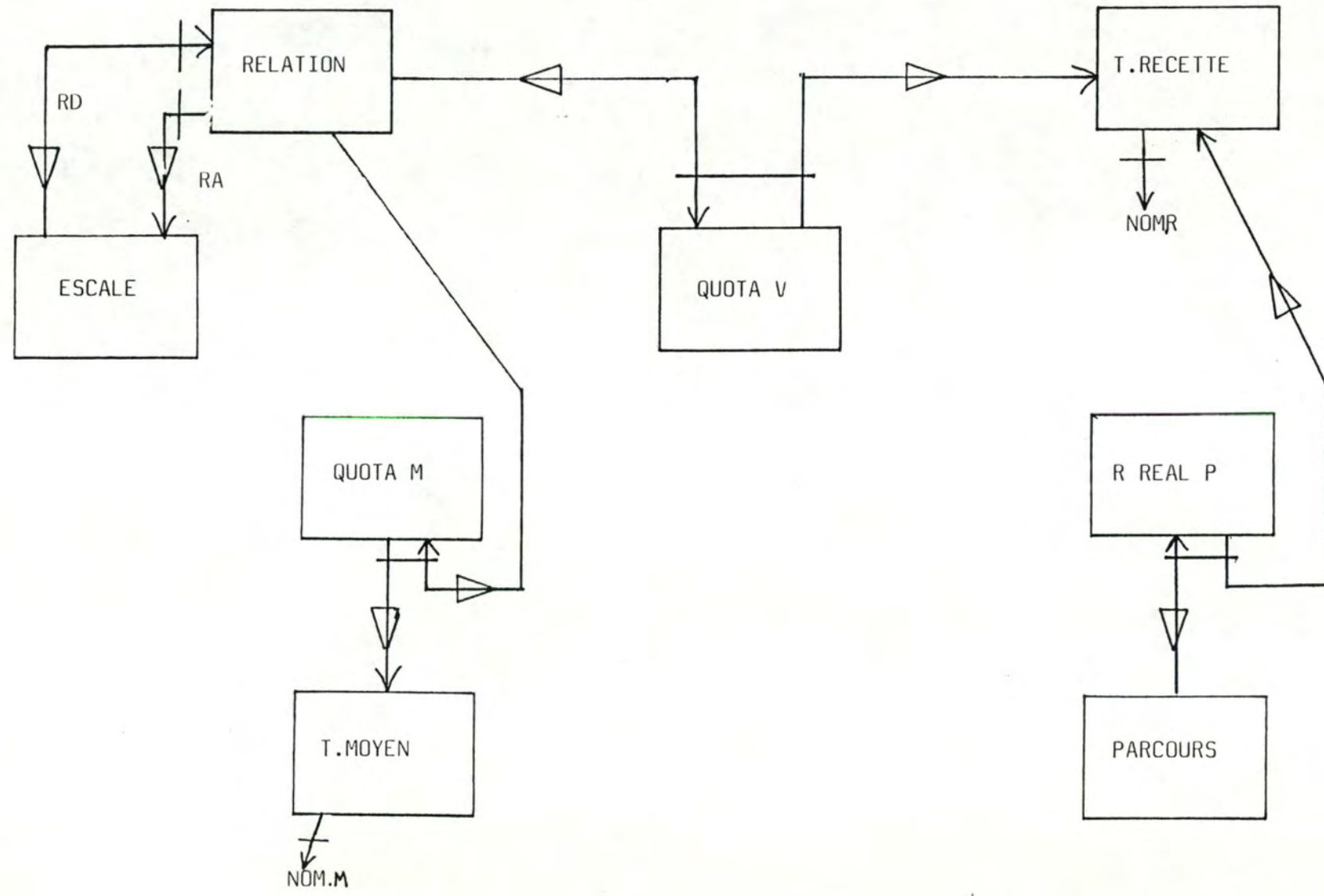
### 9.9.0. Introduction

Dans les sous-schémas ci-dessous nous ne retenons que les chemins nécessaires pour les programmes actuels. Par contre nous gardons tous les items, car nous avons retenu dans le schéma conceptuel des attributs qui seront prochainement utilisés dans de nouvelles applications développées par le service.

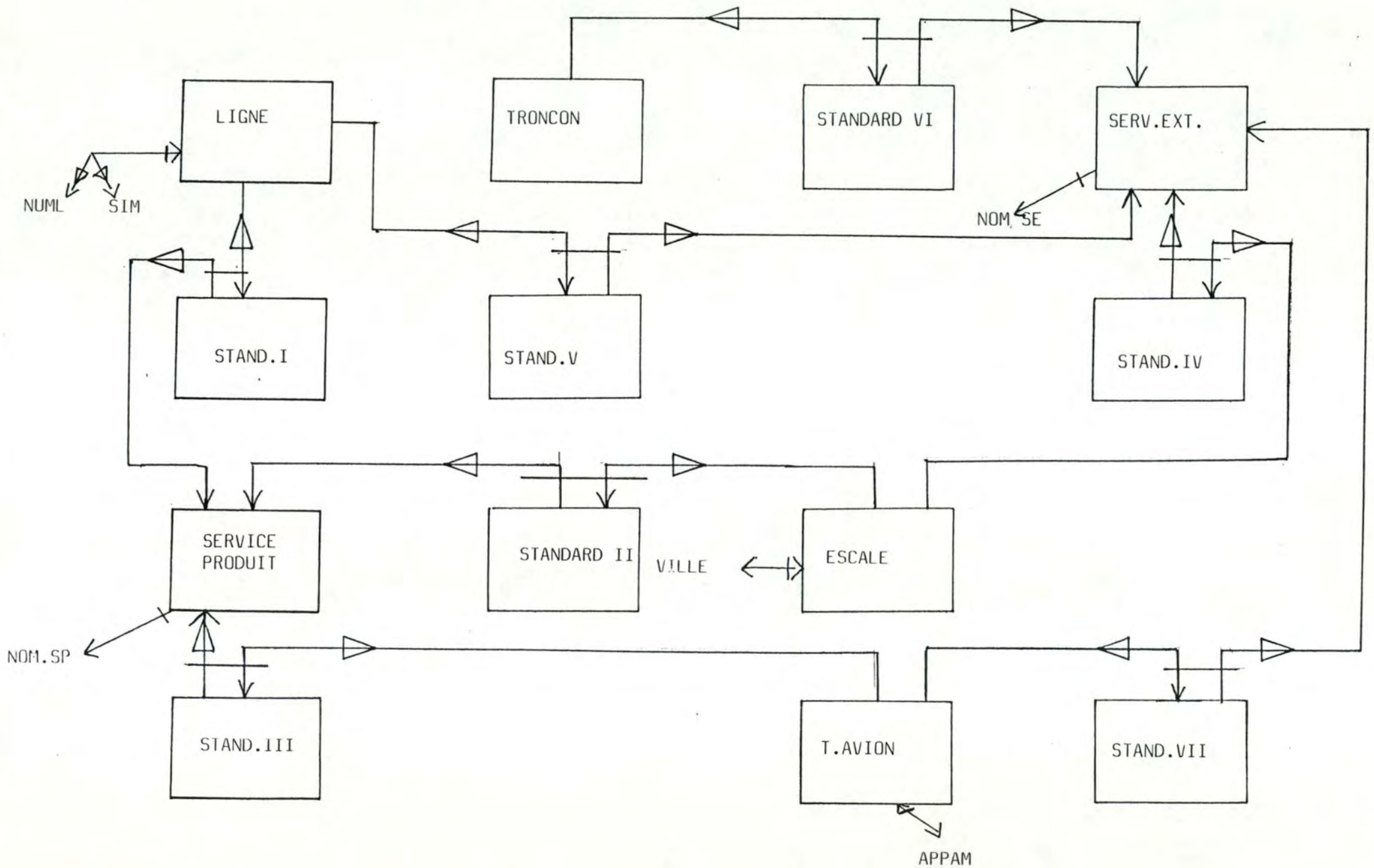
9.9.1. SOUS-SCHEMA PRODUIT ET MOYEN



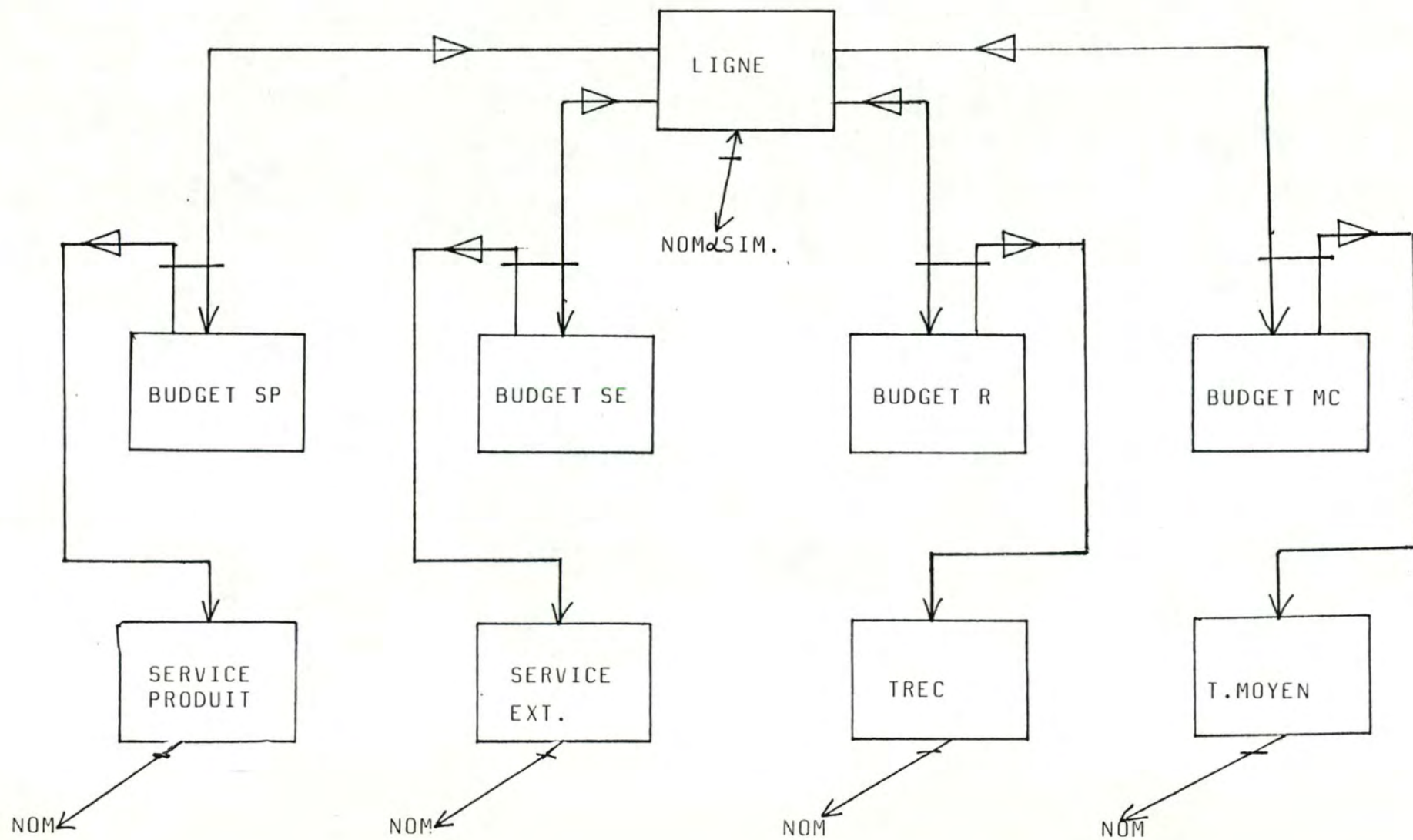
9.9.2. SOUS-SCHEMA QUOTAS ET SOUS-SCHEMA FACTEURS DE PRODUCTION



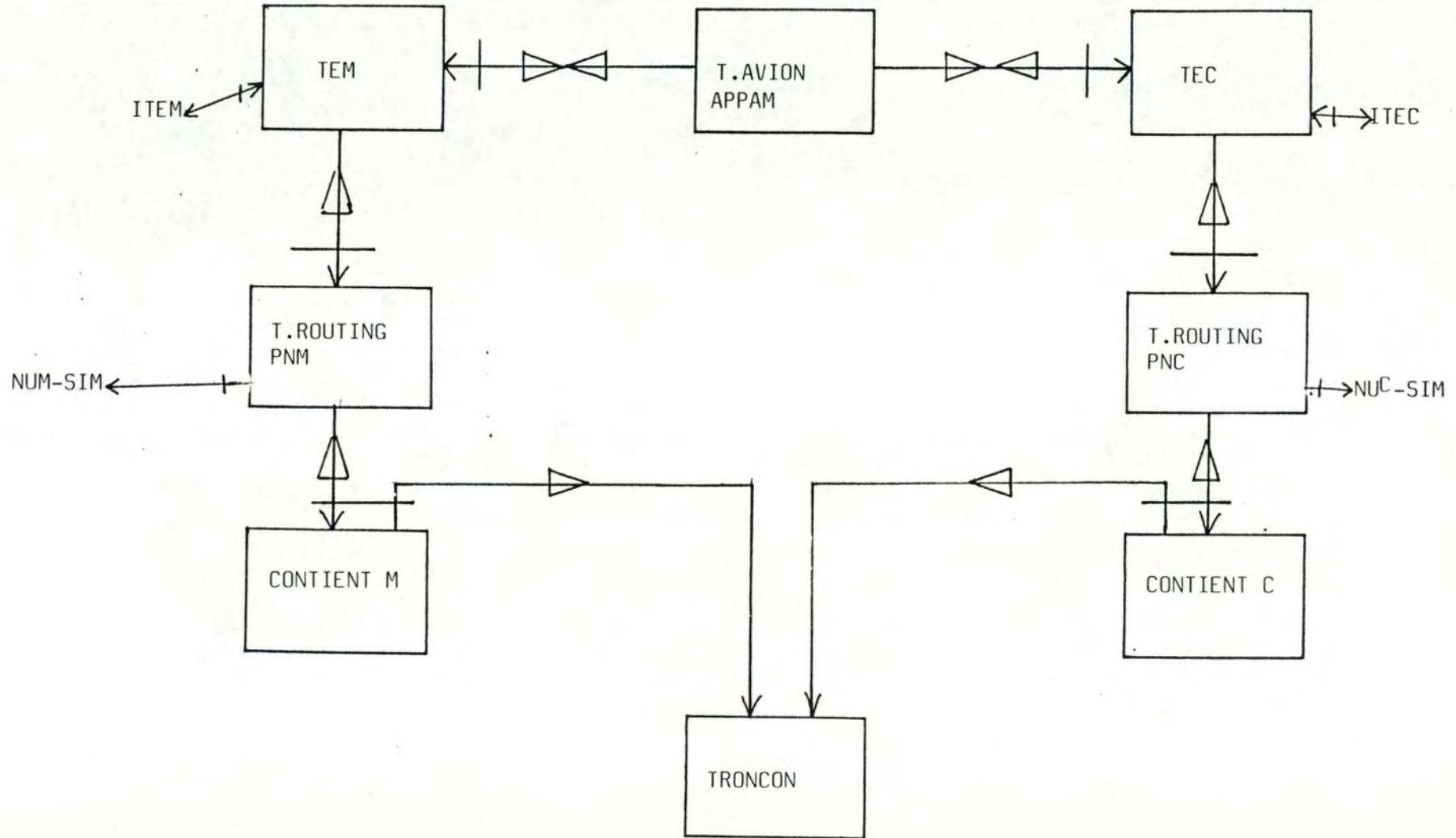
9.9.3. SOUS-SCHEMA COUTS UNITAIRES



9.9.4. SOUS-SCHEMA BUDGET



9.9.5. SOUS-SCHEMA EQUIPAGES



X SCHEMA IMAGE /HP.

### 10.1. Description de Image

Image est un SGBD de la famille TOTAL. Deux types d'articles sont définis:

- 1) MASTER SET
- 2) DETAIL SET.

Les types de chemins sont one to many de MASTER vers DETAIL sans inverse possible.

Les items peuvent être composés mais pas répétitifs. L'accès à un DETAIL se fait toujours via un MASTER. Plusieurs types d'accès sont possibles:

MASTER : - par clé identifiante

- direct
- séquentiel.

DETAIL : - chaîné à partir d'une clé d'un MASTER

- séquentiel
- direct

Il existe deux types de MASTER:

- AUTOMATIQUE
- MANUAL.

#### 1) AUTOMATIQUE

- Il ne possède qu'un item, l'identifiant.

- Il doit être relié à au moins un DETAIL.
- Rajout automatique d'une occurrence du MASTER si ajout d'un DETAIL lié n'ayant pas une valeur de clé déjà existante dans MASTER.
- Suppression automatique d'une occurrence de MASTER si plus d'éléments ayant la même clé sont présents dans le DETAIL lié.

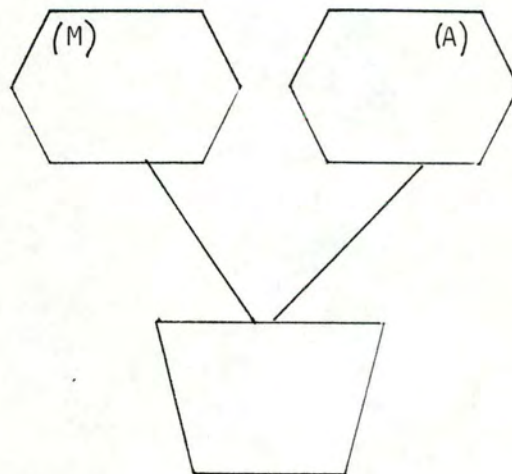
## 2) MANUAL

- Il possède plusieurs items.
- Il peut être "stand alone".
- Les ajouts ou retraits ne sont pas automatiques.

### Représentation

MASTER

MANUAL



AUTOMATIQUE

DETAIL

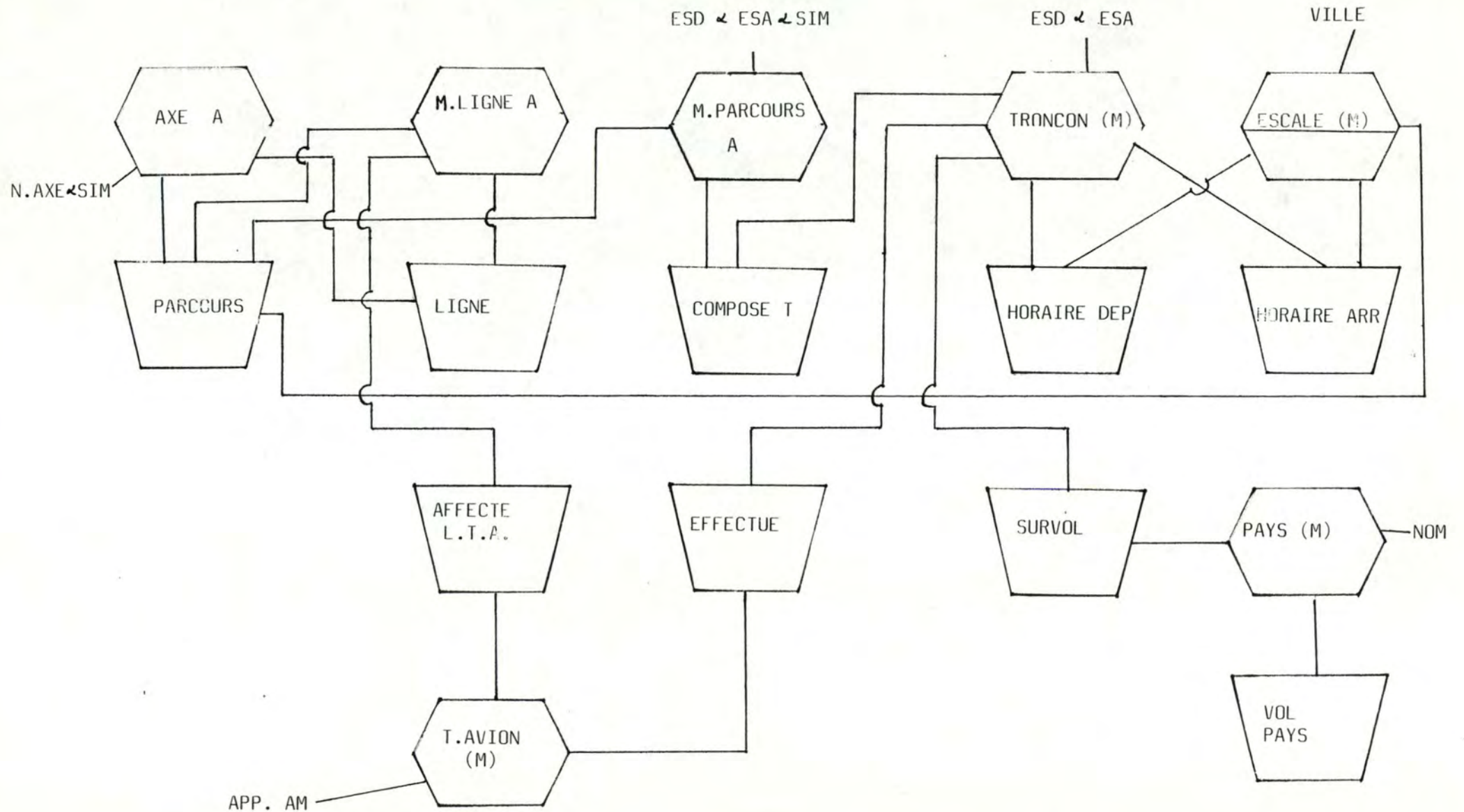
## 10.2. Schéma IMAGE

Nous avons traduit le schéma des accès nécessaires en un schéma compatible IMAGE. Cette traduction a nécessité que nous opérions des transformations, telle que la création de M.LIGNE.

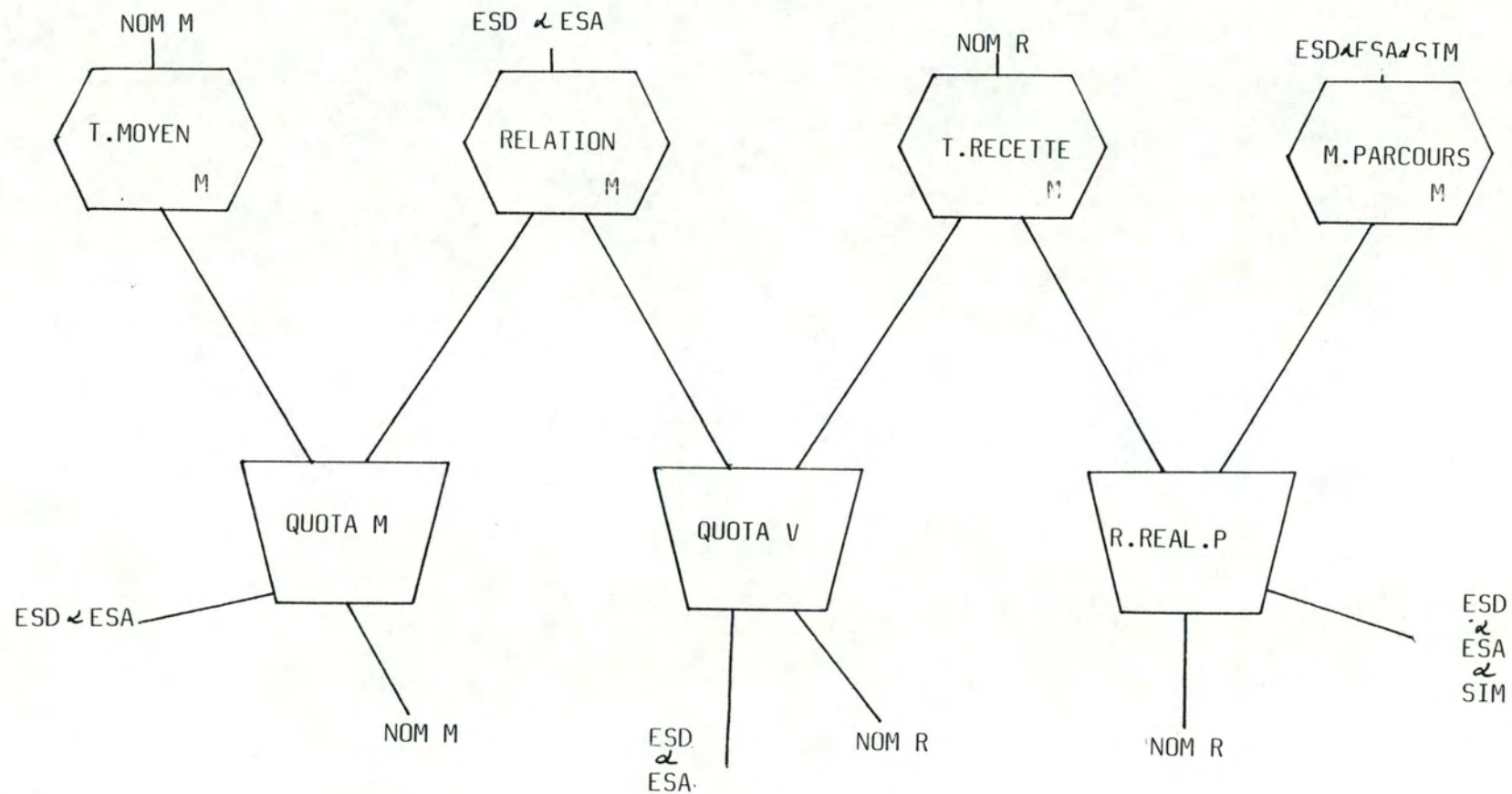
Dans le sous-schéma des produits et moyens, il serait sémantiquement plus approprié d'appeler le DETAIL PARCOURS, CONTIENT.P. Toutefois, nous avons choisi PARCOURS afin que les futurs utilisateurs ne soient pas troublés et puissent retrouver aisément les informations relatives aux parcours dans la banque de données.

Nous présentons ci-dessous les schémas simplifiés. Au paragraphe 10.3., nous donnerons les algorithmes ADL compatibles avec le schéma IMAGE.

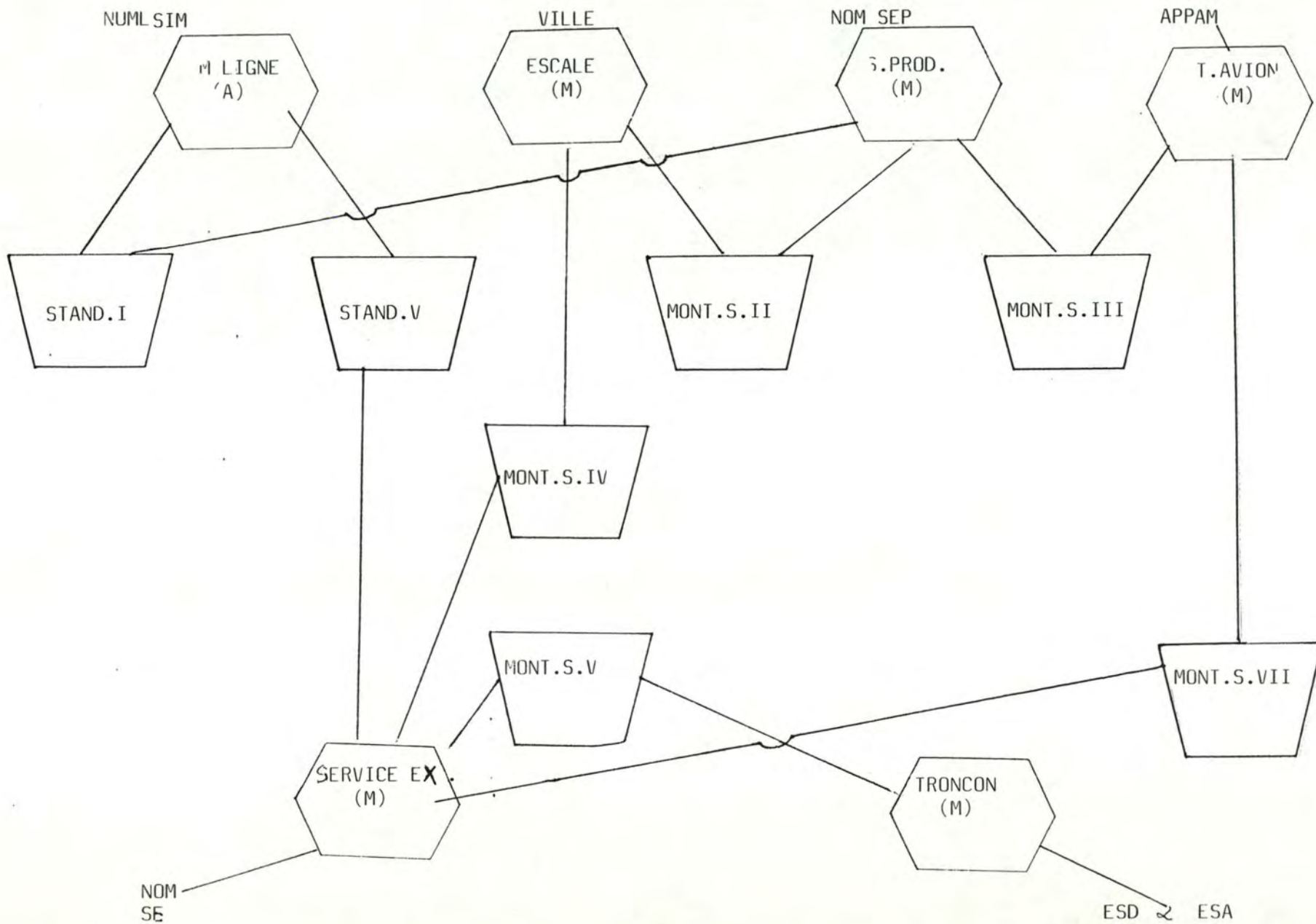
10.2.1. PRODUITS ET MOYENS



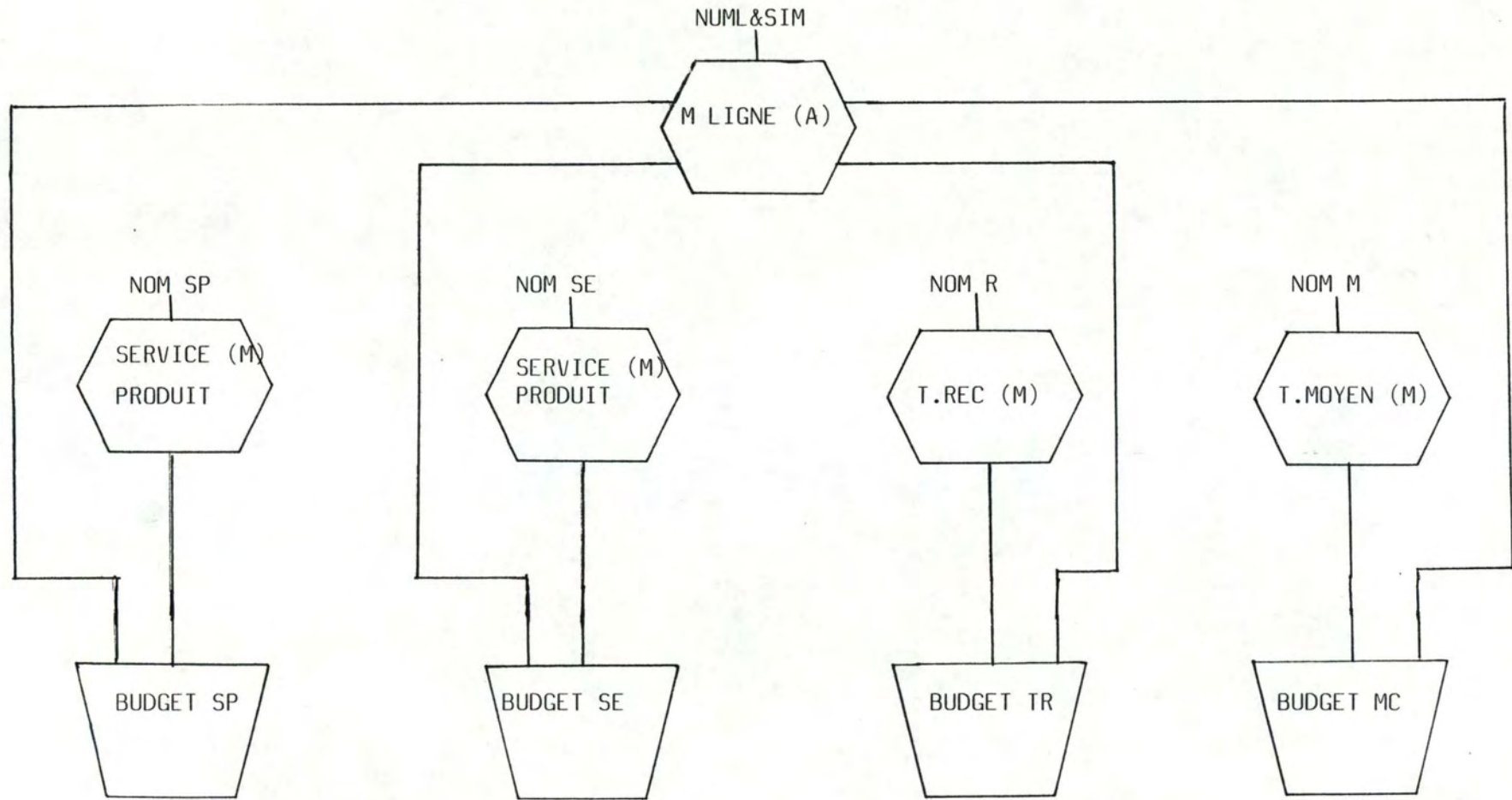
10.2.2. QUOTA ET FACTEURS DE PRODUCTION



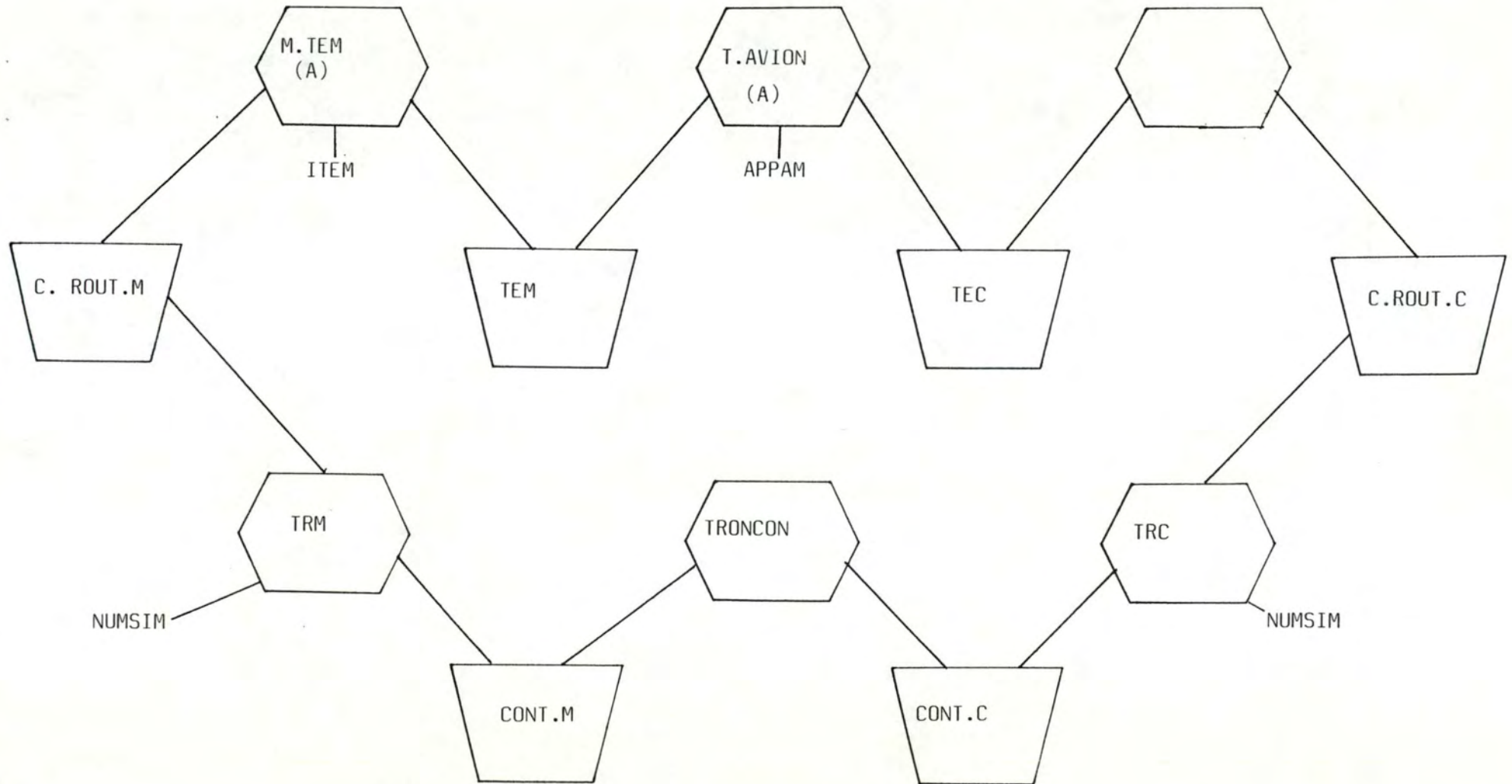
10.2.3. COÛTS UNITAIRES



10.2.4. BUDGET



10.2.5. EQUIPAGES



### 10.3. Algorithmes effectifs ADL/IMAGE.

#### 10.3.1. Constitution des axes.

1) Soit CHAXE

1) NAXE&SIM

1) NBRE-LIGNES

1) LIGNES repeat (NBRE-LIGNES, )

2) NUML&SIM

2) APPAM

1) NBRE-PARCOURS

1) PCOURS repeat (NBRE-PARCOURS, )

2) ESD&ESA&SIM

2) NBRE-TRC

2) TRC repeat (NBRE-TRC, )

3) ESD&ESA

2) NBRE-LIG

2) LIG repeat (NBRE-LIG)

3) NUML&SIM.

1) Créer pour une simulation

- un axe

- les lignes, parcours et tronçons de cet axe

- les avions de lignes.

input (CHAXE)

for NL:=1..NBRE-LIGNES

create L:=LIGNE ((: NUML&SIM := NUML&SIM(:LIGNES [NL])(:CHAXE)))

&(:NAXE&SIM:=NAXE&SIM (:CHAXE)))

```

create ALTA:=AFFECTE L.T.A((: NUML&SIM:=NUML&SIM (:LIGNE [NL](:CHAXE)))
                        &(:APPAM=APPAM(:LIGNES [NL] (:CHAXE))))
for N.P.:=1..NBRE-PARCOURS
  for NL:=1..NBRE-LIG
    create P:=PARCOURS ((:NAXE&SIM:=AXE&SIM(:CHAXE))
                      &(:NUML&SIM:=NUML&SIM(:LIG [NL] (:PCOURS [NP](:CHAXE)))
                      &(:ESD&ESA&SIM:=ESD&ESA&SIM(:PCOURS[NP] (:CHAXE))))
  endfor NL
for NT:=1..NBRE-TRC
  create CT:=COMPOSE.T ((:ESD&ESA:=ESD&ESA(:TRC [NT] (:PCOURS [NP](:CHAXE))))
                      &(:ESD&ESA&SIM:=ESD&ESA&SIM (:PCOUR [NP] (:CHAXE))))
endfor

```

2) Pour une ligne d'une simulation créer les horaires.

Soit CHHOR

- 1) NBRE-TRC
- 1) TRC repeat (NBRE-TRC, )
  - 2) ESD
  - 2) ESA
  - 2) NUML&SIM
  - 2) HEURE D
  - 2) JOUR D [1..7]
  - 2) HEURE A
  - 2) JOUR A [1..7]

Input (CHHOR)

for NT:=1..NBRE-TRC

```

create H.D.: =HORAIRE.DEP((:=NL&SIM:=NUML&SIM (:CHHOR)
                        &(HEURE:=HEURE.D(:TRC[NT])(:CHHOR)))
                        &(for N:=1..7
                          JOUR[N](:H.D)):=JOUR.D[N](:TRC[NT])(:CHHOR))
                        &( :VILLE:=ESD(:TRC[NT]))
                        &( :ESD&ESA:=ESD&ESA(:TRC[NT])(:CHHOR))))
create H.A.: =HORAIRE.ARR((:NL&SIM:=NUML&SIM (:CHHOR)
                        &(HEURE:=HEURE A(:TRC [NT])(:CHHOR)))
                        &(for N:=1..7
                          JOUR[N](:H.D)):=JOUR A[N](:TRC[NT])(:CHHOR))
                        &(VILLE:=ESA (:TRC [NT]))
                        &(ESD&ESA:=ESD&ESA (:TRC [NT])(:CHHOR))))

```

N.B. Nous supposons l'existence d'une procédure externe permettant de concaténer ESD, ESA en ESD&ESA.

10.3.2. QUOTAS

3) Créer pour une relation et une simulation les quota-vente.

Soit CHQUOT

1) ESD&ESA

1) NBRE-QVX

1) QVX repeat (NBRE-QVX )

2) SIM

2) NUN

2) RUN

2) NOM R.

3) input (CHQUOT)

for NQ:=1.. NBRE-QVX

```

create QV:=QUOTA V ((:SIM:=SIX(:QVX [NQ](:CHQUOT)))
                    &(:NUN:=NUN(:QVX [NQ] (:CHQUOT)))
                    &(:RUN:=RUN (:QVX [NQ](:CHQUOT)))
                    &(:ESD&ESA:=ESD&ESA(:CHQUOT))
                    &(:NOM:=NOM.R(:QVX [NQ](:CHQUOT))))

```

end

Même type d'algorithme pour le N°4 qui crée les quotas moyens.

10.3.3. Harmonisation

5) Pour une relation R donner la recette z pour la simulation SIX.

input (ESX&ESY, SIM, Z)

for R:=RELATION (:ESD&ESA=ESX&ESY)\*

for Q.V:=QUOTA.V (:R)

if ((SIM(:QV)=SIX)

```

      α (NOM R (:QV)=Z))
    then print NUN(:QV)
          RUN (:QV)
endfor.

```

6) Pour 1 relation d'1 simulation donner les moyens commerciaux.

```

Input (ESX&ESY,SIX)
for R:=RELATION (:ESD&ESA=ESX&ESY)*
  for Q.M.:=QUOTA.M (:R)
    if SIM (:QM)=SIX then
      print MONTANT (:QM)
          NOM.M (:QM)
    endif
  endfor
endfor

```

7) Pour une simulation donnée, donner tous les axes ayant un parcours reliant les escales X et Y.

```

newlist (Alist)
input (ESX&ESY&SIX)
for M.P:=M.PARCOURS (:ESD&ESA&SIM=ESX&ESY&SIX)*
  for P:=PARCOURS (:MP)
    putlist (Alist, NAXE&SIM(:P))
  endfor M.P.
unique list (Alist)
for A:=Alist
  print A
endfor A.

```

8) Pour un axe d'une simulation, donner tous les parcours

```

newlist (Plist)
input (AXEXSIX)

for A:=AXE(:NAXE&SIM=AXEX&SIX)*
  for P:=PARCOURS (:A)
    putlist (Plist, ESD&ESA&SIM(:P))
  endfor A
unique list (Plist)
for P:=Plist
  print P
endfor P.

```

9) Pour une ligne d'une simulation, donner les parcours.

```

Input (NUMX&SIX)
for M.L:=M.LIGNE (:NUML&SIM=NUMX&SIX)*
  for P:=PARCOURS (:M.L)
    print ESD&ESA&SIM(:P)
  endfor M.L.

```

10) Donner tous les axes d'une simulation ESX comme escale de départ d'un parcours.

Nous avons deux possibilités:

1) lecture séquentielle de M.PARCOURS, si ESD=X pointer sur parcours.

2) Créer un chemin entre ESCALE et PARCOURS.

Afin d'éviter une lecture séquentielle pénalisante de M.PARCOURS, nous choisissons la deuxième solution. Toutefois lors de la

création de PARCOURS, il ne faut pas oublier de le relier à ESCALE. C'est pourquoi le premier algorithme doit être modifié comme suit (\* nous supposons la déconcaténation de ESD&ESA&SIM de CHAXE exécutée par une procédure externe\*).

```

for N.P:=1..NBRE-PARCOURS
  for NL:=1..NBRE-LIG
    create P:=PARCOURS ((:NAXE&SIM:=AXE&SIM(:CHAXE))
                        &(:NUML&SIM:=NUML&SIM(:LIG[NL])(:PCOURS[NP])(:CHAXE)))
                        &(:ESD&ESA&SIM:=ESD&ESA&SIM(:PCOURS[NP])(:CHAXE)))
                        &(:VILLE:=ESD (:PCOURS[NP])(:CHAXE)))
  endfor NL.

```

---

10) Newlist (Alist)

```

Input (X,SIX)
for E:=ESCALE (:VILLE=X)*
  for P:=PARCOURS (:E)
    if SIM(:P)=SIX (*après déconcaténation*)
      then putlist (Alist, NAXE&SIM)
    endif
  endfor E
  uniquelist (Alist)
  for A:=Alist
    print A
  endfor A.

```

11) Pour un axe donné d'une simulation, donner la charge offerte par tronçon de toutes les lignes.

```
Newlist (Tlist)
```

```
Input (AXEX&SIX)
```

```
for A:=AXE(:NAXE&SIM=AXEX&SIX)*
```

```
  for P:=PARCOURS (:A)
```

```
    NUML&SIM(:TRL):=NUML&SIM (:P)
```

```
      for M.P:=M.PARCOURS (:ESD&ESA&SIM=ESA&ESA&SIM(:P))*
```

```
        for CT:=COMPOSE T (:M.P)
```

```
          ESD&ESA(:TRL):=ESD&ESA (:CT)
```

```
          putlist (Tlist, TRL)
```

```
    ednfor P.
```

```
  unique list (Tlist)
```

```
  for TRL:=(Tlist)
```

```
    for M.L:=(M.LIGNE(:NUML&SIM=NUML&SIM(:TRL))*
```

```
      for A:=AFFECTE (:ML)
```

```
        for T.A:= T.AVION (:APPAM=APPAM(:A))*
```

```
          for E:=EFFECTUE (:A)
```

```
            print CH.PAX (:E)
```

```
              CH.FRET (:ES)
```

```
              ESD&ESA (:E)
```

```
  endfor.
```

12) Donner pour un axe d'une simulation, le nombre d'heures de vol et mob. par ligne , ainsi que la charge offerte.

```

Input (AXEX&SIX)
for A:=AXE(:NAXE&SIM=AXEX&SIX)*
  for L:=LIGNE (:A)
    for M.L:=M.LIGNE (:NUML&SIM=NUML&SIM(:A))*
      for AF:=AFFECTE L.T.A. (:M.L)
        print NUML&SIM (:A)
          H.VOL (:AF)
          H.MOB (:AF)
          CHARGE.PAX (:AF)
          CHARGE.FRET (:AF)
      endfor.
    endfor.
  endfor.

```

13) Pour une ligne et 1 simulation, donner le routing et l'horaire.

```

Newlist (Tlist)
Input (NUMX&SIX)
for M.L:=M.LIGNE (:NUML&SIM=NUMX&SIX)
  for P:=PARCOURS (:ML)
    NUML&SIM (TRL):=NUML&SIM (:P)
    for M.P:=M.PARCOURS (:ESD&ESA&SIM=ESD&ESA&SIM (:P))*
      for C:=COMPOSE T (:MP)
        ESD&ESA(:TRL):=ESD&ESA(:C)
        putlist (Tlist, TRL)
      endfor.
    endfor.
  endfor.
endfor ML

```

```

uniquelist (Tlist)
for TR:=Tlist
  for T:=TRONCON(:ESD&ESA:=ESD&ESA(:TR))*
    for H.DP:=HORAIRE DEP (:T)
      if NUML&SIM(:HD)=NUML&SIM(:TRL)
        then print JOUR (:HD)
              HEURE (:HD)
    endfor H.D.
  for H.A.:=HORAIRE ARR(:T)
    if NUML&SIM (:HA)=NUML&SIM (:TRL)
      then print JOUR (:HA)
            HEURE (:HA)
  endfor.

```

14) Pour les parcours d'un axe, créer les recettes réalisables par parcours pour 1 simulation.

Soit CHARM

- 1) NBRE-PARC
- 1) PARC repeat (NBRE-PARC, )
  - 2) NAXE&ESD&ESA&SIM
  - 2) NBRE-REC
  - 2) REC repeat (NBRE-rec, )
    - 3) N.UN.REA
    - 3) N.UN.REF
    - 3) R.REA
    - 3) NOM.R.
- 2) ESD&ESA&SIM.

Input CHARM

for NP:=1..NBRE-PARC

for NR:=1..NBRE-REC

create RR:=R.REAL.P((:NAXE&ESD&ESA&SIM=NAXE&ESD&ESA&SIM(:PARC[NP]  
(:CHARM))))

&(:NUNR:=NUNR(:TRC NR (:PARC[NP](:CHARM))))

&(:NUNREF:=NUN.REF(:TRC [NR](:PARC[NP](:CHARM))))

&(:R.REA:=R.REA(:TRC [NR](:PARC[NP](:CHARM))))

&(:NOM.R:=NOM.R(:TRC [NR](:PARC[NP](:CHARM))))

&(:ESD&ESA&SIM:=ESD&ESA&SIM(:PARC[NP](:CHARM))))

15) Pour un tronçon donné, donner les pays survolés et le vol pour une simulation SIX.

Input (ESX&ESY&SIX)

for T:TRONCON(:ESD&ESA=ESX&ESY)\*

for S:=SURVOL (:T)

for P:=PAYS (:NOM.P=NOM.P(:S))\*

for V.P.:VOL. PAYS (:P)

if SIM(:VP)=SIX

then print NOM;P(:P)

TYPE TARIF (:V)

POND (:V)

endfor.

#### 10.3.4. Standards

16) Pour une ligne et un service produit, créer les standards I.

Soit (CHSTI)

- 1) NUML&SIM
- 1) NBRE-STAND
- 1) STAND repeat (NBRE-STAND)
  - 2) MONTANT-D
  - 2) NOM-SP

Input (CHST I)

for NS:=1..NBRE-STAND

```

create S:=STAND.I ((:MONTANT:=MONTANT-D (:STAND[NS](:CHSTI I)))
&(:NUML&SIM:=NUML&SIM (:CHSTI I))
&(:NOM SP:=NOM SP(:STAND [NS](:CHST I))))

```

endfor

17) Pour une escale donnée, créer les standards II des services produits.

Soit CHST II

- 1) ES
- 1) NBRE-STAND
- 1) STAND repeat (NBRE-STAND, )
  - 2) MONTANT-D
  - 2) SIM
  - 2) NOM-SP

Input(CHST II)

for N.S:=1..NBRE-Stand

```

create M.S:=MONT S.II((:NOM.SP:=NOM.SP(:STAND[NS] (:CHST II)))
& (:VILLE:=ES (:STAND [NS](:CHST II)))
& (:MONTANT:=MONTANT-D(:STAND[NS](:CHST II)))
& (:SIM :=SIM (:STAND [NS] (:CHST II))))

```

end

C'est le même type d'algorithme que nous utiliserons pour les autres standards.

#### 10.3.5. Rentabilité

18) Pour une ligne donnée d'une simulation, donner les recettes réalisables par parcours d'axe ainsi que les refus et les escales de départ et d'arrivée.

Input (NL&SIX)

for M.L:=M.LIGNE (:NUML&SIM=NL&SIX)\*

for P:=PARCOURS (:M.L)

for M.P.:=M.PARCOURS(:ESD&ESA&SIM=ESD&ESA&SIM (:P))\*

for R.R.:=R.REA.LP (:MP)

print (\*les items\*)

19) Pour une ligne d'une simulation donner les stand I.

Input (NUMX&SIX)

for M.L:=M.LIGNE(:NUML&SIM=NUMX&SIX)\*

for S:=STAND I (:ML)

print (\*les items\*).

20) Pour escale ESD, donner les stand II d'une simulation.

Input (ESD,SIX)

for E:=ESCALE(:VILLE=ESD)\*

for S:=MONT.S.II (:E)

if SIM(:S)=SIX

then print (\* les items\*)

10.3.6. Equipages

21) Pour une escale, une simulation, un type d'équipage de maîtrise donnés, donner les tronçons ayant cette escale comme départ, ainsi que les lignes contenant ces tronçons.

Input (ESD,SIX,I TEX)

for MT:=M.TEM (:ITEM=I TEX)\*

for CR:=C.ROUT.M (:MT)

for TR:=T.ROUTING PNM (:NUM&SIM=NUM&SIM:(CR))\*

for CM:=CONTIENT.M (:TR)

if ESD(:TM)=ESD (\*après déconcaténation de ESD&ESDA\*)

then for T:=TRONCON(:ESD&ESA=ESD&ESA(:CM))\*

for H.D.:HORAIRE DEP(:T)

if SIM (:HD)=SIX

then print (\* les items\*)

endif

endfor H.D.

for H.A.:HORAIRE ARR (:T)

if SIM(:HA)=SIX

then print (\* les itemsé)

endfor H.A.

endfor.

10.3.7. Métaalgorithmes1) Premier type

Input (CHARGE)

```

for N:=1..NBRE-C
  create cx:=C' ((:c':=c (:([N] (:CHARGE)))
                &(:a':=a(:CHARGE))
                &(:b':=b(:([N](:CHARGE))))
endfor N.

```

## 2) Deuxième type

Input (a)

```

for A:=A'(:a'=a)*
  for C:=C'(:A)
    print c'(:C)
          b'(:C)
  end
end

```

### 10.4. Transformation du schéma de base

Nous n'avons dû apporter qu'une modification au schéma-IMAGE de base. En effet, afin de ne pas être pénalisé en accès, nous avons rajouté un chemin de PARCOURS à ESCALE. D'autres modifications à ce schéma de base ne sont pas nécessaires. On aurait pu envisager des modifications visant à permettre le travail exclusif de deux modules, sur un type d'article dédoublé, tel que LIGNE. Nous pensons que cette redondance n'est pas justifiée si un environnement software de gestion de concurrence est mis sur pied.

## 10.5. Schéma interne

### 10.5.1. Data Base

Nous proposons qu'une seule DATA BASE. En effet, aucune contrainte physique nous impose le dédoublement de la base. En ce qui concerne les problème de concurrence, on pourrait envisager de dédoubler la base en deux, l'une pour le long terme, l'autre pour le court terme. La justification de ce dédoublement serait que souvent des transactions concurrentes vont apparaître, telles que:

- 1) la constitution des axes à long terme,
- 2) le calcul rentabilité à court terme ou l'harmonisation à court terme. Lors de la constitution des axes, la base sera souvent verrouillée complètement, afin de créer les axes. Or seuls des verrous sur toute la base sont disponibles sur IMAGE. Nous voyons donc qu'il peut être tentant, voire acceptable, de dédoubler la base, la première contenant les simulations court terme, l'autre les longs termes. Toutefois, nous pensons qu'un logiciel de gestion de concurrence permettrait d'obtenir un compromis honorable entre temps nécessaire pour une transaction et espace mémoire utilisée.

Pour qu'un tel logiciel soit établi, il faut que le service définisse les priorités des différentes applications et/ou utilisateurs. Actuellement, aucun niveau de priorité n'est défini. C'est pourquoi nous proposerons les niveaux d'accès par défaut.

10.5.2. Items

Nous proposons momentanément 1 et 10 comme niveau respectif de confidentialité à la lecture et à l'écriture.

NUMLSIM	X 5
=	
Nom de l'item	X = caractères ASC II
	I = integer
	R = real

ITEMS

NAXESIM,	I1
ROT	I1
SYMT,	X3
NUMLSIM,	I1
NAXEESDESASIM	X18
VILLE	X3
ESDESA	X6
ESAD	I1
DIST	I1
AEROPORT	X4
INSALUBRE	X1
HOTEL	R2
TRANS.	I1
SIM	I1
CH.PAX	R2
CH.FRET	R2
H.V.	R2
HMTA	
HMP	

TOW	I1
TEMPS BL	R2
NL	I1
JOUR,7	X1
HEURE,	R2
APPAM	I1
BLOCK FUEL,	R2
AVIA	R2
RESEAU	I1
SIEGES	I1
CHAR.OFF.PAX	R1
PRCT FRET	R2
PRCT PAX	R2
CHARG.FRET	R2
POIDS BAG.	R2
CH.TOT.MAX,	R2
COEFF.OBJ.PAX,	R2
COEFF.OBJ.CHAR,	R2
H.VOL,	R2
H.MOB,	R2
CHARGE PAX	R2
CHARGE FRET	R2
DIST	R2
NOM P	X10
DEWISE	X5
TYP.TAR.	X1
VAL.TAR.	R2
POND	R2

ITEM	I1
ITEC	I1
NUM SIM	I1
SLIP	X1
NUCSIM	I1
NOM.M	X4
MONTANT.M	R2
N°UN	R2
RUN	R2
NOM.R	X4
NUN.REF.P	R2
NUN.REA.P	R2
R.REA.P	R2
N.UNI.T.	R2
N.REF.T	R2
NOM.SP,	X6
NOM.SE,	X5
VPNM,	R2
VPNC,	R2
VSCB,	R2
VESP,	R2
VMAT,	R2
VATL,	R2
CATE,	R2
SPNM,	R2
SPNC,	R2
SSCB,	R2
REPR,	R2

AUXI, R2  
 SATL, ré  
 LBAT, R2  
 LEQT, R2  
 PEXT, R2  
 ESDESAR, X6.

### 10.5.3. Master Data Sets

NAME : AXE :: ,A = nom du master  
 ENTRY : NAXESIM (2)= nombre de chemins partant de cette clé  
 de ce master  
 CAPACITY : 75.X x = nombre de simulations

NAME : M.LIGNE:: ,A  
 ENTRY : NUMLSIM (9)  
 CAPACITY : 300.X

NAME : M.PARCOURS :: ,A  
 ENTRY : NAXE ESD ESA SIM (3)  
 CAPACITY : 500.X

NAME : TRONCON :: M  
 ENTRY : ESDESA (9)  
 ESAD  
 DIST  
 CAPACITY : 250

NAME : ESCALE :: M  
 ENTRY : VILLE (6)  
 AEROPORT  
 : INSALUBRE  
 CAPACITY 120

NAME : T.AVION ::M  
 ENTRY : APPAM (6)  
 BLOCKFUEL  
 AVIA  
 AVIB  
 RESEAU  
 SIEGES  
 CHARG.OFF.PAX  
 CHARG.FRET  
 POIDS BAG.  
 CH.TOT.MAX.  
 COEFF.OBJ.PAX  
 COEFF.OBJ.CHA.  
 CAPACTIY : 40  
 NAME : PAYS :: ,M  
 ENTRY : NOM.P (2)  
 DEVISE  
 CAPACITY : 200  
 NAME : T.MOYEN :: ,M  
 ENTRY : NOM.M (2)  
 CAPACITY : 6  
 NAME : RELATION :: ,M  
 ENTRY : ESDESAR (2)  
 CAPACITY : 300  
 NAME : T.RECETTE :: ,M  
 ENTRY : NOM.R (4)  
 CAPACITY : 5

NAME : S.PRODUIT :: ,M  
 ENTRY : NOM.SP (4)  
 CAPACITY : 13

NAME : M TIE.M :: A  
 ENTRY : ITEM (1)  
 CAPACITY : 40

NAME : T.E.C. ::A  
 ENTRY : ITEC (1)  
 CAPACITY : 60

NAME : T.ROUT.PNM ::M  
 ENTRY : NUMSIM (1)  
 CAPACITY : 200 . X

NAME : T.ROUT.PNC :: M  
 ENTRY : NUC SIM (1)  
 CAPACITY : 300 . X

NAME : EXT.::M  
 ENTRY : NOM S.E. (2)  
 CAPACITY : 19

#### 10.5.4. Detail Data Sets

NAME : LIGNE, D = nom du DETAIL  
 ENTRY : ROT  
           NAXESIM (AXE)  
           NUMLSIM (M.LIGNE)  
           SYMT  
           CHARGE PAX  
           CHARGE FRET  
 CAPACITY : 300 . X

NAME : CONTIENT P, D  
 ENTRY : NAXESIM (AXE)  
 NUMLSIM (M.LIGNE)  
 NAXEESDESASIM (M.PARCOURS)  
 VILLE (ESCALE)

CAPACITY : 500 . X

NAME : COMPOSE.T

ENTRY : ESDESA (TRONCON)  
 NAXEESDESASIM (M.PARCOURS)  
 SIM

CAPACITY : 1000 . X

NAME : EFFEFCTUE

ENTRY : ESDESA (TRONCON)  
 APPAM (T.AVION)  
 HMP  
 TOW  
 CH.PAX  
 HV  
 CH.FRET  
 HMTA  
 TEMPS BL

CAPACITY : 2000

NAME : HORAIRE DEP

ENTRY : ESDESA (TRONCON)  
 SIM .  
 NL  
 JOUR  
 HEURE HEURE GMT  
 VILLE (ESCALE)

CAPACITY : 300 . X  
NAME : HORAIRE ARR  
ENTRY : ESDESA (TRONCON)  
SIM  
NL  
JOUR  
HEURE  
VILLE (ESCALE)

CAPACITY : 300 . X  
NAME : AFFECTE L.T.A.  
ENTRY : NUMLSIM (M.LIGNE)  
APPAM (T.AVION)  
H.VOL  
H.MOB  
CHARGE PAX  
CHARGE FRET

CAPACITY : 300 . X  
NAME : SURVOL , D  
ENTRY : ESDESA (TRONCON)  
DIST  
NOM P (PAYS)

CAPACITY : 1000  
NAME : VOL PAYS, D  
ENTRY : NOM P (PAYS)  
TYPE. TAR  
VAL TAR  
POND

CAPACITY : 1000 X

NAME : LOGEMENT , D  
 ENTRY : VILLE (ESCALE)  
         HOTEL  
         TRANS  
         SIM  
 CAPACITY : 120 . X

NAME : QUOTA M , D  
 ENTRY : ESDESA (RELATION)  
         SIM  
         MONTANT-M  
         NOM-M (T.MOYEN)  
 CAPACITY : 1800 . X

NAME : QUOTA V , D  
 ENTRY : ESDESAR (RELATION)  
         SIM  
         NUN  
         RUN  
         NOM R (T.RECETTE)  
 CAPACITY : 1500 .X

NAME : R.REAL P , D  
         NAXEESDESASIM (M.PARCOURS)  
         N.UN.REF.P.  
         NUN. REA. P.  
         R. REA. P.  
         NOM.R. (T.RECETTE)  
 CAPACITY : 2500 . X

NAME : BUDGET , SP , D

ENTRY : NUMLSIM (M.LIGNE)

NOM.SP (S.PRODUIT)

VPNM

VPNC

VSCB

VESP

VMAT

VATZ

CATE

SPNM

SPNC

SSCB

REPR


AUXI

SATL

LBAT

LEAT

PEXT



= A

CAPACITY : 3900 . X

NAME : BUDGET SE , D

ENTRY : NUMLSIM (M.LIGNE)

NOM.SE (S.EXT)

MONTANT

CAPACITY : 5700 . X

NAME : BUDGET R ,, D  
 ENTRY : NUMLSIM (M.LIGNE)  
 NOM.R (T.RECETTE)  
 MONTANT  
 CAPACITY : 1500 .x

NAME : BUDGET M , D  
 ENTRY : NUMLSIM (M.LIGNE)  
 NOM.M (T.MOYEN)  
 MONTANT  
 CAPACITY : 1800 . X

NAME : STAND.I,, D  
 ENTRY : NUMLSIM (M.LIGNE)  
 NOM SP (S.PRODUIT)  
 A  
 CAPACITY : 900 . X

NAME : STAND II ,, D  
 ENTRY : NOM SP (S.PRODUIT)  
 VILLE (ESCALE)  
 SIM  
 A  
 CAPACTIY : 240 . X

NAME : STAND IV ,, D  
 ENTRY : VILLE (ESCALE)  
 NOM SE (S.EXT)  
 MONTANT  
 SIM  
 CAPACITY : 960 . X

NAME : STAND V,, D  
 ENTRY : NOM SE (S.EXT)  
         NUMLSIM (M.LIGNE)  
         MONTANT  
 CAPACTIY : 1500 .X  
  
 NAME : STAND VI,, D  
 ENTRY : ESDESA (TRONCON)  
         NOM SE (S.EXT)  
         MONTANT  
         SIM  
 CAPACITY : 500 . X  
  
 NAME : STAND III ,, D  
 ENTRY : NOM SP (S.PRODUIT)  
         APPAM (T.AVION)  
         A  
         SIM  
 CAPACITY : 320 . X  
  
 NAME : STAND VII,, D  
 ENTRY : APPAM (T.AVION)  
         NOM SE (S.EXT)  
         MONTANT  
         SIM  
 CAPACITY : 160 . X  
  
 NAME : NECESSITE.M ,,D  
 ENTRY : ITEM (T.E.M.)  
         APPAM (T.AVION)  
         RESEAU  
 CAPACITY : 80 .

NAME : NECESSITE C,, D  
ENTRY : ITEC (T.E.C.)  
APPAM  
RESEAU  
CAPACITY : 120

NAME : CONT.M ,,D  
ENTRY : NUMSIM (T.ROUT.PNM)  
NL  
SLIP  
ESDESA  
CAPACTIY : 800 . X

NAME : CONTIENT C ,, D  
ENTRY : NUC SIM (T.ROUT.PNC)  
ESDESA (TRONCON)  
NL  
SLIP  
CAPACITY : 1120 . X

NAME : C.ROUTING M  
ENTRY : ITEM (M.TEM)  
NUMSIM (T.ROUT.PNM)  
CAPACITY : 200 . X

NAME : C.ROUTING C  
ENTRY : ITEC ( M.TEC)  
NULSIM (T.ROUT.PNC)  
CAPACITY : 300 . X

### 10.5.3. Les algorithmes PSEUDO-FORTRAN/IMAGE

#### 10.5.3.1. Introduction

Nous allons dans ce paragraphe développer en pseudo FORTRAN les appels des routines de IMAGE. Dans la première partie de ce mémoire, nous avons suggéré que des modules d'accès indépendants du SGBD soient développés. Ces modules ne peuvent être établis que dans le cadre d'un environnement logiciel global d'utilisation de la banque de données. Comme nous ne pouvons préjuger si le service va développer un tel environnement, nous exposons ci-dessous les appels qui peuvent être introduits dans les programmes existants. Toutefois, à l'issue de ce paragraphe, nous donnerons un exemple de module que nous proposons.

Dans les algorithmes ci-dessous nous proposons l'existence des routines suivantes:

put X in [Y]

mettre la valeur X dans l'array Y

check STAT 1

routine qui analyse le résultat d'un appel à la Data Base et qui prend les actions nécessaires si une anomalie est signalée par IMAGE.

Résultats: - accès illicite  
 - item inexistant  
 - dernier élément de chaîne  
 - article inexistant  
 - appel réussi  
 - etc.

Actions: - informer le programme appelant  
 - continuer  
 - etc.

Si - pa - fin DETAIL go to X.

condition booléenne : sur l'élément courant de la chaîne des articles de DETAIL. Si l'élément courant n'est pas la dernière condition vraie.

putflag (Tlist)

mettre un flag comme dernier élément de la Tlist.

si - pas - flag Tlist go to X

condition booléenne sur l'élément courant de la Tlist.

Si l'élément courant = Flag condition fausse.

Nous proposons que Tlist soit en mémoire centrale. Par conséquent, le traitement de retrait des doubles est très rapide et de plus des fichiers ne doivent pas être manipulés.

En ce qui concerne les verrous, il y a possibilité d'acquérir:

- lock exclusiv en lecture/écriture  
 lors de l'ouverture de la Data Base.
- intend lock en lecture/écriture  
 lors de l'ouverture de la Data Base.
- shared lock en lecture  
 lors de l'ouverture de la Data Base.

L'intend lock est transformé en exclusiv lock par un appel du programme appelant.

Nous proposons qu'un lock manager responsable de ces verrous soit développé.

Description des appels.

DBFND (B, MASTER, IM, STAT, CLE, VAL.CLE)

Accès par clé de valeur VAL-CLE ou l'article MASTER de la banque B.

DBGET (B, DETAIL, 5, STAT, LIST, BUF, IARG)

Accès chaîné (5) aux articles de DETAIL pointés par l'appel précédent à un MASTER.

LIST: liste des items à lire.

DBPUT (B, DETAIL, 1, STAT, LIST, BUF)

Création d'une occurrence du DETAIL et création des masters qui lui sont liés.

### 10.5.3.2. Constitution des axes

1) Créer pour une simulation

- un axe
- les lignes, parcours et tronçons de cet axe
- les avions des lignes

put LIGNE	in	[DETAIL 1]
NAXE&SIM, NUML&SIM	in	[LIST 1 ]
AFFECTE L.T.A.	in	[DETAIL 2]
NUML&SIM, APPAM	in	[LIST 2 ]
PARCOURS	in	[DETAIL 3]
NAXE&SIM, NUML&SIM, ESD&ESA&SIM	in	[LIST 3 ]
COMPOSE	in	[DETAIL 4]
ESD&ESA, ESD&ESA&SIM	in	[LIST 4 ]

```

NL:=0
10 NL:=NL + 1
   put NAXE&SIM (:CHAXE), NUML&SIM (:LIGNES [NL](:CHAXE)),,
      in BUF 1
   call DBPUT (B, DETAIL 1, 1, STAT 1, LIST 1, BUF 1)
   check STAT 1
   put NUML&SIM (:LIGNE [NL](:CHAXE)), APPAM (:LIGNES [NL](:CHAXE)),,,
      in BUF 2.
   call DBPUT (B, DETAIL 2, 1, STAT 2, LIST 2, BUF 2)
   check STAT 2
   if NL < NBRE-LIGNE
      then go to 10
   endif
   NP:=0
20 NP:=NP + 1
   NL:=0
30 NL:=NL + 1
   put NAXE&SIM (:CHAXE), NUML&SIM (:LIG [NL](:PCOURS [NP](:CHAXE))))) ,
      ESD&ESA&SIM (:PCOURS [NP](:CHAXE))), ESD (:PCOURS [NP](:CHAAXE))))
      in BUF 3
   CALL DBPUT (B, DETAIL 3, 1, STAT 3, LIST 3, BUF 3)
   check STAT 3
   if NL < NBRE-LIGNES
      then go to 30
   endif
   NT:=0
40 NT:=NT + 1
   put ESD&ESA (:TRC [NT](:PCOUR [NP](:CHAXE))),
      ESD&ESA&SIM (:PCOUR [NP](:CHAXE))),
      in BUF 4

```

```

call DBPUT (B, DETAIL 4, 1, STAT 4, LIST 4, BUF 4)
check STAT 4
if NT < NBRE-TRC
    then go to 40
endif
if NP < NBRE-PRCOURS
    then go to 20
end.

```

2) Pour une ligne d'une simulation, créer les horaires.

Input (CHHOR)

```

put HORAIRE DEP                in    [DETAIL 1]
    NL&SIM, HEURE, JOUR 1, JOUR 2, ..
    JOUR 7, VILLE, ESD&ESA      in    [LIST 1]
    HORAIRE ARR                in    [DETAIL 2]
    NL&SIM, HEURE, JOUR 1, ...
    JOUR 7, VILLE, ESD&ESA      in    [LIST 2]

```

NT:=0

10 NT:= NT + 1

```

put NUML&SIM (:CHHOR), HEURE D (:TRC [NT](:CHHOR)),
    JOUR D [1](:TRC [NT](:CHHOR)), JOUR D [2](:TRC [NT](:CHHOR)),
    ... JOUR D [7](:TRC [NT](:CHHOR)), ESD (:TRC [NT],
    ESD&ESA (:TRC [NT](:CHHOR)),
    in BUF 1.

```

call DBPUT (B, DETAIL 1, 1, STAT-D 1, LIST 1, BUF 1)

check STAT -D1

```

put NUML&SIM (:CHHOR), HEURE A (:TRC [NT](:CHHOR)),
    JOUR A [1] (:TRC [NT](:CHHOR)),...

```

```

..., JOUR A [7](:TRC [NT](:CHHOR)),
ESA (:TRC [NT](:CHHOR)), ESD&ESA (:TRC [NT](:CHHOR)),
in BUF 2
call DBPUT (B, DETAIL 2, 1, STAT -D2, LIST 2, BUF 2)
check STAT-D2
if NT < NBRE-TRC go to 10
end

```

#### 10.5.3.3. Quotas

3) Créer pour une relation et une simulation les quotas vente.

```

input CHQUOT
put QUOTA.V in [DETAIL]
SIM, NUN, RUN, NOM R, ESD&ESA in [LIST]
meta 1 (cf métaalgorithme ci-dessouso)

```

4) Créer pour une relation et une simulation les quotas moyens.

```

input
put QUOTA.M in [DETAIL]
SIM, MONT, NOM M, ESD&ESA in [LIST]
meta 1.

```

#### 10.5.3.4. Harmonisation

5) Pour une relation R, donner la recette Z pour la simulation six.

```

put RELATION in [MASTERS]
ESD&ESA in [CLE-M]
ESX&ESY in [VAL-CLE-M]
QUOTA in [DETAIL]
'1 in [IM]

```

```

SIM, NUN, RUN, NOM R           in   [ LIST ]
QLC                             in   [ I ARG ]
call DBFND (B, MASTER, IM, STAT-M, CLE-M, VAL-CLE-M)
check STAT-M
10 call DBGET (B, DETAIL, 5, STAT-D, LIST, BUF, IARG)
check STAT-D
dec BUF (*déconcaténation)
if ((SIM (:BUF)=six) and (NOM R (:BUF)=Z))
then print NUN (:BUF)
           RUN (:BUF)
endif
si - pas - fin - DETAIL : go to 10.
end

```

6) Pour 1 relation d'1 simulation donner les moyens commerciaux.

```

put RELATION                     in   [ MASTER ]
ESD&ESA                          in   [ CLE-M ]
ESX&ESY                          in   [ VAL-CLE-M ]
QUOTA-M                          in   [ DETAIL ]
SIM, MONTANT, NOM.M              in   [ BUF   ]
QLC                              in   [ IARG  ]
1                                in   [ IM   ]
call DBFND (B, MASTER, IM, STAT-M, CLE-M, VAL-CLE-M)
check STAT-M
10 call DBGET (B, DETAIL, 5, STAT-D, LIST, BUF, IARG)
check STAT-D
dec BUF

```

```

if SIM (:BUF) = six then print MONTANT
                                NOM.M

```

```
endif
```

```
si -pas - fin - de DETAIL - go - to 10
```

```
end.
```

7) Pour une simulation donnée, donner tous les axes ayant un parcours reliant les escales X et Y.

```

put M. PARCOURS                in    [MASTER]
    ESD&ESA&SIM                in    [CLE-M]
    ESX&ESY&SIX                in    [DETAIL]
    1                          in    [IM]
    NAXE&SIM                    in    [LIST]
call DBFND (B, MASTER, IM, STAT-M, CLE-M, VAL-CLE-M)
check STAT-M
10 call DBGET (B, DETAIL, 5, STAT-D, LIST, BUF, IARG)
check STAT-D
putlist (Alist, NAXE&SIM (:BUF))
si - pas - fin    DETAIL    go to 10
unique list (Alist)
print list (A list)
end.

```

8) Pour un axe d'une simulation, donner tous les parcours.

```

put AXE                        in    [MASTER]
    NAXE&SIM                    in    [CLE-M]
    PARCOURS                    in    [DETAIL]
    1                          in    [IM]
    QLC                         in    [IARG]
    ESD&ESA&SIM                in    [LIST]
    AXEX&SIX                    in    [VAL-CLE-M]

```

```

newlist (Plist)
call DBFND (B, MASTER, IM, STAT-M, CLE-M, VAL-CLE-M)
check STAT-M
10 call DBGET (B, DETAIL, 5, STAT-D, LIST, BUF, IARG)
check STAT-D
putlist (Plist, ESD&ESA&SIM (:BUF))
si - pas - fin DETAIL go to 10
uniquelist (Plist)
printlist (Plist)

```

9) Pour une ligne d'une simulation donner les parcours

put M.LIGNE	in	[MASTER]
PARCOURS	in	[DETAIL]
1	in	[IM]
QLC	in	[IARG]
NUML&SIM	in	[CLE-M]
NUMX&SIX	in	[VAL-CLE-M]
ESD&ESA&SIM	in	[LIST]

```

call DBFND (B, MASTER, IM, STAT-M, CLE-M, VAL-CLE-M)
check STAT-M
10 call DBGET (B, DETAIL, 5, STAT-D, LIST, BUF, IARG)
check STAT-D
print (ESD&ESA&SIM (:BUF))
si -pas - fin -DETAIL go to 10

```

10) Donner tous les axes d'une simulation ayant ESX comme  
 escale de départ d'un parcours

```

put ESCALE                in      [MASTER]
    PARCOURS              in      [DETAIL]
    VILLE                 in      [CLE-M]
    x                    in      [VAL-CLE-M]
    l                    in      [IM]
    QLC                  in      [IARG]
    NAXE&SIM             in      [LIST]
  
```

newlist (Alist)

call DBFND (B, MASTER, IM, STAT-M, CLE-M, VAL-CLE-M)

check STAT-M

10 call DBGET (B, DETAIL, 5, STAT-D, LIST, BUF, IARG)

dec BUF

if SIM (:BUF)= six then putlist (Alist, NAXE (:BUF))

endif

si - pas - fin DETAIL go to 10

uniquelist (Alist)

printlist (Alist).

1) Pour un axe donné d'une simulation, donner la charge  
 offerte par tronçon de toutes les lignes de cet axe.

```

put AXE                  in      [MASTER 2]
    PARCOURS            in      [DETAIL 1]
    AXE&SIM             in      [CLE-M 1]
    M.PARCOURS         in      [MASTER 2]
    ESD&ESA&SIM        in      [CLE-M2]
    COMPOSE-T          in      [DETAIL 2]
    M.LIGNE            in      [MASTER 3]
    NUML&SIM           in      [CLE-M 3]
  
```

AFFECTE	in	[DETAIL 3]
T.AVION	in	[MASTER 4]
APPAM	in	[CLE-M4]
EFFECTUE	in	[DETAIL 4]
NUML&SIM, ESD&ESA&SIM	in	[LIST 1]
ESD&ESA	in	[LIST 2]
APPAM	in	[LIST 3]
CH.PAX,CH.FRET,ESD&ESA	in	[LIST 4]

DBFND (B, MASTER 1, IM, STAT-M1, CLE-M1, VAL-CLE-M1)  
check STAT-M1

10 DBGET (B, DETAIL 1, 5, STAT-D1, LIST 1, BUF 1, IARG)  
check STAT-D1  
NUML&SIM (:TRL):=NUML&SIM (:BUF1)  
put ESD&ESA&SIM (:BUF 1) in [VAL-CLE-M2]

20 DBFND (B, MASTER 2, IM, STAT-M2, CLE-M2, VAL-CLE-M2)  
check STAT-M2

30 DBGET (B, DETAIL 2, 5, STAT-D2, LIST 2, BUF 2, IARG)  
check STAT-D2  
ESD&ESA (:TRL)=ESD&ESA (:BUF2)  
putlist (Tlist, TRL)  
si - pas - fin DETAIL 2 go to 30  
si - pas - fin DETAIL 1 go to 10  
putflag (Tlist)  
uniquelist (Tlist)  
N:=1

40 put NUML&SIM (:Tlist [N]) in [VAL-CLE-M3]  
DBFND (B, MASTER 3, IM, STAT-M3, CLE-M3, VAL-CLE-M3)  
check STAT-M3

```

50 DBGET (B, DETAIL 3, 5, STAT-D3, LIST-3, BUF 3, IARG)
   check STAT-D3
   put APPAM (:BUF 3)           in [VAL-CLE-M4]
   DBFND (B, MASTER 4, IM, STAT-M4, CLE-M4, VAL-CLE-M4)
   check STAT-M4
60 DBGET (B, DETAIL 4, 5, STAT-D3, LIST-3, BUF 4, IARG)
   check STAT-D4
   print CH.PAX (:BUF 4)
       CH.FRET (:BUF 4)
       ESD&ESA (:BUF4)
   si pas fin DETAIL 4 go to 60
   si pas fin DETAIL 3 go to 50
   N:=N + 1
   si pas flag Tlist go to 40
end.

```

12) Donner pour un axe d'une simulation, le nombre d'heures de vol et mob. par ligne, ainsi que la charge offerte.

```

put AXE           in [MASTER 1]
  NAXE&SIM       in [CLE-M1]
  AXEX&SIX       in [VAL-CLE-M1]
  LIGNE          in [DETAIL 1]
  NUML&SIM       in [LIST 1]
  M.LIGNE        in [MASTER 2]
  NUML&SIM       in [CLE-M2]
  AFFECTE        in [DETAIL 2]
  H.VOL,H.MOB, CHARGE PAX,
  CHARGE FRET    in [LIST 2]

```

```

DBFND (B, MASTER 1, IM, STAT-M1, CLE-M1, VAL-CLE-M1)
check STAT-M1

10 DBGET (B, DETAIL 1, 5, STAT-D1, LIST 1, BUF1, IARG)
check STAT-D1
put NOML&SIM (:BUF1)           in      VAL-CLE-M2
DBFND (B, MASTER 2, IM, STAT-M2, CLE-M2, VAL-CLE-M2)
check STAT-M2

20 DBGET (B, DETAIL 2, 5, STAT-D2, LIST2, BUF2, IARG)
print NUML&SIM (:BUF 1)
      H.VOL (:BUF 2)
      CHARGE PAX (:BUF 2)
      CHARGE FRET (:BUF'2)

si - pas - fin DETAIL 2 go to 20
si - pas - fin DETAIL 1 go to 10
end.

13) put
      M.LIGNE           in      [MASTER 1]
      NUML&SIM           in      [CLE M1]
      NUMX&SIX           in      [VAL-CLE-M1]
      PARCOURS           in      [DETAIL 1]
      NUML&SIM           in      [LIST 1]
      M.PARCOURS         in      [MASTER 2]
      ESD&ESA&SIM        in      [CLE M2]
      COMPOSE T          in      [DETAIL 2 ]
      ESD&ESA            in      [LIST 2]
      TRONCON           in      [MASTER 3]
      ESD&ESA            in      [CLE M3]
      HORAIRE DEP        in      [DETAIL 3]
      NUML&SIM, JOUR, HEURE in    [LIST 3]
                                [LIST 4]

```

```

HORAIRE ARR                                in      DETAIL 4

newlist (Tlist)

DBFND (B, MASTER 1, IM, STAT-M1, CLE-M1, VAL-CLE-M1)
check STAT-M1
10 DBGET (B, DETAIL 1, 5, STAT-D1, LIST 1, BUF 1, IARG)
check STAT-D1
NUML&SIM (:TRL):=NUML&SIM (:BUF1)
put ESD&ESA&SIM(:BUF1)                    in      VAL-CLE-M2

DBFND (B, MASTER 2, IM, STAT-M2, CLE-M2, VAL-CLE-M2)
check STAT-M2
20 DBGET (B, DETAIL 2, 5, STAT-D2, LIST 2, BUF 2, IARG)
check STAT-DE
ESD&ESA (:TRL):ESD&ESA (:BUF2)
putlist (Tlist, TRL)
si pas fin DETAIL 2 go to 20
si pas fin DETAIL 1 go to 10
putflag (Tlist)

uniquelist (Tlist)
N:=1
30 put ESD&ESA (:Tlist [N])                in      VAL-CLE-M3
DBFND (B, MASTER 3, IM, STAT M3, CLE-M3, VLA-CLE-M3)
check STAT-M3
40 DBGET (B, DETAIL 3,5, STAT-D3, LIST 3, BUF 3, IARG)
check STAT-M4
if NUML&SIM (:BUF 3) = NUML&SIM (:Tlist [N])
then print JOUR (:BUF3)
HEURE (:BUF 3)

```

```

endif
si pas fin DETAIL 3 go to 40
50 DBGET (B, DETAIL 4, 5, STAT-D3, LIST 4, BUF 4, IARG )
check STAT-D3
if NUML&SIM (:BUF 4) = NUML&SIM (:Tlist [N] )
    then print JOUR (:BUF 3)
        HEURE (:BUF3)
endif
si pas fin DETAIL 4 go to 50
N:=N + 1
si pas flag Tlist go to 30.

```

14) Pour les parcours d'un axe créer les recettes réalisables par parcours pour une simulation.

```

input CHARM
put R.REAL.P                in    [DETAIL]
    NAXE&ESD&ESA,SIM, NUNREA,
    NUNREF,R.REA,NOM.P,ESD&ESA&SIM in    [LIST]
meta 1 (*inf appel 1 des métaalgorithmes)

```

15) Pour un tronçon donné, donner les pays survolés et le vol pour une simulation six.

```

input (ESX&ESY,SIX)
put TRONCON                in    [MASTER 1]
    ESD&ESA                in    [CLE-M1]
    ESX&ESY                in    [VAL-CLE-M1]
    SURVOL                 in    [DETAIL 1]
    NOM.P                  in    [LIST 1]
    PAYS                   in    [MASTER 2]
    NOM.P                  in    [CLE-M2]

```

```

VOL.PAYS                                in      DETAIL 2
SIM,NOM.P,TYPE TARIF,POND              in      LIST 2

DBFND (B, MASTER 1, IM, STAT-M1, CLE-M1, VAL-CLE M1)
check STAT-M1
10 DBGET (B, DETAIL 1, 5, STAT-D1, LIST 1, BUF 1, IARG)
check STAT-D1
put NOM.P (:BUF 1)                       in      VAL-CLE-M2
DFND (B, MASTER 2, IM, STAT-M2, CLE-M2, VAL-CLE-M2)
check STAT-M2
20 DBGET (B, DETAIL 2, 5, STAT-D2, LIST 2, BUF 2, IARG)
if SIM (:BUF2) = SIX
then print NOM.P (:BUF2)
      TYPE TARIF (:BUF 2)
      POND (:BUF 2)

znsid
si pas fin DETAIL 2 go to 20
si pas fin DETAIL 1 go to 10

16) Pour une ligne et un service produit créer les standards I.
input (CHST I)
put STAND I                               in      [DETAIL ]
      MONTANT, NUML&SIM, NOM SP           in      [LIST ]
méta 1

```

10.5.3.5. Standards

17) Pour une escale donnée créer les standards II des services produits.

input (CHST II)

```
put MONT S II                in    [DETAIL]
    NOM SP, VILLE, MONTANT, SIM in    [LIST]
```

10.5.3.6. Rentabilité

18) Pour une ligne donnée d'une simulation, donner les recettes réalisables par parcours d'axe ainsi que les refus et les escales de départ et d'arrivée.

input (NL&SIX)

```
put M.LIGNE                  in    [MASTER 1]
    NUML&SIM                  in    [CLE M1]
    NL&SIX                     in    [VAL-CLE-M1]
    PARCOURS                   in    [DETAIL 1]
    ESD&ESA&SIM                in    [LIST 1]
    M.PARCOURS                 in    [MASTER 2]
    R.REAL.P                   in    [DETAIL 2]
    ESD&ESA&SIM                in    [CLE-M2]
```

DBFND (B, MASTER 1, IM, STAT-M1, CLE-M1, VAL-CLE-M1)

check STAT-M1

10 DBGET (B, DETAIL 1, 5, STAT-D1, LIST 1, BUF 1, IARG)

check STAT-D1

```
put ESD&ESA&SIM (:BUF 1)      in    [VAL-CLE-M2]
```

DBFND (B, MASTER 2, IM, STAT-M2, CLE-M2, VAL-CLE-M2)

check STAT-M2

20 DBGET (B, DETAIL 2, 5, STAT-D2, LIST 2, BUF 2, IARG)

check STAT-D2

print BUF 2

si pas fin DETAIL 2 go to 20

si pas fin DETAIL 1 go to 10

19) Pour une ligne d'une simulation donner les standards I

input (NUMX&SIX)

put M.LIGNE	in	[MASTER]
STAND I	in	[DETAIL]
NUML&SIM	in	[CLE M]
NUMX&SIX	in	[VAL-CLE-M]
MONTANT&NOM.SP	in	[LIT 1]

méta 2

20) Pour une escale ESD donner les standards II

input (ESD,SIX)

put ESCALE	in	[MASTER]
VILLE	in	[CLE M]
ESD	in	[VAL-CLE-M]
MONT.S II	in	[DETAIL]
SIM,MONTANT,NOM SP	in	[LIST]

DBFND (B, MASTER, IM, STAT M, CLE-M, VAL-CLE-M)

check STAT-M

10 DBGET (B, DETAIL, 5, STAT-D, LIST, BUF, IARG)

check STAT-D

if SIM (:BUF)=SIX

then print MONTANT (:BUF)

NOM SP (:BUF)

endif

si pas fin DETAIL go to 10

### 10.5.3.7. Equipages

21) Pour une escale, une simulation, un type d'équipage de maîtrise donnés, donner les tronçons ayant cette escale comme départ ainsi que les lignes contenant ces tronçons.

input (ESD, SIX, ITEX)

put M.TEM	in	[MASTER 1]
ITEM	in	[CLE-M1]
ITEX	in	[VAL-CLE-M1]
C.ROUT.	in	[DETAIL 1]
NUM&SIM	in	[LIST 1]
T.ROUTING PNM	in	[MASTER 2]
NUM&SIM	in	[CLE-M2]
CONTIENT M	in	[DETAIL 2]
ESD&ESA	in	[LIST 2]
TRONCON	in	[MASTER 3]
ESD&ESA	in	[CLE M3]
HORAIRE DEP	in	[DETAIL 3]
SIM, JOUR 1, ..., HEURE, NUML	in	[LIST 3]
HORAIRE ARR	in	[DETAIL 4]
SIM, NL, JOUR 1, ..., HEURE	in	[LIST 4]

DBFND (B, MASTER 1, IM, STAT-M1, CLE-M1, VAL-CLE-M1)

check STAT-M1

10 DBGET (B, DETAIL 1, 5, STAT-D1, LIST 1, BUF 1, IARG)

check STAT-D1

put NUM&SIM (:BUF1) in [VAL-CLE-M2]

```

DBFND (B,MASTER 2,IM,STAT-M2,CLE-M2,VAL-CLE-M2)
check STAT-M2
20 DBGET (B,DETAIL 2,5,STAT-D2,LIST 2,BUF 2,IARG)
check STAT-D2
put ESD (:BUF 2) in [VAL-CLE-M3]
DBFND (B,MASTER 3,IM,STAT-M3,CLE-M3,VAL-CLE-M3)
check STAT-M3
30 DBGET (B,DETAIL 3, 5, STAT-D3,LIST 3, BUF 3,IARG)
check STAT-D3
if SIM (:BUF 3) = SIX
then print BUF 3
endif
si pas fin DETAIL 3 go to 30
40 DBGET (B,DETAIL 4,5, STAT-D4,LIST 4,BUF 4,IARG)
check STAT-D4
if SIM -:BUF 4)=SIX
then print BUF 4
endif
si pas fin DETAIL 4 go to 40
si pas fin DETAIL 2 go to 20
si pas fin DETAIL 1 to go 10

```

#### 10.5.3.8. Métaalgorithme

1) input (CHARGE)

```

put C' in [DETAIL ]
a',b',c' in [LIST ]

```

```

N:=0
10 N:= N + 1
  put c (:C [N](:CHARGE)), b(:CHARGE), c (: C [N](:CHARGE))
    in BUF
  call DBPUT (B, DETAIL, 1, STAT, LIST, BUF)
  check STAT
  if N < NBRE-C
  then go to 10
  end.

2) input (a)
  put A'           in [MASTER]
     C             in [DETAIL]
     a'           in [CLE-M]
     a             in [VAL-CLE-M]
     c',b'        in [LIST]
  CALL DBFND (B, MASTER, IM, STAT-M, CLE-M, VAL-CLE-M)
  check STAT-M
10 call DBGET (B, DETAIL, 5, STAT-D, LIST, BUF, IARG)
  check STAT-D
  print BUF
  si pas fin DETAIL go to 10
  end.

```

### 10.5.3.9. Modules d'accès

A partir de l'algorithme effectif MAG, nous développons un algorithme indépendant du SGBD qui appelle des modules d'accès comptables IMAGE/HP. Si le SGBD est changé, seuls ces modules devront être réécrits. Nous rappelons d'abord l'algorithme effectif, ensuite nous donnons la procédure des modules, ensuite le programme.

Accès: Pour une ligne, donner le stand I et le nom du service produit.

#### Algorithme effectif

```
for L:= LIGNE (:NUML&SIM=NUMX&SIX)*
  for SI:= STANDARD I (:L)
    print MONTANT (:SI)
  for SP:= SERVICE PRODUIT (:SI)
```

#### Procédure

```
input (NUMX&SIX)
GET-LIGNE (L, NUMX&SIX) FOUND-L
if FOUND-L
  then GET-FIRST-ST I (S, L, FOUND-S)
  while FOUND-S do
    GET-MONTANT-ST-I (S,M)
    print MONTANT
    GET-SER.PR (SP,S,FOUND-SP)
    if FOUND = True
      GET-NOM-PR (SP,N)
      print NOM-PR
      GET-NEXT-ST I (S,L,FOUND-S)
    endif
  endwhile
endif
```

Explication des modules

GET-LIGNE (L, NUMX&SIX, FOUND-L)

données : NUMX&SIX=integer

résultat : L:LIGNE

: FOUND: boolean

fonction : obtenir l'article de la séquence L:=LIGNE  
 (:NUML&SIM=NUMX&SIX)\* obtenu par la clé  
 d'accès NUML&SIM. Si cet article est trouvé,  
 il est désigné par la variable L et FOUND-L  
 se trouve à la valeur "true" sinon FOUND-L=  
 "false".

GET-FIRST-STI (S,L,FOUND-S)

données : L:LIGNE

résultats : S:STANDARD I

FOUND-S : boolean

fonction : obtenir le premier article STANDARD I du chemin  
 ayant origine L, soit STANDARD I (:L). Si cet  
 article est trouvé, il est désigné par la variable  
 S et FOUND-S est="true", sinon FOUND-S = "false".

GET-NEXT-ST I (S,L,FOUND-S)

données : S : STANDARD I

L : LIGNE

résultats : S:

FOUND-S : boolean

GET-MONTANT-ST I (S,M)

données : S : STANDARD I

résultat : M : real

fonction : ranger dans M la valeur de MONTANT de  
l'article désigné par S.

GET-SER-PR (SP, S, FOUND-SP)

données : S : STANDARD I

résultat : SP : SERVICE PRODUIT

FOUND-SP : boolean

fonction : obtenir l'article SERVICE PRODUIT cible du  
chemin ayant S comme origine. Cet article est  
désigné par SP. S'il est trouvé FOUND-SP="true",  
sinon "False".

GET-NOM-PR (P,N)

donnée : P:

résultat : N : caractère

fonction : ranger dans N la valeur du NOM SP de l'article  
désigné par S.

Programme pseudo FORTRAN/IMAGE

input (NUMX&SIX)

\*GET-LIGNE (L,NUM&SIX)

go to 110 (\* appel du module GET-LIGNE (L,NUM&SIX,FOUND-L) \*)

5 if FOUND-L="true" then

    then go to 10

    else go to 50

\*GET-FIRST-ST I (S,L)

10 go to 120

```
* while FOUND-S do
20 if FOUND-S="true"
    then go to 25
    else go to 50

* GET-MONTANT(ST I
25 go to 125

* PRINT MONTANT
30 print M

* GET-SER-PR
go to 140

* if FOUND-SP="true" then
33 if FOUND-SP ="true"
    then go to 35
    else go to 50

* GET-NOM-PR
35 go to 130

* PRINT -NOM-PR
40 print N

* GET-NEXt-ST I (S,L)
go to 120

* endwhile
go to 20

50 end.
```

Modules

```

* GET-LIGNE (L,NUX&SIX,FOUND-L)

110 put NUX&SIX in [VAL CLE] (*la valeur de NUX&SIX est entrée
                             en input du module*)

    put LIGNE   in [MASTER]
        NUML&SIM in [CLE]

    DBFND (B,MASTER,IM,STAT,CLE,VAL-CLE)

    if STAT[o] = 0
        then FOUND-L ="true"
        else FOUND-L ="false"
        check STAT
    endif

go to 5 (* retour au programme appelant*)

* GET-FIRST(ST I (S,L, FOUND-S)
* GET-NEXT-ST I

120 put STANDARD I  in [DETAIL]
        MONTANT&NOM SP in [LIST]

    DBGET (B,DETAIL,5,STAT,LIST,BUF,IARG)

    if STAT [o] = 0
        then FOUND-S ="true"
            S :=BUF
        else FOUND-S ="false"
        check STAT
    endif

go to 120

* GET-MONTANT-ST I (S,M)

125 M:=MONTANT (:S)

go to 30

```

\* GET-NOM-PR (P,N)

130 N:=NOM.SP (:P)

go to 40

\* GET-SER-PR (SP,S,FOUND-SP)

140 SP:=S

FOUND-S="true"

go to 33.

10.5.4. Place mémoire

Soit x: nombre de simulations

Data base: 738 870 . X + 119.110 mots

Tables du SGBD                    +    1884    mots

---


$$738.870 . X + 120\ 994 \text{ mots}$$

ou encore 5772,42 X + 945,27 blocks

Les fichiers actuels occupent 15.407 blocks par saison,  
non compris la banque de données des programmes et  
trafic qui, à elle seule, prend 18.240 blocks.

Le disque disponible actuellement a une capacité de  
78.912 blocks.

La place pour la nouvelle banque de données est de:

78.912 blocks

- 19.728 blocks                    Système

- 8.964 blocks                    Programmes d'application

---

50.220 blocks

Nombre de simulations possibles:

$$X = \frac{50.220 - 945,27}{5772,42}$$

$$= 8,54.$$

Le service envisage d'acquérir prochainement un disque  
offrant une capacité de 157.824 blocks, ce qui permet-  
trait de disposer de 22 simulations.

En attendant que le service n'acquière ce nouveau disque, il est possible d'économiser de la place en limitant le nombre de simulations simultanées, présentes dans le système. En effet, lors de la constitution des axes, il n'est pas nécessaire de conserver en permanence les 10 simulations des standards ni des équipages. Par conséquent, une bonne gestion du nombre de simulations ne nécessiterait pas l'acquisition d'un nouveau disque.

Au paragraphe 10.5.1., nous avons parlé de l'éventualité de créer 2 Data Bases identiques, l'une pour le long terme, l'autre pour le court terme. Ce dédoublement n'est pas fort pénalisant en terme de place gaspillée par les redondances.

Nombre de simulations possibles dans chacune des bases:

$$2 X = \frac{50.220 - (945,25).2}{5772,42}$$

$$= 8,37$$

$$X = 4,19$$

Toutefois, il ne faut pas perdre de vue que ce dédoublement ne résout pas tous les problèmes de concurrence.

## XI - CONCLUSION

L'objet de ce chapitre est de voir dans quelle mesure le présent travail répond aux objectifs et contraintes que le service s'est assigné. Ensuite nous exposerons quelques réflexions sur la démarche suivie.

1) Rappelons brièvement les objectifs et contraintes du service.

### Objectif général

Automatiser l'élaboration des simulations budgétaires avec comme corollaire disposer des informations qui correspondent plus adéquatement aux besoins du département.

### Contraintes

- Réduire les redondances dans les fichiers et l'espace mémoire nécessaire.
- Avoir des temps de traitement aussi rapides que dans le système existant.

Le système d'information proposé au chapitre VI de la deuxième partie répond à nos yeux à l'objectif général. De plus, nous y ajoutons des objectifs opérationnels qui, une fois atteints, permettent la réalisation de l'objectif général. Dans ce même chapitre VI, nous avons vu qu'un moyen pour atteindre ces objectifs opérationnels était de développer un nouveau système de gestion des

données. C'est pourquoi, compte tenu des objectifs et des contraintes, nous avons proposé une banque de données aux caractéristiques suivantes:

1) Information

La banque de données contient à nos yeux tous les objets conceptuels nécessaires. Chacun de ces objets a été défini et possède un identifiant exempt d'ambiguïté.

2) Place

En ce qui concerne la place mémoire nécessaire, un gain important est obtenu. Ce gain peut encore être amélioré si une bonne gestion du nombre de simulations est mise en oeuvre.

3) Temps de traitement

Nous avons choisi les chemins d'accès les plus performants. Toutefois, afin d'améliorer les temps de traitement, nous proposons qu'une bonne gestion de la concurrence soit également mise en oeuvre.

4) Extensions

4.1. Calcul des standards

La banque proposée contient déjà les composants permettant le calcul des standards suivants:

- eurocontrol
- survol
- logement
- catering passager

- fuel
- documents de vol.

#### 4.2. Devises

La présente banque ne contient aucune information relative à l'aspect devise, excepté pour le survol. Nous pouvons envisager que le concept de devises devienne un attribut des associations: recettes réalisables, quotas vente, quotas moyens, standards, selon le schéma suivant:

##### QUOTA.V :

Définition: si TR E T.RECETTE et si R E RELATION  
 <TR,R> indique l'objectif de vente pour une recette TR pour la relation R.

Atributs: - QV (R&D)

- recette unitaire en \$
- nombre d'unités de recette en \$
- recette unitaire en FB
- nombre d'unités de recette en FB
- recette unitaire en d'autres devises (exprimée en FB)
- nombre d'unités de recettes en autres devises
- SIM

où tous les taux de change sont définis par la simulation.

Une fois ces attributs définis, on peut les retrouver dans les items des différents schémas.

#### 4.3. Budget intégré

Pour le budget intégré, il y a lieu de définir de nouvelles entités et associations, telles que:

##### I. recette industrielle

Définition: type de recette associée à une prestation de service qui n'est pas de transport.

Attribut : NOM RI : nom de la recette industrielle

MONTANT: montant exprimé en FB.

Si ces nouveaux objets respectent les contraintes d'intégrité des objets déjà définis, nous pouvons les retrouver dans les différents schémas.

#### .2. Démarche suivie

Nous pensons qu'il faut étudier la démarche suivie sur deux plans:

- A partir d'un système informatique classique de traitement de données (EDP), réaliser une banque de données qui permet le développement ultérieur de systèmes d'aide à la décision (D.S.S.).
- La méthode suivie pour la conception de la banque de données.

##### 2.1. De l'EDP vers le DSS.

Réaliser une banque de données comme première étape de l'installation d'un DSS offre à nos yeux les avantages suivants:

- améliorer la qualité de l'information
- une fois que les concepts sont rigoureusement définis, l'utilisateur voit les extensions possibles et par conséquent, la puissance informationnelle dont il peut disposer.
- une fois la banque réalisée, l'utilisateur ne pouvant ou ne voulant pas développer un DSS peut toujours l'utiliser dans les applications EDP existantes.

Par contre, l'étendue de l'application nécessite une longue analyse conceptuelle, ce qui empêche de fournir rapidement un produit exécutable et par conséquent, nous risquons de perdre la motivation et l'intérêt de l'utilisateur.

#### 11.2.2. Conception de la banque de données

L'approche très top-down de la méthodologie et l'indépendance des différents niveaux permet au concepteur de réaliser un produit qui peut à tout moment évoluer et s'adapter soit aux besoins de l'utilisateur, soit au SGBD choisi. De plus, le rapport détaillant les différentes étapes constitue une excellente base pour la documentation, la maintenance et même l'extension du produit.

Dans notre réalisation, nous avons identifié des moments critiques dans l'élaboration de la banque de données.

En effet, vu la pauvreté de la documentation existante et l'étendue du problème, la réalisation et la validation

du schéma conceptuel est longue et nécessite de nombreuses entrevues avec les utilisateurs.

Le caractère fastidieux de ces entretiens peut amener à une perte d'intérêt et de motivation de l'utilisateur. Par conséquent, nous pensons que c'est au concepteur de percevoir ce danger. S'il lui semble que ce risque devient éminent, il doit, selon nous, fournir immédiatement une première version du produit exécutable, limitée à une partie de l'application. Ensuite revenir au modèle conceptuel et continuer la démarche.

Une fois qu'une partie de l'application est opérationnelle, nous pensons qu'un deuxième danger peut survenir. En effet, le concepteur peut être tenté de passer directement du modèle conceptuel au schéma du SGBD et ce, à cause des contraintes de temps, liées au développement d'un DSS. Ce raccourci, quoique tentant, est à nos yeux, extrêmement dangereux, car l'optimisation des chemins et l'indépendance par rapport au SGBD et à la machine ne sont plus assurées.

Finalement, nous pensons que la réalisation d'une banque de données peut être une bonne approche comme première étape de développement d'un DSS, de même la méthode de conception utilisée. Toutefois, le concepteur doit dès le début limiter l'étendue de l'application et être vigilant à tout moment au niveau de la motivation et de l'intérêt de l'utilisateur.

BIBLIOGRAPHIE

- (1) F.BODART, Y. PIGNEUR : Modèles, outils et méthodes d'aide  
au développement d'un système d'infor-  
mation. Première partie: Etude d'oppor-  
tunité et analyse conceptuelle.  
Namur 1983
- (2) BODART : An introduction to decision support systems.  
Namur 1983.
- (3) HAINAUT J.L. : Cadre de référence pour la conception  
de bases de données.  
Namur 1983.
- (4) HAINAUT J.L. : Programmation d'application sur bases  
de données.  
Namur 1984.
- (5) HEWLETT & PACKARD : Image/1000. Reference manual.