

# **Troisième partie**

## **Niveau organisationnel**



# Chapitre 6

## Modèle organisationnel des traitements (MOT)

Après avoir défini le niveau conceptuel, qu'on pourrait résumer au quoi faire, il convient de définir le comment.

Dans le MOT, interviennent des notions nouvelles :

- le poste de travail
- le personnel (éventuellement spécialisé dans différents rôles)

À ce niveau, on considère généralement plusieurs solutions qu'il s'agit d'évaluer (prise en compte de critères économiques, techniques, ergonomiques et sociaux). Certaines solutions peuvent induire des changements dans l'entreprise. Il est à noter que le MOT se trouve à la charnière entre le système organisationnel de l'entreprise et son système organisationnel informatisé. En général, c'est le personnel qui servira d'interface entre les deux.

### 6.1 Éléments constitutifs du MOT

#### 6.1.1 postes de travail

Ils se caractérisent par

- les compétences des employés qui le servent
- les caractéristiques du matériel qui lui est associé
- leur aménagement
- leur localisation

Un poste de travail type peut comporter une ou plusieurs personnes qui y travaillent, voire même zéro (lecteur automatique de carte ou de badge) et avoir ou non du matériel informatique. En outre, à un même poste type peuvent correspondre plusieurs occurrences de poste.

#### 6.1.2 tâches

Les opérations conceptuelles sont cette fois découpées en tâches, concourant à un but précis. Chaque tâche se caractérise par :

- un **seul poste type** (on peut toutefois parler de tâches similaires sur des postes différents).
- son degré d'automatisation :
  - **M**anuelle,
  - **C**onversationnelle ou
  - **A**utomatique)

- son délai de réponse :
  - **I**mmédiate ou
  - **D**ifférée
- son mode de fonctionnement :
  - **U**nitaire (les ressources ne sont libérées qu'à la fin d'une occurrence de tâche) ou par
  - **L**ot (toutes les occurrences sont traitées en une seule fois).

La tâche peut aussi être décrite par des règles de traitement (sous forme d'un algorithme appliqué à un certain nombre de données). On assigne souvent à une tâche un sous-ensemble de données repris dans le schéma entité-relation et auquel la tâche fait référence (mise à jour ou lecture). Ce sous-schéma ne doit reprendre que les informations pertinentes pour la tâche. Ce type de schéma servira de base à la définition des autorisations d'accès aux bases de données.

Présentés sous la forme de messages, les événements/résultats font l'objet d'une double analyse :

- en terme de succession d'événements que le niveau conceptuel ne distinguait pas (ex. demande  $\Rightarrow$  demande initiale, demande modifiée, demande finale).
- en terme d'alternatives : des événements similaires sont maintenant distingués, parce qu'ils entraînent des tâches différenciées.

Des événements non spécifiés par le MCT apparaissent à présent, notamment les décisions du pilote (début de mois, fin de semaine...) ou les décisions d'un acteur (faisant partie ou non du poste de travail).

On peut ou non représenter les ressources à ce niveau (ce peut être par exemple un terminal disponible). Généralement, cet aspect est laissé de côté. Pour certaines tâches, il est plus clair de les mentionner.

### 6.1.3 phases

On entend par là « une succession de tâches exécutées consécutivement au sein d'un même poste ». Comme les ressources seront mobilisées pendant tout son déroulement, la phase a une importance sur le rythme de travail, sur les risques de file d'attente.

### 6.1.4 procédures organisationnelles

Elles permettent de donner une vision plus globale des traitements : elles partent d'un événement initial et structurent un ensemble de tâches/phases qui concourent à produire un résultat final. Notons qu'à ce niveau, on est souvent amené à parler de variantes, dont il sera bon de spécifier la fréquence.

### 6.1.5 représentation graphique

La complexité de la représentation fait qu'une triple représentation est souvent nécessaire : en termes de procédures (pour dégager les grandes lignes), en termes de phases et en termes de tâches.

On réservera des colonnes pour les postes ou les acteurs externes (parfois tous dans la même colonne). L'axe vertical représentera la chronologie. Les tâches, dont le formalisme est proche des opérations du MCD, figureront dans les colonnes réservées aux postes (voir exemple plus loin).

## 6.2 Eléments importants pour la formalisation du MOT

### 6.2.1 importance du facteur temps :

On va définir à ce niveau une série de spécifications de fréquences, de durée, de délai. On citera en vrac :

- fréquence d'un événement,
- durée d'une tâche,
- nombre maximum d'occurrences qu'un événement type peut accepter (capacité),
- nombre d'occurrences d'événements nécessaires au déclenchement d'une opération,
- temps maximum entre la survenance d'un événement et sa consommation par une opération,
- durée limite de synchronisation (avant purge),
- délai de synchronisation,
- nombre de duplications d'un résultat...

### 6.2.2 éléments de la description

Parmi les documents produits, si les tâches et les postes sont nombreux, on fera des listes et des documents de synthèse, comme par exemple (liste non exhaustive) :

- description des postes,
- schémas d'enchaînement des tâches,
- listes des phases par poste,
- pour chaque tâche :
  - messages d'entrée (écrans, bordereau)
  - règles de traitement sous forme d'algorithmes
  - sous-schéma conceptuel ou organisationnel de données
  - actions sur les données mémorisées
  - résultats produits (écrans ou états)

**Bien que ce paragraphe paraissent bien discret, les éléments de la descriptions représentent un travail considérable, n'utilisant pas de formalisme unique. Tous les moyens sont bons pour opérer la description. On veillera cependant à garder une certaine cohérence dans la présentation des différentes tâches..**

### 6.2.3 prise en compte de l'ergonomie

Si le projet présente une certaine ampleur, on fera appel à des théoriciens et des spécialistes de l'ergonomie qui évalueront les solutions proposées en prenant en compte les opinions des travailleurs concernés et l'intégration de ces solutions dans la vie de l'entreprise. Pour ce qui concerne les interfaces logicielles, on veillera à leur cohérence, au respect des habitudes déjà ancrées et de la possibilité d'une nouvelle formation des utilisateurs.

## 6.3 Exemple

L'exemple développé montre le cas d'une société, vendant des articles d'électricité. Le MCT avait isolé une opération de «vente au comptoir» qui se terminait par un échec (en cas

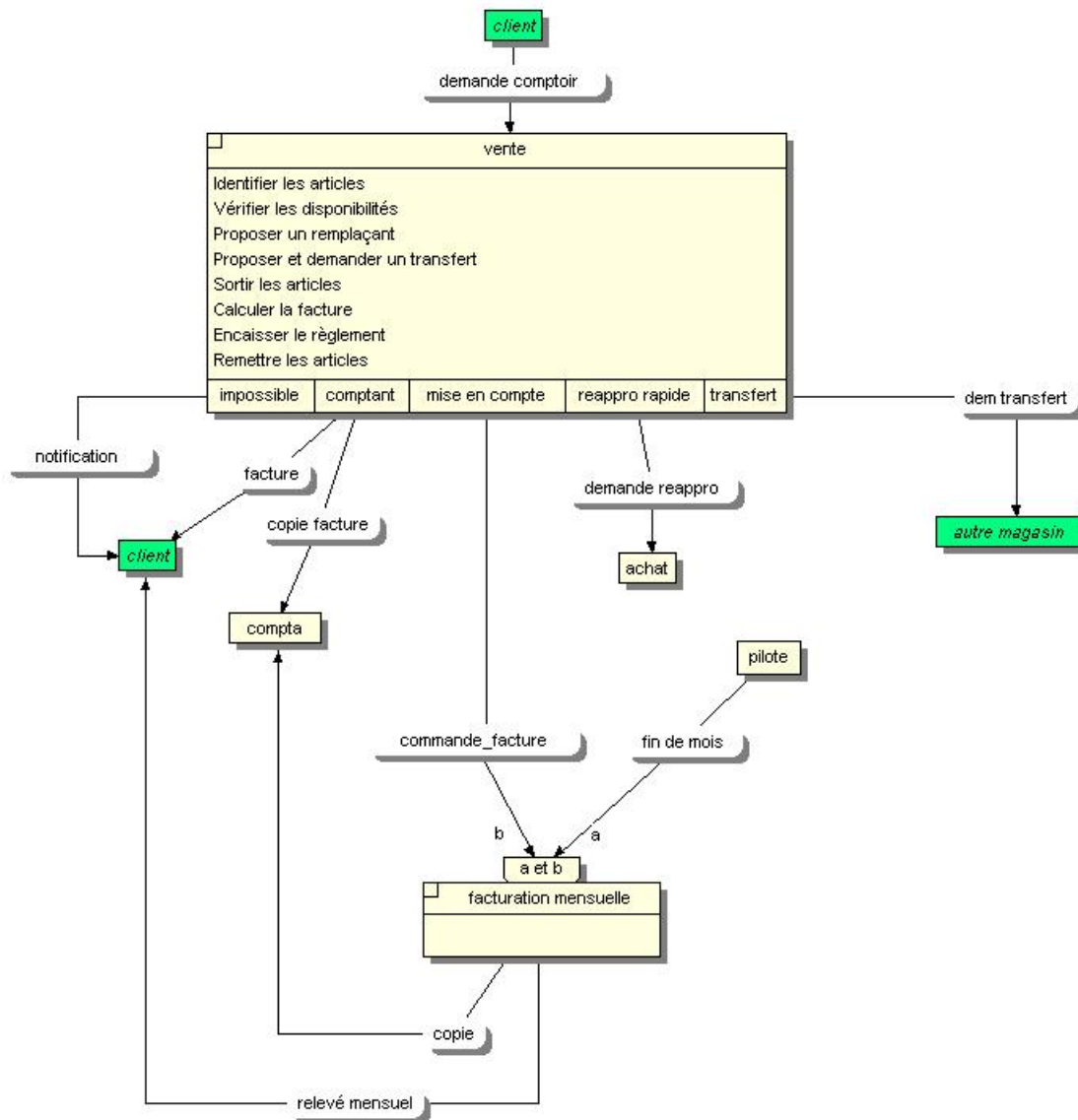


FIG. 6.1 – MCD partiel reprenant les ventes au comptoir

d'inexistence d'un article), par une décision de transfert (d'un autre magasin de la firme), par une demande de réapprovisionnement rapide, une vente au comptant ou une vente par facturation du compte du client.

On va proposer ici quatre MOT différents<sup>1</sup>.

### 6.3.1 4 postes de travail

On distingue 4 postes de travail (voir Fig. 6.2), auxquels on peut placer du personnel aux compétences diverses :

1. un comptoir, occupé par des magasiniers aux compétences techniques importantes, capable d'interroger et de renseigner le client ;
2. à la caisse, un caissier ayant la responsabilité commerciale et financière ;
3. au stockage, des manutentionnaires qui vont chercher les articles ;

<sup>1</sup>L'exemple et ses solutions est repris de NANCY et ESPINACE.

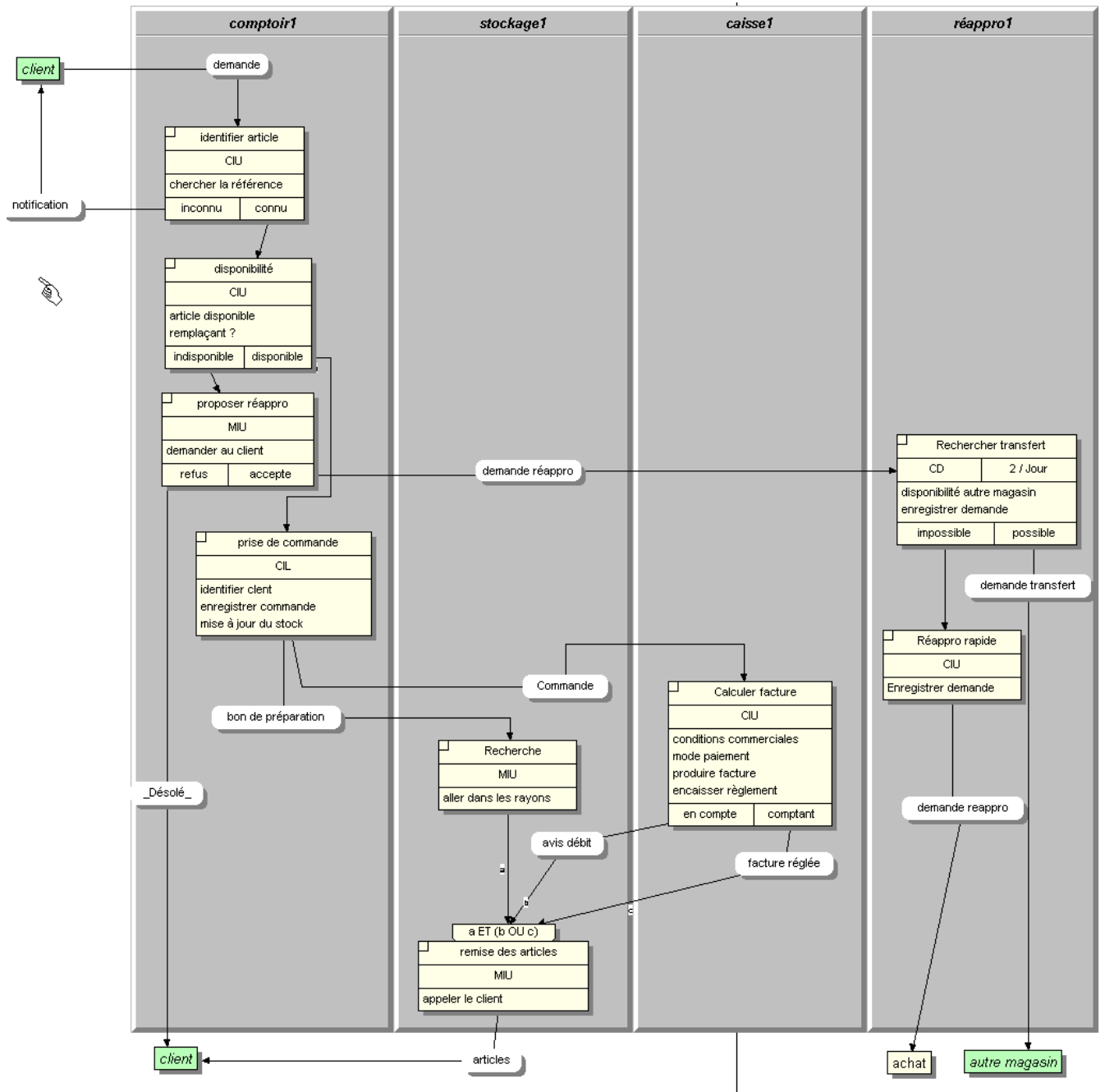


FIG. 6.2 – Le magasin avec 4 postes de travail

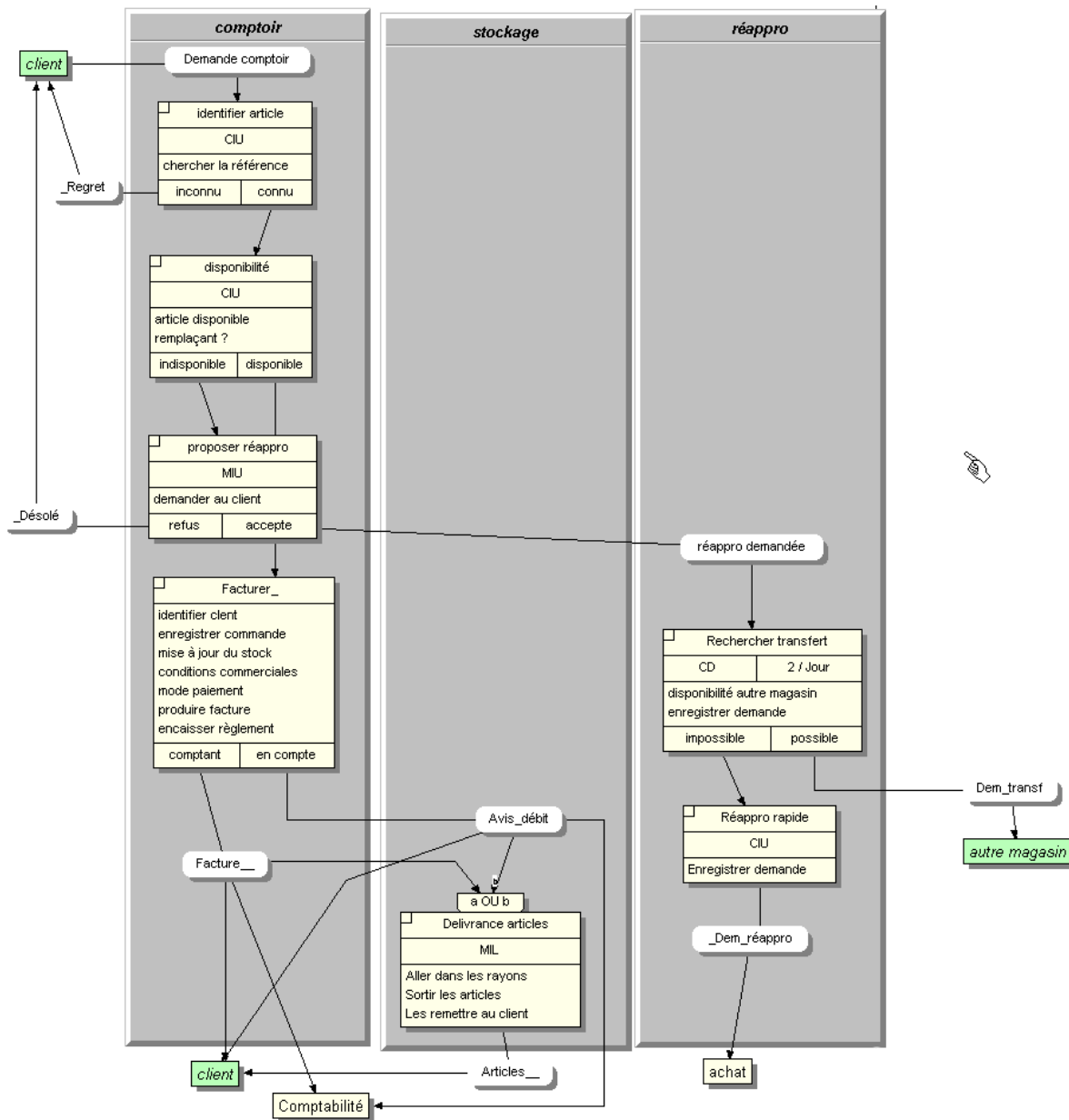


FIG. 6.3 – Le magasin sans poste-caisse

4. au réappro, un personnel chargé de trouver un article manquant soit par transfert d'un autre magasin soit par commande rapide.

L'un des inconvénients est que la facturation est faite avant que le client ait vu ses articles, de même que la mise à jour des stocks. Si le client change d'avis au moment de la livraison (non-conformité), on devra annuler des opérations, d'où perte de temps.

### 6.3.2 Suppression du poste caisse

Dans cette optique (voir Fig. 6.3), on suppose que la plupart des clients sont en compte et que les conditions commerciales sont bien connues du système. L'inconvénient est que les opérations de recherche et de facturation sont effectuées par le même employé, ce qui allonge le temps consacré à chaque client et risque d'occasionner des files. On rattrape un peu de temps sur l'absence de paiement.

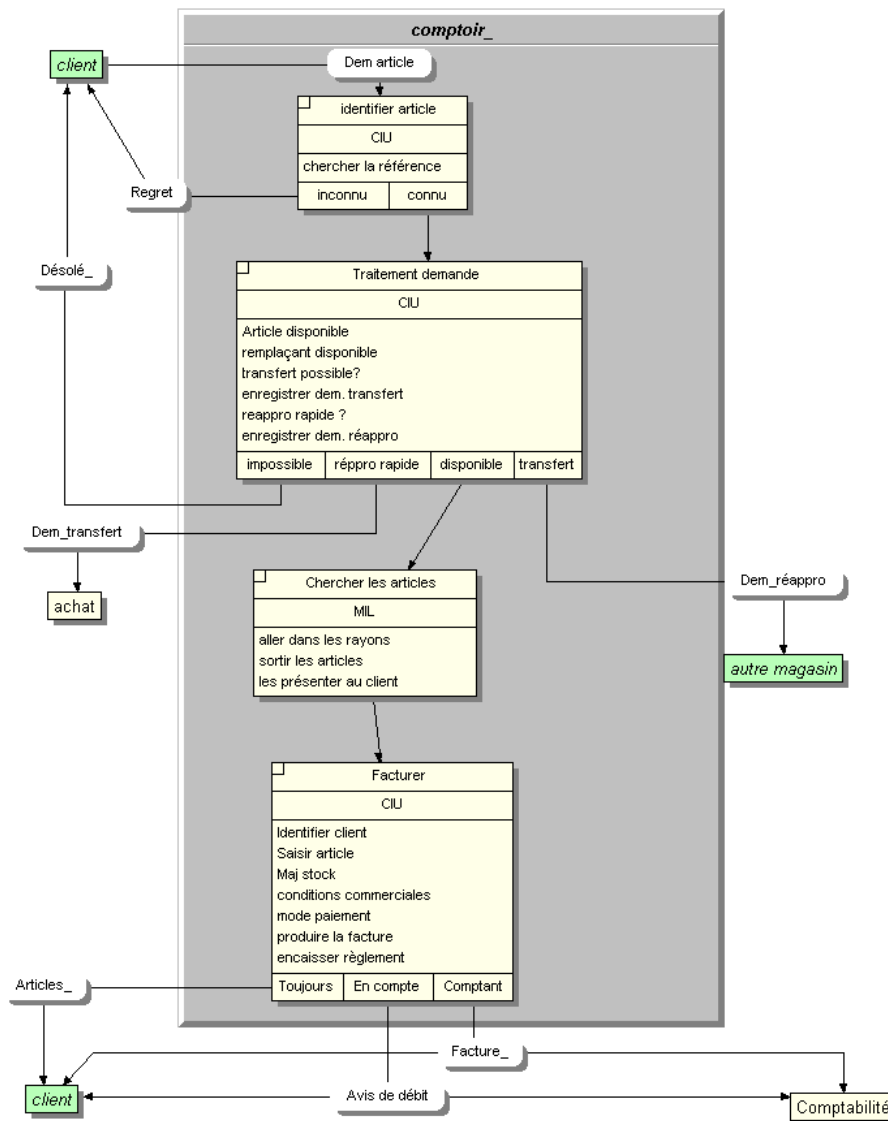


FIG. 6.4 – Le magasin avec un seul poste de travail

Les principaux défauts de la solution n°1 subsistent (retour d'une marchandise non approuvée par le client).

### 6.3.3 Concentration sur un seul poste

Chaque employé va dispenser les conseils, aller chercher la marchandise, établir la facture et, éventuellement, encaisser le paiement (voir Fig. 6.4). Les défauts des deux premières solutions disparaissent parce que le client verra sa marchandise avant de confirmer l'achat. Par contre, chaque poste informatisé va être sous-employé pendant la recherche et, si celle-ci est salissante, l'utilisation du clavier va poser problème.

### 6.3.4 Création d'un libre service

Cette solution (voir Fig. 6.5) remet en cause toute l'organisation du magasin (frais d'aménagement accrus, notamment pour garantir la sécurité, espace probablement plus grand...). La solution risque d'être mal adaptée si les clients ont souvent besoin de conseils. Notons que

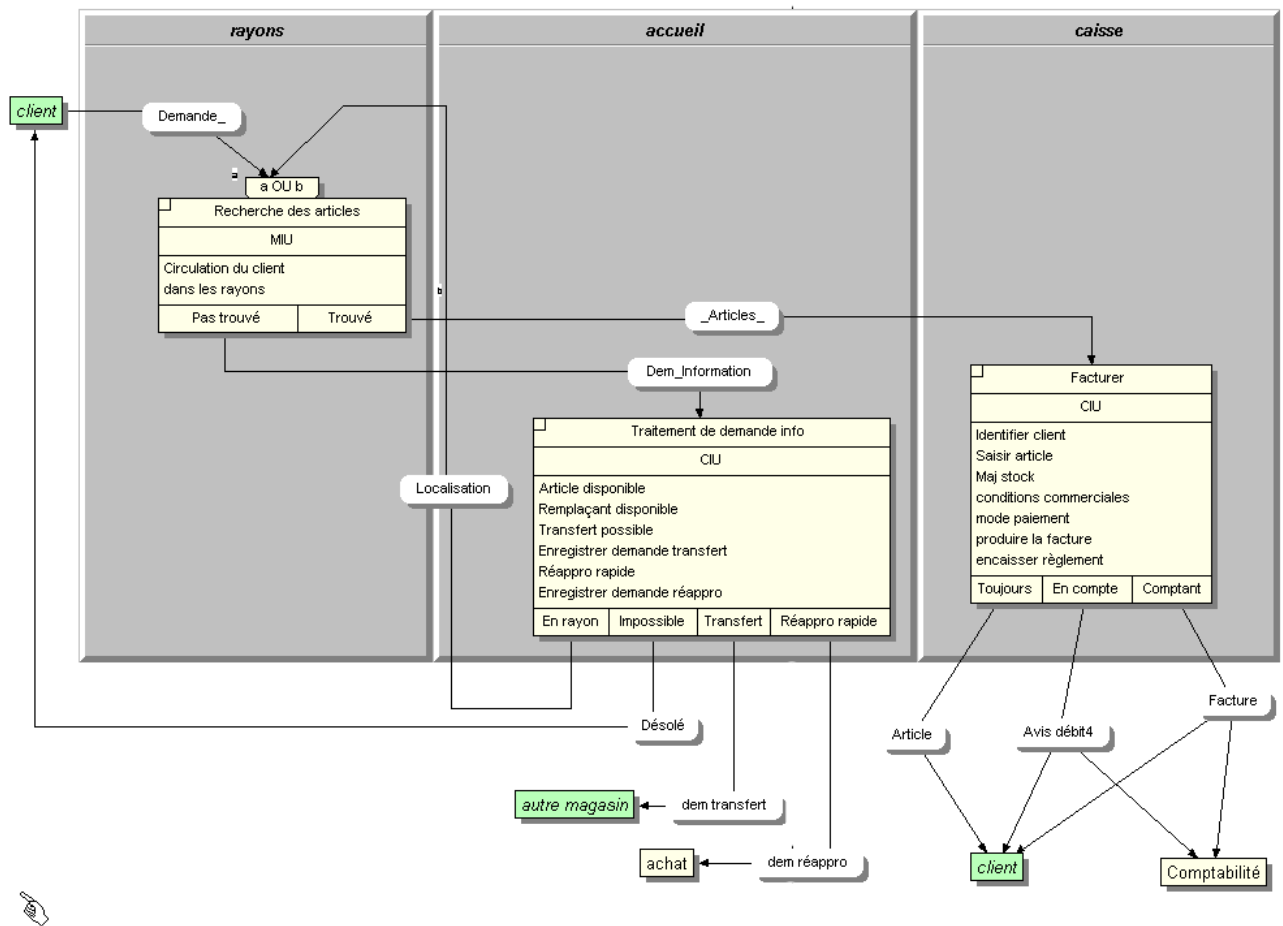


FIG. 6.5 – Le magasin en libre service

dans cette optique, le client devient lui-même une ressource.

# Chapitre 7

## Modèle organisationnel des données (MOD)

Après avoir déterminé, dans le modèle conceptuel de données, quelles données permettent de représenter le domaine auquel s'applique l'analyse, on va passer à des aspects plus pratiques. Le modèle organisationnel des données va

- déterminer quelles données précises devront réellement figurer dans le système informatique
- préciser leur taille et leur nombre afin d'évaluer les besoins matériels à prévoir
- évaluer la durée de vie des données pour prévoir l'évolution des besoins matériels
- organiser la répartition de données entre les différents postes de travail déterminés par l'analyse organisationnelle des traitements.

### 7.1 Choix des données à mémoriser

Le MOD va apporter des modifications par rapport au MCD :

- élimination des entités, relations et propriétés qui ne nécessitent aucun traitement informatique (par exemple celles qui figurent sur des documents de travail donnés à des manutentionnaires) ;
- modifications d'autres éléments induites par ces suppressions ;
  - ajout d'informations ;
  - référence à des données qui continueront à faire l'objet d'un traitement manuel (numéros de bons, fiches ou dossiers) ;
- mémorisation d'états du système après certains traitements du MOT.

### 7.2 Quantification du MOD

#### 7.2.1 Taille

Il est extrêmement difficile de quantifier la taille des données à ce niveau. Ce devait être plus facile au moment où la première méthode Merise a été définie, puisqu'on pouvait réduire toute représentation à celle d'une chaîne de caractères ou à l'un des types numériques classiques :

- $An$  (champ alphanumérique de  $n$  caractères)
- $Nn$  (champ numérique de  $n$  chiffres)
- $Nn, p$  (champ numérique de  $n$  chiffres significatifs et  $p$  chiffres après la virgule)

Les programmes modernes font usage des SGBDR, dans lesquels la taille prise par les données est plus difficile à quantifier avec certitude (méconnaissance de la représentation interne, poids apporté par les index et les clés). En outre, avec l'introduction de langages à objets, les données elles-mêmes ne sont plus indépendantes du code. Dans tous les cas, la non-connaissance des représentations choisies aux niveaux logique et physique rend ce calcul hasardeux. On peut néanmoins risquer une approximation. On ne cherche pas une quantification exacte, il suffira d'évaluer le volume des données à traiter, et surtout, à stocker, afin de prévoir les ressources matérielles nécessaires à leur gestion.

## 7.2.2 Nombres d'occurrences des entités et des relations

### a. durée de vie

On doit ici distinguer la mémoire immédiate de la mémoire à long terme (par recours aux archives). Des données vont subsister en mémoire immédiate jusqu'au moment où leur présence ne s'imposera plus. On les archivera (en éliminant peut-être certains « détails » devenus inutiles) sur des supports moins coûteux, mais d'accès moins aisé. Se pose ainsi la question de la durée de vie des entités et des relations. Celle-ci peut être :

**indéterminée** (dépend des aléas de la réalité : exemple du compte qui ne sera archivé qu'un an après sa clôture) : le nombre d'occurrences dépend du nombre d'éléments introduits lors de la création plus ceux que permet de prédire la fréquence de création ;

**déterminée** (fixée lors de sa création) : le nombre d'occurrences y est proportionnel à la fréquence de création et à la durée de vie.

La prise en compte de la durée de vie impose en outre un certain nombre de réajustements du MCD. Certaines données peuvent résulter d'un calcul et ne doivent normalement pas être mémorisées (par exemple, le chiffre d'affaires annuel d'un client). A partir du moment où les informations nécessaires au calcul disparaissent après un certain laps de temps, il devient nécessaire de reconsidérer le modèle et d'y introduire la donnée sous une forme primaire (nouvelle propriété, voire nouvelle entité ou relation).

Dans un programme de facturation, on décidera par exemple de garder les données concernant une facture durant le nombre d'années prescrit par la loi (10 ans en Belgique). Après ce laps de temps, on peut décider de supprimer les données. Cependant, cette suppression entraîne l'impossibilité d'établir des statistiques sur les achats réalisés par les clients, sur le nombre d'articles vendus etc...

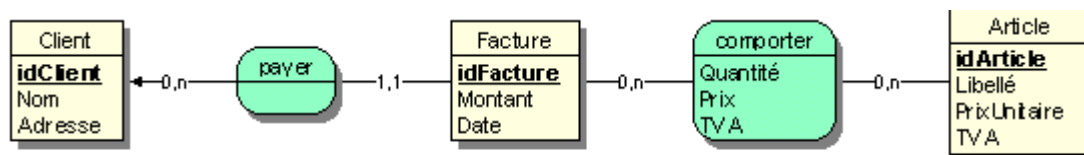


FIG. 7.1 – Modélisation de factures (MCD)

On peut y suppléer en ajoutant des propriétés comme les chiffres d'affaire et le nombre d'années sur lequel il porte pour le client, le nombre de ventes pour les articles (voir figure 7.2).

Si on veut une analyse plus fine, on pourra créer une entité *Année* et des relations *Achats* et *Ventes* capables des restituer des statistiques annuelles concernant les données supprimées (voir figure 7.3).

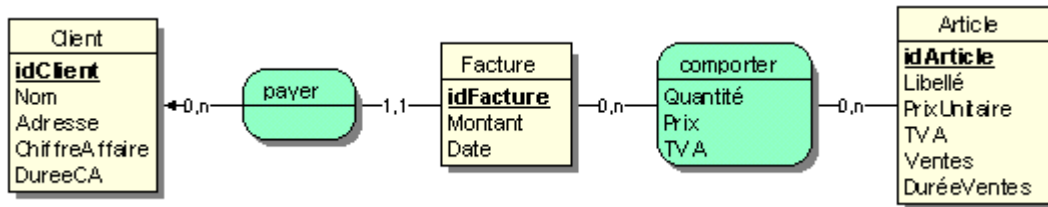


FIG. 7.2 – Factures avec conservation des CA et volumes de ventes moyens (MOD)

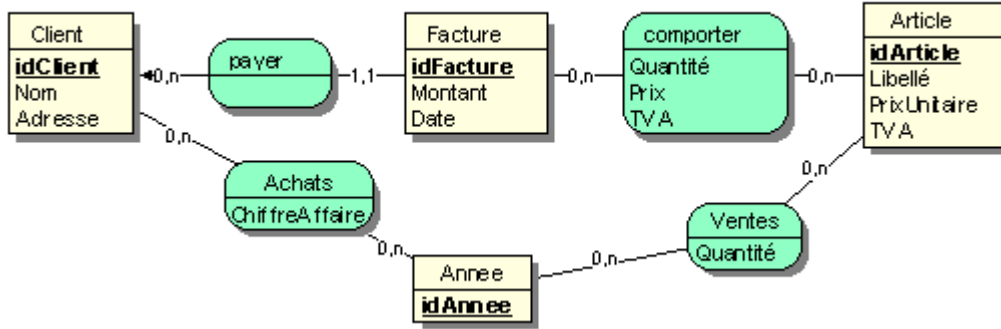
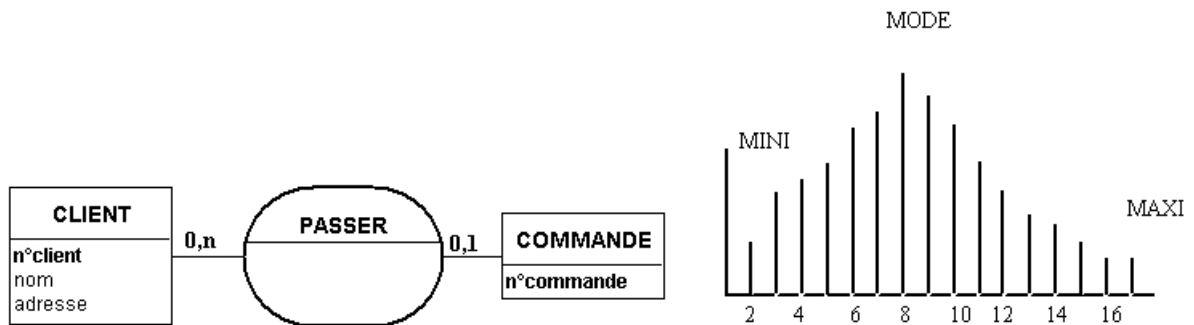


FIG. 7.3 – Factures avec conservations de CA et volumes de ventes annuels (MOD)

**b. cardinalités**

La cardinalité spécifiée dans le MCD se limitait à 0, 1 et N. On va maintenant tâcher de préciser d’avantage le nombre d’occurrences des individus et des relations, de manière à pouvoir disposer d’informations utiles pour le choix de moyens d’archivage (taille prévue des fichiers ou des tables).

Des formules complexes sont mises en œuvre. On va déterminer une valeur moyenne pour la cardinalité qui se basera sur l’observation des données réelles.



Si on examine la passation d’une commande, il convient de chiffrer le nombre de clients et de commandes (des données objectives). Dans la pratique, on peut supposer une répartition des occurrences approximativement triangulaire. En examinant un échantillon des commandes, on va déterminer quatre nombres :

- m** : le nombre minimum de commandes par client (en excluant 0),
- N** : le nombre maximum de commandes
- M** : la valeur modale (celle qui a le plus fort pourcentage)

$P$  : le taux de participation (100% - pourcentage de clients sans commande).

La cardinalité moyenne de clients sera donnée par la formule :

$$\frac{m+2M+N}{4} \times P$$

En multipliant cette cardinalité par le nombre de clients, on pourra se faire une idée raisonnable du nombre d'occurrences de la relation Passer.

## 7.3 Répartition

### 7.3.1 Le MOD local

Le MOD local correspond à une unité organisationnelle (service ou entité géographique) dont tous les utilisateurs partagent une vue commune d'un ensemble de données. Le découpage peut se faire en termes d'entités, relations et propriétés, mais aussi en terme d'occurrences (une agence ne gère que les dossiers qui lui sont propres). Ces limitations sont exprimées :

- par un schéma des entités, relations et propriétés utilisées ;
- par une liste des restrictions sur les occurrences.

On en viendra ainsi à distinguer :

- les données communes à l'ensemble du domaine ;
- les données partagées entre certaines unités ;
- les données privées de certaines unités.

### 7.3.2 Accessibilité

On précisera en outre quel type d'accès est souhaitable pour les utilisateurs des données :

<b>C</b> réation	<b>L</b> ecture
<b>M</b> odification	<b>S</b> uppression

Ces différents modes d'accès seront spécifiés sur chaque entité et chaque relation des schémas partiels<sup>1</sup>.

## 7.4 Sécurité

Il convient ici de spécifier plus avant les modes d'accès aux données en apportant des restrictions supplémentaires dans la perspective d'une sécurisation. Lors de l'étude de la répartition des données, on a surtout vu les accès nécessaires à un traitement. Du point de vue de la sécurité, il faut considérer plutôt ce qui est interdit. On peut par exemple limiter l'accès à certaines occurrences d'individus, voir même prévoir des restrictions sur les valeurs possibles. Dans l'exemple suivant, une même propriété *NiveauDécouvert* sera accessible en lecture par un employé et en lecture/écriture par le directeur. Le chef de service aura un accès en lecture et en écriture, mais en imposant une limite maximale sur la valeur du découvert. Pour les deux premiers utilisateurs, les droits définis par SQL seront suffisants, pour imposer des limites plus fines, il faudra recourir à de la programmation (idéalement, procédures stockées ou *triggers*).

<sup>1</sup>On remarquera que ces quatre types d'accès correspondent précisément aux quatre verbes de SQL : INSERT, SELECT, UPDATE et DELETE. Ce langage dispose des commandes GRANT et REVOKE dont l'effet est de donner ou reprendre le droit d'utiliser chacun des quatre commandes.

Profil utilisateur : Employé	
Entité-Relation	Restrict./autor.
CLIENT <i>Propriété</i> niveau découvert	Lecture seule

Profil utilisateur : Directeur	
Entité-Relation	Restrict./autor.
CLIENT <i>Propriété</i> niveau découvert	Modif. autorisée

Profil utilisateur : Chef de service	
Entité-Relation	Restrict./autor.
CLIENT <i>Propriété</i> niveau découvert	Modif. autor. si $\leq 10.000F$

## Exercices

**Exercice 1 :** Établir le MCD et le MOD d'une entreprise vendant des ordinateurs :

- pour chaque client on établit un bon de commande reprenant les spécifications du PC (cartes, mémoires, processeur, périphériques) ;
- les employés monteurs reçoivent chacun ces bons marqués à leur nom par le chef d'atelier et procèdent au montage ;
- un testeur vérifie la conformité du PC et le met au banc d'essai ;
- les PC, une fois conformes, sont livrés aux clients par camions, ces camions participent à des tournées et sont pilotés par des chauffeurs.

A la réflexion, on s'aperçoit qu'il n'est pas nécessaire de mémoriser le nom des monteurs. Cependant, les bons de montage sont archivés dans des classeurs.

**Exercice 2 :** En reprenant le MCD de la bibliothèque examiné lors des chapitres précédents, établir le MOD des utilisateurs effectuant des recherches sur les ouvrages.

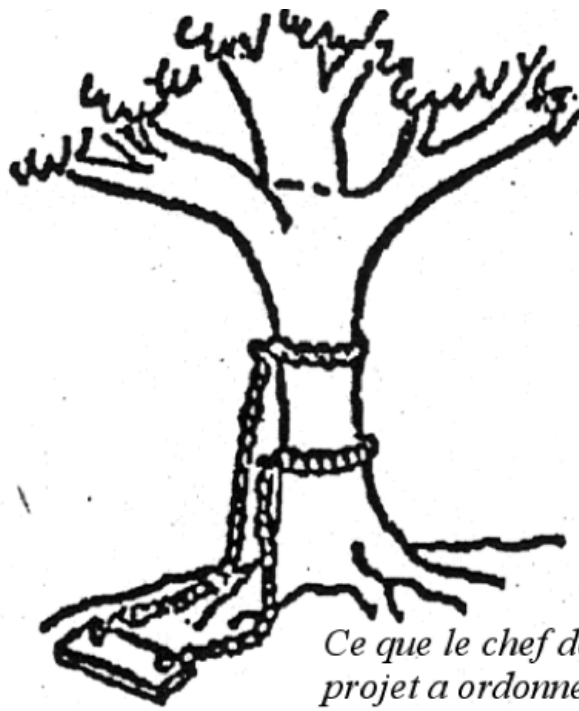
**Exercice 3 :** Définir les règles de sécurité à mettre en œuvre dans une bibliothèque si on veut que les lecteurs puissent effectuer eux-mêmes les réservations (un lecteur pourra gérer ses propres réservations, voir qu'un ouvrage est réservé, mais pas nécessairement connaître celui qui l'a réservé).

**Exercice 4 :** Dans la firme de montage du premier exercice, on distinguera trois services distincts :

- le service commande
- le service livraison
- le service après-vente (en cas de panne sous garantie)

Établir les MOD locaux des trois services sur base du MOD global. Définir les protections mises en place.

3



*Ce que le chef de projet a ordonné*

# Chapitre 8

## Confrontation des données et des traitements

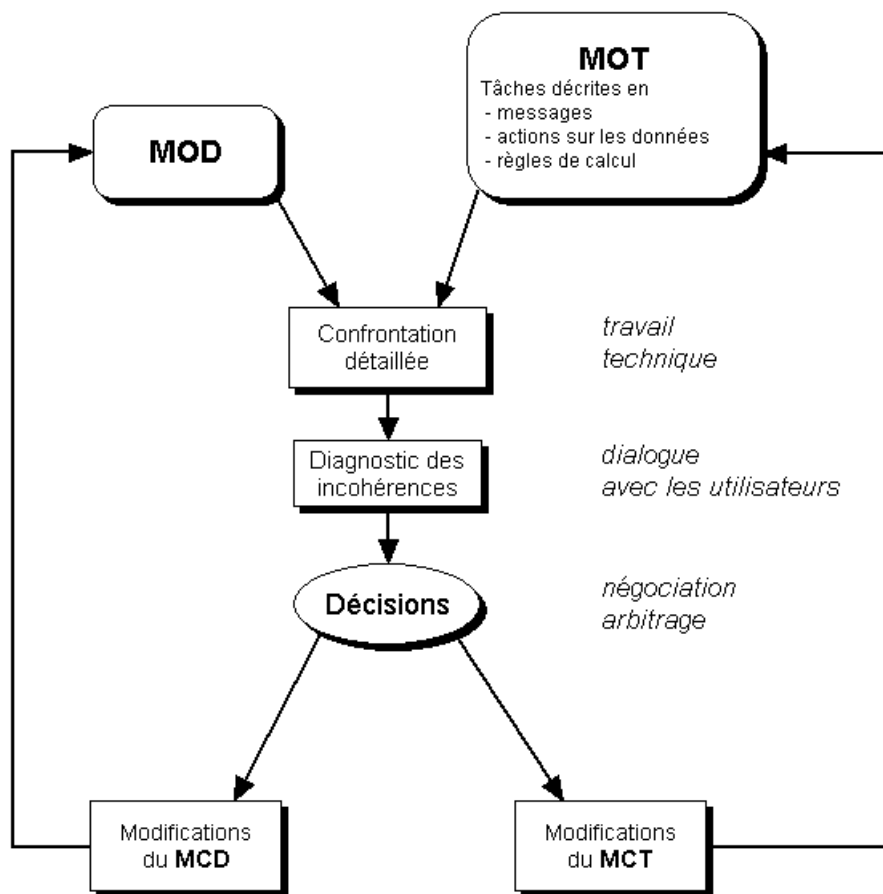


FIG. 8.1 – La confrontation des données et des traitements

La distinction de deux modèles distincts pour les données et les traitements est l'un des points centraux de la méthode MERISE. Elle tend à lutter contre deux excès inverses :

- réduire le travail de l'informaticien à la seule conception des algorithmes<sup>1</sup>,
- se concentrer sur le seul modèle entité/association et considérer les traitements comme allant de soi.

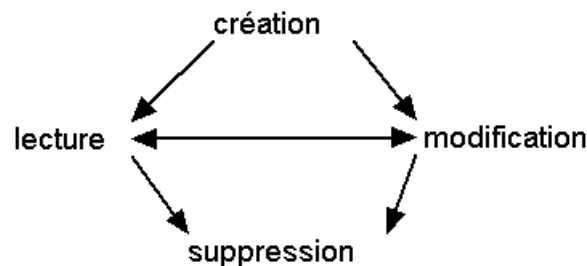
<sup>1</sup>C'est contre cette idée que Wirth, le concepteur de Pascal a lancé sa célèbre équation :  
*Programs = Algorithms + Data*

Il est certain que cette indépendance a ses limites. Le récent développement de la programmation objet vient d'ailleurs poser la question de la légitimité de cette démarche. Mais les concepteurs de Merise n'ont pas attendu la venue de ces nouveaux modes de programmation pour se pencher sur la cohérence entre les deux modèles. Le schéma de la figure 8.1 illustre la démarche. Les MOD/MOT sont confrontés et toute découverte d'incohérence entraîne la remise en cause d'un des deux modèles sous-jacents (MCD ou MCT). Ces modifications entraînent une relecture des MOD/MOT et une nouvelle confrontation.

## 8.1 Relecture croisée MCD/MCT

Il s'agit d'une opération relativement élémentaire, mais néanmoins très efficace. En reliant, le MCT, on vérifie que les entités et les associations évoquées dans les opérations sont présents dans le MCD. A l'inverse, toutes les données du MCT doivent être manipulées par au moins une opération. On la réservera à l'**étude préalable**.

## 8.2 Grille de cohérence globale MCD-MOD/MOT



Cette grille sera élaborée aussi bien en **étude préalable** que dans la première phase de l'**étude détaillée**. Chaque occurrence d'entité ou d'association doit passer par différents stades (création, lecture, modification et suppression). On établit une grille reprenant respectivement les tâches et les entités et associations. A chaque intersection, on place les opérations concernées (CRE-LEC-MOD-SUP). On peut ainsi facilement repérer les problèmes :

- entités/rerelations jamais créées (« oubli »).
- entités/rerelations créées par plusieurs opérations. Théoriquement possible, cette situation pose néanmoins le problème de la responsabilité de création.
- entités/rerelations jamais modifiées. A nouveau possible en théorie, mais uniquement pour les données sensibles. Dans la plupart des cas, on a tout simplement oublié de prévoir cette opération (par exemple, changement d'adresse d'un client). Dans certains cas prévus par la loi, il est interdit de modifier certaines données comptables. Si ce respect est à prendre en compte, il ne faut pas bloquer l'utilisateur par un programme interdisant toute erreur. Il faut alors prévoir des procédures de modification dont l'utilisation sera limitée dans le temps ou mieux encore par une validation définitive intervenant après une vérification de l'encodage.
- entités/rerelations jamais relues. On peut ici se demander si leur présence dans le système se justifie.
- entités/rerelations jamais supprimées. On risque ici de saturer peu à peu la mémoire. Il faudra prévoir des tâches d'archivage des données qui ne sont plus utiles.

	ASSURE	CONTRAT	GARANTIE	SINISTRE	Comporter	Échéance	Rattacher	Mise en jeu
Vérification								
Contrôle	LEC	LEC	LEC		LEC	LEC		
Ouverture							CRE	
Analyse								
Saisie				MOD				CRE
Informé	LEC	LEC	LEC	LEC			LEC	LEC

## 8.3 Confrontation algorithmique détaillée

Cette technique, à réserver à la deuxième phase de l'étude détaillée, va confronter de manière détaillée les MCD-MOD avec les modèles externes (expression formalisée des messages, règles de traitement et actions d'une tâche).

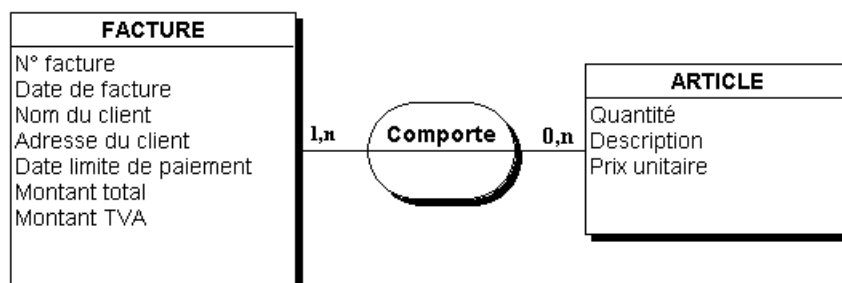
### 8.3.1 modélisation des messages

Il s'agit ici de modéliser uniquement les informations qui figurent dans les messages servant d'entrée ou de sortie à des tâches. La modélisation qu'on a opérée dans le MCT était purement verbale. La modélisation faite ici devra se limiter à ce qui apparaît dans le message (on parle ici de modèle *externe*). A ce modèle externe, on pourra associer déjà un dessin d'écran (entrée) ou d'état (sortie) qui s'élaborera de manière plus complète dans le MLT. Notons que si plusieurs messages sont synchronisés :

- ils donneront naissance à deux modèles externes s'ils sont synchronisés à l'aide de OU
- ils donneront naissance à un seul modèle externe s'ils sont synchronisés à l'aide de ET.

Cette modélisation reprend le formalisme entité/association en ne gardant que la règle de non répétitivité des informations.

Prenons l'exemple d'une facture. Celle-ci va se diviser en deux entités et une relation externes. En effet, les multiples articles qui figurent sur la facture ne peuvent pas se considérer comme propriété de la facture.



Le modèle ainsi obtenu ne correspond évidemment pas au MOD partiel défini pour les factures (notamment il intègre les données du client dans la facture). L'utilité est de pouvoir vérifier si ces données sont disponibles. En outre, l'entité *Facture* définie ici pourrait correspondre à une vue du modèle logique.

### 8.3.2 expression des actions sur les données mémorisées

Les quatre actions déjà évoquées plus loin se retrouvent ici (création, modification, consultation et suppression). Notons que selon qu'elles s'appliquent à des entités ou à des relations, elles donnent lieu à des variantes.

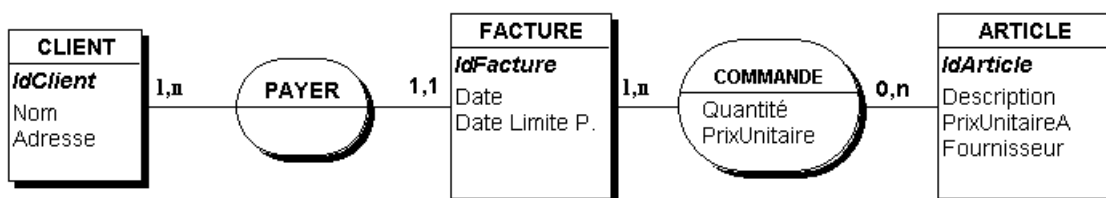
1. **créer une entité** : il faut déterminer la valeur de son identifiant et de quelques propriétés.
2. **modifier une entité** : après avoir recherché l'occurrence d'entité par son identifiant, on initialise ou modifie la valeur d'une ou plusieurs propriétés (pas celle de l'identifiant).
3. **supprimer une entité** : ici aussi, il faut accéder à l'entité avant de la supprimer. Ce ne sera possible que si les relations liées à cette entité ont déjà été supprimées.
4. **consulter une entité** : la consultation se fait par intérêt pour la valeur des propriétés soit pour simplement y faire référence au niveau des relations.
5. **créer une relation** : avant de procéder à la création de l'occurrence de relation, il faut que les occurrences d'entités qu'elle relie existent déjà. D'éventuelles valeurs de propriétés peuvent être enregistrées.
6. **modifier une relation** : on va initialiser ou modifier les valeurs des propriétés, après avoir réalisé un accès à cette occurrence. Remarque importante : on ne peut pas modifier la collection<sup>2</sup> de l'occurrence. Dans ce cas, il faut supprimer l'occurrence et en recréer une nouvelle.
7. **supprimer une relation** : la suppression (après accès) concerne aussi les propriétés éventuellement contenues.
8. **consulter une relation** : deux raisons de consulter une relation : soit pour accéder à une de ses propriétés, soit pour parcourir la relation pour atteindre une occurrence d'entité.

### 8.3.3 algorithme de confrontation détaillée<sup>3</sup>

Partie très fastidieuse mais néanmoins nécessaire du travail de confrontation, cette étape dispose de trois éléments de base :

- le MCD/MOD
- la tâche qui est en cours de confrontation
- le/les messages formalisés en termes d'entités et relations externes.

Le travail consiste à vérifier que les propriétés, les entités et les relations du modèle externe ont leurs équivalents dans le MOD.



<sup>2</sup>Rappelons que la *collection* d'une relation désigne les entités qu'elle relie.

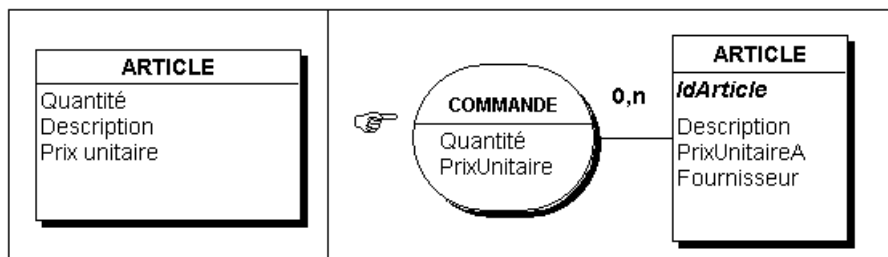
<sup>3</sup>Je reprends le terme exposé dans la théorie, bien que je doute de sa nature algorithmique.

**(1) équivalence propriété externe/propriété conceptuelle** La propriété externe peut intervenir dans trois cas de figures :

- correspondance **directe** avec une propriété du MCD (exemple le numéro de facture figure dans l'entité *Facture*, c'est l'identifiant)
- correspondance **via une relation** à une propriété conceptuelle (le nom du client dans une facture peut être trouvé de manière unique grâce à la relation *Payer* qui relie l'entité *Facture* à l'entité *Client*).
- correspondance via **une règle de traitement** : il faut alors expliciter la manière dont s'opère le « calcul » (le montant de la facture sera égal à la somme des produits de chaque *PrixUnitaire* par la *Quantité* qu'on peut trouver dans la relation *Commande*).

L'impossibilité de trouver une équivalence amènera l'analyste à ajouter une propriété à son modèle. Dans des cas plus rares, il ne parviendra pas à placer la propriété dans son modèle, ce qui l'obligera à le remettre en cause.

**(2) correspondance entité externe/sous ensemble conceptuel** Le travail consiste ici à trouver un sous-ensemble du MOD qui correspondent à l'entité externe. On partira d'une entité ou d'une relation pivot qui permettra d'atteindre une seule occurrence des éléments de l'ensemble. On verra que l'entité externe *Article* correspond en réalité à la portion suivante du MCD, où la relation *Commande* sert de pivot.



Notons qu'un échec de la confrontation à ce niveau entraînera la remise en question de tous les MOD, sa reconversion en MCD et la nécessité de recommencer toute la confrontation.

**(3) correspondance relation externe/relation conceptuelle** La correspondance entre les relations externes et des éléments du MOD se fera d'une manière plus variée :

- relation aux centres des entités pivots correspondant aux entités externes,
- patte d'une relation conceptuelle si l'un des pivots est une relation conceptuelle (voir notre exemple),
- partage d'entités dans des cas plus complexes<sup>4</sup>.

L'examen des relations externes doit de toutes façons vérifier que les cardinalités maximales sont cohérentes entre les deux modèles.

**(4) prise en compte des actions** L'examen de la tâche va permettre pour chaque entité ou relation de spécifier quelle action va s'opérer (création, modification, consultation et suppression). On veillera à vérifier que les identifiants sont disponibles.

<sup>4</sup>Je n'ai pas trouvé d'exemple pour ces cas, pas plus semble-t-il que les auteurs de mes ouvrages de référence habituels.

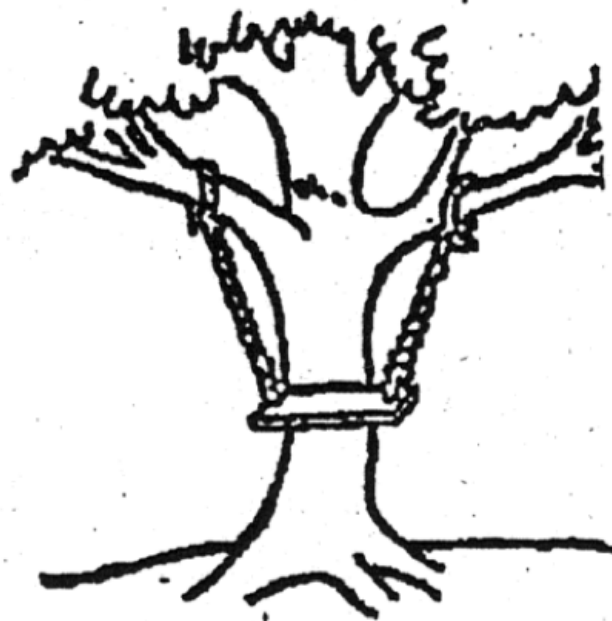
## 8.4 Maquettage

Cette technique fait appel à des outils spécifiques. Nous y reviendrons plus tard. Il est bien certain que l'utilisation par les futurs utilisateurs d'une maquette préfigurant le programme fini permet, dans bien des cas, de souligner des problèmes restés inaperçus. Parmi ceux-ci, figurent des incohérences entre les MCT et MCD.

## 8.5 Conclusion

Au terme de ces confrontations et modifications, nous disposons d'un double modèle : MOD et MOT. Ce modèle a été non seulement validé par les analystes, mais soumis au comité directeur et aux utilisateurs pour approbation (nous verrons ce point plus en détail dans le cycle de vie). Nous sommes à présent en mesure de passer aux niveaux suivants qui concernent les seuls informaticiens : les modèles logiques et physiques.

4



*Ce que les programmeurs ont réalisé*