

**INGENIERIE DES SYSTEMES D'INFORMATION :  
MERISE DEUXIEME GENERATION**

4<sup>e</sup>édition - 2001

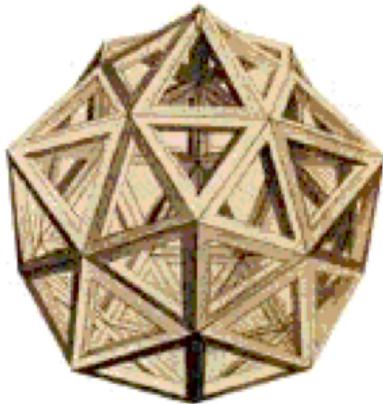
**Dominique NANCI - Bernard ESPINASSE**

*Avec la collaboration de Bernard COHEN, Jean-Claude ASSELBORN et Henri HECKENROTH*

## Sommaire

- 1 - Diagrammes des flux du cas de la société X .....3**
  - Diagramme des flux.....3
  - Modélisation des acteurs .....4
  - Délimitation du domaine d'activité .....4
- 2 - Modèles conceptuels des traitements du cas X.....6**
  - Modèle conceptuel des traitements actuel.....6
  - Modélisation des processus .....6
  - Modélisation des opérations .....7
  - Modélisation des événements internes .....10
  - Modélisation orientée processus ou orientée état .....11
  - Modèle conceptuel des traitements futur .....13
- 3 - Modèles conceptuels des données du cas X.....16**
  - Modèle conceptuel de données actuel .....16
  - Modèle conceptuel de données futur .....19
- 4 - Modèles organisationnels des traitements du cas X .....21**
  - Modèles organisationnels des traitements futur .....21
  - Organisation synchrone ou asynchrone .....26
- 5 - Modèles organisationnels des données du cas X.....27**
  - Modèle organisationnel de données global .....28
  - Modèles organisationnels de données répartis .....31
- 6 - Modèles logiques de données du cas X.....34**
  - Modèle logique de données relationnel brut .....34
  - Transformation des concepts avancés .....34
  - Modèles logiques de données relationnels répartis .....35
  - Mise en cohérence de données dupliquées .....36
- 7 -Optimisation du modèle logique de données du cas X et modèle physique..38**
  - Modèle logique optimisé .....38
  - Modèle logique Magasin.....39
  - Modèle logique Siège .....39
  - Modèle physique.....40
- 8 - Modèle logique de traitements du cas X .....48**
  - Modèle logique de traitements futur .....48
  - Procédure logique non répartie .....48
  - Détail des ULT .....49
  - Répartition logique des traitements .....55

# SOLUTIONS MERISE DEVELOPPEES DU CAS SOCIETE X



*Cas issu de l'ouvrage : « Ingénierie des systèmes d'information : MERISE 2<sup>e</sup> génération » D.NANCI, B.ESPINASSE avec la collaboration de B.COHEN, H.HECKENROTH, J-C ASSELBORN, ", 4<sup>e</sup>édition, 2001, Vuibert Informatique éditeur ( ISBN : 2-7117-8674-9).*

## 1 - DIAGRAMMES DES FLUX DU CAS DE LA SOCIÉTÉ X

Nous avons choisi de développer, au fur et à mesure de cet ouvrage, un cas dont l'exposé est présenté à la fin du livre.

Cette étude de cas aurait dû normalement se dérouler en suivant la démarche. Toutefois, à titre d'illustration, nous avons choisi de traiter le cas au fur et à mesure de l'étude des modèles types, en nous situant généralement en étude préalable, c'est-à-dire en nous préoccupant essentiellement des activités principales.

### Diagramme des flux

L'analyse des flux, en étude préalable, a pour objectif de faciliter la prise de connaissance générale de l'activité à travers les échanges d'informations, et d'appréhender les frontières du domaine étudié.

Dans le cas de la société X, les domaines d'activités semblent assez bien déterminés :

- Comptabilité
- Personnel
- Achats
- Vente et stocks

Ce dernier domaine est celui qui sera étudié en détail dans cet ouvrage pour illustrer la méthode.

Dans le diagramme des flux de la situation existante, nous cherchons à mettre en évidence des unités actives ou acteurs qui échangent des flux d'information. A ce niveau, un acteur peut représenter :

- un partenaire extérieur à l'entreprise,
- un domaine d'activité de l'entreprise,
- un service,
- un poste (un ensemble de personnes exerçant une activité commune).

Dans le cas, nous avons perçu les acteurs suivants :

- Client
- Fournisseur
- Comptabilité
- Achat
- Comptoir
- Caisse
- Stock
- Secrétariat

L'ensemble des flux retenus est représenté sur le diagramme brut des flux de la figure 5.6. Notons que l'acteur "Direction commerciale" n'a pas été retenu car il n'est présent que dans un processus secondaire (hypothèse d'étude préalable annoncée en début de paragraphe).

### Modélisation des acteurs

Tous les acteurs retenus *exercent une activité* ; d'une façon générale, les acteurs traduisent des groupes de personnes exerçant une activité. Il faut cependant adapter le niveau de détail à l'étude pour ne pas fragmenter les acteurs ; par exemple nous n'avons pas distingué *responsable des stocks* et *employé des stocks*, nous les avons confondus dans un même acteur générique *stock*. Ce terme peut paraître ambigu ; il ne désigne pas le contenu du stock mais le service où s'exercent les activités de gestion des stocks.

Enfin, nous n'avons considéré ni *catalogue* ni *fichier client* comme des acteurs, bien qu'il y ait flux d'informations. En effet, dans le diagramme des flux de type Merise, les acteurs représentent des ensembles d'activités, ce qui n'est pas le cas des objets évoqués (un fichier ne fait rien, il est). Ils représentent en fait la mémoire collective du système d'information du domaine, et nous convenons de ne pas les représenter dans un diagramme des flux de type Merise. Toutefois, si l'on s'orientait vers des diagrammes de flux de données de type « analyse structurée », ces objets (catalogue, fichier client...) se représenteraient par des dépôts de données, donc par des symboles différents.

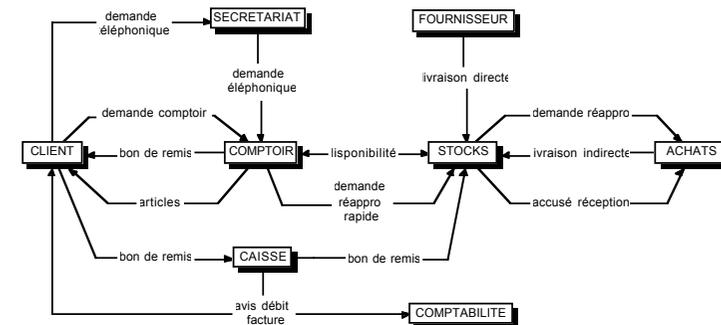


Figure 5.6 : Diagramme des flux brut de la société X.

### Délimitation du domaine d'activité

A la lecture du cas et en reconduisant les frontières actuelles, on peut considérer que les activités exercées par le comptoir, le secrétariat, la caisse et le stock relèvent de notre domaine vente-stock.

Le diagramme des flux brut (figure 5.6) illustre l'organisation de notre domaine gestion commercial actuel. Sur celui-ci nous pouvons mieux délimiter les frontières de notre domaine d'étude.

Le diagramme des flux brut est un outil souvent suffisant pour se faire une bonne compréhension du fonctionnement macro-organisationnel de l'existant. Au cas où cette description s'avérerait insuffisante, nous avons toujours la possibilité de décrire l'organisation existante à l'aide de modèles organisationnels de traitement (MOT existant).

Pour préparer la modélisation conceptuelle des traitements, on établit alors le diagramme conceptuel des flux de la figure 5.7.

Le diagramme des flux conceptuel est un outil très intéressant. Il nous permet de commencer à mener le processus d'abstraction qu'illustre bien la courbe du soleil, à nous échapper de cette description organisationnelle de l'existant pour passer à un niveau conceptuel toujours associé à cet existant.

En particulier, ce modèle permet d'une part de distinguer acteurs externes et domaines, d'autre part de mettre plus particulièrement en évidence le domaine étudié et ses flux.

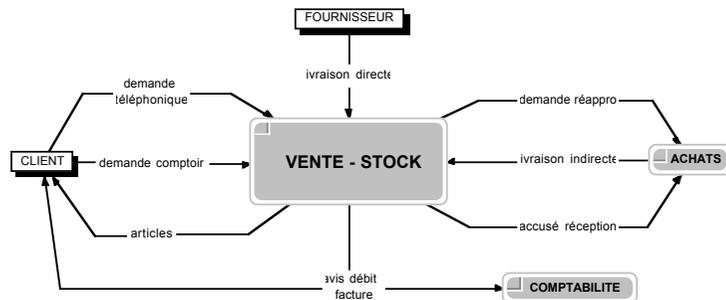


Figure 5.7 : Diagramme des flux conceptuel actuel de la société X.

Ce diagramme des flux conceptuel peut aussi nous aider à mieux appréhender la modélisation conceptuelle des traitements associée à l'activité future du domaine. En effet, il est souvent possible, sur un tel diagramme, de faire apparaître les évolutions conceptuelles, c'est à dire les nouvelles orientations de gestion. Ainsi, pour notre domaine de la gestion commerciale, deux évolutions majeures pourraient être les suivantes :

- dans le futur, le domaine produira les relevés de fin de mois élaborés à partir des avis de débits et destinés aux clients en compte; ces factures mensuelles sont jusqu'à présent éditées par la comptabilité;
- en cas de rupture de stock, le domaine peut actuellement recourir à un réapprovisionnement rapide auprès du service Achats. Dans le futur, il pourrait aussi, par exemple si le délai associé à ce réapprovisionnement était trop long, demander à un autre magasin de le dépanner en effectuant un transfert.

Le diagramme de flux conceptuel de la figure 5.8 met en évidence ces nouvelles orientations et modifie ainsi les frontières de notre domaine. Ceci pourra nous aider dans l'élaboration du modèle conceptuel des traitements futur, comme nous le verrons par la suite.

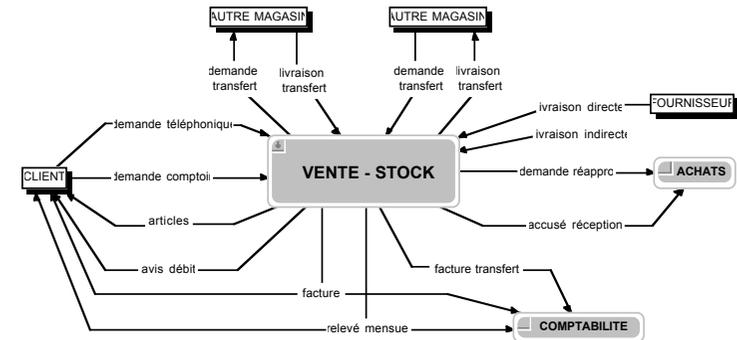


Figure 5.8 : Diagramme des flux conceptuel futur de la société X.

## 2 - MODELES CONCEPTUELS DES TRAITEMENTS DU CAS X

### Modèle conceptuel des traitements actuel

L'élaboration d'un MCT a pour objectif la mise en évidence des activités exercées au sein du domaine, en interaction avec son environnement.

#### Modélisation des processus

Les activités concourant à une même finalité de gestion (niveau conceptuel) sont regroupées en processus. Rappelons qu'un processus est déclenché par un ou plusieurs événements initiaux (externes ou internes), regroupe un ensemble d'activités concourant à un même but et donne lieu à l'émission d'un ou plusieurs résultats finaux. L'identification des processus est importante mais parfois délicate car elle nécessite un effort d'abstraction. Le passage du diagramme des flux brut au diagramme des flux conceptuel nous avait déjà permis de mener une telle abstraction en définissant les frontières du domaine Vente - Stock; aussi l'identification des processus peut en partie s'appuyer sur ce dernier diagramme.

Dans la modélisation des activités actuelles du domaine Gestion commerciale du cas X, nous avons retenu les deux processus principaux annoncés dans l'exposé du cas:

- le processus de vente au comptoir;
- le processus de réapprovisionnement.

Le processus de vente au comptoir est initié par l'arrivée d'une demande client et est considéré comme achevé (la finalité globale du processus est atteinte) lorsque l'avis de débit ou la facture sont émis.

Le processus de réapprovisionnement est initié par une décision interne de suivi quotidien des stocks (on retrouve le rôle d'acteur du système de pilotage, même lorsque ces décisions sont programmées dans le temps) et s'achève lorsque l'accusé de réception de livraison est retourné au service Achats.

La figure 6.9 représente un schéma général des processus (voir Modularité des modèles conceptuels de traitements).

*Remarque* : Certains flux présents sur le diagramme des flux conceptuel (demande réappro, livraisons directe et indirecte) n'apparaissent plus sur le schéma général des processus. En effet, ces flux ne traduisent que des situations intermédiaires du processus; ainsi, l'émission d'une demande de réappro ne peut être considérée ni comme le déclencheur du processus de réapprovisionnement, ni comme la conclusion de ce processus.

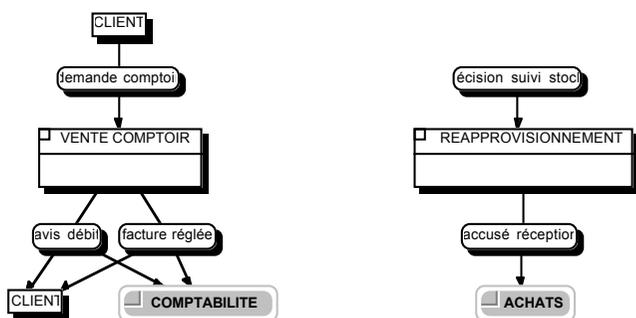


Figure 6.9 : Schéma général des processus

*Remarque* : La simplicité relative du cas X induit un schéma des processus élémentaire. Dans la pratique, un tel schéma peut comporter une dizaine de processus et permet de présenter une cartographie générale des activités du domaine, avant de rentrer dans le détail de chaque processus.

### Modélisation des opérations

Par rapport à la description de l'activité de la société X, nous proposons le modèle conceptuel des traitements actuel à la figure 6.10. Compte tenu de la simplicité de la modélisation, nous avons réuni dans le même schéma les opérations composant les deux processus, en délimitant toutefois les processus. Dans la pratique, chaque processus principal donnerait généralement lieu à une décomposition sur un schéma.

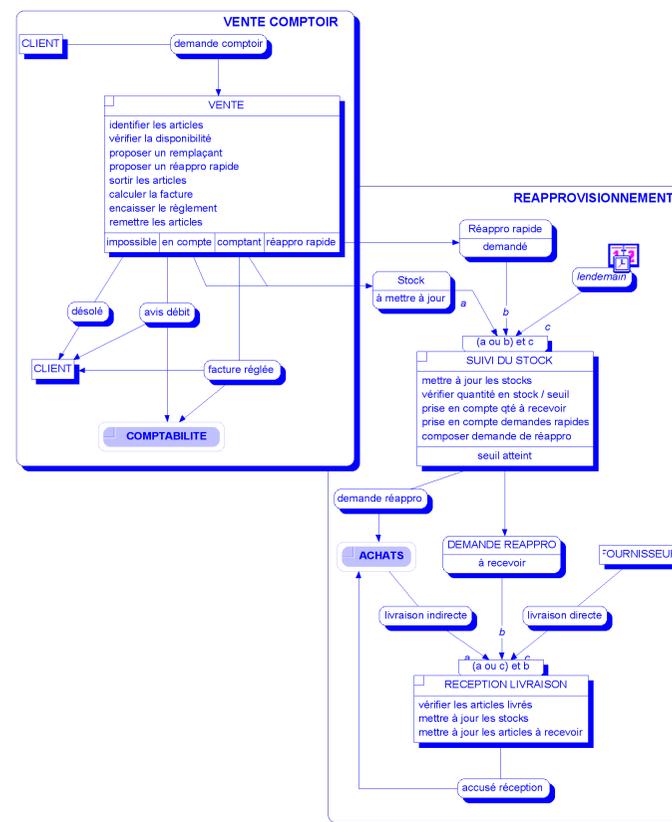


Figure 6.10 : MCT actuel - Approche Processus

### Opération Vente au comptoir

Cette opération regroupe les activités réalisées au comptoir, dans les rayonnages et à la caisse. En effet, à partir de la demande du client, le domaine dispose de toutes les informations (dans sa mémoire) pour gérer cette demande jusqu'à la délivrance des articles et des documents associés.

Dans cette opération, il a fallu cependant s'interroger sur le rôle du bon de remis. La figure 6.11 illustre la suite du propos.

Une première modélisation aurait pu faire apparaître l'aller-retour du bon de remis vers le client, et induire deux opérations.

Cependant, le concepteur doit s'interroger sur la valeur ajoutée apportée par le client sur le message bon de remis. S'il y a apport d'informations auparavant inconnues, alors cette modélisation est correcte ; il faudrait cependant modifier les appellations des événements pour indiquer au lecteur l'évolution contenu du message. A ce sujet, nous rappelons qu'il faut distinguer le nom du support (qui peut rester constant au cours des divers traitements) et le nom de l'événement et message associé qui déclenche ou résulte d'un traitement. S'il n'y a pas de différence entre le message adressé à l'acteur et celui retourné, alors il n'a pas besoin d'un aller-retour vers l'environnement.

La modélisation évolue, et le bon de remis est alors représenté comme un événement interne. Or, au niveau conceptuel, deux opérations enchaînées seulement par un événement interne doivent être fusionnées. On obtient alors une modélisation regroupant l'ensemble des activités en une seule opération.

Cette dernière modélisation montre bien que le bon de remis, par rapport au client, ne sert qu'à transmettre de l'information entre le comptoir et la caisse, qui sont des notions organisationnelles (postes). La modélisation conceptuelle des traitements nous conduit naturellement à en faire abstraction. Dans les échanges avec le client, ce bon de remis n'a pas la même stabilité que la demande ou la facture; on pourra éventuellement le remettre en cause au niveau organisationnel futur.

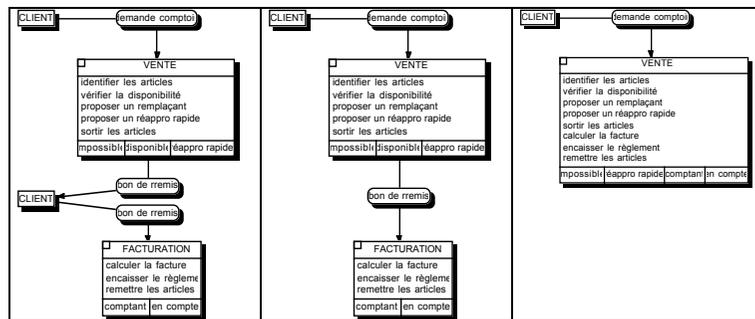


Figure 6.11 : Evolutions de la modélisation conceptuelle du Bon de remis

### Opération Suivi de stock

La mise à jour des stocks suite aux ventes, la prise en compte des demandes de réapprovisionnement rapide et la surveillance du seuil de réapprovisionnement sont des activités qui, a priori, ne nécessitent pas l'attente d'informations en provenance d'acteurs ; on devrait donc mettre ces activités dans l'opération amont, c'est-à-dire Vente.

Cependant, les activités de vente sont rythmées par la survenance des demandes ; à chaque demande, les différentes activités contenues dans l'opération de vente peuvent être exécutées. Par contre, les fonctions de suivi des stocks ont un rythme différent ; on ne met pas à jour le stock à chaque demande, on ne surveille pas le seuil de réapprovisionnement à chaque demande. L'ensemble de ces activités est donc déclenché par un événement complémentaire qui les cadence ; il s'agit d'une décision du pilote (système de pilotage) dont les interactions avec le système d'information peuvent s'assimiler au comportement d'un acteur.

Ce pilote intervient assez fréquemment dans les systèmes d'information. Lorsque la décision du pilote est rythmée par un cycle de temps régulier (tous les jours, semaines, mois...), il est représenté par un événement interne temporel. En toute rigueur théorique, on peut évidemment s'interroger sur la nature « conceptuelle » de l'intervention d'un tel acteur que l'on pourrait considérer comme agent d'organisation.

### Opération Réception livraison

Cette opération concerne la prise en compte d'une livraison effectuée soit par un fournisseur (livraison directe) soit par le service des achats de la société X, et consécutive à une demande de réapprovisionnement en attente (représentée par un état). Cette opération regroupe les activités de vérification des articles livrés et la mise à jour des stocks. Elle se termine par l'envoi d'un accusé de réception au service des achats.

### Modélisation des événements internes

Nous avons vu que l'opération Suivi des stocks était déclenchée par l'événement interne temporel "lendemain" et qu'elle devait prendre également en compte les quantités livrées (figurant sur le bon de remis) pour mettre à jour les stocks et les demandes de réapprovisionnement rapide. Ces flux d'informations entre deux opérations conceptuelles peuvent être modélisés comme des événements internes (c'est à dire sans flux avec l'extérieur du domaine). C'est cette solution qui a été appliquée dans la première génération de la méthode Merise.

Dans la deuxième génération, les événements internes qui permettent de conditionner l'enchaînement d'opérations conceptuelles se modélisent désormais par la notion d'état. Ces événements internes qui traduisaient des échanges internes relèvent alors du niveau organisationnel. Dans cette nouvelle modélisation, les événements et résultats doivent seulement exprimer soit des échanges avec l'extérieur du domaine, soit l'intervention du système de pilotage (événement interne décisionnel ou temporel), respectant ainsi l'approche systémique de Merise.

La figure 6.12 illustre les différences de modélisation entre les deux générations.

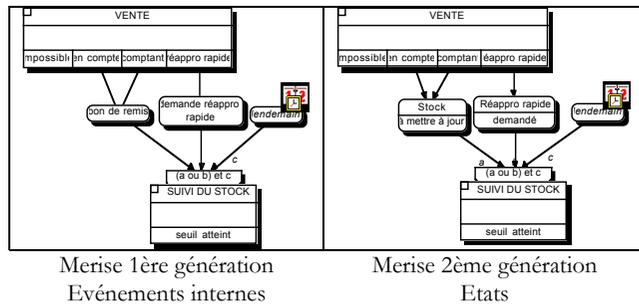


Figure 6.12 : Evolution de la modélisation des événements internes

### Modélisation orientée processus ou orientée état

L'introduction de la notion d'état dans la modélisation conceptuelle des traitements que nous venons de voir se substituer au rôle d'événement interne, entraîne une importante évolution dans la technique, l'approche et la représentation des MCT.

Dans une approche orientée processus, l'analyse et la construction du MCT est conduite par l'enchaînement chronologique des activités, à partir d'un ou plusieurs événements initiaux. On déroule ainsi le processus, en respectant toutefois les règles de modélisation des opérations (on regroupe dans une opération toute les activités réalisables à partir des informations fournies par le(s) événement(s) sans avoir besoin d'autres informations nécessitant des événements complémentaires).

La description des processus est alors un enchaînement d'opérations, cet enchaînement étant assuré par les états (ex événements internes); la similitude graphique (voulue) entre événement et état permet de conserver l'aspect général et la présentation des MCT de la première génération. Dans cette modélisation, une même opération peut être représentée dans plusieurs processus. Le MCT présenté en figure 6.10 est une illustration de cette approche orientée processus.

Dans une approche orientée état, chaque opération est analysée indépendamment des autres opérations; ses conditions de déclenchement étant seulement soumises à la présence d'événement (au moins un) et d'état. Pour les états, le concepteur ne cherche pas à modéliser, voire à connaître les opérations qui ont antérieurement produit ces états; L'opération ainsi modélisée regroupe un ensemble d'activités dont l'exécution ne demande pas d'échange avec l'environnement du domaine; on formalise essentiellement ses entrées et ses sorties.

La représentation graphique des MCT est alors une simple juxtaposition d'opérations, sans indication d'enchaînements entre opérations. Dans cette modélisation, une opération n'est représentée qu'une seule fois. La figure 6.13, représentant le MCT actuel, illustre cette approche orientée état.

Ces deux approches sont complémentaires, voire duales. A partir d'un MCT orienté processus, on peut obtenir un MCT orienté état en dupliquant puis segmentant tous les états qui matérialisent des enchaînements d'opérations. A partir d'un MCT orienté état, on peut reconstituer l'ensemble des processus possibles en reliant les opérations qui partagent le même état (état résultant, opération précédente - état préalable, opération suivante). Les figures 6.10 et 6.13 l'illustrent partiellement (le nombre limité d'opérations, d'états et de processus ne permet pas de mettre en évidence les différences de présentation).

L'approche processus peut apparaître plus "naturelle" aux utilisateurs; on la retrouve d'ailleurs dans le BPR. L'approche par état permet d'isoler les opérations, facilitant ainsi l'analyse ultérieure de son contenu; les informaticiens y retrouveront une certaine "approche objet".

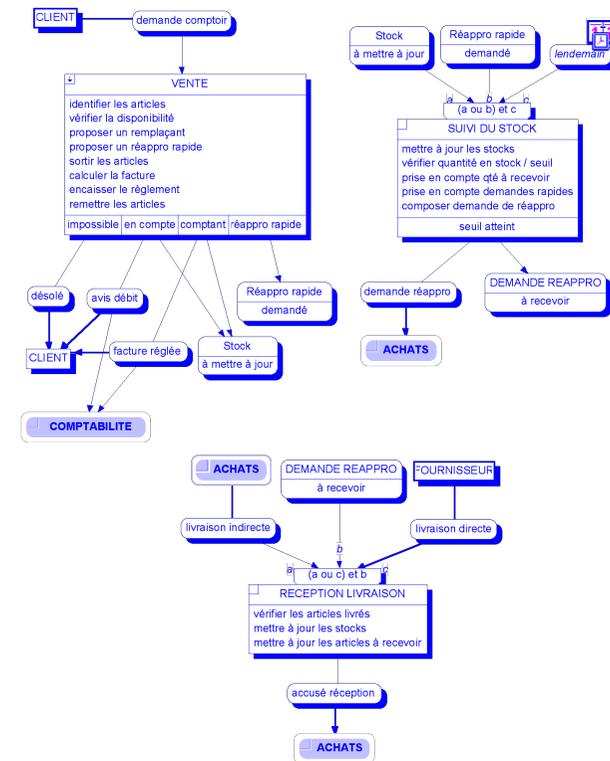


Figure 6.13 : MCT actuel - Approche Etat

## Modèle conceptuel des traitements futur

Au niveau d'un modèle conceptuel de traitements, les évolutions entre un système actuel et un système futur peuvent se traduire par :

- le déplacement d'activités entre opérations ;
- la modification de synchronisations ou de conditions de sortie ;
- le changement d'origine d'un événement ou de destination d'un résultat ;
- le changement de rythme de gestion ;
- l'ajout ou la suppression d'une opération et/ou de fonctions.

Ce dernier cas conduit généralement à une redéfinition de la frontière du domaine par la prise en charge de nouvelles activités, ou l'abandon d'activités actuelles à d'autres domaines.

De telles orientations peuvent déjà être mises en évidence dans un diagramme des flux conceptuel futur, comme nous l'avons fait précédemment.

Pour la modélisation des activités futures du domaine Gestion commerciale du cas X, nous proposons le modèle conceptuel des traitements de la figure 6.14. Les nouvelles orientations de gestion, de niveau conceptuel, sont les suivantes :

- ① la mise à jour des stocks est maintenant effectuée à chaque vente;
- ② les demandes de réapprovisionnement rapide sont maintenant directement adressées au service achat, au fur et à mesure des demandes client;
- ③ la vente prend en charge les activités de production des relevés de fin de mois élaborés à partir des avis de débits et destinés aux clients en compte ; ces activités étaient antérieurement assurées par la comptabilité;
- ④ en cas de rupture de stock lors de la vente, les magasins ont maintenant la possibilité de s'adresser directement à un autre magasin pour demander un transfert;
- ⑤ l'informatisation permettra au service Achats de connaître l'état des stocks de l'ensemble des magasins, on envisage de confier au service Achats les activités de suivi des stocks des magasins; ainsi, lorsqu'un article est à réapprovisionner dans un magasin, le service Achats pourrait apprécier les besoins dans les autres magasins et faire une commande fournisseur globale; ceci éviterait que des demandes de réappro concernant le même article se succèdent à quelques jours près au service Achats.

L'orientation ① se modélise en déplaçant l'activité "mettre à jour les stocks" dans l'opération Vente.

L'orientation ② se modélise en adressant le résultat "demande réappro rapide" au domaine Achats.

L'orientation ③ se modélise par une nouvelle opération Facturation mensuelle, déclenchée en fin de mois et qui reprend les Commande\_Facture d'état "à facturer".

L'orientation ④ nous conduit tout d'abord à modifier l'opération Vente au comptoir en ajoutant le déclenchement d'une demande de transfert vers un autre magasin, puis à introduire un nouveau processus Transfert pour gérer une demande de transfert en provenance d'un autre magasin, comportant l'opération de transfert et sa facturation interne, enfin à enrichir l'opération Réception livraison pour prendre en compte la réception des marchandises provenant d'un autre magasin suite à une demande de transfert.

L'orientation ⑤ nous conduit à supprimer l'opération Suivi du stock.

Ainsi, le modèle conceptuel des traitements du domaine actuel a été modifié de façon significative. La plupart des opérations de ce dernier sont touchés et de nouvelles opérations et processus apparaissent, d'autres disparaissent.

Cette reconception des activités de gestion, formalisée par le passage d'un modèle conceptuel de traitements actuel à un modèle futur, relève aujourd'hui du B.P.R. (Business Process Reengineering).

### Opération Facturation mensuelle

Cette nouvelle opération rassemble les activités permettant d'éditer chaque mois, à partir des commandes à facturer (avis de débit), les facturations mensuelles des clients en compte du magasin. Cette opération se termine par l'envoi d'un relevé mensuel au client et au service de la comptabilité. On supposera que ce dernier se chargera de suivre leur règlement.

### Processus Transfert

Ce nouveau processus rassemble les activités relatives à la prise en compte d'une demande de transfert provenant d'un autre magasin. Il concerne ainsi la prise en compte de la demande, la consultation des stocks, le déstockage des articles et la livraison. Il se termine par l'envoi au service de la comptabilité d'une facture de transfert. Ce processus se décompose en deux opérations, en effet nous avons une attente associée à l'événement "bon de livraison signé" provenant du magasin livré.

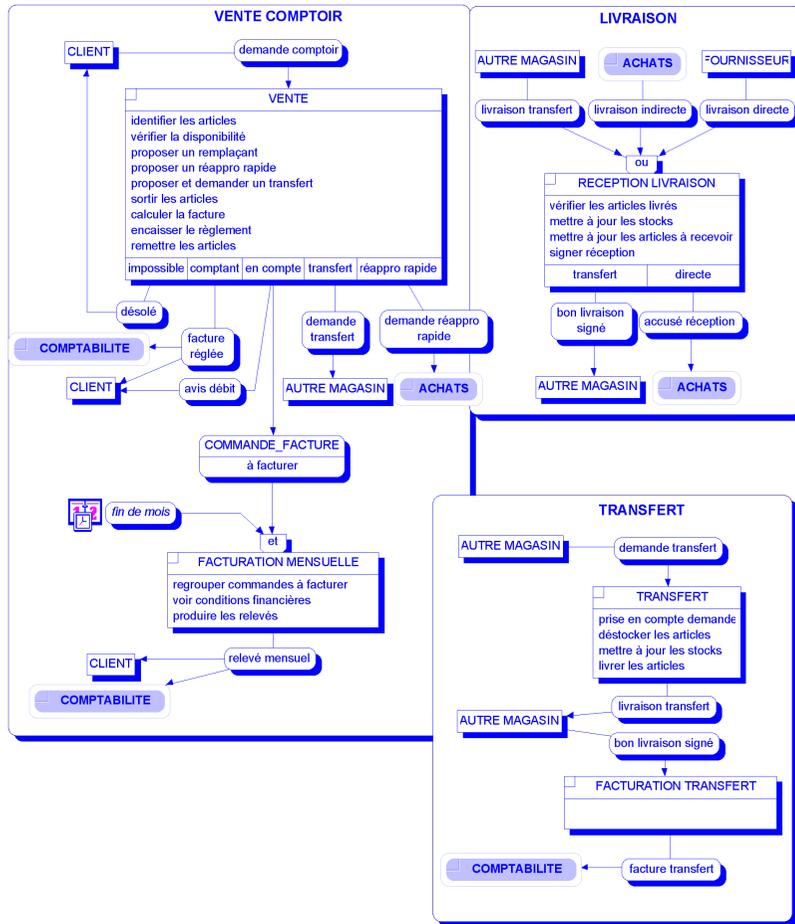


Figure 6.14 : MCT futur de la société X.

### 3 - MODELES CONCEPTUELS DES DONNEES DU CAS X

#### Modèle conceptuel de données actuel

Pour élaborer ce modèle conceptuel de données actuel, nous nous sommes appuyés sur une liste d'informations partielle. Ces informations ont été recueillies dans la phase d'analyse de l'existant et figurent sur des documents (voir le cas de la société X).

Notre compréhension des informations utilisées dans la société X nous a conduits à proposer le modèle conceptuel des données actuel de la figure 7.57. Nous allons maintenant expliquer et justifier certains choix de modélisation.

Entité commande (facture)

En première analyse, les notions de commande et facture évoquent des « objets » différents. Mais le texte du cas nous indique que n° commande = n° facture. Cet indice nous suggère que commande et facture sont de même distinguabilité (une commande correspond à une facture et réciproquement), et que la transformation d'une commande en une facture n'est qu'un simple changement d'état. Il est probable que dans le type d'activité de la société X, commande et facture soient sur le même support (liasse autocopiante). Nous avons donc modélisé une seule entité COMMANDE avec une propriété ETAT.

Relation facturer

Le texte indique qu'un client autre que le donneur d'ordre peut être facturé. Deux interprétations sont possibles :

- La désignation du client à facturer est indépendante de la commande, c'est-à-dire constant pour un client donné ; son attribution est probablement faite lors de l'ouverture du compte client. On modélisera alors comme suit :

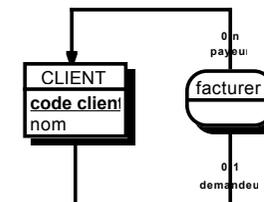


Figure 7.56 : Autre modélisation de la facturation à un tiers.

- La désignation d'un client tiers à facturer s'effectue à chaque commande et peut ainsi changer ; c'est la modélisation que nous avons retenue en indiquant avec une cardinalité mini égale à 0 que cette relation s'applique exceptionnellement.

**Relations valable pour et substituer**

La répétitivité des informations « matériel » et « référence article substituable », relatives à l'article, conduit à la modélisation de relations à cardinalités maxi n.

**Cumul des quantités vendues**

Cette information peut être calculée ; il s'agit de la somme des quantités livrées pour un article, à partir d'un dépôt, au cours d'un mois. Vérifions qu'une valeur de quantité livrée n'est imputable qu'à un dépôt, un article et un mois :

- A une occurrence de composer, ne correspond qu'une valeur de quantité livrée.
- A une occurrence de composer, ne correspond qu'une occurrence d'article.
- A une occurrence de composer, ne correspond qu'une occurrence de commande.
- A une occurrence de commande, ne correspond qu'une valeur de date, une valeur de date n'appartient qu'à un mois.
- A une occurrence de commande, ne correspond qu'une occurrence de dépôt par la relation auprès de.

Au niveau d'un modèle conceptuel de données, et en particulier en étude préalable, nous déconseillons de faire figurer des propriétés calculées à partir de cumuls, sauf sur des types de systèmes d'informations statistiques ou pour certaines informations importantes pour l'utilisateur. Compte tenu de la durée de vie, certaines informations calculées seront prises en compte dans le modèle organisationnel de données.

**Prix de vente net**

Le texte du cas laisse libre cours à toute interprétation pour la modélisation de cette propriété. Nous l'avons compris comme étant le prix d'un article propre à un magasin et choisi de l'affecter à la relation « stocker » ; ce prix, que nous appelons désormais prix magasin, sera lié au prix tarif par une règle indiquant la remise maxi de 25%.

**Relations stocker et réappro**

Les propriétés quantité en stock, date prochaine livraison et quantité à recevoir sont déterminées par rapport à article et dépôt, et pourraient formellement figurer dans la même relation. Cependant, nous considérons qu'elles ne concernent pas les mêmes préoccupations : date prochaine livraison et quantité à recevoir sont liées au problème de réapprovisionnement, tandis que quantité en stock est lié au stock. Pour ces motifs, nous choisissons de les affecter dans des relations distinctes.

**Relation préférer**

Cette relation modélise la notion de dépôt préférentiel ; elle est porteuse d'une dépendance fonctionnelle client x article → dépôt, exprimant que, pour un article, un client doit s'adresser de préférence à un dépôt unique.

Nous avons également choisi de présenter deux niveaux de modélisation conceptuelle des données; une première en utilisant les concepts de base de la première génération de Merise (figure 7.57), une seconde en mettant en oeuvre les concepts de modélisation avancés (figure 7.58).

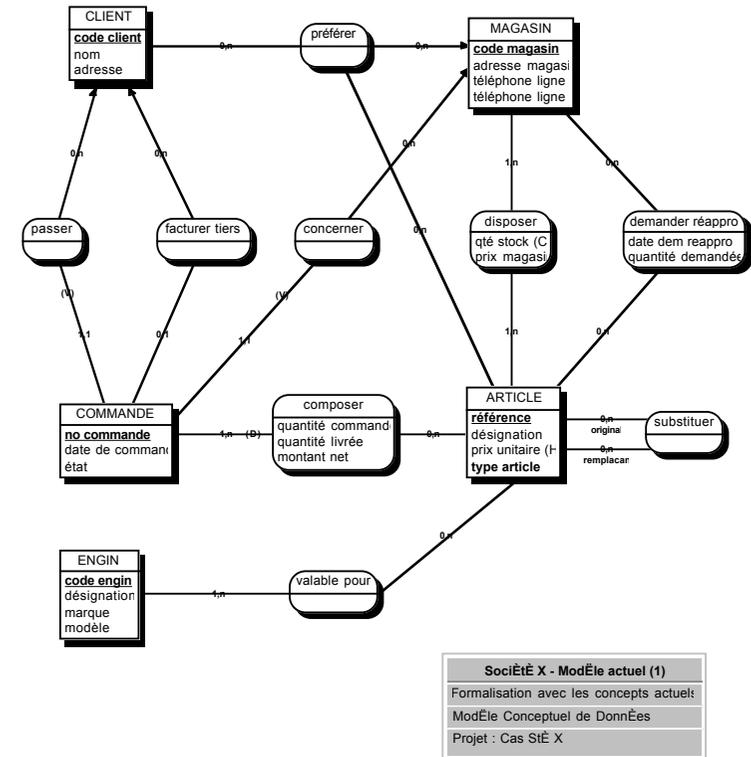


Figure 7.57 : Modèle conceptuel des données actuel de la société X. Formalisation basique

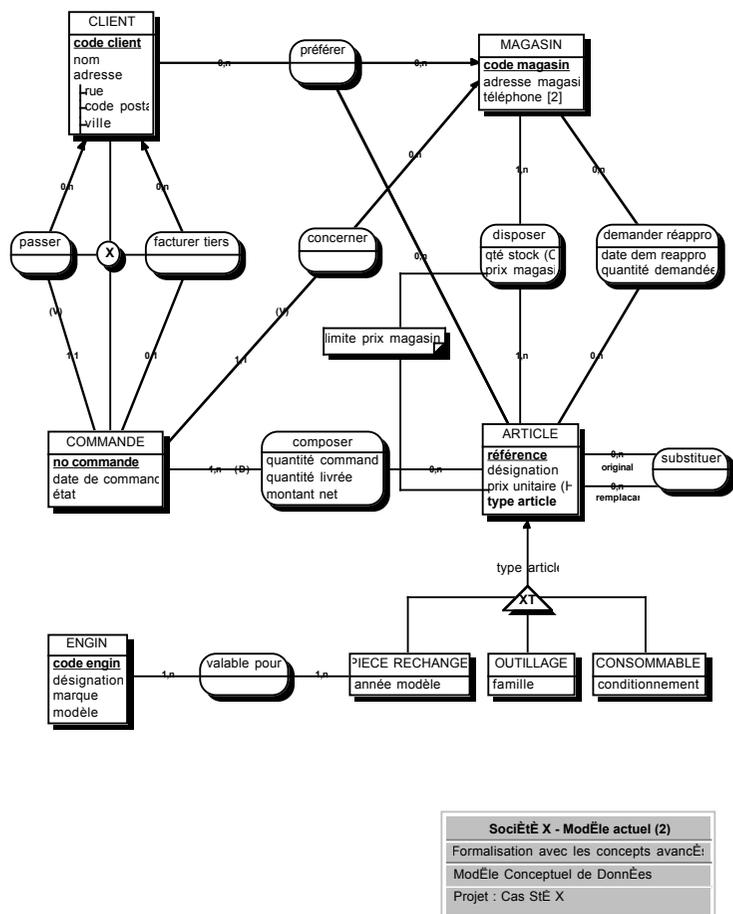


Figure 7.58 : ModÈle conceptuel des donnÈes actuel de la sociÉTÉ X.  
Formalisation avancÈe

### ModÈle conceptuel de donnÈes futur

Dans la pratique, on constate assez peu d'Èvolution entre l'actuel et le futur au niveau du modÈle conceptuel de donnÈes ; on retrouve son aspect de grande stabilitÈ sÈmantique.

En ce qui concerne notre cas, nous avons volontairement souhaitÈ qu'il en soit autrement. Ainsi, la prise en compte des nouvelles orientations conceptuelles dÈjÀ ÈvoquÈes pour le modÈle conceptuel des traitements futur nous conduit à amender de faÇon significative notre modÈle conceptuel des donnÈes (cf. figure 7.59). On voit notamment apparaître dans le modÈle de donnÈes l'entitÈ "Transfert".

Enfin, en confrontant le modÈle conceptuel de donnÈes actuel au modÈle conceptuel de traitements par la technique de la relecture croisÈe (voir chapitre 10), nous aurions constatÈ que la partie rÈapprovisionnement Ètait abordÈe sommairement et ne permettait pas une rÈelle gestion du processus de rÈapprovisionnement ; ou alors, ce processus disparaissait du MCT futur en Ètant confiÈ par exemple aux achats. Pour mettre en accord les deux modÈles futurs, (rappelons que le domaine Vente Stock conserve les rÈapprovisionnements rapides), nous proposons d'amender le MCD futur par la formalisation d'une demande de rÈapprovisionnement et des articles concernÈs. Les nouvelles propriÈtÈs proposÈes peuvent se substituer aux prÈcÈdentes propriÈtÈs de la relation de rÈapprovisionnement (voir figure 7.58).

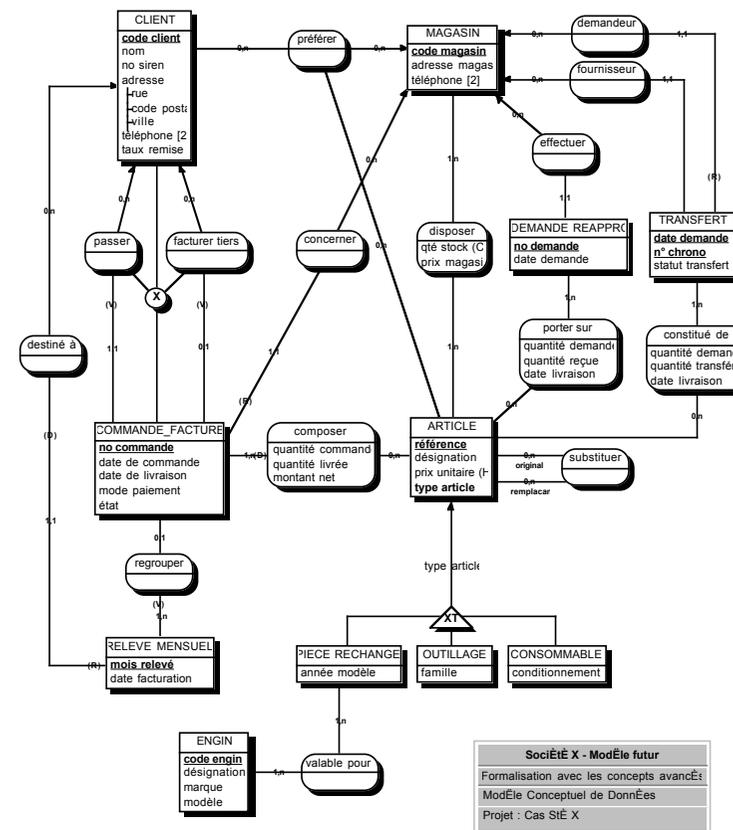


Figure 7.59 : ModÈle conceptuel des donnÈes futur de la sociÉTÉ X.

## 4 - MODELES ORGANISATIONNELS DES TRAITEMENTS DU CAS X

### Modèles organisationnels des traitements futur

Nous aurions pu, dans la pratique, construire pour la société X le modèle organisationnel des traitements actuel ; hormis l'aspect purement formel, celui-ci ne présenterait que peu d'intérêt. Nous avons préféré proposer plusieurs variantes de modèles organisationnels de traitements futurs, toujours dans une optique d'étude préalable.

Rappelons que le modèle organisationnel de traitements a pour objectif de décomposer les opérations du modèle conceptuel de traitements en tâches, et de les organiser en fonction des ressources (enchaînement, postes, informatisation).

Pour l'illustration du cas société X, nous avons choisi de décomposer l'opération Vente en plusieurs solutions organisationnelles. On remarquera que, globalement, chaque solution organisationnelle est une décomposition possible de l'opération conceptuelle : on retrouve en entrée et en sortie les événements et résultats principaux, on retrouve dans les tâches ou leur contenu, les fonctions de l'opération. Familièrement, on fait la même chose, mais différemment...

#### Première solution

Cette solution (voir figure 8.10) est axée sur la spécialisation des compétences par poste :

- Au comptoir, le magasinier a les compétences techniques pour identifier la demande du client et le conseiller.
- A la caisse, le caissier a la responsabilité commerciale et financière.
- Au stockage, le manutentionnaire n'a que des tâches manuelles.
- Au réappro, le personnel est chargé de trouver un réapprovisionnement rapide, par transfert d'un autre dépôt ou par demande au service Achats.

Pour accélérer la procédure, la caisse et le stockage travaillent en parallèle. Un bon de préparation est édité au stockage.

Dans cette solution, la mise à jour des stocks est antérieure à leur sortie effective, et la facturation est faite avant que le client ait vu ses articles. En cas de non-conformité, il faudrait prévoir une procédure de réintégration qui peut être pénalisante, donc à prendre en considération dans l'appréciation de la solution, en fonction de la probabilité d'un tel cas.

#### Deuxième solution

Cette solution (voir figure 8.11) supprime le poste caisse, si l'on suppose que les conditions commerciales sont connues du système et que la quasi-totalité des clients est en compte.

Cette solution concentre en une seule phase au comptoir l'ensemble des tâches informatisées de vente (conséquences sur les équipements et la conception de l'application). La procédure est toutefois plus longue pour le client. Le découpage en deux phases sur deux postes permet une optimisation des postes et un équilibrage des ressources humaines en fonction du temps de chaque phase (2 magasiniers pour 3 manutentionnaires par exemple).

Cette solution soulève les mêmes critiques quant à la remise des articles après la facturation et la mise à jour des stocks.

#### Troisième solution

Cette solution (voir figure 8.12) essaie de pallier la critique précédente concernant la mise à jour des stocks avant la visualisation des articles et cherche à tout concentrer en un seul poste.

Le comptoir effectue toutes les tâches ; après avoir vérifié la disponibilité, propose un réappro rapide (par transfert ou via les achats) , il va chercher les articles puis les facture.

Cette solution concentre les tâches informatisées. Elle nécessite probablement un écran par magasinier avec cependant une sous-utilisation durant la tâche manuelle de recherche des articles.

Toutefois, si la tâche de recherche des articles dans les rayons est salissante, cette solution risque d'être inadaptée ergonomiquement avec la manipulation d'un clavier lors de la facturation (conséquences sur l'équipement).

#### Quatrième solution

Nous avons également voulu proposer une solution (voir figure 8.13) qui remette complètement en cause l'organisation actuelle : un libre-service.

Le client recherche lui-même ses articles; il constitue d'ailleurs la ressource humaine de ce poste. L'accueil peut le renseigner sur la disponibilité et la localisation des articles, et prendre éventuellement une demande de réapprovisionnement rapide (par transfert ou via les achats). La facturation s'effectue à la caisse de sortie.

L'intérêt d'une telle solution est à évaluer en fonction du type d'activité ; possible quand il s'agit de bricolage mais, à notre avis, peu probable dans le cas de la société X. A travers les quatre solutions, nous avons cherché à mettre en relief leur diversité avec, pour chacune, ses avantages et ses inconvénients ; aucune n'est la meilleure, le choix dépend de la pondération des facteurs d'évaluation.

Nous avons également choisi de ne pas informatiser la tâche d'identification des articles à partir des indications fournies par le client ; « à la demande des utilisateurs », nous conservons les catalogues avec leurs schémas. Par ailleurs, l'informatisation de tels catalogues (schémas compris) et le développement informatique de la tâche d'aide à la recherche de la référence seraient d'un coût disproportionné par rapport à l'informatisation des autres tâches.

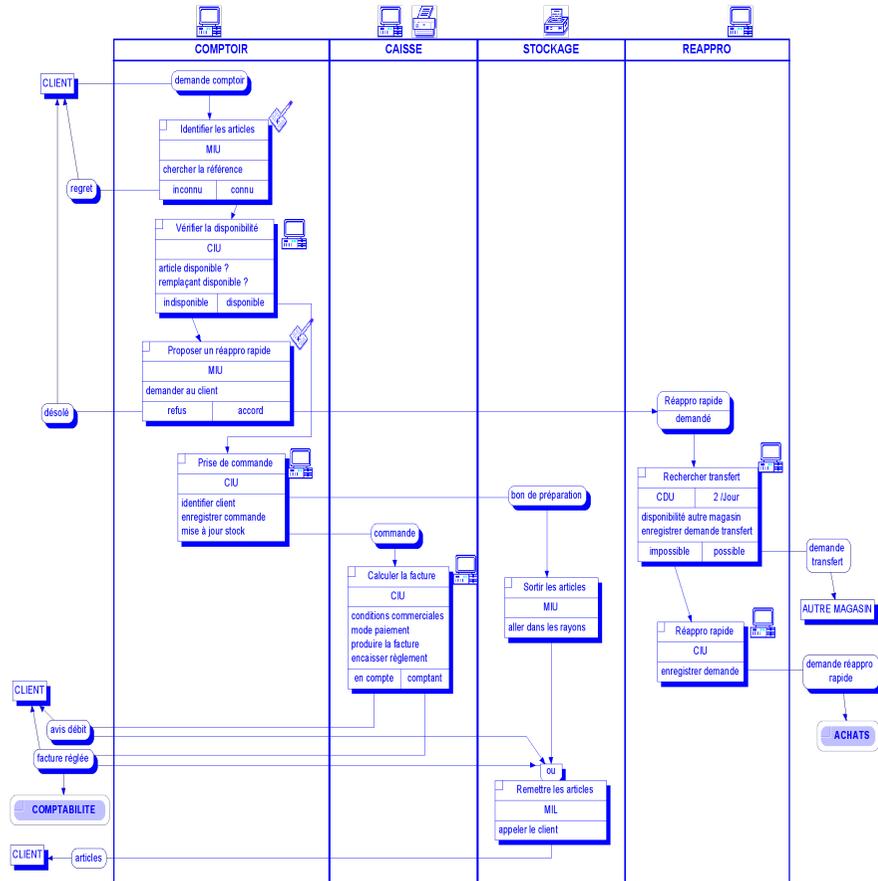


Figure 8.10 : Modèle organisationnel des traitements associé à la solution 1.

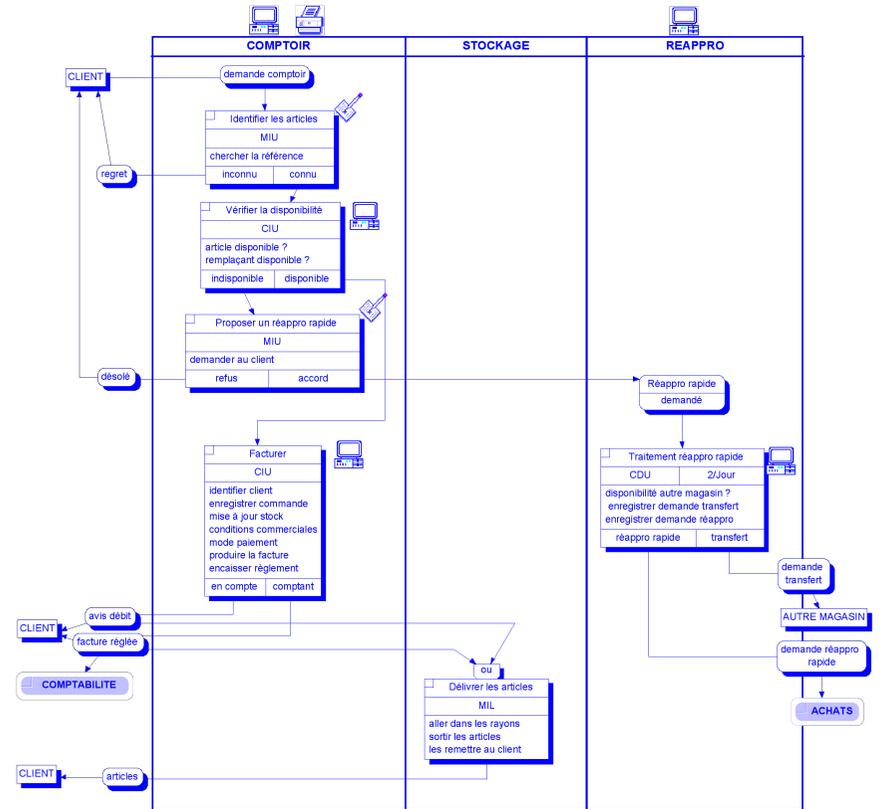


Figure 8.11 : Modèle organisationnel des traitements associé à la solution 2.

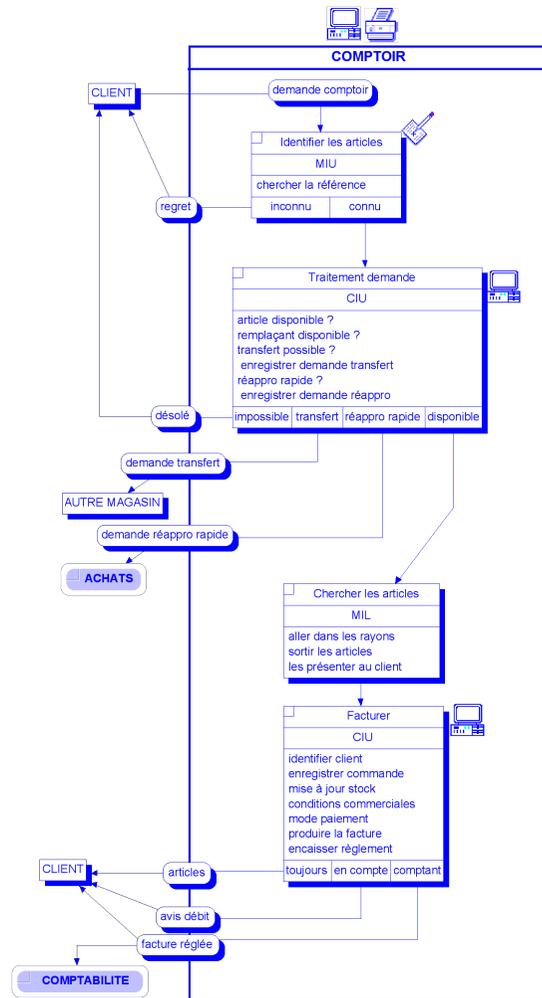


Figure 8.12 : Modèle organisationnel des traitements associé à la solution 3.

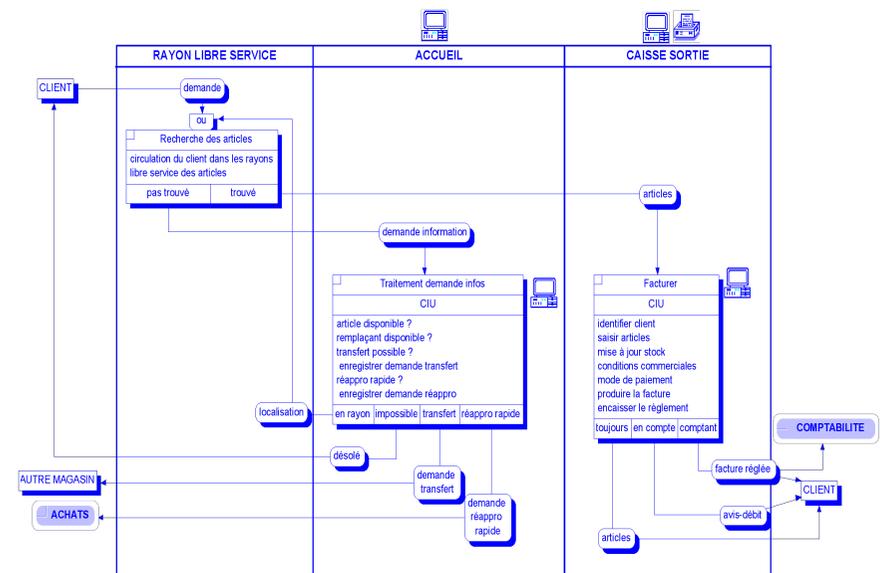


Figure 8.13 : Modèle organisationnel des traitements associé à la solution 4.

### Organisation synchrone ou asynchrone

Dans les modèles organisationnels de traitements, l'enchaînement entre les tâches est un aspect important de l'expression de la procédure organisationnelle. La modélisation de ces enchaînements peut s'effectuer de diverses manières qui, chacune, traduisent des fonctionnements différents.

A sein d'une phase, l'enchaînement entre deux tâches se modélise par une simple flèche (pas d'événement ou d'état intermédiaire), exprimant ainsi la continuité des activités dans la phase (non interruptible).

Par contre, lorsque l'enchaînement s'effectue entre des tâches appartenant à des phases différentes (au sein d'un même poste ou entre des postes distincts), l'enchaînement doit être formalisé par l'intermédiaire soit d'un événement soit d'un état. Le choix de l'une ou l'autre modélisation exprimera un fonctionnement synchrone ou asynchrone de la procédure.

Dans le cas d'un fonctionnement synchrone, modélisé par un événement (quel que soit son support), l'événement émis par l'une des tâches vient "stimuler" l'autre tâche qui la prendra en compte généralement immédiatement (si ses ressources sont disponibles). Le rythme de fonctionnement de la tâche réceptrice est totalement synchronisé par celui de la tâche émettrice.

Dans le cas d'un fonctionnement asynchrone, la tâche émettrice produit en résultat un état qui est repris en entrée de la tâche réceptrice, synchronisé par un événement temporel ou décisionnel; à la différence de l'événement, l'état n'a aucun rôle de stimulus. Le rythme de fonctionnement de la tâche réceptrice est totalement désynchronisé par rapport à celui de la tâche émettrice; la tâche réceptrice doit prendre l'initiative du déclenchement des traitements.

Par exemple, dans le cas de la solution 1, il semble préférable de choisir un fonctionnement synchrone entre la tâche Prise de commande au comptoir et la tâche Calcul de la facture à la caisse. L'enchaînement est donc modélisé par un événement, que ce soit par la transmission d'un document ou par un message informatisé entre les deux postes. La personne à la caisse sera avertie immédiatement qu'une commande est à facturer; normalement la personne à la caisse devrait traiter dans la séquence la commande transmise.

Toujours dans le cas de solution 1, il semble par contre préférable de choisir un fonctionnement asynchrone entre la tâche Proposer un réappro rapide au comptoir et la tâche Rechercher un transfert au réappro. L'enchaînement sera modélisé par un état; la tâche Rechercher transfert étant déclenchée à l'initiative de la personne du réappro, par exemple en fin demi-journée.

Dans le cas d'une modélisation de fonctionnement asynchrone, il n'est pas nécessaire de faire obligatoirement figurer l'enchaînement entre les tâches. On retrouve alors la même problématique que celle évoquée au niveau des MCT entre l'approche orientée processus et l'approche orientée états.

Le choix d'une organisation synchrone ou asynchrone relève également des préoccupations du B.P.R. et trouve aujourd'hui un écho particulier dans les nouvelles approches de "workflow".

## Modèle organisationnel de données global

### Volume

Nous supposons que les informations permettant d'estimer le volume ont été recueillies auprès des futurs utilisateurs. ces informations sont récapitulées dans le tableau de la figure 9.11.

RELATION	ENTITE	cardinalité moyenne	taux de participation
composer	COMMANDE	4,2	1
	ARTICLE	2,8	≈0,5
concerner	MAGASIN	6 000	1
	COMMANDE	1	1
constitué de	ARTICLE	0,01	≈0,01
	TRANSFERT	1,3	1
cumul ventes	ARTICLE	25	1
	MAGASIN	220 000	1
	MOIS	30 000	1
destiné à	CLIENT	3	≈1
	RELEVÉ	1	1
disposer	ARTICLE	3	1
	MAGASIN	27 000	1
effectuer	MAGASIN	20	1
	DEMANDE	1	1
	REAPPRO		
facturer tiers	COMMANDE	1	0,01
	CLIENT	0,12	0,005
fournisseur	MAGASIN	60	1
	TRANSFERT	1	1
passer	COMMANDE	1	1
	CLIENT	12	≈1
porter sur	DEMANDE	1,3	1
	ARTICLE	0,003	
préférer	CLIENT	0,4	0,04
	ARTICLE	0,02	≈0,02
	MAGASIN	200	≈1
stocker	ARTICLE	3	1
	MAGASIN	27 000	1
regrouper	COMMANDE	1	1
	RELEVÉ	4	1
substituer	ARTICLE	0,3	0,2
	ARTICLE	0,3	≈0,2
valable pour	ARTICLE	6,4	0,75
	MATERIEL	725	1

Figure 9.11 a : Tableaux récapitulatifs d'évaluation du volume

## 5 - MODELES ORGANISATIONNELS DES DONNEES DU CAS X

Dans le cas de la société X, on peut considérer que le MOD actuel est sans intérêt, il s'agira donc d'un MOD futur.

L'élaboration d'un modèle organisationnel de données a pour objectif la prise en compte, à partir du modèle conceptuel, des aspects de :

- volume et durée de vie des informations,
- choix de mémorisation des informations,
- choix de répartition organisationnelle des informations.

Ce modèle s'exprime dans le même formalisme que le MCD.

Pour illustrer différents aspects de la modélisation organisationnelle des données, nous avons choisi différentes hypothèses successives.

ENTITE - Relation	Total taille propriétés	Nombre d'occurrences	Volume
ARTICLE	170	45 000	7 650 000
CLIENT	250	2 500	625 000
COMMANDE-FACTURE	90	30 000	2 700 000
CONSOMMABLE	15	5 000	75 000
DEMANDE REAPPRO	40	100	4 000
ENGIN	30	400	12 000
MAGASIN	20	5	100
MOIS	4	36	144
OUTILLAGE	30	10 000	300 000
PIECE	6	30 000	180 000
RELEVÉ MENSUEL	45	7 500	337 500
TRANSFERT	50	300	15 000
composer	40	125 000	5 000 000
concerner	0	30 000	0
constitué de	20	400	8 000
cumul ventes	10	1 100 000	11 000 000
demandeur	0	300	0
destiné à	0	7 500	0
disposer	20	135 000	2 700 000
effectuer	0	60	0
facturer tiers	0	3 000	0
fournisseur	0	300	0
passer	0	30 000	0
porter sur	20	130	2 600
préférer	0	1 000	0
regrouper	0	30 000	0
substituer	0	12 500	0
valable pour	0	290 000	0
<b>Total</b>			<b>30 609 344</b>

Soit environ **30 millions** de caractères

Figure 9.11 b : Tableaux récapitulatifs d'évaluation du volume

### Impact de la durée de vie des informations

Supposons que, pour des motifs de volume, nous limitons la durée de vie des commandes - factures à trois mois. En conséquence, une information telle que cumul des quantités vendues /article/dépôt/mois, à conserver sur trois années, est incalculable au-delà de trois mois. Il faut donc faire de ce cumul une propriété, ce qui conduit à la création de la relation cumul ventes et de l'entité MOIS (relation qui intègre la chronique des valeurs).

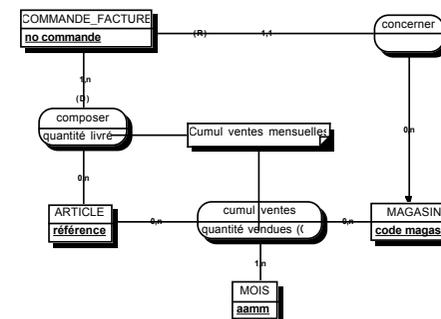


Figure 9.12 : Impact de la durée de vie sur les informations calculées

### Choix de mémorisation

Nous avons fait l'hypothèse que les coordonnées des magasins (adresse et n° de téléphone) étaient parfaitement connues de l'ensemble du personnel et que leur mémorisation était superflue.

Le MOD global résultant est représenté par la figure 9.13

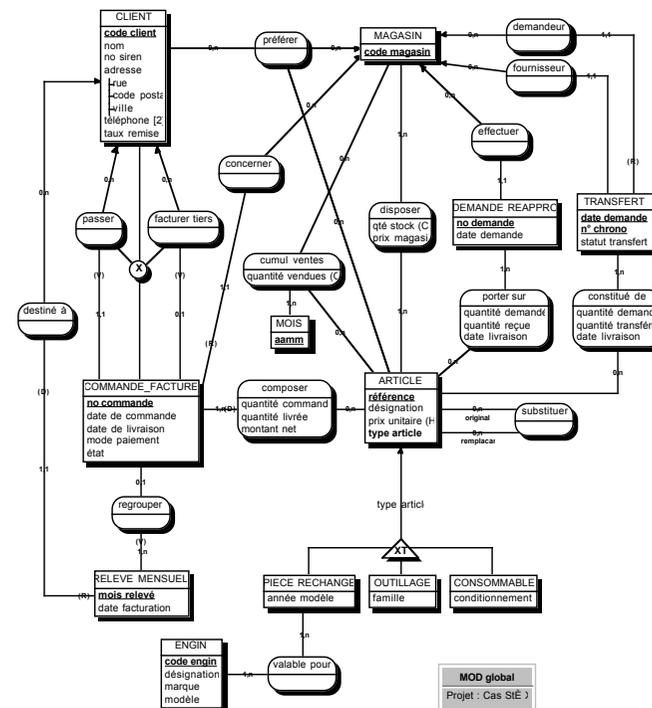


Figure 9.13 : Modèle organisationnel de données global

### Modèles organisationnels de données répartis

Nous avons fait l'hypothèse de trois unités organisationnelles :

- l'unité «Vente» gérant les ventes au niveau de chaque magasin,
- l'unité «Réapprovisionnement», gérant les livraisons, les transferts et les réapprovisionnements rapides au niveau de chaque magasin ,
- l'unité «Direction commerciale» gérant le catalogue des articles.

*Remarque.* Ces hypothèses organisationnelles correspondent aux solutions 1, 2 et 4 proposées dans la modélisation organisationnelle des traitements. Par contre, le choix de la solution 3 impliquerait une autre répartition organisationnelle des données qui, ultérieurement pourrait influencer la répartition logique des données.

Les modèles organisationnels de données locaux correspondant aux trois unités organisationnelles sont respectivement présentés aux figures 9.14, 9.15 et 9.16.

On remarquera que les informations liées à la validité des pièces selon les engins (entité Engin, relation valable pour) sont absentes du MOD Vente, la tâche d'identification des articles où ces informations sont utilisées étant non informatisée (voir commentaires sur les solutions organisationnelles proposées). Par contre, la constitution des catalogues est à la charge de la direction commerciale.

Le figure 9.17 récapitule les accessibilités des différentes entités et relations en termes d'actions autorisées (Création, Modification, Suppression, Lecture) et de restriction sur les occurrences. En cas de partage de données en mise à jour (C,M,S) entre différentes unités organisationnelles, on indique quelle est l'unité qui détient les données de référence (R).

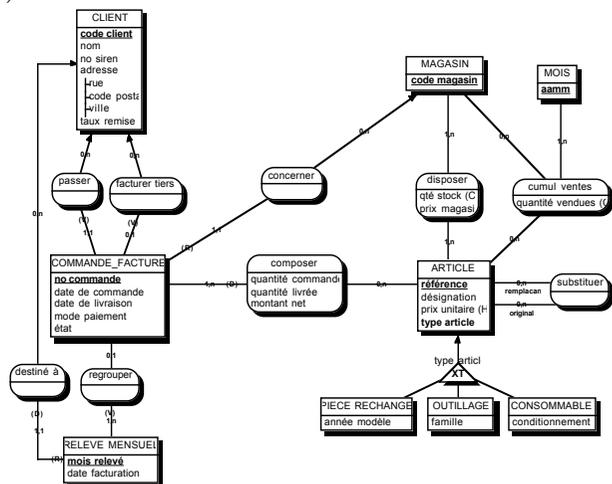


Figure 9.14 : Modèle organisationnel de données local - Unité organisationnelle Ventes

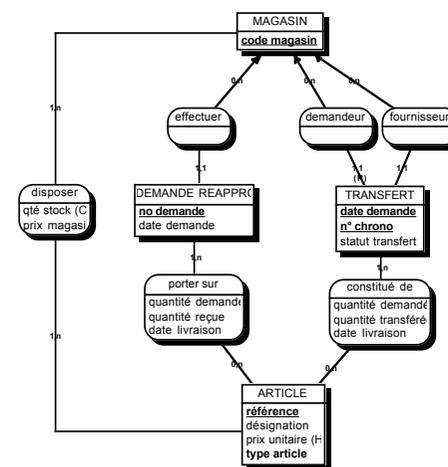


Figure 9.15 : Modèle organisationnel de données local - Unité organisationnelle Réapprovisionnement

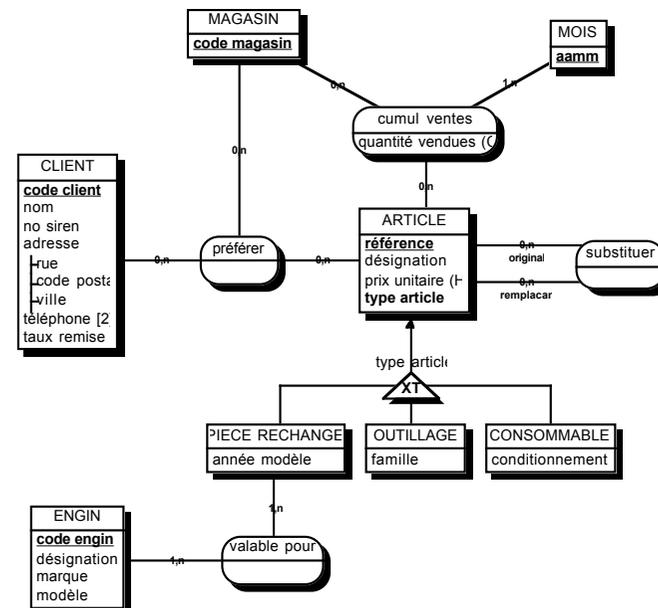


Figure 9.16 : Modèle organisationnel de données local - Unité organisationnelle Direction commerciale

Entité-Relation	VENTES			REAPPRO			DIRECTIO N	
	accès	restriction		accès	restriction		accès	restriction
ARTICLE	L			L			CMSL	
CLIENT	CML	ceux du magasin	R				CLS	
COMMANDE-F.	CMSL	celles du magasin						
CONSOMMABLE	L						CMSL	
DEMANDE REA.				CMSL	celles du magasin			
ENGIN							CMSL	
MAGASIN	L			L			L	
MOIS	CL						SL	
OUTILLAGE	L						CMSL	
PIECE	L						CMSL	
RELEVÉ MENS.	CMSL	ceux du magasin						
TRANSFERT				CMSL ML	ceux demandés ceux fournis			
composer	CMSL	ceux du magasin						
concerner	CMSL	ceux du magasin						
constitué de				CMSL ML	ceux demandés ceux fournis			
cumul ventes demandeur	CML	ceux du magasin	R				LS	
destiné à	CMLS	ceux du magasin						
disposer	ML	ceux du magasin	R	CMSL L	ceux du magasin autres magasins			
effectuer				CMSL	ceux du magasin			
facturer tiers fournisseur	CMLS	ceux du magasin		CMSL L	ceux demandés ceux fournis			
passer	CMSL	ceux du magasin						
porter sur				CMSL	ceux du magasin			
préférer	L						CMSL	
regrouper	CMSL	ceux du magasin						
substituer	L						CMSL	
valable pour							CMSL	

Figure 9.17 : Tableau des accessibilité des MOD locaux

## 6 - MODELES LOGIQUES DE DONNEES DU CAS X

### Modèle logique de données relationnel brut

Pour élaborer ce modèle logique de données relationnel, nous sommes partis du modèle organisationnel de données global (fig. 9.13), auquel nous avons appliqué les règles de transformation. Le modèle logique de données obtenu est dit brut (non optimisé).

Le MLD relationnel brut global est représenté à la figure 12.50

#### Transformation des concepts avancés

##### Propriété multivaluée

La propriété multivaluée Téléphone [2] de client a été transformée en Téléphone\_1 et Téléphone\_2.

##### Propriété composée

Dans la propriété composé Adresse de client, seules les propriétés composantes ont été transformées.

##### Sous-types (Article, Pièce rechange, Outillage, Consommable)

Nous avons choisi de rassembler l'ensemble des sous-types dans le sur-type, avec éventuellement recours à la définition de vues au niveau physique.

##### Historisation (prix tarif d'Article)

Il donne lieu à la création de la table H\_PU\_TARIF\_ART dont la clé primaire est composée de référence article et de la date de changement de prix tarif.

##### Règle (Limite prix magasin)

Les règles sont reconduites telles quelles au niveau logique où elles s'appliquent sur les tables et les attributs.

##### Contrainte d'intégrité fonctionnelle (relation préférer)

La contrainte conceptuelle Client x Article → Magasin se traduit dans la table PREF par une clé primaire composée uniquement de Nocli et Ref\_art (alors que la relation d'origine est ternaire) et Code\_mag seulement en clé étrangère.

##### Contraintes inter relation (exclusion entre passer et facturer tiers)

Cette contrainte conceptuelle s'exprimera par un trigger au niveau du modèle physique de données. Elle exprime que, dans la table CMD\_FACT, Nocli\_fact\_tiers doit être différent de Nocli\_passer.

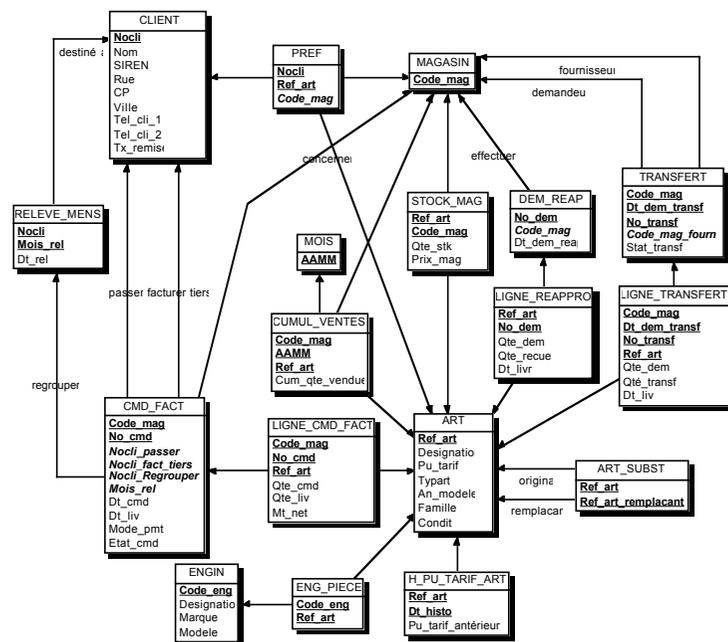


Figure 12.50 : Modèle logique de données relationnel brut global

### Modèles logiques de données relationnels répartis

Nous avons voulu illustrer, sur le modèle relationnel, un exemple de répartition logique des données. Rappelons encore la différence entre répartition organisationnelle et répartition logique des données. La répartition organisationnelle est perceptible par l'utilisateur et concerne l'organisation de ses tâches à travers le besoin et la disponibilité des données; nous l'avons vu dans l'exemple des MOD répartis (Vente, Réapprovisionnement, Direction commerciale).

La répartition logique est liée à l'implantation informatique des données sur des machines logiques différentes. La répartition logique est transparente à l'utilisateur, n'a aucun impact sur son organisation mais concerne exclusivement l'informatique.

La problématique de la répartition informatique (logique et physique) des données est encore plus d'actualité avec le développement des architectures client / serveur. Les critères de répartition reposent souvent sur des considérations techniques (volume des échanges, capacité des systèmes). On peut toutefois tirer parti de la répartition organisationnelle pour tenter de rapprocher le plus possible les données de leur lieu d'utilisation, en distinguant entre autre la mise à jour et la lecture. Le chapitre 27 «Merise et le client serveur» développe quelques réflexions dans ce domaine.

De toute façon, la répartition des données au niveau logique est un arbitrage entre des considérations techniques (caractéristiques des matériels) et l'utilisation des données mémorisées (localisation, fréquence et nature des accès) qui provient du niveau organisationnel.

Dans le cas de la société X, nous faisons l'hypothèse d'une architecture matérielle suivante :

- dans chaque magasin, un serveur et des postes clients ,
- au siège, un serveur suffisamment dimensionné pour accueillir en outre l'ensemble des applications spécifiques au siège (achats, comptabilité, personnel) et des postes clients.
- le siège et les magasins sont reliés en réseau.

Pour ce qui concerne la répartition informatique des données de l'activité commerciale étudiée, nous avons retenu les arguments suivants :

- nous n'avons a priori aucune contrainte technique nous obligeant à implanter les données sur un site précis.
- les données utilisées pour l'activité de vente (voir MOD Ventes) sont spécifiques à chaque magasin et demandent un accès assez fréquent en mise à jour; il est donc normal de les implanter essentiellement sur les machines Magasin, avec éventuellement une duplication des clients et des statistiques au Siège.
- les données de réapprovisionnement, bien qu'utilisées au niveau de chaque magasin (voir MOD Réapprovisionnement) sont partagées non seulement entre les différents magasins mais également avec les Achats situés au siège (voir hypothèse du MCT futur); nous avons choisi de les centraliser au niveau du serveur du siège, mais avec une duplication des informations sur les stocks au niveau de chaque magasin.
- les données techniques sur les articles d'une part sont fortement utilisées en lecture au niveau des magasins, d'autre part sont gérées sous la responsabilité de la direction commerciale qui seule en effectue la mise à jour; nous avons donc choisi de les centraliser au niveau du serveur du siège avec duplication partielle dans les magasins.

Les modèles logiques de données locaux sont présentés en figures 12.51 et 12.52.

### Mise en cohérence de données dupliquées

Il y a duplication de données lorsque des occurrences identiques de tables sont mémorisées sur des sites différents. Il est alors nécessaire de préciser les mécanismes qui permettront d'assurer la cohérence des valeurs; leur choix dépend des utilisations de ces données (fréquence, action).

Pour les tables partagées entre Magasin et Siège, nous justifions succinctement les solutions adoptées.

### Article et article substituable

Les magasins utilisent ces informations en lecture très fréquemment; mais c'est le siège qui les fait vivre (création, mise à jour des tarifs, ...) à une fréquence plus épisodique. Nous avons choisi une mise à jour en différé des données des magasins à partir du siège, suite aux opérations de mise à jour du catalogue.

### Cumul ventes

Le siège utilise régulièrement ces statistiques qui sont élaborées mensuellement au niveau des magasins. Nous avons choisi un transfert mensuel en différé des magasins au siège.

### Stock magasin

Ces données dupliquées doivent être toujours en cohérence. Leur mise à jour est faite au magasin (sortie dans l'activité de vente ou transfert, entrée dans l'activité de réception livraison) mais doit être immédiatement transmise au siège. Nous avons choisi une mise à jour instantanée (commit en deux phases).

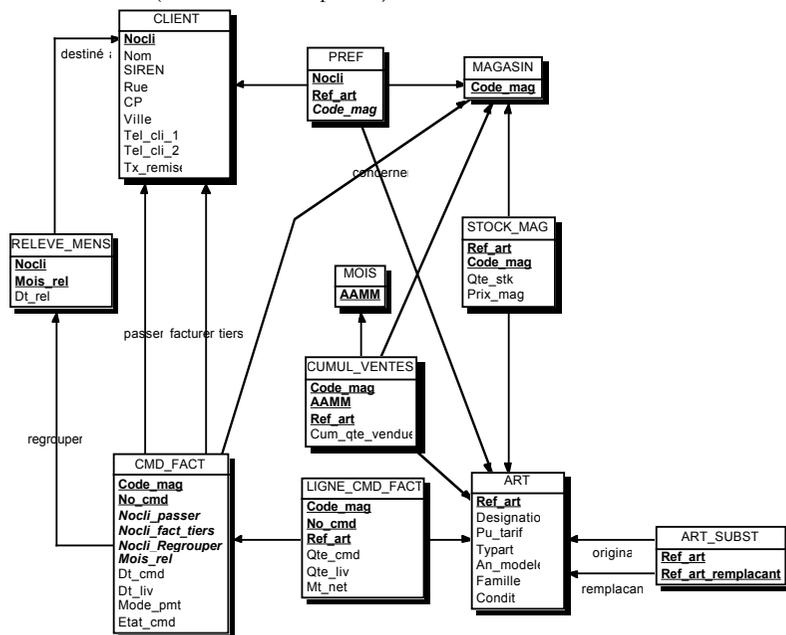


Figure 12.51 : Modèle logique de données relationnel local Site Magasin

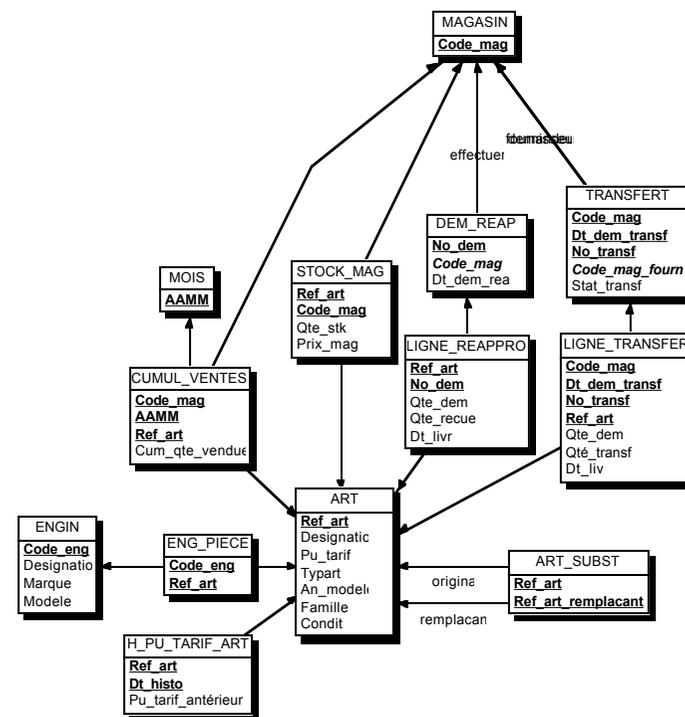


Figure 12.52 : Modèle logique de données relationnel local Site Siège

## 7 - OPTIMISATION DU MODELE LOGIQUE DE DONNEES DU CAS X ET MODELE PHYSIQUE

### Modèle logique optimisé

L'optimisation des modèles logiques de données se met en oeuvre à partir des modèles logiques bruts, éventuellement répartis, en appliquant les différentes techniques présentées, compte tenu des conditions d'utilisation des données mémorisées.

Rappelons que dans le cas X, nous avons abouti à deux modèles logiques bruts : Magasin et Siège (voir chap. 12). Sur ces derniers, nous avons réalisés quelques optimisations que nous justifions.

**Modèle logique Magasin**

**Migration de la table Magasin**

Au niveau de chaque magasin, cette table qui ne contient que le code magasin n'a aucun intérêt en terme référentiel. On supprime donc les liens référentiels des tables CUMUL\_VENTES, STOCK\_MAG, CMD\_FACT et PREF. Les ex clés étrangères deviennent alors de simples attributs sauf s'ils participaient à la clé primaire. On supprime ensuite la table Magasin qui n'a plus d'utilité.

Remarque: Sachant que ce MLD est local à chaque magasin, on pourrait estimer que l'attribut Code\_mag est superflu dans les différentes tables où il apparaît, en particulier dans les tables CUMUL\_VENTES et STOCK\_MAG. Dans le cas de STOCK\_MAG, on pourrait même envisager de fusionner cette table avec la table ART (elles auraient alors la même clé primaire). Cependant, en transformant ainsi ces tables qui sont partagées entre les magasins et le siège, la mise en cohérence (par répllication ou commit deux phases) serait plus difficile à assurer.

**Migration de la table Mois**

Le rôle référentiel joué par la table MOIS n'a aucun intérêt. On applique donc le même processus que ci-dessus.

**Index**

Les index permettent un accès rapide aux occurrences d'une table selon certains attributs. En dehors des attributs participants aux clés primaires et étrangères, nous avons choisi de définir un index sur les attributs suivants :

- Nom de CLIENT
- Dt\_cmd de CMD\_FACT

Nous avons également choisi de ne pas implémenter d'index sur certaines clés primaires ou étrangères ou d'en modifier la composition compte tenu de la faible utilisation supposée de ces accès :

- tous les Code\_mag intervenant dans des index,
- Réf\_art\_remplaçant dans ART\_SUBST.

**Modèle logique Siège**

**Migration des tables Magasin et Mois**

On réalise les mêmes optimisations que dans le modèle logique Magasin

**Index**

- Nous avons choisi de définir un index sur les attributs suivants :
- Dt\_dem\_reapp de DEM\_REAP
  - Stat\_transf de TRANSFERT

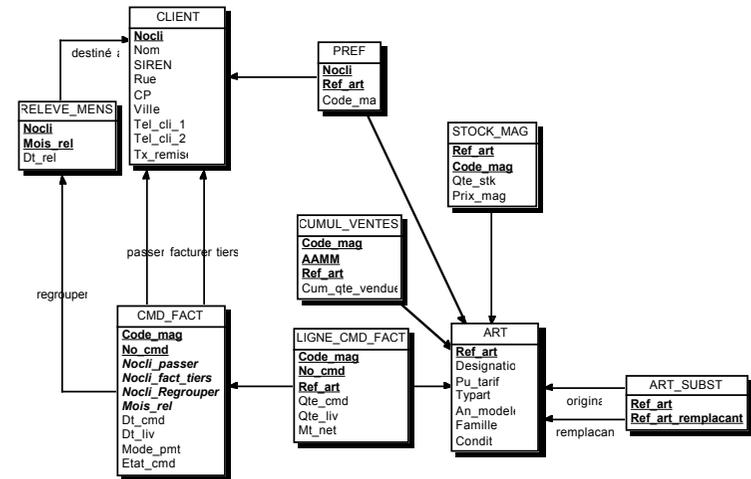


Figure14.x1 : Modèle logique optimisé - Site Magasin

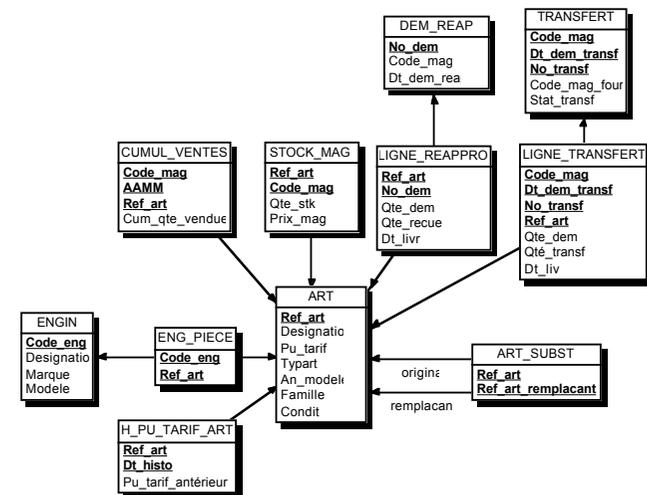


Figure14.21 : Modèle logique optimisé - Site Siège

**Modèle physique**

Le modèle physique est concrétisé par le script de création (ou de modification) de la base de données dans la syntaxe du S.G.B.D. choisi.

Pour illustrer l'aboutissement de la modélisation du cas X, nous présentons l'essentiel du script de création de la base Magasin en Oracle 7.

Remarque. Pour la gestion des contraintes d'intégrité référentielles, la version 7 d'Oracle (au premier semestre 96) ne dispose que de l'option ON DELETE CASCADE. Par ailleurs, nombre d'utilisateurs préfèrent recourir malgré tout à des triggers. Pour ces motifs, nous présentons également des extraits du script de création des triggers permettant de gérer ces contraintes référentielles des principales tables

```

-----
-- Génération d'une base de données pour
-- Oracle Version 7.0
-- (30/4/1996 16:43:06)
-----
-- Nom de la base : BD_Magasin
-- Projet : Cas Sté X
-- Auteur :
-- Date de dernière modification : 30/4/1996 16:15:57
-----
-- TABLE : ART
-----

CREATE TABLE ART
(
  Ref_art CHAR(32) NOT NULL,
  Designation CHAR(32) ,
  Pu_tarif NUMBER(10,2) ,
  Typart CHAR(16)
  DEFAULT 'ME' NOT NULL
  CONSTRAINT ck_in_Typart_ART CHECK (Typart IN ('PR',
'OU', 'CO')),
  An_modele NUMBER(4) ,
  Famille NUMBER(4) ,
  Condit CHAR(16)
)
  CONSTRAINT PK_ART PRIMARY KEY (Ref_art)
);

COMMENT ON TABLE ART
  IS 'ensemble des articles vendues par la société X';

COMMENT ON COLUMN ART.Ref_art
  IS 'code identifiant un article';

COMMENT ON COLUMN ART.Pu_tarif
  IS 'prix unitaire tarif catalogue';

COMMENT ON COLUMN ART.Typart
  IS 'permet de distinguer les types d'article (cf domaine de
valeurs)';

-----
-- TABLE : CLIENT
-----

CREATE TABLE CLIENT
(
  Nocli NUMBER(6) NOT NULL,
  Nom CHAR(32) ,
  SIREN NUMBER(12,0) ,
  Rue CHAR(32) ,
  CP CHAR(5) ,
  Ville CHAR(32) ,
  Tel_cli_1 NUMBER(10,0) ,
  Tel_cli_2 NUMBER(10,0) ,
  Tx_remise NUMBER(5,2)
)
  CONSTRAINT PK_CLIENT PRIMARY KEY (Nocli)
);

COMMENT ON TABLE CLIENT
  IS 'client de la société X.';

-----
-- INDEX DE LA TABLE CLIENT
-----

CREATE INDEX I_nom_cli
  ON CLIENT (Nom ASC);

-----
-- TABLE : CMD_FACT
-----

```

```

CREATE TABLE CMD_FACT
(
  Code_mag CHAR(5) NOT NULL,
  No_cmd NUMBER(4) NOT NULL,
  Nocli_passer NUMBER(6) NOT NULL,
  Nocli_fact_tiers NUMBER(6) ,
  Nocli_Regrouper NUMBER(6) ,
  Mois_rel CHAR(6) ,
  Dt_cmd DATE ,
  Dt_liv DATE ,
  Mode_pmt CHAR(4) ,
  Etat_cmd CHAR(16)
)
  CONSTRAINT PK_CMD_FACT PRIMARY KEY (Code_mag,
No_cmd)
);

COMMENT ON TABLE CMD_FACT
  IS 'commande d'articles par un client dans un dépôt';

COMMENT ON COLUMN CMD_FACT.No_cmd
  IS 'no identifiant de la commande';

-----
-- INDEX DE
LA TABLE CMD_FACT
-----

CREATE INDEX I_FK_passer
  ON CMD_FACT (Nocli_passer ASC);

CREATE INDEX I_FK_FACT_TIERS
  ON CMD_FACT (Nocli_fact_tiers ASC);

CREATE INDEX I_FK_Regrouper
  ON CMD_FACT (Nocli_Regrouper ASC, Mois_rel ASC);

CREATE INDEX I_dt_cmd
  ON CMD_FACT (Dt_cmd ASC);

-----
-- TABLE : RELEV_MENS
-----

CREATE TABLE RELEV_MENS
(
  Nocli NUMBER(6) NOT NULL,
  Mois_rel CHAR(6) NOT NULL,
  Dt_rel DATE
)
  CONSTRAINT PK_RELEV_MENS PRIMARY KEY (Nocli,
Mois_rel)
);

-----
-- TABLE : CUMUL_VENTES
-----

CREATE TABLE CUMUL_VENTES
(
  Code_mag CHAR(5) NOT NULL,
  AAMM NUMBER(6) NOT NULL,
  Ref_art CHAR(32) NOT NULL,
  Cum_qte_vendues NUMBER(4)
)
  CONSTRAINT PK_CUMUL_VENTES PRIMARY KEY (AAMM,
Code_mag, Ref_art)
);

-----
-- TABLE : LIGNE_CMD_FACT
-----

CREATE TABLE LIGNE_CMD_FACT
(

```

Merise : Solutions du Cas Société X

```

Code_mag CHAR(5) NOT NULL,
No_cmd NUMBER(4) NOT NULL,
Ref_art CHAR(32) NOT NULL,
Qte_cmd NUMBER(4) ,
Qte_liv NUMBER(4) ,
Mt_net NUMBER(10,2)

CONSTRAINT PK_LIGNE_CMD_FACT PRIMARY KEY
(Code_mag, No_cmd, Ref_art)
);

COMMENT ON TABLE LIGNE_CMD_FACT
IS 'une commande est composée d'au mois une ligne de
commande';

COMMENT ON COLUMN LIGNE_CMD_FACT.Qte_cmd
IS 'quantité commandée d'un article dans une ligne article
d'une commande';

COMMENT ON COLUMN LIGNE_CMD_FACT.Qte_liv
IS 'quantité livrée d'un article commandé';

COMMENT ON COLUMN LIGNE_CMD_FACT.Mt_net
IS 'montant net après remise de la ligne de commande';

-----
-- TABLE : STOCK_MAG
-----

CREATE TABLE STOCK_MAG
(
Ref_art CHAR(32) NOT NULL,
Code_mag CHAR(5) NOT NULL,
Qte_stk NUMBER(4) ,
Prix_mag NUMBER(10,2)

CONSTRAINT PK_STOCK_MAG PRIMARY KEY (Code_mag,
Ref_art)
);

COMMENT ON COLUMN STOCK_MAG.Qte_stk
IS 'quantité en stock d'un article dans un dépôt.';

-----
-- TABLE : PREF
-----

CREATE TABLE PREF
(
Nocli NUMBER(6) NOT NULL,
Ref_art CHAR(32) NOT NULL,
Code_mag CHAR(5)

CONSTRAINT PK_PREF PRIMARY KEY (Nocli, Ref_art)
);

-----
-- TABLE : ART_SUBST
-----

CREATE TABLE ART_SUBST
(
Ref_art CHAR(32) NOT NULL,
Ref_art_replacant CHAR(32) NOT NULL

CONSTRAINT PK_ART_SUBST PRIMARY KEY
(Ref_art_replacant, Ref_art)
);

COMMENT ON TABLE ART_SUBST
IS 'articles substituables à d'autres articles en cas de rupture
de stock';

-----
-- CREATION DES REFERENCES DE TABLE
-----

ALTER TABLE CMD_FACT ADD (
CONSTRAINT FK_passer
FOREIGN KEY (Nocli_passer)
REFERENCES CLIENT (Nocli));

ALTER TABLE CMD_FACT ADD (
CONSTRAINT FK_FACT_TIERS
FOREIGN KEY (Nocli_fact_tiers)
REFERENCES CLIENT (Nocli));

ALTER TABLE CMD_FACT ADD (
CONSTRAINT FK_Regrouper
FOREIGN KEY (Nocli_Regrouper, Mois_rel)
REFERENCES RELEV_MENS (Nocli, Mois_rel));

ALTER TABLE RELEV_MENS ADD (
CONSTRAINT FK_destiné_à
FOREIGN KEY (Nocli)
REFERENCES CLIENT (Nocli));

ALTER TABLE CUMUL_VENTES ADD (
CONSTRAINT FK_ART
FOREIGN KEY (Ref_art)
REFERENCES ART (Ref_art));

ALTER TABLE LIGNE_CMD_FACT ADD (
CONSTRAINT FK_ART1
FOREIGN KEY (Ref_art)
REFERENCES ART (Ref_art));

ALTER TABLE LIGNE_CMD_FACT ADD (
CONSTRAINT FK_CMD_FACT
FOREIGN KEY (Code_mag, No_cmd)
REFERENCES CMD_FACT (Code_mag, No_cmd));

ALTER TABLE STOCK_MAG ADD (
CONSTRAINT FK_ART2
FOREIGN KEY (Ref_art)
REFERENCES ART (Ref_art));

ALTER TABLE PREF ADD (
CONSTRAINT FK_ART3
FOREIGN KEY (Ref_art)
REFERENCES ART (Ref_art));

ALTER TABLE PREF ADD (
CONSTRAINT FK_CLIENT
FOREIGN KEY (Nocli)
REFERENCES CLIENT (Nocli));

ALTER TABLE ART_SUBST ADD (
CONSTRAINT FK_ART4
FOREIGN KEY (Ref_art)
REFERENCES ART (Ref_art));

ALTER TABLE ART_SUBST ADD (
CONSTRAINT FK_replacant
FOREIGN KEY (Ref_art_replacant)
REFERENCES ART (Ref_art));

-----
-- FIN DE GENERATION
-----

```

Merise : Solutions du Cas Société X

```

-----
-- Génération des triggers de la base
-- de données : BD_Magasin
-- (30/4/1996 17:06:57)
-----

-----
-- Table : ART
-----

-- Trigger de suppression -----
create trigger TD_ART
after delete on ART for each row
declare numrows INTEGER;
begin

-- Supprimer les occurrences correspondantes de la table
CUMUL_VENTES.

delete from CUMUL_VENTES
where
CUMUL_VENTES.Ref_art = :old.Ref_art;
-- Interdire la suppression d'une occurrence de ART s'il existe
des
-- occurrences correspondantes de la table
LIGNE_CMD_FACT.

select count(*) into numrows
from LIGNE_CMD_FACT
where
LIGNE_CMD_FACT.Ref_art = :old.Ref_art;
if (numrows > 0) then
raise_application_error(
-20001,
'Impossible de supprimer "ART". Des occurrences de
"LIGNE_CMD_FACT" existent. ');
end if;
-- Supprimer les occurrences correspondantes de la table
STOCK_MAG.

delete from STOCK_MAG
where
STOCK_MAG.Ref_art = :old.Ref_art;
-- Supprimer les occurrences correspondantes de la table
PREF.

delete from PREF
where
PREF.Ref_art = :old.Ref_art;
-- Supprimer les occurrences correspondantes de la table
ART_SUBST.

delete from ART_SUBST
where
ART_SUBST.Ref_art = :old.Ref_art;
-- Supprimer les occurrences correspondantes de la table
ART_SUBST.

delete from ART_SUBST
where
ART_SUBST.Ref_art_replacant = :old.Ref_art;

end;
/

-- Trigger de modification -----
create trigger TU_ART
after update on ART for each row
declare numrows INTEGER;
begin

-- Ne pas modifier la clé primaire de la table ART s'il existe des
-- occurrences correspondantes dans la table
CUMUL_VENTES.

if
:old.Ref_art <> :new.Ref_art

```

Merise : Solutions du Cas Société X

```

ART_SUBST.Ref_art = :old.Ref_art;
end if;
-- Répercuter la modification de la clé primaire de ART sur les
-- occurrences correspondantes de la table ART_SUBST.

if
:old.Ref_art <> :new.Ref_art
then
update ART_SUBST
set
ART_SUBST.Ref_art_remplacant = :new.Ref_art
where
ART_SUBST.Ref_art_remplacant = :old.Ref_art;
end if;

end;
/

-----
-- Table : CLIENT
-----

-- Trigger de suppression -----
create trigger TD_CLIENT
after delete on CLIENT for each row
declare numrows INTEGER;
begin

-- Interdire la suppression d'une occurrence de CLIENT s'il
existe des
-- occurrences correspondantes de la table RELEVEMENTS.

select count(*) into numrows
from RELEVEMENTS
where
RELEVEMENTS.Nocli = :old.Nocli;
if (numrows > 0) then
raise_application_error(
-20001,
'Impossible de supprimer "CLIENT". Des occurrences de
"RELEVEMENTS" existent.);
end if;
-- Interdire la suppression d'une occurrence de CLIENT s'il
existe des
-- occurrences correspondantes de la table CMD_FACT.

select count(*) into numrows
from CMD_FACT
where
CMD_FACT.Nocli_passer = :old.Nocli;
if (numrows > 0) then
raise_application_error(
-20001,
'Impossible de supprimer "CLIENT". Des occurrences de
"CMD_FACT" existent.);
end if;
-- Interdire la suppression d'une occurrence de CLIENT s'il
existe des
-- occurrences correspondantes de la table CMD_FACT.

select count(*) into numrows
from CMD_FACT
where
CMD_FACT.Nocli_fact_tiers = :old.Nocli;
if (numrows > 0) then
raise_application_error(
-20001,
'Impossible de supprimer "CLIENT". Des occurrences de
"CMD_FACT" existent.);
end if;
-- Supprimer les occurrences correspondantes de la table
PREF.

delete from PREF
where

```

45

```

PREF.Nocli = :old.Nocli;
end;
/

-- Trigger de modification -----
create trigger TU_CLIENT
after update on CLIENT for each row
declare numrows INTEGER;
begin

-- Répercuter la modification de la clé primaire de CLIENT sur
-- occurrences correspondantes de la table RELEVEMENTS.

if
:old.Nocli <> :new.Nocli
then
update RELEVEMENTS
set
RELEVEMENTS.Nocli = :new.Nocli
where
RELEVEMENTS.Nocli = :old.Nocli;
end if;
-- Répercuter la modification de la clé primaire de CLIENT sur
-- occurrences correspondantes de la table CMD_FACT.

if
:old.Nocli <> :new.Nocli
then
update CMD_FACT
set
CMD_FACT.Nocli_passer = :new.Nocli
where
CMD_FACT.Nocli_passer = :old.Nocli;
end if;
-- Répercuter la modification de la clé primaire de CLIENT sur
-- occurrences correspondantes de la table CMD_FACT.

if
:old.Nocli <> :new.Nocli
then
update CMD_FACT
set
CMD_FACT.Nocli_fact_tiers = :new.Nocli
where
CMD_FACT.Nocli_fact_tiers = :old.Nocli;
end if;
-- Répercuter la modification de la clé primaire de CLIENT sur
-- occurrences correspondantes de la table PREF.

if
:old.Nocli <> :new.Nocli
then
update PREF
set
PREF.Nocli = :new.Nocli
where
PREF.Nocli = :old.Nocli;
end if;
end;
/

-----
-- Table : CMD_FACT
-----

-- Trigger de suppression -----
create trigger TD_CMD_FACT
after delete on CMD_FACT for each row
declare numrows INTEGER;

```

```

begin

-- Supprimer les occurrences correspondantes de la table
LIGNE_CMD_FACT.

delete from LIGNE_CMD_FACT
where
LIGNE_CMD_FACT.Code_mag = :old.Code_mag and
LIGNE_CMD_FACT.No_cmd = :old.No_cmd;

end;
/

-- Trigger de modification -----
create trigger TU_CMD_FACT
after update on CMD_FACT for each row
declare numrows INTEGER;
begin

-- Interdire la modification de la clé étrangère référençant la
table
-- CLIENT.

-- Interdire la modification de la clé étrangère référençant la
table
-- CLIENT.

-- Interdire la modification de la clé étrangère référençant la
table
-- RELEVEMENTS.

-- Ne pas modifier la clé primaire de la table CMD_FACT s'il
existe des
-- occurrences correspondantes dans la table
LIGNE_CMD_FACT.

if
:old.Code_mag <> :new.Code_mag or
:old.No_cmd <> :new.No_cmd
then
select count(*) into numrows
from LIGNE_CMD_FACT
where
LIGNE_CMD_FACT.Code_mag = :old.Code_mag and
LIGNE_CMD_FACT.No_cmd = :old.No_cmd;
if (numrows > 0)
then
raise_application_error(
-20005,
'Impossible de modifier "CMD_FACT" car
"LIGNE_CMD_FACT" existe.);
end if;
end if;

end;
/

-- Trigger d'insertion -----
create trigger TI_CMD_FACT
after insert on CMD_FACT for each row
declare numrows INTEGER;
begin

-- Interdire la création d'une occurrence de CMD_FACT s'il
n'existe pas
-- d'occurrence correspondante dans la table CLIENT.

select count(*) into numrows
from CLIENT
where
:new.Nocli_passer = CLIENT.Nocli;
if
(
numrows = 0
)
then
raise_application_error(

```

46

Merise : Solutions du Cas Société X

```

-20002,
'Impossible d'ajouter "CMD_FACT" car "CLIENT" n'existe
pas.);
end if;
-- Interdire la création d'une occurrence de CMD_FACT s'il
n'existe pas
-- d'occurrence correspondante dans la table CLIENT.

select count(*) into numrows
from CLIENT
where
:new.Nocli_fact_tiers = CLIENT.Nocli;
if
(
:new.Nocli_fact_tiers is not null and
numrows = 0
)
then
raise_application_error(
-20002,
'Impossible d'ajouter "CMD_FACT" car "CLIENT" n'existe
pas.);
end if;
-- Interdire la création d'une occurrence de la table CMD_FACT
s'il existe
-- déjà des occurrences de la table CMD_FACT correspondant
à la même
-- table RELEVEMENTS.

-- Interdire la création d'une occurrence de CMD_FACT s'il
n'existe pas
-- d'occurrence correspondante dans la table RELEVEMENTS.

select count(*) into numrows
from RELEVEMENTS
where
:new.Nocli_Regrouper = RELEVEMENTS.Nocli and
:new.Mois_rel = RELEVEMENTS.Mois_rel;
if
(
:new.Nocli_Regrouper is not null and
:new.Mois_rel is not null and
numrows = 0
)
then
raise_application_error(
-20002,
'Impossible d'ajouter "CMD_FACT" car "RELEVEMENTS"
n'existe pas.);
end if;

end;
/

-----
-- Table : LIGNE_CMD_FACT
-----

-- Trigger de modification -----
create trigger TU_LIGNE_CMD_FACT
after update on LIGNE_CMD_FACT for each row
declare numrows INTEGER;
begin

-- Interdire la modification de la clé étrangère référençant la
table
-- ART.

-- Interdire la modification de la clé étrangère référençant la
table
-- CMD_FACT.

end;
/

```

```

-- Trigger d'insertion -----
create trigger TI_LIGNE_CMD_FACT
after insert on LIGNE_CMD_FACT for each row
declare numrows INTEGER;
begin

-- Interdire la création d'une occurrence de LIGNE_CMD_FACT
s'il n'existe pas
-- d'occurrence correspondante dans la table ART.

select count(*) into numrows
from ART
where
:new.Ref_art = ART.Ref_art;
if
(
numrows = 0
)
then
raise_application_error(
-20002,
'Impossible d'ajouter "LIGNE_CMD_FACT" car "ART"
n'existe pas. ');
end if;
-- Interdire la création d'une occurrence de LIGNE_CMD_FACT
s'il n'existe pas
-- d'occurrence correspondante dans la table CMD_FACT.

select count(*) into numrows
from CMD_FACT
where
:new.Code_mag = CMD_FACT.Code_mag and
:new.No_cmd = CMD_FACT.No_cmd;
if
(
numrows = 0
)
then
raise_application_error(
-20002,
'Impossible d'ajouter "LIGNE_CMD_FACT" car
"CMD_FACT" n'existe pas. ');
end if;

end;
/

```

```

select count(*) into numrows
from ART
where
:new.Ref_art = ART.Ref_art;
if
(
numrows = 0
)
then
raise_application_error(
-20002,
'Impossible d'ajouter "STOCK_MAG" car "ART" n'existe
pas. ');
end if;

end;
/

```

```

-----
-- Table : STOCK_MAG
-----

-- Trigger de modification -----
create trigger TU_STOCK_MAG
after update on STOCK_MAG for each row
declare numrows INTEGER;
begin

-- Interdire la modification de la clé étrangère référençant la
table
-- ART.

end;
/

-- Trigger d'insertion -----
create trigger TI_STOCK_MAG
after insert on STOCK_MAG for each row
declare numrows INTEGER;
begin

-- Interdire la création d'une occurrence de STOCK_MAG s'il
n'existe pas
-- d'occurrence correspondante dans la table ART.

```

## 8 - MODELE LOGIQUE DE TRAITEMENTS DU CAS X

### Modèle logique de traitements futur

Nous avons choisi, pour illustrer la modélisation logique des traitements, de détailler une phase du MOT correspondant à l'enchaînement des tâches Vérifier disponibilité puis Facturer de la solution 2 (figure 8.11).

Comme nous le reverrons dans le modèle logique de données, nous avons fait l'hypothèse d'une architecture matérielle suivante :

- dans chaque magasin, un serveur et des postes clients ,
- au siège, un serveur suffisamment dimensionné pour accueillir en outre l'ensemble des applications spécifiques au siège (achats, comptabilité, personnel) et des postes clients.
- le siège et les magasins sont reliés en réseau.

Conformément à la démarche de répartition, nous présentons d'abord un MLT non réparti, puis nous aborderons la problématique de sa répartition.

### Procédure logique non répartie

La procédure logique présente l'enchaînement des différentes ULT permettant le traitement des tâches composant la phase.

Dans la conception de cette procédure, nous avons supposé que les postes clients disposaient d'une interface graphique.

La figure 11.16 illustre la procédure logique Vente comptoir.

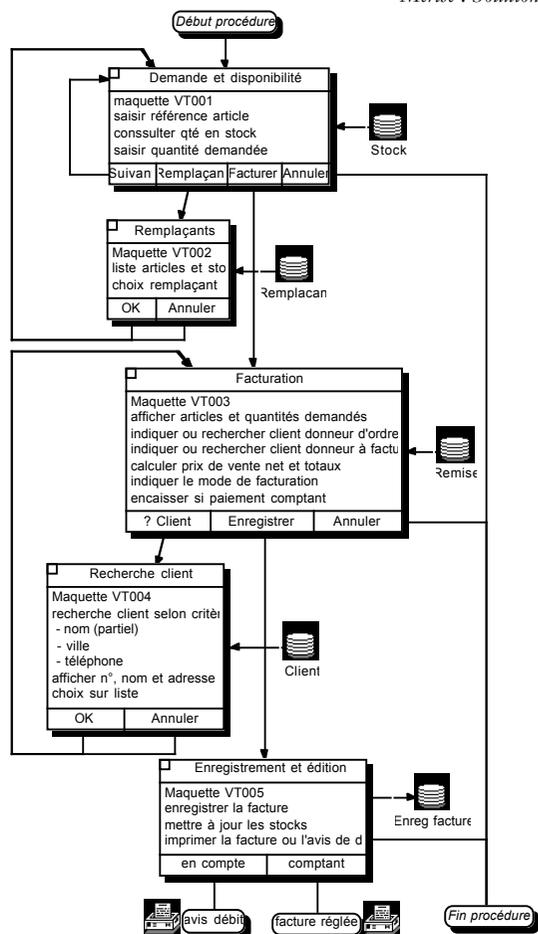


Figure 11.16 : Procédure logique Vente comptoir

Détail des ULT

La description détaillée de chaque unité logique de Traitement comporte les éléments suivants :

- la présentation sous la forme d'une maquette d'écran ou de boîte de dialogue,
- la logique de dialogue, c'est à dire les règles de gestion et les contrôles appliquer sur les données de la boîte de dialogue,
- la logique fonctionnelle, c'est à dire l'enchaînement général des traitements au sein de l'ULT,
- les règles de traitements, c'est à dire les algorithmes,

- les actions sur données et les sous-schémas de données associés,
- l'enchaînement vers les autres ULT et leur conditions de déclenchement.

Demande et disponibilité

- Présentation

Maquette d'écran VT001

- Logique de dialogue

Saisir la référence, ↵, afficher libellé et quantité en stock correspondant.

Si référence inconnue, afficher panneau d'erreur : *Article inconnu.*

Saisir la quantité demandée

- Logique fonctionnelle

Sans commentaire.

- Règle

Si quantité demandée > quantité en stock, afficher panneau d'erreur.

- Sous-schéma Stock



Lecture sur toutes les tables

- Enchaînement

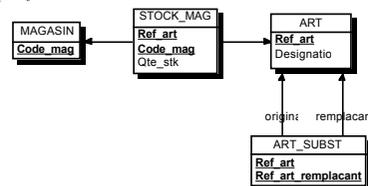
condition	action	résultat
Suivant	bouton	conserver la quantité demandée, effacer les zones pour ressaisie
Remplaçant	bouton	si l'article possède des remplaçants, appeler l'ULT Remplaçants avec passage de la référence article, sinon afficher message : <i>Pas de remplaçant</i>
Facturer	bouton	appeler l'ULT Facturation avec passage des références et quantités des articles demandés
Annuler	bouton	fin de procédure

### Remplaçants

- Présentation

Maquette d'écran VT002

- Logique de dialogue
  - Afficher référence, libellé et quantité en stock des articles remplaçants.
  - Choix sur liste.
- Logique fonctionnelle
  - Sans commentaire.
- Règle
  - Aucune
- Sous-schéma Remplaçant



Lecture sur toutes les tables

- Enchaînement

condition	action	résultat
OK	bouton	Appeler l'ULT Demande et disponibilité, avec transmission de la référence sélectionnée
Annuler	bouton	Appeler l'ULT Demande et disponibilité,

### Facturation

- Présentation

Maquette d'écran VT003

- Logique de dialogue
  - Saisir n° ou bouton ? pour le client livré, ↵, afficher nom et adresse
  - Saisir n° ou bouton ? pour le client facturé, ↵, afficher nom et adresse
  - Si inconnu, afficher le message : Client inconnu.
  - Saisir mode de paiement
- Logique fonctionnelle
  - Afficher les articles et les quantités demandées provenant de l'ULT précédente.
  - Chiffrer la facture (voir règles)

- Règle

#### Détermination du % remise

Le % remise appliqué est le plus favorable des deux taux suivants:

- % remise propre au client sur tous ses achats
- % remise appliqué sur certains articles par le dépôt

$$\% \text{ remise appliqué} = \text{Max} (\% \text{ remise client}, \% \text{ remise article})$$

Calcul montant total ligne

$$\text{total ligne} = \text{prix unitaire} * \text{quantité livrée} * (1 - \% \text{ remise appliquée})$$

Calcul montant total facture

$$\text{total facture HT} = \sum \text{lignes} * (\text{total ligne})$$

- Sous-schéma Remise



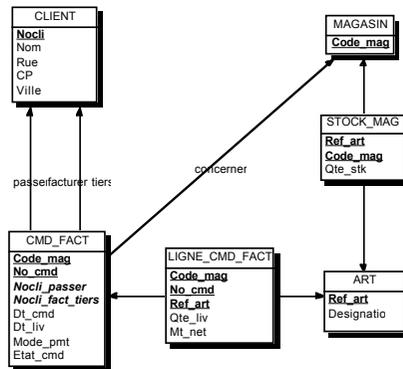
Lecture sur toutes les tables

- Enchaînement

condition	action	résultat
? Client	bouton	Appeler l'ULT Recherche client



• Sous-schéma Enreg facture



Création CMD\_FACT, LIGNE\_CMD\_FACT

Modifier STOCK\_MAG

Lecture sur les autres tables

• Enchaînement

condition	action	résultat
OK	bouton	Appeler l'ULT Facturation, avec transmission du client sélectionné
Annuler	bouton	Appeler l'ULT Facturation,

**Répartition logique des traitements**

Pour ce qui concerne la procédure logique étudiée, la répartition ne porte éventuellement que sur les données. En effet, l'architecture retenue (serveur de données classique) et la simplicité des traitements mis en oeuvre ne nécessite pas une répartition des traitements entre le poste client et le serveur.

Pour ce qui concerne les données, seule la table STOCK\_MAG est partagée entre les deux sites (voir au chap. 12 le MLD réparti). La mise à jour de cette table s'effectue à partir de l'ULT Enregistrement et édition avec une mise à jour simultanée des deux bases (serveur Magasin et serveur Siège) par la technique de "Two phases commit".

En conséquence, pour la procédure étudiée, il n'est pas nécessaire de formaliser un MLT réparti.