

CHAPITRE I

LE MODELE CONCEPTUEL DES DONNEES : M.C.D

I/ INTRODUCTION

Le niveau conceptuel apporte des réponses aux questions suivantes : Quelles actions entreprendre et avec quelles données ? L'analyse conceptuelle des données a pour objet de recenser toutes les informations du champ de l'étude et de décrire les liens entre ces informations. Elle aboutit à la mise en place d'un modèle conceptuel des données (M.C.D). Le M.C.D a pour objectif :

- établir le DD (dictionnaire des données)
- donner la liste des entités et éventuellement leurs propriétés
- établir les relations entre les entités en précisant leurs propriétés
- construire le graphe de dépendance fonctionnelle (ou la structure d'accès théorique, SAT) ou la matrice de dépendance fonctionnelle
- construire le schéma du M.C.D.

II/ DEFINITION ET FORMALISME DU M.C.D

1°- ENTITE, RELATION ET PROPRIETES

a- Entité : une entité ou individu (objet) est la représentation dans le système d'information d'un objet matériel ou immatériel dans l'univers de discours :

Exemple :

dans l'UDD, ITES, les entités classes, élèves, professeurs, matières etc...

b- Relation : association entre les entités (objets) dans l'UDD

Exemple :

« Jean Paul fréquente à l'ITES » exprime la relation « fréquente à » une association entre les entités Jean Paul et ITES.

C- Une propriété :

c'est une rubrique attribut d'une entité ou d'une relation. Les propriétés sont utilisées pour décrire les entités et les relations.

Exemple : Nom de l'Etudiant

Prénom de l'Etudiant $\Rightarrow$ nom et prénom sont des propriétés

3°- TYPES ET OCCURRENCES

a- Un type :

est un ensemble d'éléments ayant les mêmes caractéristiques.

b- Une concurrence d'un type :

est un élément particulier appartenant à cet ensemble.

c- Entité - type :

c'est une classe d'entités particulières ayant des propriétés analogues.

d- Une occurrence d'un type :

est un élément particulier appartenant à cet ensemble.

Exemple : Etudiant est une entité - type

L'Etudiant Koffi et l'étudiant Konan sont des occurrences de cette entité type.

N.B : L'identifiant (clé d'identification) permet de distinguer une entité d'une autre. Il caractérise chaque occurrence de l'entité - type.

e- Relation - type :

est une relation (association) entre plusieurs entités - types.

La collection est la liste des entités - types participant à cette relation.

Exemple : « fréquent à » est une relation - type définie sur la collection des entités - types. Etudiant et Ecole.

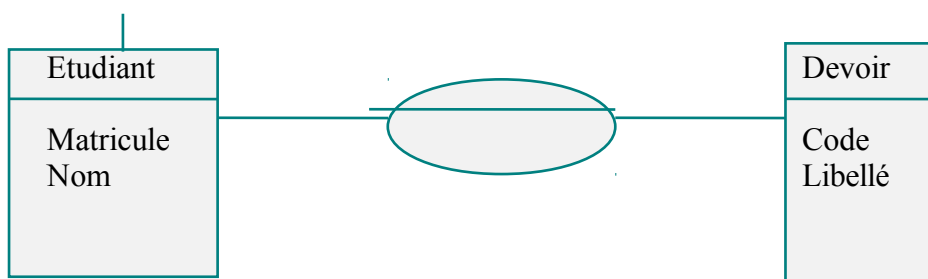
f- Propriété - type : est une classe de propriétés semblables. Une occurrence d'une propriété - type est une valeur prise par cette propriété.

Exemple : Nom - Etudiant est une propriété - type de l'entité - type étudiant et représente la classe des noms de tous les étudiants.

Une propriété - type peut être de type code, libellé ou montant. Elle peut être élémentaire ou concaténée, mémorisée ou calculée. Elle est caractérisée par une certaine structure :

- Sa classe
 - * numérique (chiffres)
 - * alphabétique (lettres ou espaces)
 - * alphanumériques (tous caractères)
- Sa longueur
 - * nombre de caractères

3°- REPRESENTATION SCHEMATIQUE



4°- CARACTERISTIQUE D'UNE RELATION

a- Collection :

la collection d'une relation - type est la liste des entités - types sur laquelle la relation est définie.

Exemple : la relation « effectue » est définie sur la collection : {étudiant, devoir}

b- Dimension :

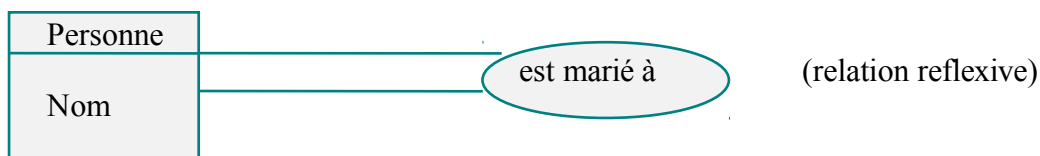
la dimension d'une relation - type est le nombre d'occurrence d'entités concernés par une occurrence de la relation type.

Elle est supérieure ou égale au nombre d'entités de la collection.

Exemple :



Collection : {étudiant, filière}
dimension : 2



Collection : personne (entité - type)
dimension 2 (2>1)

(relation reflexive)

Pour qu'il est mariage, il faut 2 personnes donc il faut 2 occurrences de la relation personne pour une occurrence de la relation est marié à

N.B :

- * une relation de dimension 2 est une relation binaire
- * une relation de dimension 3 est une relation ternaire
- * une relation de dimension n est dite n-aire.

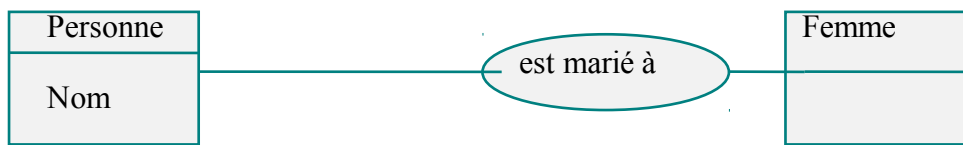
c- Fonctionnalité

Elle est définie par rapport à 2 entités - types X et Y. On distingue les relations :

* 1 à 1 (1 - 1)

A toute occurrence de X ne correspond qu'une seule occurrence de Y et réciproquement.

Exemple :

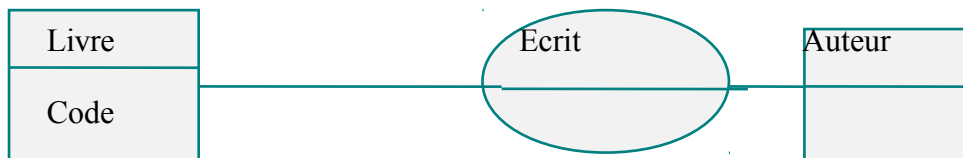


un homme n'est marié qu'à une femme et une femme n'est mariée qu'à un homme.

* 1 à Plusieurs (1-N)

A toute occurrence de X correspond une ou plusieurs occurrences de Y et à toute occurrence de Y une seule de X.

Exemple



Un livre a été écrit par un seul auteurs, mais un auteur a écrit un ou plusieurs livres.

* Plusieurs a plusieurs (m, n)

A toute occurrence de X correspond une ou plusieurs occurrences de Y et réciproquement.

Exemple :



un client peut commander plusieurs produits et chaque produit peut être commandé par plusieurs clients.

d- Totalité / Partialité

Une relation est dite totale si aucune occurrence de X et aucune occurrence de Y ne peuvent exister sans participer à une occurrence de la relation.

Une relation est dite partielle si certaines occurrence de X ou certaines occurrences de Y peuvent n'être impliquées dans aucune occurrence de la relation.

5°- CARDINALITES :

La cardinalité (minimum / maximum) permet d'exprimer la fonctionnalité et la totalité / partialité d'une relation

- Cardinalité minimum :

c'est le nombre minimum de fois où chaque occurrence d'une entité - type participe à la relation type

La cardinalité minimum correspond à une relation partielle

La cardinalité minimum **1** signifie qu'une occurrence d'entité type ne peut exister sans participer à une occurrence de la relation.

La cardinalité minimum **n** implique que toute occurrence d'entité type participe obligatoirement à n occurrences de la relation de la relation.

Les cardinalités minimum non nulles correspondent à des relations totales.

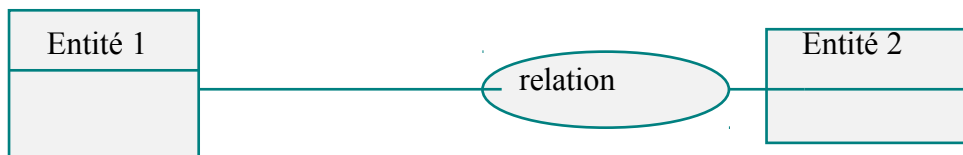
- Cardinalité maximum

C'est le nombre maximum de fois où chaque occurrence d'entité -type peut participer à une occurrence de la relation - type

La cardinalité maximum **1** signifie que toute occurrence de l'entité - type ne peut participer qu'à une occurrence de la relation, au plus.

La cardinalité maximum **n** signifie qu'une occurrence de l'entité type peut être impliquée dans un maximum n occurrences de la relation.

Représentation graphique



6) LES REGLE DE GESTION

Les règles de gestion du M..C.D précisent les contraintes qui doivent être respectées par le modèle.

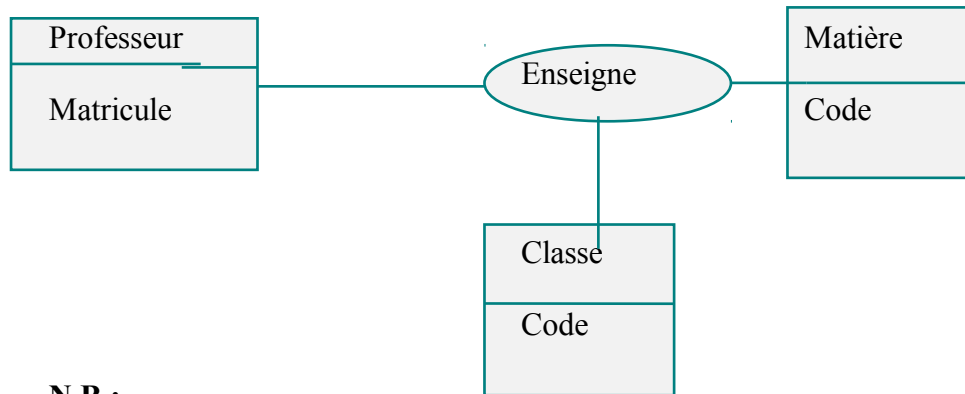
Exemple : Dans une école, on peut avoir les règles suivantes :

RG₁ : tout professeur enseigne au moins une matière, mais certains d'entre eux peuvent être dispensés d'enseignement.

RG₂ : toute matière est enseignée dans au moins une classe

RG₃ : toute classe a au moins 3 enseignements.

Le M..C. D devient



N.B. :

Les règles de gestion expriment les contraintes d'intégrité du modèle. Elles représentent les lois de l'univers réel modélisé dans le S.I.

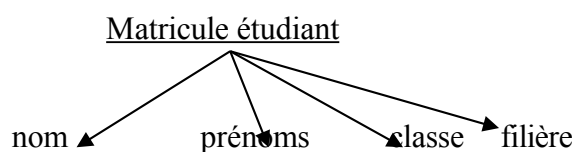
III/ DEPENDANCES FONCTIONNELLES

1°) DEPENDANCES FONCTIONNELLES ENTRE PROPRIETES

Définition : une propriété dépend fonctionnellement d'une propriété A ce qui se note : $A \rightarrow B$ (ou $A \xrightarrow{\text{df}} B$), lorsque la connaissance de A entraîne celle de B. (à partir de A on peut déterminer B).

Exemple : Matricule étudiant $\rightarrow$ Nom
Code client $\rightarrow$ adresse

La réciproque est fausse. Connaissant l'adresse, on ne peut déterminer le code d'un client.



a) Dépendance fonctionnelle élémentaire

$A \rightarrow B$ est une dépendance fonctionnelle élémentaire si et seulement si :

A n'est pas un sous-ensemble de B et si
il n'existe pas une partie A' de A telle que $A' \rightarrow B$

Exemples :

1°) Code produit alimentaire $\rightarrow$ Code produit n'est pas une dépendance fonctionnelle élémentaire car l'ensemble des produits alimentaires est sous - ensemble des produits.

2°) Code client plus non client $\rightarrow$ adresse client n'est pas élémentaire puisque la connaissance de code client (partie de code client + nom client) suffit à déterminer l'adresse.

Code client $\rightarrow$ adresse client est élémentaire.

b°) Dépendance fonctionnelle élémentaire directe

On dit qu'il y a une dépendance fonctionnelle élémentaire directe et on note

$A \rightarrow B$ si cette DF est élémentaire et s'il n'existe pas de propriété C telle que :

$A \rightarrow C$ et $C \rightarrow B$ en d'autres termes, les transitivités n'existent pas.

Exemple :

Matricule professeur $\rightarrow$ code matière

Code matière $\rightarrow$ nom matière

Matricule professeur $\rightarrow$ nom matière

Les 2 premières dépendances fonctionnelles sont directes, mais la 3^{ème} ne l'est pas compte tenue de la transitivité.

Matricule professeur $\rightarrow$ code matière $\rightarrow$ nom matière

N.B

clé (d'identification) d'une entité : c'est la propriété (ou une concaténation de propriétés) de cette entité telle que les autres propriétés de l'entité dépendent d'elle fonctionnellement et telle que ceci ne soit plus vrai pour aucune de ses propriétés.

Exemple :

Code client plus nom n'est pas une clé bien que code client plus nom $\Rightarrow$ adresse car ceci reste vrai pour la partie code client de code client plus nom puisque code client détermine adresse.

En revanche : code client est une clé car :

code client $\rightarrow$ nom

code client $\Rightarrow$ adresse

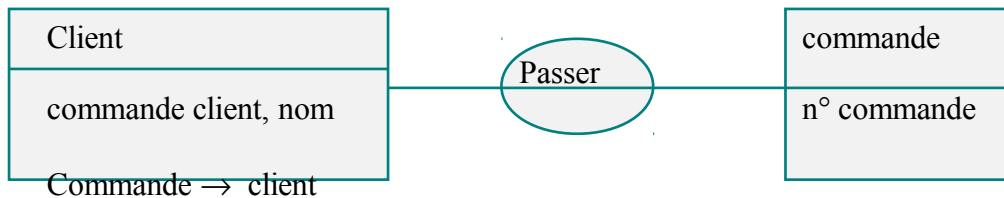
En entité peut avoir plusieurs clés. Mais on choisira la clé qui sera l'identifiant.

Les dépendances fonctionnelles entre propriétés sont à considérer par rapport aux entités et relations.

2°) DEPENDANCES FONCTIONNELLES ENTRE ENTITES :

on dit qu'il existe une DF entre 2 entités A et B et on note $A \rightarrow B$ si toute occurrence de A détermine une et une seule occurrence de B.

Exemple :



Les cardinalités 1,1 de commande dans cette relation expriment que toute commande détermine un et un seul client ; Il s'agit d'un client ayant passé commande.

N.B : La cardinalité maximum 1 correspond toujours à une D.F. Les dépendances fonctionnelles entre entités s'assimilent aux dépendances fonctionnelles entre les identifiants de ces entités.

Exemple : Comme $\rightarrow$ client

est assimilable à $n^\circ \text{ commande} \Rightarrow \text{code client}$

3°) PROPRIETES DES DEPENDANCES FONCTIONNELLES (D.F)

Soient A,B,C,D les propriétés définies dans notre D.D

- La reflexivité : $A \rightarrow A$

- Augmentation

si $A \rightarrow B \Leftrightarrow AC \rightarrow BC$

- La transitivité

si $A \rightarrow B$ et $B \rightarrow C \Rightarrow A \rightarrow C$

- L'union (ou additivité)

si $A \rightarrow B$ et $A \rightarrow C \Rightarrow A \rightarrow BC$ (ou $A \rightarrow B + C$)

- Projection

$A \rightarrow BC (B + C) \Rightarrow A \rightarrow B$ et $A \rightarrow C$

- La décomposition

Si $A \rightarrow B$ et $C < B$ (C sous ensemble de B) $\rightarrow A \rightarrow C$

Exemple : adresse (n° rue, nom rue, ville, code postal, pays)

Code fournisseur → adresse ⇒ code fournisseur → pays fournisseur

IV/ LES CONTRAINTES D'INTEGRITE

Les contraintes d'intégrité représentent les lois de l'univers modélisé dans le S.I Existe 3 contraintes d'intégrité qui sont :

la contrainte d'intégrité fonctionnelle
la contrainte d'intégrité syntaxique
la contrainte d'intégrité sémantique

a- Contrainte d'intégrité fonctionnelle et de stabilité

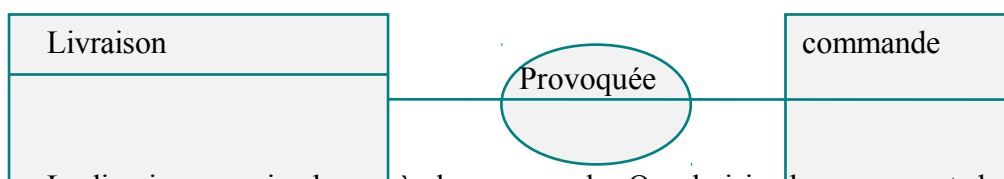
La notion de contrainte d'intégrité fonctionnelle (CIF) correspond à celle de DF forte et stable (stable signifie que les occurrences des entités mises en jeu ne pourront jamais changer).

Exemple :



Remarque :

DF et C.I.F des relations 1 à 1. Lorsqu'il y a une relation fonctionnelle (DF ou CIF) entre 2 entités avec la cardinalité maximum 1 et des 2 côtés, il y a à priori, dépendance fonctionnelle dans les 2 sens. Il faut faire intervenir le temps. La contrainte sera prise dans le sens allant de l'occurrence la plus récente l'occurrence la plus ancienne.



La livraison surviendra après la commande. On choisira le sens avant de l'occurrences récente à l'occurrence ancienne (card minimum) c'est à dire DF forte ou la CIF.

b) Contraintes d'intégrité syntaxiques :

Elles portent sur une propriété et peuvent concerner soit la forme, soit le domaine des valeurs de la propriété, son type (numérique, alphanumérique, alphabétique), la longueur.

Exemple :

Le prix d'un article doit être un nombre strictement positif.

La date doit prendre la forme : JJ/MM/AA JJ/NN/AN NA

c- Contraintes d'intégrité sémantique

Elles s'appliquent à plusieurs rubrique d'une entité type ou d'une association.

Exemple : soit l'entité « vol » ayant comme propriétés. Vol (numéro vol, date du vol, heure départ, heure d'arrivée)

La contrainte ''heure d'arrivée'' > heure départ s'applique à 2 propriétés de l'entité vol

La C.I sémantique permet de contrôler la cohérence entre les différentes données du SI.

Ces différentes contraintes d'intégrité peuvent être : statiques ou dynamiques

* Les contraintes statiques

Généralement elles portent :

Sur une propriété (forme, liste de valeurs possibles, fourchette de valeurs admissibles).
Sur diverses propriétés d'une même relation ou entité.

Exemple :

commande (n° commande, date commande, date livraison)
on doit avoir date - commande < date livraison. (contrainte d'intégrité sémantique et statique).

Sur des propriétés d'occurrences distinctes d'une relation ou entité.

Exemple :

Ligne Ecriture (n° écriture, libellé, montant, sens). La somme des montants des lignes de sens ''D'' (débit) doit être égale à celle des lignes de sens « C ».

Sur des propriétés d'entités relations différentes.

Exemple :

La somme des CA (compte achat) des produits doit être égale à celle des CA des clients.

* Contraintes dynamiques

Les C.I dynamiques expriment les règles d'évolution et portent directement sur le passage du SI d'un état dans un autre.

Exemple : le salaire d'un employé ne doit pas diminuer.

V/ REGLES RELATIVES AU M.C.D

1/ NORMALISATION DES ENTITES

a° 1^{ère} forme normale (1FN)

Dans une entité, toutes les propriétés sont élémentaires et il y a au moins une clé caractérisant chaque occurrence de l'entité représentée.

Exemple :

Etudiant
Nom Adresse

Cette entité n'est pas en 1FN, car il n'y a pas de clé (plusieurs étudiants peuvent avoir le même nom) ensuite adresse étudiant est une propriété concaténée composée de rue, ville etc.

b° Deuxième forme normale (2FN)

Toute propriété d'une entité doit dépendre de la clé par une dépendance fonctionnelle élémentaire. En d'autres termes toute propriété de l'entité doit dépendre de tout l'identifiant.

Etudiant
Matricule Nom Rue Ville

est en 2FN car nom, rue, ville dépendent de la clé matricule par une DF élémentaire

c° Troisième forme normale (3 FN)

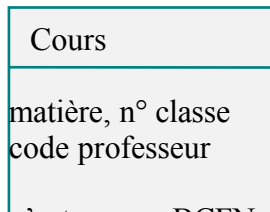
Dans une entité toute propriété doit dépendre de la clé par une dépendance fonctionnelle élémentaire directe.

Exemple :

Client
code client code catégorie, non catégorie

d° Forme normale de BOYCE - CODD (BCFN)

Si une entité a un identifiant concaténé, un des éléments composant cet identifiant ne doit pas dépendre d'une autre propriété.



n'est pas en BCFN car matière (partie de l'identifiant concaténé matière + n° classe) dépend de la propriété code- prof

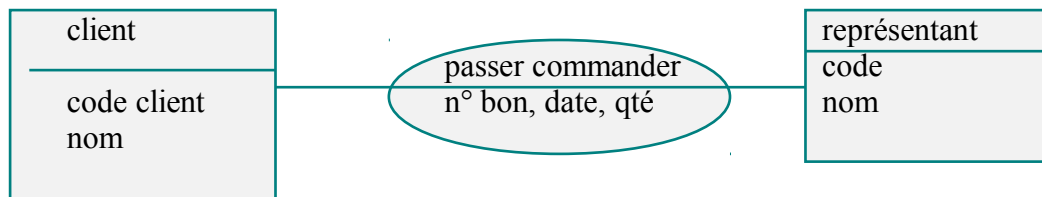
e° Respect des contraintes d'intégrité

Le M.C.D doit respecter les RG qui expriment ces C.I

3) VERIFICATION

Dans toute occurrence d'entité ou de relation type, il ne doit y avoir qu'une seule valeur de chaque propriété (non répétitivité, pour les entités, cette règle résulte de la 1 FN. Elle doit rester vraie pour les relation.

Exemple :



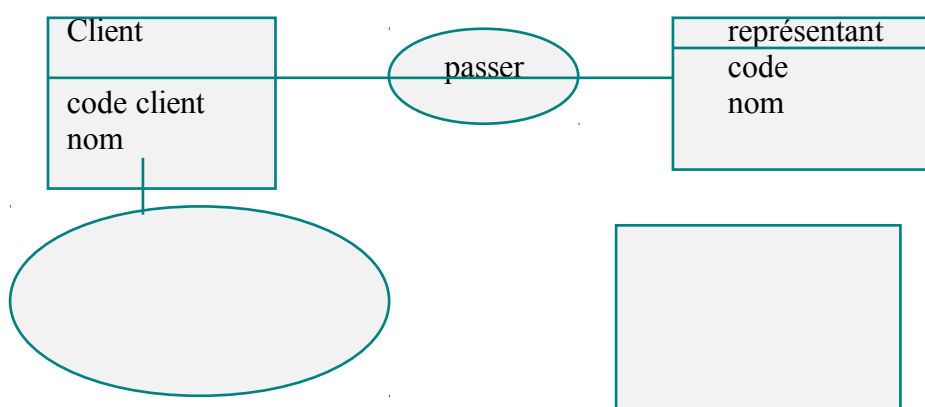
La relation pour passer commande n'est pas vérifiée, car il peut y avoir plusieurs valeurs de la quantité dans une commande passée par un client à un représentant. La quantité ne dépend pas seulement du client et du représentant mais aussi du produit commandé.

En d'autres termes :

Dans une relation, les propriétés doivent dépendre fonctionnellement des identifiants des entités concernées par la relation. La concaténation de ces identifiants constitue l'identifiant de la relation.

Améliorons le M.C.D

Première étape

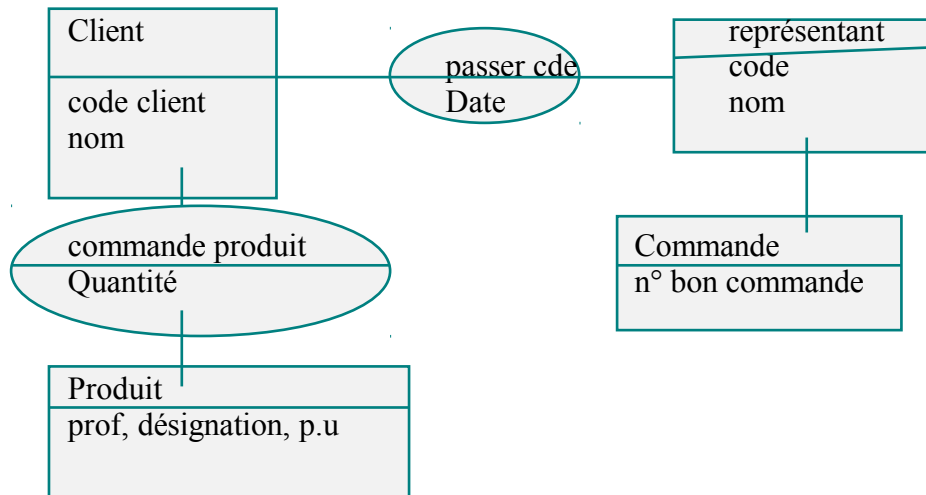


commande produit	produit
produit	réf , désignation p.u

Dans la relation commander produit, la quantité ne dépend pas seulement du client et du produit mais aussi du n° de bon de commande (un client peut passer plusieurs commandes du même produit).

Le numéro de commande n'est pas connu connaissant client, produit et représentant car il peut y avoir plusieurs bons de commande pour un client donné. La règle de vérification n'est donc pas respectée.

Deuxième étape (il faut créer l'entité commande d'où le M.C.D)



4) NORMALISATION DES RELATIONS

Chaque propriété de la relation doit dépendre fonctionnellement de l'ensemble des identifiants des entités qui participent à la relation, mais d'aucun sous - ensemble de cet ensemble.

Il doit y avoir une dépendance pleine des propriétés de la relation par rapport aux entités.

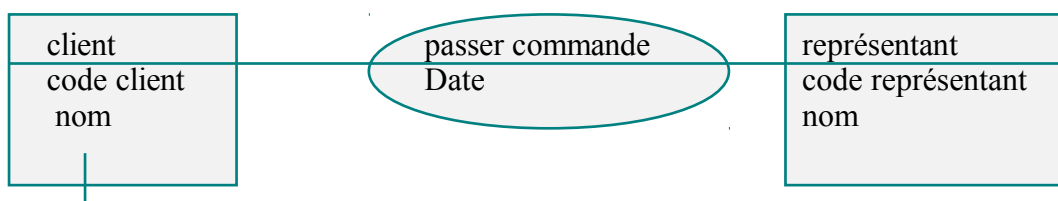
Exemple :

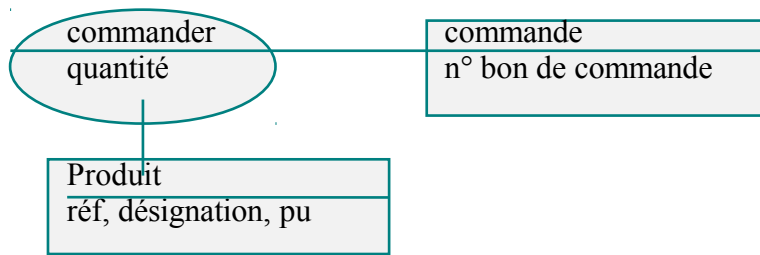
Dans le M.C.D précédent :

n° bon de commande → date car la date de la commande peut être connue si on connaît le n° de bon de commande. La propriété date dépend du n° de bon de commande et donc d'un sous ensemble et il n'y a pas de dépendance pleine par rapport à l'ensemble des entités client représentant et commande qui participent à la relation passer commande.

La date doit migrer dans l'entité commande en plus la propriété quantité doit être affectée à une relation.

Le composé de qui ne met en jeu que les entités commande et produit d'où le M.C.D.





5° DECOMPOSITION D'UNE RELATION

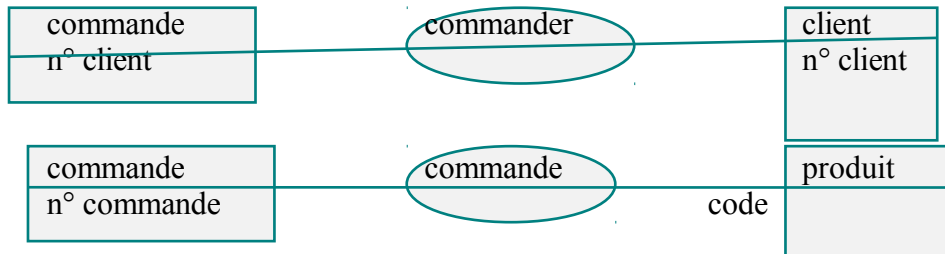
La décomposition consiste à remplacer une relation de dimension n en plusieurs relations de dimensions plus petites en utilisant les DF que l'on peut détecter sur la relation.

Exemple :

Dans la relation commander produit du M.C.D précédent, on a la D.F : commande $\rightarrow$ client.

En effet, une commande est passée par un et un seul client.

On peut donc éclater cette relation en 2 relations :



N.B : La décomposition n'est possible qu'à 2 conditions :

1. la cardinalité minimum des entités à gauche dans la DF doit être dans la relation à décomposer (relation totale pour les entités).
2. si la DF provient d'une autre relation que la relation à décomposer, il faut qu'elle concerne les mêmes occurrences d'entités que la relation à décomposer.

VI APPLICATION : CONSTRUCTION DU M.C.D

Soit la facture suivante :

N° bon		Date		
Nom du client				
Adresse				
Nom du représentant				
Réf.	Désignation	Quantité	Prix Unitaire	Montant
.....				
				Total

et les règles de gestion du SI.

RG₁ : Un client peut passer une ou plusieurs commandes ou aucune commande.

RG₂ : Une commande peut concerner un ou plusieurs produits.

RG₃ : Une commande est passée à un représentant qui n'est pas toujours le même pour un client donné.

LE MODELE CONCEPTUEL DES TRAITEMENTS **(M.C.D)**

I/ OBJECTIFS

Etre capable d'énumérer et d'expliquer les éléments fondamentaux du M.C.T à savoir :

- la notion d'événement
- la notion d'acteur
- la synchronisation des événements
- la notion d'opération
- la notion de résultat (d'opération)
- la notion de règle de détermination de résultat (règle de gestion pour les traitements
- le formalisme du M.C.T

Etre capable à partir d'un texte donné décrivant une activité dans une entreprise d'établir :

- la liste des acteurs
- la liste des événements
- le graphe des flux (diagramme des flux, graphe de visualisation des infos
- la matrice des flux
- le graphe d'ordonnancement des événements
- le graphe du M.C.T pour chaque processus identifié

N.B : il peut être demandé la construction du graphe d'ordonnancement des flux.

II/ APPROCHE INTUITIVE

Exemple : un établissement scolaire (ITES).

Un établissement scolaire a pour objectif principal la formation des élèves ou étudiants. Pour atteindre celui-ci, il est nécessaire voire primordial de dérouler un certain nombre d'actions.

Avant le début des cours

- * définir les filières
- * préparer les programmes
- * prendre contact avec les profs
- * établir les emplois du temps
- * inscrire les étudiants
- * repartir dans les classes (définir l'occupation des locaux)

Pendant l'année scolaire

- * dispenser les cours
- * définir la période d'évaluation
- * faire les évaluations
- * relever les notes
- * établir les bulletins et les distribuer

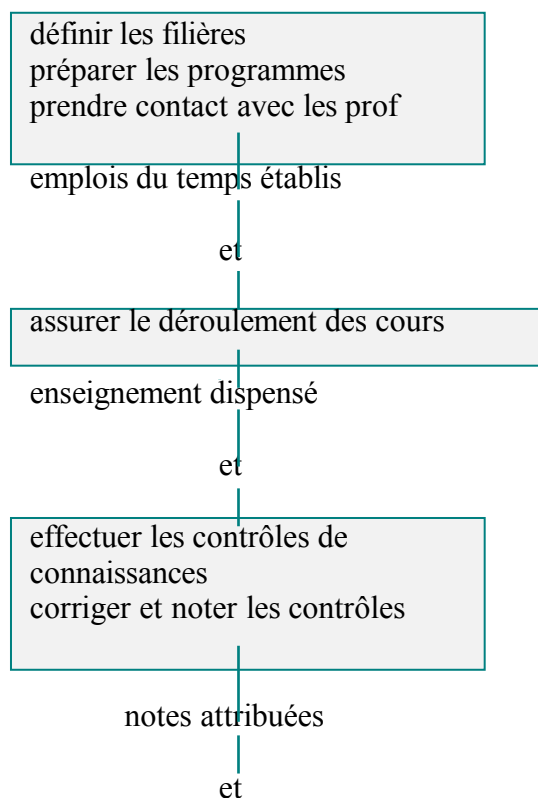
A la fin de l'année scolaire

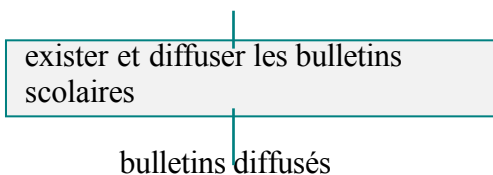
- * calculer les moyennes générales
- * établir les derniers bulletins
- * tenir le conseil des classes
- * faire le bilan de l'année

Cette liste qui vient d'être établie essaie de répondre à la question suivante : Qui fait quoi ? C'est à dire quels sont les traitements réalisés dans notre S.I. ?

Nous pouvons schématiser de la façon suivante :

Pré rentrée





III/ CONCEPTS FONDAMENTAUX

1) EVENEMENT

Un événement est un fait dont la venue à pour effet de déclencher l'exécution d'une ou plusieurs actions. En d'autres termes un événement est une situation ou un fait qui lorsqu'il se produit participe au déclenchement d'une opération.

Exemple : la fin du mois (événement temporel) déclenche dans l'entreprise le paiement des salariés

a- Evénement externes

Ils proviennent de l'environnement extérieur au S.I et provoquent donc le déclenchement d'une opération.

Un tel événement n'apparaîtra donc jamais à la fois comme déclencheur d'une opération et résultat d'une autre.

Exemple : les événements « commande reçues », « échéance atteinte » ou « départ en congé d'un employé » seront en général des déclencheurs externes.

b- Evénements internes

Un événement interne est un événement qui est à la fois résultat d'opérations et déclencheur d'opérations. Ce type d'événement apparaît en général sur le schéma parce qu'il est synchronisé à un événement. L'événement interne matérialise sur le papier l'enchaînement des opérations.

2) SYNCHRONISATION

C'est une condition booléenne, traduisant les règles de gestion que doivent vérifier les événements pour déclencher des actions. Elle exprime les conditions requises (pré - conditions) pour déclencher une opération.

Exemple : (stock en rupture) et (demande à satisfaire)

3) OPERATION

Ensemble d'actions dont l'enchaînement ininterrompible n'est conditionné par l'attente d'aucun événement autre que le déclencheur initial.

Exemple :

l'opération 'préparation d'une commande' regroupe les actions ininterruptibles suivantes :

- détermination des produits et des quantités commander
- choix du fournisseur
- rédaction d'un brouillon de commande

4) REGLE D'EMISSION

Condition, traduisant les règles de gestion, à laquelle est soumise l'émission des résultats d'une opération.

Exemple : si c'est conforme alors

5) RESULTAT

Produit de l'exécution d'une opération. Le résultat fait réel de même nature que l'événement, pourra être le déclencheur d'une autre opération.

Exemple : commande transmise
bulletins de paye édités

IV/ LE FORMALISME

Comme pour les données, il est souhaitable de définir un formalisme à la fois clair et rigoureux.

- Evénement :

Les événements sont identifiés par un nom et symbolisés par un ovale, relié à l'opération qu'ils déclenchent ou dont ils résultent.

Exemple :

rupture de stock

- Opération :

On ne fera figurer sur le schéma que le nom de l'opération de façon à ne pas le surcharger. Ce nom identifiera l'opération et sera encadré d'un rectangle.

Exemple :

préparation
commande

- Synchronisation :

La proposition logique de déclenchement sur les événements est décrite à l'intérieur d'un synchronisateur liée à l'opération.

On utilisera les symboles : $\wedge$ pour et

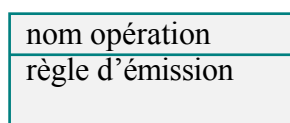
$\vee$ pour ou

et si le nombre d'événements est important, des lettres symboliques pour les représenter.

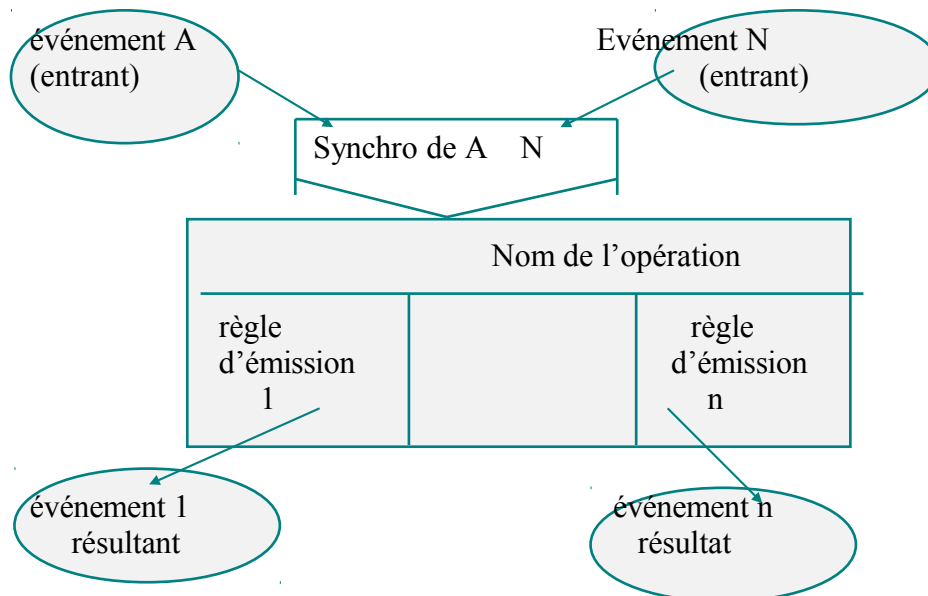


- Règle d'émission :

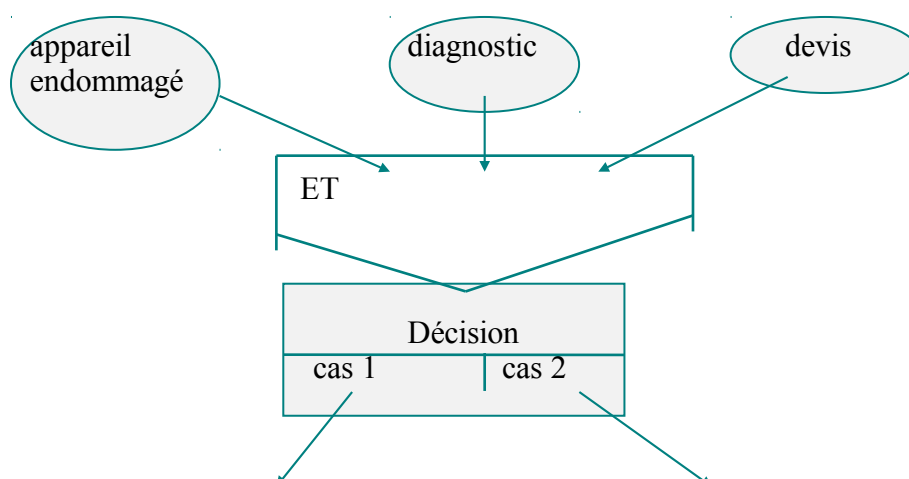
La condition d'émission du résultat pourra être précisée dans le rectangle de l'opération. On trouvera essentiellement des messages tels que : ok, ok, si, sinon



Le formalisme complet est donc le suivant :



Exemple



ordre de
réparation

ordre de
destruction

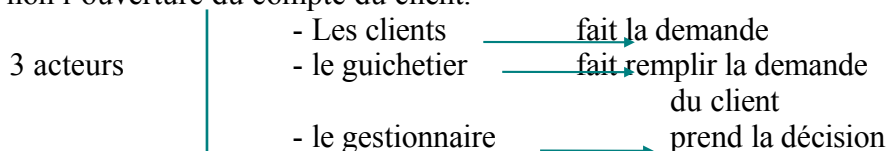
V/ CONCEPTS COMPLEMENTAIRES

1) ACTEUR :

Un acteur est terme générique qui représente tout élément impliqué dans un processus. Il peut s'agir d'activités, de services, de personnel etc...

Exemple : opération bancaire

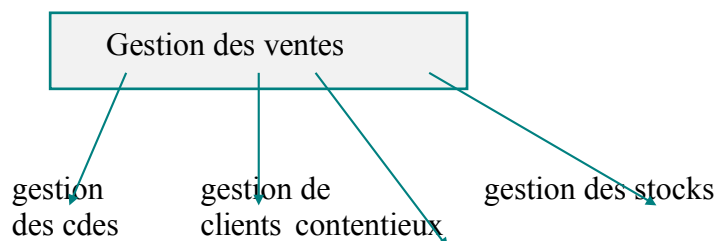
Les clients d'une banque qui veulent ouvrir un compte se rendent à un des guichets où ils remplissent un formulaire. Ce formulaire est transmis à un gestionnaire qui va prendre la décision d'accorder ou non l'ouverture du compte du client.



Les acteurs peuvent être internes à l'entreprise (se sont les agents de celle-ci) où ils peuvent être externes à l'entreprise (partenaires de l'entreprise).

2) LE DOMAINE

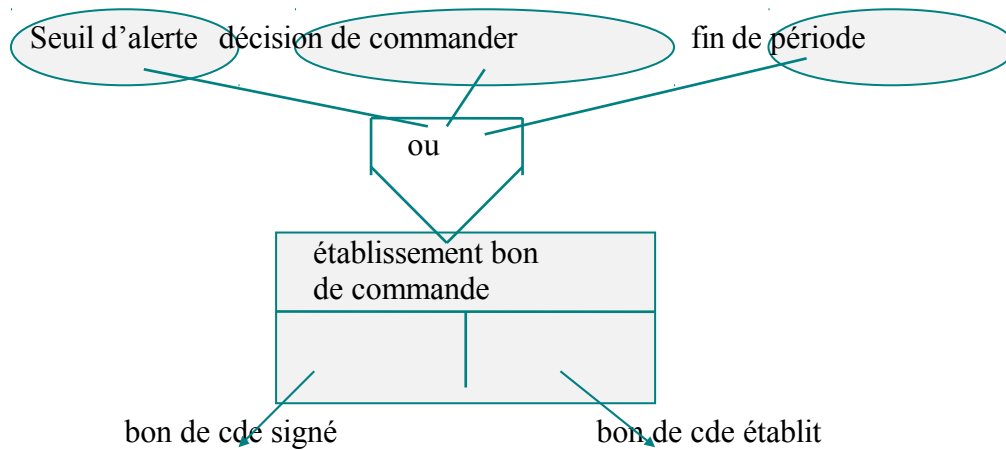
Englobe les activités ou les services concernés par une étude.



Dans le cadre de l'étude sur le M.C.T, nous nous intéressons soit à un domaine d'activité de l'entreprise soit à plusieurs domaines d'activités.

3) UN PROCESSUS

Un processus correspond à l'enchaînement des principales opérations concourant à une finalité et déclenchées par des événements (externe ou interne) dans un même domaine d'activité.



VI/ APPLICATIONS

Exemple :

Pour construire le M.C.T il convient :

- d'établir la liste des acteurs
- d'établir la liste des événements
- d'établir le graphe des flux et/ou la matrice des flux
- d'établir le graphe d'ordonnancement des événements
- de donner le schéma du M.C.T

* Représentation de la matrice des flux

Acteurs	A1	A2	A3	A4
acteurs				
A1				
A2				
A3				
A4				

* Le graphe d'ordonnancement des événements

Le graphe des flux met en évidence tous les acteurs et es échanges d'événements entre ces différents acteurs. On peut dans la mesure du possible passer du graphe des flux au graphe du M.C.T mais sans l'ordre d'enchaînement des événements et leurs synchronisation, ce passage n'est pas toujours aisé.

Ainsi, il est intéressant de construire une représentation qui ordonne les événements internes tout en les associant là où il faut aux événements externes ou événements temporels.

Le GOE se présente de la façon segments verticaux

Les segments sont les événements verticaux représentant les opérations.

LE M.C.T

VII/ APPLICATIONS

Exemple 1 : traitement des commandes clients

1°) SITUATION ACTUELLE DU SYSTEME

Les commandes des clients jugés non solvables sont refusées (par le service commercial).

Les commandes acceptées sont confrontées (au magasin) à l'état du stock pour déterminer quels sont les manquants et quelles sont les commandes disponibles.

En cas de manquant, on (le service achats) devra prendre toutes dispositions pour réapprovisionner le stock si ce n'est pas encore fait.

Dès la livraison du fournisseur, les commandes devenues disponibles subissent le même traitement que celles qui l'étaient dès le départ.

Les commandes disponibles donnent lieu à la confection de bons de livraisons destinés aux clients.

A la livraison, ceux-ci peuvent refuser la marchandise, auquel cas il y a retour de marchandises.

Si le client accepte la livraison, la comptabilité émet une facture qui ne sera soldée qu'après complet règlement les clients qui n'ont pas réglé à l'échéance doivent recevoir une relance. Les factures soldées sont archivées.

A partir de la situation actuelle, nous dégageons les règles de gestion suivantes (événements) .

RG₁ : toute commande de client non solvable est refusée

RG₂ : les commandes non disponibles sont mises en attente et devront déclencher un réapprovisionnement par le fournisseur.

RG₃ : les commandes en attente seront déclarées disponibles lorsque le réapprovisionnement sera suffisant

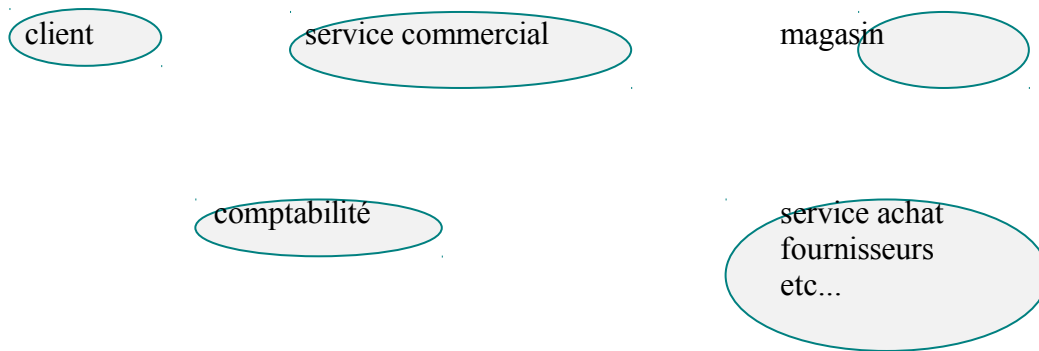
RG₄ : les commandes disponibles donnent lieu à livraison au client

RG₅ : les livraisons refusées par le client donnent lieu à retour de marchandises

RG₆ : les livraisons acceptées donnent lieu à des factures qui sont conservées jusqu'à complet règlement

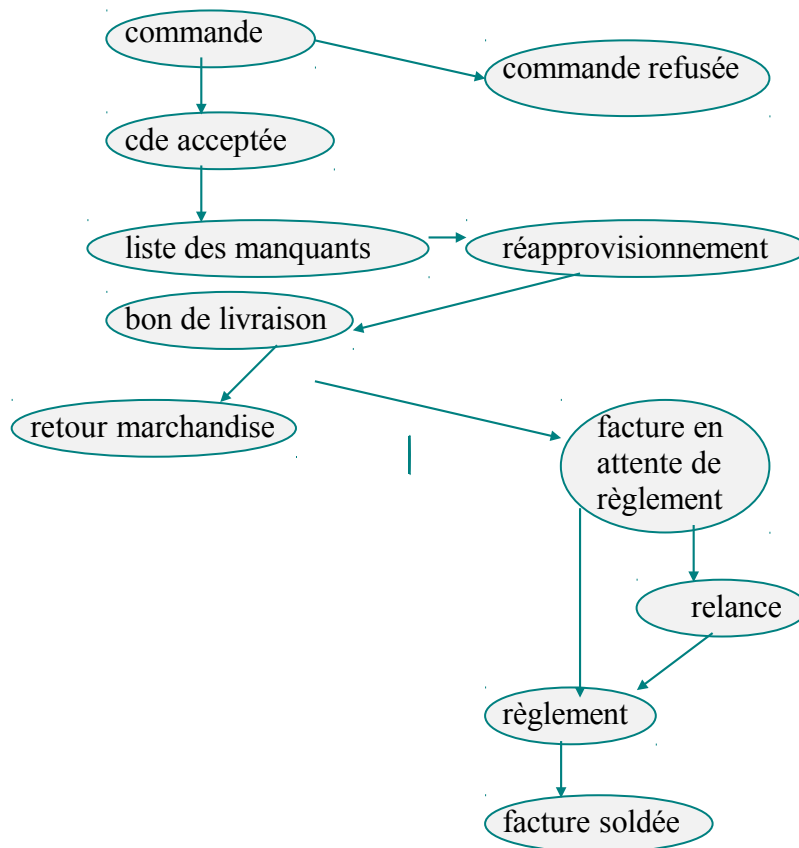
RG₇ : toute facture non réglée à échéance donne lieu à relance.

2°) GRAPHE DE CIRCULATION DES INFORMATIONS



3°) On peut déduire le graphe d'ordonnancement des flux en prenant soin d'éliminer tout ce qui est de nature organisationnelle.

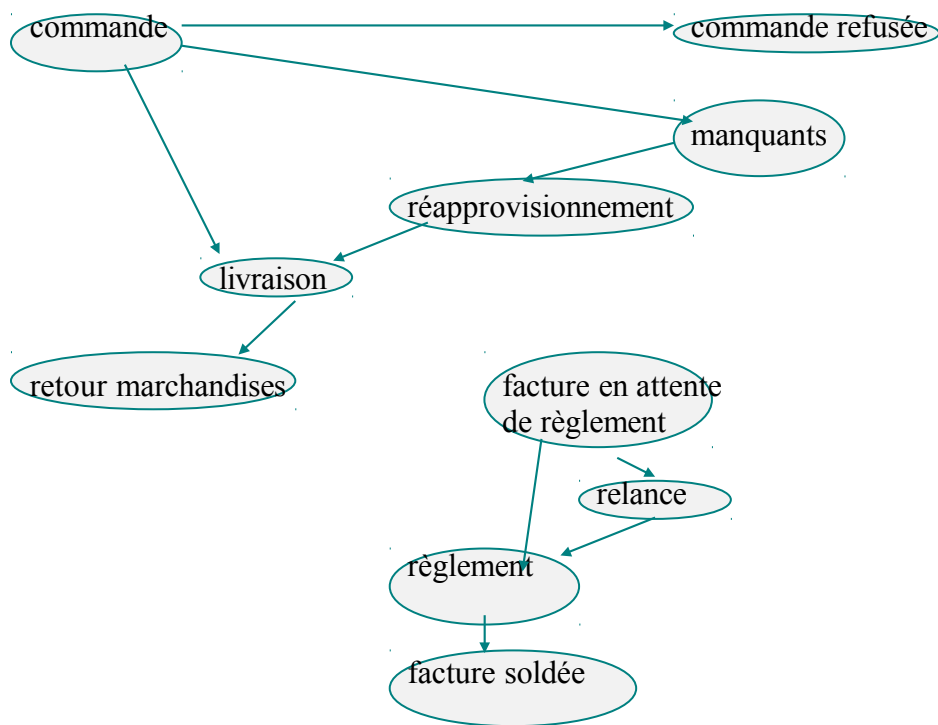
a- Première approche



Le fait que les commandes acceptées transitent du service commercial au magasin relève de choix organisationnels : « commande acceptée » n'est donc pas un événement conceptuel. On le retire du graphe.

On évitera des expressions telles que « bon de livraison » ou « liste manquants » qui font penser à l'organisation existante.

b- Graphe d'ordonnancement des flux (2^{ème} approche)



4°) CONSTRUCTION DU M.C.T

Les événements d'un même niveau sont alignés sur la même verticale.

Exemple de construction de M.C.T (TD3)
(cas organisme mutualiste)

1- liste des acteurs

- * mutualiste (client)
- * réparateur
- * point service après vente
- * agent régional

2- liste des événements

- | | |
|----------------------------|-------------------------------|
| * panne déclarée | * matériel réparé transmis |
| * matériel de prêt expédié | * ordre de réparation |
| * matériel neuf expédié | * matériel endommagé transmis |
| * accord client | * diagnostic |
| * panne enregistrée | * estimation du coût |
| * matériel endommagé reçu | * ordre de destruction |
| | * matériel réparé |

L'agent régional est l'acteur de l'organisme qui effectue des échanges avec les acteurs externes.

3°) GRAPHE DES FLUX (schéma de circulation des informations) ou diagramme des flux

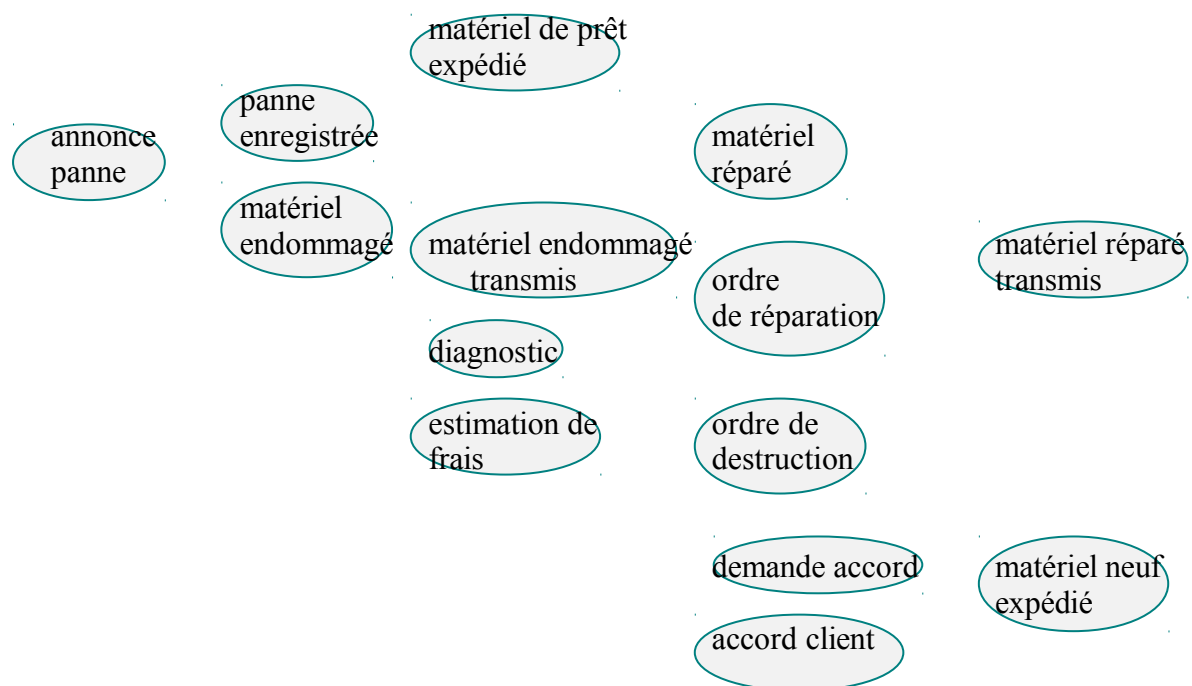
Le graphe des flux est récapitulatif des échanges entre l'agent régional et les acteurs externes :

4°) MATRICE DES FLUX

La matrice des flux est une représentation des flux sous forme de tableau. Aussi bien en ligne qu'en colonne sont spécifiés tous les acteurs.

De \ Vers	client	agent régional	réparateur	service après vente
client		annonce panne accord		
agent régional	matériel neuf prêt	panne enregistrée	matériel en panne ordres	matériel réparé
réparateur		diagnostic estimation du coût		
service après vente		matériel endommagé		

5°) LE GRAPHE D'ORDONNANCEMENT



6°) GRAPHE M.C.T

Le modèle conceptuel des traitements suivant résulte du GOE

N.B :

L'opération d'expédition comporte autant de règles d'émission (r, p, n, e) qu'il y a d'événements alternatives : matériel réparé transmis, matériel de prêt expédié, matériel neuf envoyé, matériel endommagé transmis.

L'opération de décision de réparation comporte 2 règles d'émission traduisant soit une réponse affirmative, soit une réponse négative.

Ok produit l'événement d'ordre de réparation

non ok () produit l'événement d'ordre de destruction

VIII/ NOTIONS COMPLEMENTAIRES

1°) La non redondance des opérations

Une même règle de gestion ne doit pas apparaître dans plusieurs opérations. On peut toujours regrouper les règles de gestion similaires dans une même opération en définissant la synchronisation et les règles d'émission adéquates.

Exemple :

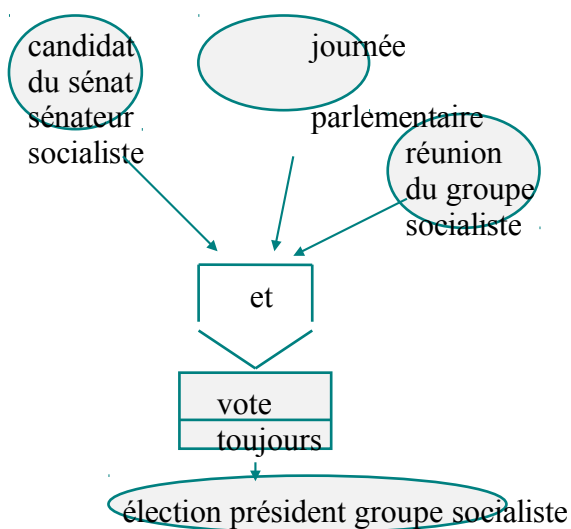


FIGURE 1

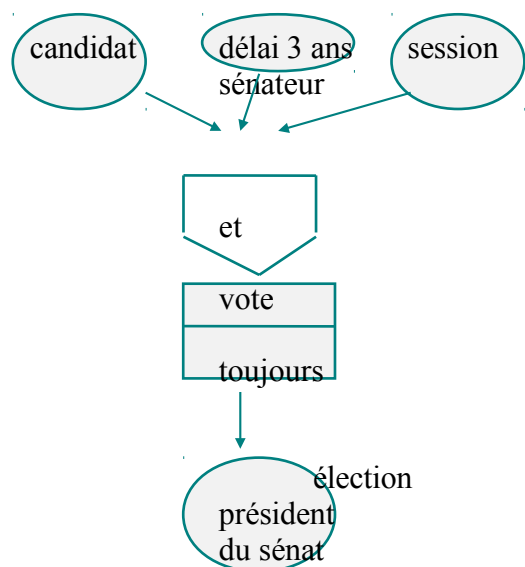


FIGURE 2

On peut les regrouper en un seul modèle

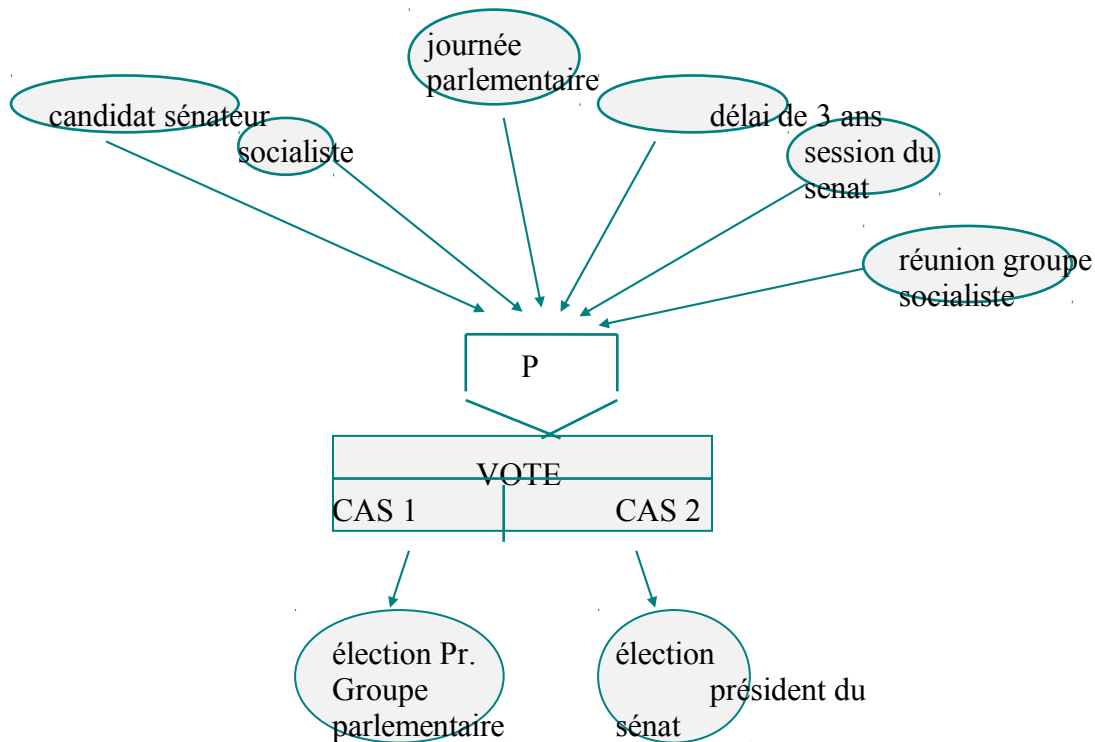


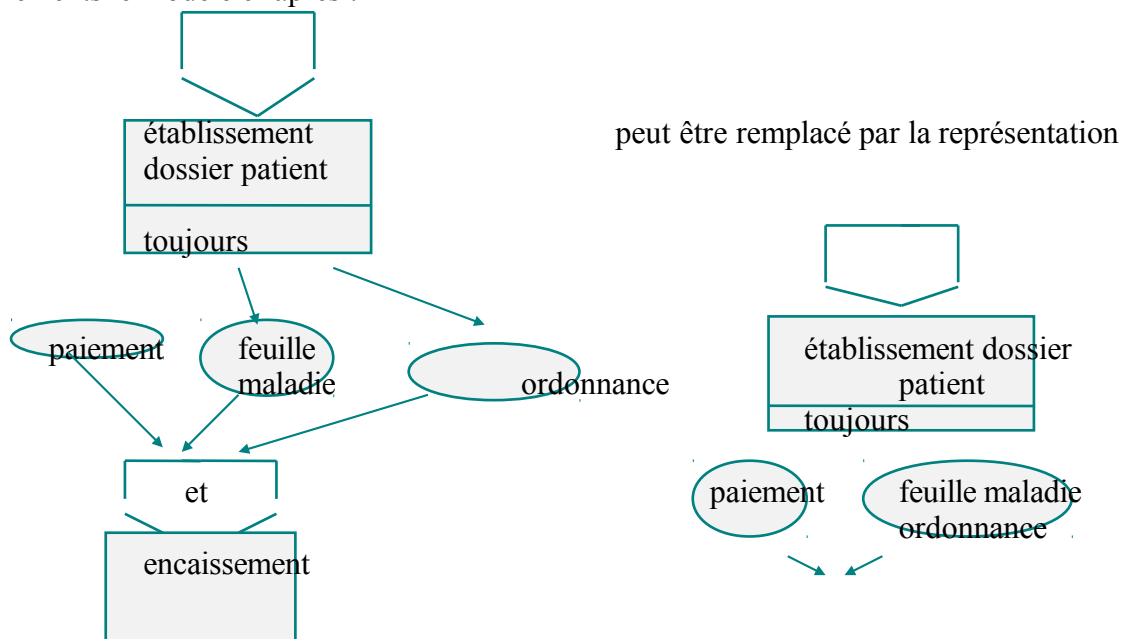
FIGURE 3

2°) La Non - Redondance des Evénements

Pour éviter les synchronisations qui n'ont pas lieu d'être, il est opportun de regrouper dans le même symbole des événements créés simultanément et destinés à la même opération.

Exemple :

Lors d'une consultation médicale, les services hospitaliers encaissent le paiement du patient une fois que la feuille de maladie et l'ordonnance ont été produites. Pour éviter une séparation inutile des événements le modèle ci-après :

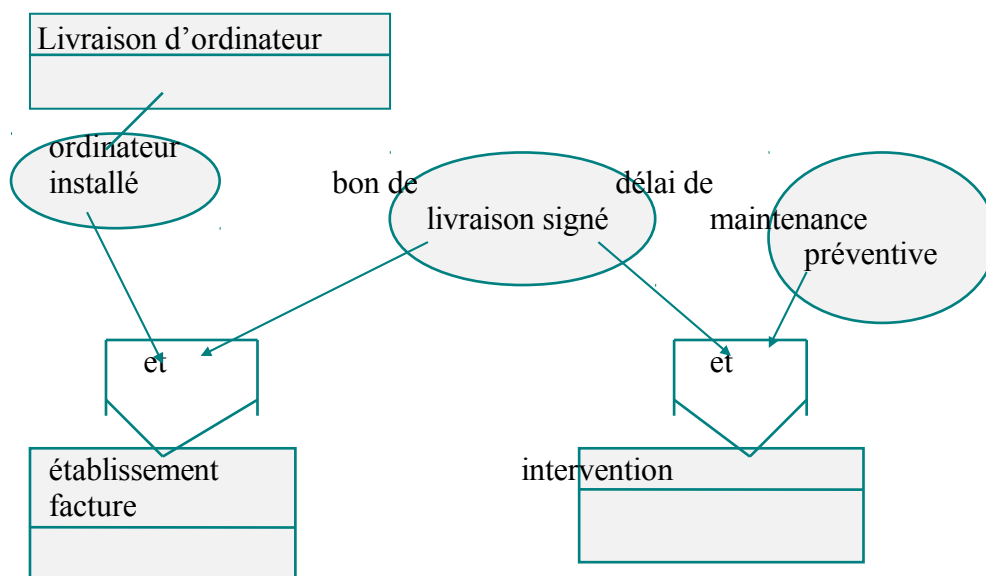




3°) Cas de Conflit

Les situations de conflit se produisent lorsqu'un événement est sollicité au même moment dans plusieurs synchronisations.

Exemple : suite à une installation de matériel l'opération de facturation est déclenchée une fois que la signature du client est apposée sur le bon de livraison, cette condition est aussi nécessaire pour l'opération d'intervention.



La restriction à apporter à ce schéma est l'impossibilité d'une exploitation simultanée de l'événement bon de livraison signé par les règles de gestion facturation et intervention.

4°) Homogénéité des Opérations

Ils vaut mieux obtenir des opérations petites mais homogènes vis à vis des résultats.

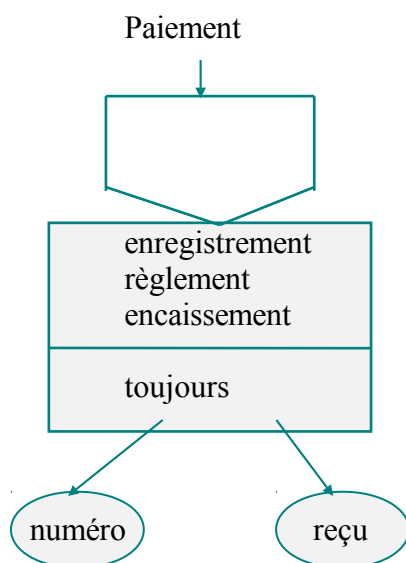
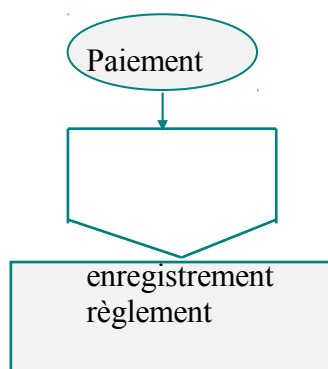
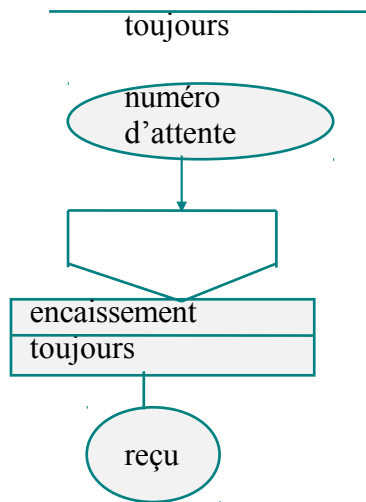


Figure 1





CHAPITRE 3

LE MODELE ORGANISATIONNEL DES TRAITEMENTS

I/ OBJECTIFS

A la vue purement fonctionnelle de l'entreprise, fournie par le modèle conceptuel, doit à présent succéder une vue plus concrète s'appuyant sur l'organisation. Par rapport au "quoi" qui était l'objectif essentiel auquel devrait répondre le M.C.T (par la définition des acteurs de l'organisme et les acteurs externes qui agissent sur le système), le modèle organisationnel devra répondre aux questions suivantes :

- qui fait le traitement
- où est-il exécuté
- quand se déroule-t-il ?
- de quelles ressources dispose t- on pour réaliser les actions ?
- selon quelle périodicité se déroulent les traitements ?
- quelle est la durée d'exécution des traitements ?

La procédure intègre l'ensemble de ces préoccupations et représente la suite des règles qui se déroulent sans attente au niveau d'un poste de travail.

II/ APPROCHE INTUITIVE

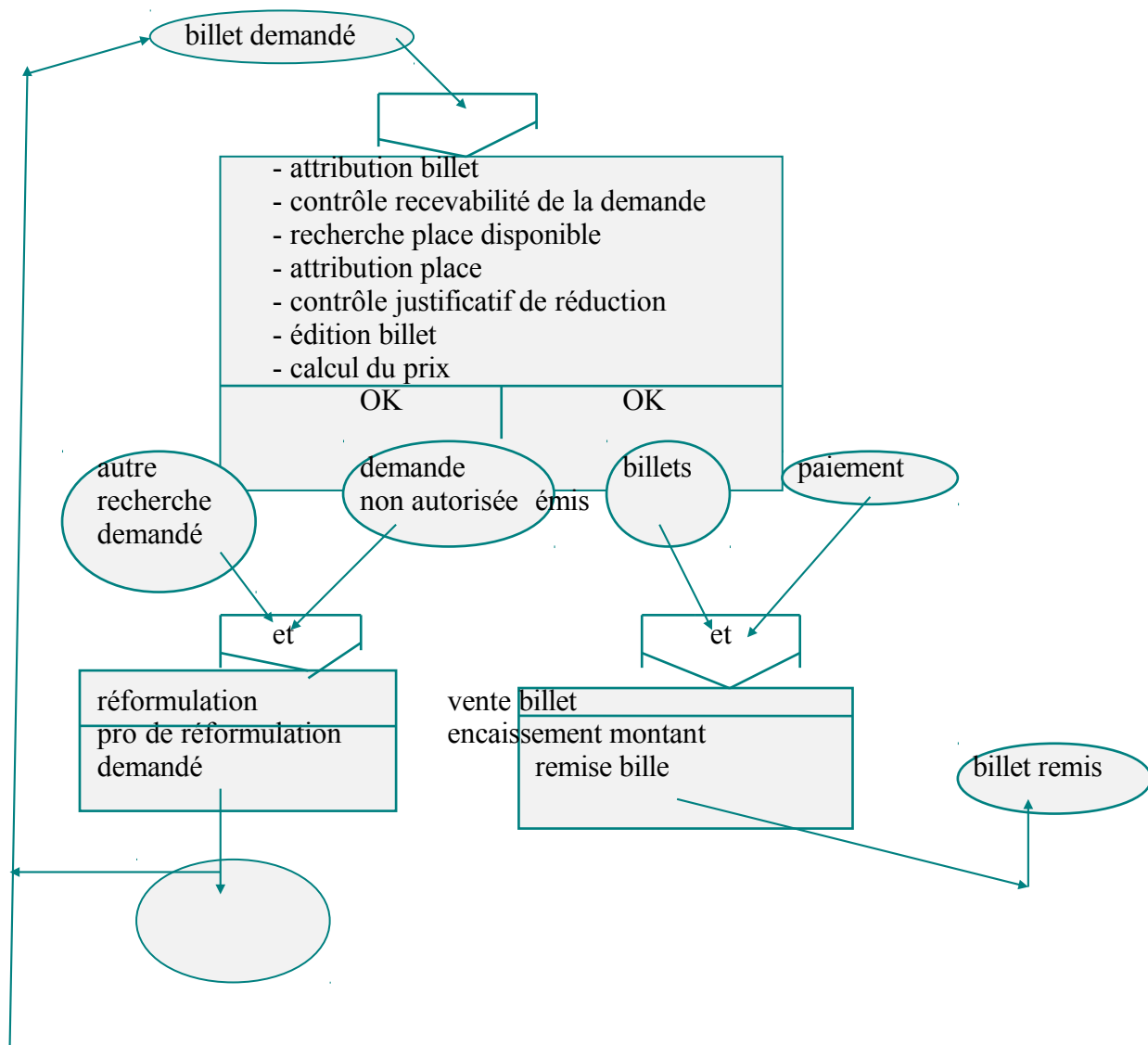
Soit l'exemple suivant : vente de places avec possibilité de réservation (théâtre, Sitarail...).

Les règles de gestion sont les suivantes :

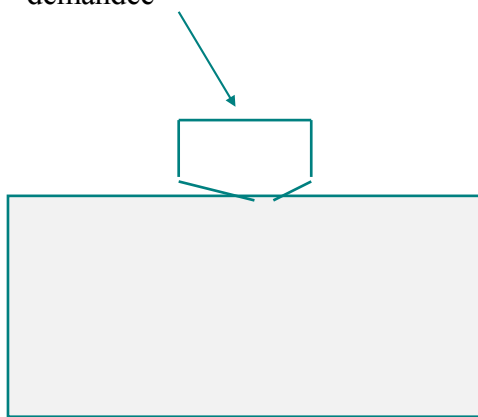
- aux heures d'ouverture, l'organisme peut délivrer, soit des billets à l'avance, soit des billets pour l'entrée immédiate
- les réservations de place sont possibles sous certaines conditions (moins de 2 mois à l'avance)
- pour toute attribution de place un billet doit être émis
- des réductions sont attribuées sur présentation d'un justificatif (militaires, étudiants)

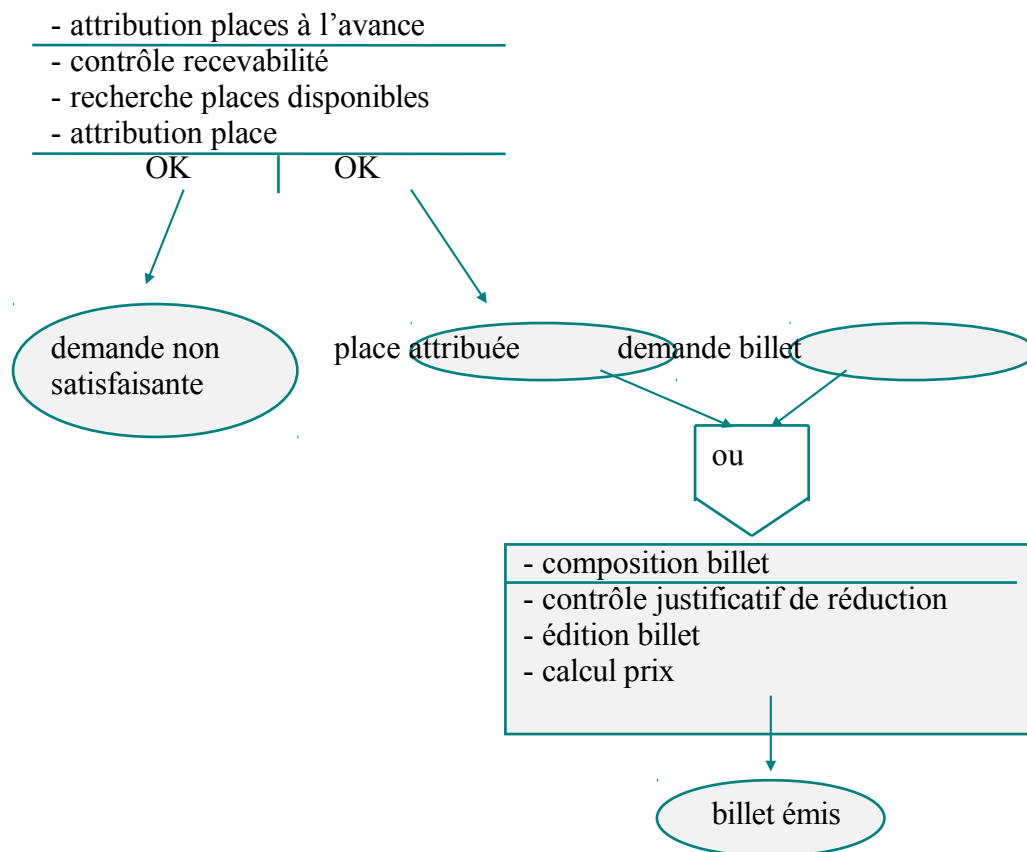
- aucun billet ne peut être délivré si son paiement n'a pas été perçu au préalable
- pour les entrées immédiates les billets sont délivrés sans attribution précise d'une place.

Le M.C.T simplifié sera le suivant :



réservation
demandée





Nous appuyant sur le modèle conceptuel ci-dessus (vente de places), cherchons à organiser chacune des opérations.

Parmi les objectifs et contraintes dégagés lors des entretiens certains concernent précisément le niveau où nous nous situons.

Par exemple, seront affirmés des règles d'organisation :

- Les attributions de billet seront dévolues à 2 portes
 - * un guichet public devra être organisé de façon à satisfaire immédiatement les demandes
 - * un bureau de gestion pour les demandes par correspondance souvent adressées par les associations
- Le guichet public devra être organisé de façon à satisfaire immédiatement les demandes

- Le bureau de gestion disposera d'un délai de réponse qui en aucun cas ne saurait dépasser 3 jours ouvrables
- La gestion des places et des tarifs pourra, si cela paraît de nature à améliorer le service, être automatisée

Il apparaît donc que l'opération d'attribution de billet qui, d'un point de vue conceptuel forme un tout homogène, devient, si on l'organise, un ensemble de tâches disparates telles que :

- recherche des places disponibles au guichet du public par un traitement automatisé déclenché à chaque demande
- édition des billets au bureau de gestion un traitement automatisé déclenché chaque soir
- contrôle du justificatif de réduction au guichet du public par un traitement

manuel, déclenché à chaque demande.

Pour chaque action du niveau conceptuel sont donc précisés le poste de travail, la nature du traitement et sa chronologie de déclenchement. Les tâches pour lesquelles ces 3 points sont identiques seront regroupées pour constituer le nouveau bloc homogène servant de base à l'organisation. Un tel ensemble cohérent sera baptisé phase. Les opérations seront donc découpées en phases.

On aura donc

III/ DEFINITION ET FORMALISME

1°) REGLE D'ORGANISATION

Expression de l'organisation mise en place en termes de poste de travail, de la nature du traitement et de la chronologie.

Exemple :

- Le guichet du public transmettra les dossiers de demande en fin de journée au bureau de gestion
- Le gérant sera tenu de vérifier chaque semaine les commandes non encore livrées

2°) TACHE

Action ou sous - définition d'action pourvue d'une organisation définie par les règles d'organisation.

En d'autres termes, une action est un constituant d'une procédure fonctionnelle et est composée d'un ensemble d'actions élémentaire soit par l'homme (soit par l'homme) soit par la machine.

Exemple

- saisie automatisée en début de journée des bordereaux
- transmission, au fil des arrivées, des dossiers au service du

personnel

3°) EVENEMENT

Fait réel dont la venue a pour effet de déclencher l'exécution d'un ou plusieurs tâches.

Exemple : dossier reçu
bordereau enregistré

4°) SYNCHRONISATION

Condition booléenne, traduisant les règles de gestion et d'organisation, que doivent vérifier les événements pour déclencher les tâches.

Exemple : (dossier transmis) & (contrôle effectué)

5°) PHASE

Ensemble de tâches dont l'enchaînement ininterrompible, compte tenu de l'organisation mise en place, n'est conditionné par l'attente d'aucun événement autre que le déclencheur initial.

En conséquence, le poste de travail, la nature du traitement ainsi, que son déroulement dans le temps seront communs à toutes les tâches d'une même phase.

Exemple :

Enfin de journée au centre de calcul, un traitement par lots assurera la phase "d'édition" qui regroupe les tâches :

- calcul du prix
- édition du billet
- édition de la facture

6°) REGLE D'EMISSION

Condition, traduisant les règles de gestion et d'organisation, à laquelle est soumise l'émission des résultats d'une phase.

Exemple : si contrôle positif, alors...

7°) RESULTAT

Produit de l'exécution d'une phase. Le résultat, fait réel de même nature que l'événement, pourra être déclencheur d'une autre phase.

Exemple : - formulaire transmis
- bordereau édité

Le formalisme complet est le suivant :

Le formalisme est le même que celui du modèle conceptuel des traitements à conditions de remplacer les opérations par les phases. A celles-ci des noms sont attribués de façon à les identifier sur le schéma.

Trois colonnes seront ajoutées au graphique de façon à préciser pour chaque phase :

- le déroulement chronologique
- la nature du traitement
- le poste de travail concerné

IV/ CONSTITUTION DU MODELE

A) CREATION DES TACHES

1° Les Composantes de l'Organisation

Comme nous venons de le voir, une tâche est une action pourvue d'une organisation. Il convient donc de préciser les trois composants de toute organisation.

a- Le poste de travail

On pourra l'identifier soit par la fonction qu'il remplit (comptabilité, atelier de saisie, relations avec le public), soit par les moyens qu'il emploie (profil de l'agent et type de la machine par exemple).

La 2^{ème} solution, plus analytique, permet de rattacher simplement tout poste de travail à l'une des 3 feuilles suivantes :

- un homme sans moyen informatique
- un homme doté de moyen informatique (par exemple un micro-ordinateur)
- une machine informatique seule

b- La nature du traitement

On précisera si le traitement reste manuel ou doit être informatisé. Pour le dernier cas, il faudrait indiquer le type d'automatisation :

- traitement conversationnel
- traitement par lots

C- Période de déroulement

Cette information définira la période de temps pendant laquelle on accepte d'exécuter l'action.

2° Règles d'Organisation

La mise en place d'une organisation satisfaisante est un travail capital dans la réussite du projet. L'expérience et l'image du concepteur sont donc importants. Cependant, celui-ci n'est pas totalement libre de ses choix et il devra appuyer sa réflexion sur 3 axes.

a- Partir du modèle concepteur

Comme vu précédemment, celui-ci impose les points d'attente et propose un découpage en processus.

b- Respecter les objectifs

Ceux-ci sont souvent exprimés en des termes où se trouvent mêlés des choix à caractère conceptuel, organisationnel et même parfois physique ("vous devez utiliser la machine x").

C- Tenir compte des moyens dont dispose l'entreprise

Il s'agit de rester réaliste et d'intégrer tant les moyens financiers que matériels et humains susceptibles d'être réellement utilisables.

Il faut être précis sur les effectifs et leur niveau de qualification pour cela l'observation de l'organisation existante fournira de précieuses indications.

Remarque

A la différence des règles de gestion, les règles d'organisation seront essentiellement des règles d'actions et non de calcul.

B) PHASES

1° Passage des tâches aux phases

Pour chaque opération a donc été défini un ensemble de tâches, traduction fidèle des règles d'organisation issues d'objectifs et contraintes créés par l'équipe de conception. Celles-ci, comme les règles de gestion, exprimant à la fois :

2° L'enchaînement des tâches entre elles

3° L'organisation appliquée aux tâches elles-mêmes

La création d'une phase consiste alors à regrouper, en un ensemble ininterrompu, des tâches respectant la règle des 3 unités :

- unité de lieu (même poste de travail)
- unité d'action (même opération, même nature de traitement)
- unité de temps (même période de déroulement)

On attribuera ensuite un nom à chaque phase de façon à l'identifier.

3° Notion de Procédure : (équivalent de processus)

Définition :

Une procédure est un enchaînement de phases dont les opérations originelles appartiennent au même processus et permettent de parcourir celui-ci en totalité.

4° Réparation des Taches Selon la Nature des Procédures Fonctionnelles

1) Procédures fonctionnelles manuelles

Ici, les tâches sont toutes composées exclusivement de consignes et de supports propres au travail manuel.

2) Procédures fonctionnelles temps réel

Elle se composent de tâches manuelle exécutées par l'homme et des tâches automatisées exécutées par l'ordinateur.

Les 1ères incluent :

- des consignes données par l'utilisateur
- des consignes d'utilisation de terminal
- des états et imprimés nécessaires.

Et les secondes peuvent concerner :

- la génération d'états
- les contrôles des entrées
- les algorithmes de traitement informatique.

3° Procédures fonctionnelles différées

Elle sont constituées d'algorithmes qui s'exécutent automatiquement.

V/ CONSTRUCTION DE MODELES EXTERNES

1°) GENERALITES

Les modèles conceptuels de données (M.C.D brut) sont des vues panoramiques de l'univers tel qu'il apparaît au concepteur à un moment donné.

On peut donc améliorer cela en introduisant de nouvelles règles organisationnelles.

Le M.C.D brut nous présente des données superflues et des données manquantes à l'ensemble des procédures fonctionnelles.

La validation des données comporte 2 grandes phases :

- La construction des modèles de données propres à chaque procédure fonctionnelle appelées modèles externes ou vues externes. Le qualificatif externes pour bien préciser qu'ils trouvent leur origine à l'extérieur du groupe d'étude des données.

- La confrontation de ces modèles externes avec le modèle conceptuel brut.

* Procédés d'élaboration des modèles externes

On peut s'appuyer soit sur une approche événementielle, soit sur le langage naturel ou sur un schéma des flux. Le procédé déductif classique se fonde sur l'analyse des documents.

Un document est un repère d'informations écrit (bordereau de saisie, état de sortie), virtuel (écran de saisie) ou oral (transmission téléphonique) produit ou utilisé régulièrement et rattaché à des acteurs. A chaque procédure fonctionnelle est rattaché un ensemble de documents.

2° Exemple de Construction de Modèles Externes

Encaissement au magasin.

Une des solutions organisationnelles visant à faciliter la tâche du caissier et à mieux satisfaire le client a été de concevoir un ticket de caisse élaboré.

Société AMACAM Magasin n°					n° ticket date	
produit	code	désignation	prix unit	qté achetée	prix total	
				somme à payer montant du paiement..... mode de paiement reste du.....		
La grille d'analyse correspondante se présente comme suit :						

Application : encaissement code document : Doc 11			PF : 02
			Diffusion (+) : 2
libellé de rubrique	propriétés correspondantes	nature	entité de rattachement
n° ticket	1) n° ticket d'achat	SI	ticket
date	2) date achat	SI	ticket
numéro magasin	3) numéro magasin	SI	magasin
code produit	4) code produit	SI	produit
désignation	5) désignation produit	SI	produit
qté achetée	6) qté achetée du produit	M	achat / produit
prix unitaire	7) prix unitaire	SI	produit
prix total	6), 7	C	-
somme à payer	6), 7	C	-
montant paiement	8) montant paiement	M	achat
type paiement	type paiement	M	achat
reste dû	6), 7), 8)	C	
A partir de la liste des propriétés de la grille d'analyse, on obtient le modèle externe suivant :			

VI/ **APPLICATIONS** Cas agence immobilière chanteloup

Travail à faire :

- 1) A partir d'une analyse conceptuelle, présenter le modèle organisationnel des traitements relatant les opérations à effectuer vis à vis des propriétaires.
- 2) Même question pour les opérations à effectuer vis à vis des locataires.

Réponse :

Démarche : * construire d'abord le M.C.T

* mais avant construire le graphe des flux et G.O.E

savoir les événements émis par le propriétaire. Donc connaître d'abord la liste des acteurs.

1° M.C.T des opérations à effectuer vis à vis des propriétaires

Liste des événements émis par le propriétaire :

- * offre de location
- * accord de principe (sur le tarif)
- * mandat de location signé

Liste des événements émis vers le propriétaire

- * tarif
- * mandat de location
- * chèque solde
- * chèque acompte

Liste des événements internes non ventilés vers l'extérieurs

- * appartement enregistré
- * état détaillé édité

Liste des événements temporels

- * au 30 octobre de l'année
- * avant le 30 mai, après la saison

Graphe du M.C.T propriétaire

2) MOT des opérations à effectuer vis à vis des locataires.

Tout comme les opérations vis à vis des propriétaires, c'est le responsable des locataires qui a la charge de toutes les procédures fonctionnelles ici. La première procédure P1 dont le but est d'aboutir à la location est constituée par les procédures fonctionnelles PF₁, PF₂, et PF₃ qui sont respectivement : recherche disponibilité, pré - location, location.

L'objet de la procédure P₂ est de préparer le début du séjour du client. La procédure P₃ se termine par la clôture du compte du client arrivé et fin de séjour.

Liste des événements émis par le locataire

- * demande de location
- * acompte
- * caractéristiques locataires

- * acompte
- * solde
- * exemplaire n° 1 signé
- * assentiment du locataire (par rapport à l'état des lieux)
- * clés rendues par le locataire

Liste des événements émis vers le locataire :

- * proposition location
- * avis indisponibilité
- * convention location
- * clés remises au locataire
- * caution remboursée partiellement ou en totalité
- * somme à rembourser

Liste des événements non ventilés vers l'extérieur

- * client et acompte enregistrés
- * location effectuée
- * solde enregistré
- * compte clôturé

Liste des événements temporels

- * jour de l'arrivée
- * fin de séjour

M.C.T locataire

VII/ APPLICATION 2 Cas gestion de stock et approvisionnements

A) DESCRIPTION DU NIVEAU CONCEPTUEL

1° Rôle du Système

Il s'agit ici d'un système de gestion des stocks comportant 2 processus :

- tenue de stock
- approvisionnement

Ce système est en étroite liaison avec le système de gestion des commandes.

2° Règles de Gestion

RG₁ : un produit peut être en stock dans plusieurs magasins

RG₂ : un produit en magasin peut être mouvementé plusieurs fois par diminution ou augmentation de la quantité en stock

RG₃ : un produit est vendu par un seul fournisseur pour tous les magasins

RG₄ : le système concerne une entreprise de distribution qui achète des produits aux fournisseurs pour les revendre à ses clients

RG₅ : une commande de réapprovisionnement concerne un fournisseur

RG₆ : on passe une commande à un fournisseur dans l'un des 2 cas suivants :

- un produit commandé par un client à un magasin est en rupture de stock dans ce magasin
- dans un magasin, on a pour un produit :
 $\text{stock} + \text{total commandé aux fournisseurs} < \text{stock minimum}$.

On commande alors $Q = \text{stock maximum} - (\text{stock} + \text{total commande})$. Autrement dit pour chaque produit d'un magasin, on définit un stock maximum et un stock minimum et dès que le niveau du stock tombe en dessous du stock minimum on commande ce qu'il faut pour remonter au stock maximum.

RG₇ : les livraisons des fournisseurs sont contrôlés par comparaison avec les commandes. Toute livraison non conforme est refusée et retournera chez le fournisseur.

RG₈ : on tient à jour un stock théorique d'après les mouvements du stock

RG₉ : à période fixe on fait un inventaire pour déterminer les écarts entre le stock physique réel et stock théorique déterminé par le S.I.

RG₁₀ : les mouvements du stock sont :

a) Hors période d'inventaire

- livraison fournisseur : $\text{stock} = \text{stock} + \text{quantité livrée}$
- bon de livraison : $\text{stock} = \text{stock} - \text{quantité livrée}$
- retour de marchandise client : $\text{stock} = \text{stock} + \text{quantité retournée}$

b) Pendant ou hors période d'inventaire

ajustement (suite à inventaire ou à écart occasionnel constaté)

$\text{stock} = \text{stock} \pm \text{écart entre stocks réel et théorique}$

Les retours de marchandises fournisseurs n'entrent pas en jeu car les marchandises sont retournées avant d'avoir été prises en compte dans le stock théorique.

B) CONSTRUCTION DU MODELE CONCEPTUEL DES DONNEES (M.C.D)

C) CONSTRUCTION DU MCT

2° Processus Tenue Stock

C) CONSTRUCTION DU MODELE ORGANISATIONNEL DES TRAITEMENTS (MOT)

1°) Règles d'organisation

- RO₁ : le service achats et les magasins sont équipés de micro-ordinateurs compatibles pouvant s'échanger des disquettes. Le service commercial dispose d'un matériel analogue.
- RO₂ : pour la détermination des commandes à passer aux fournisseurs, le micro du service achats édite toutes propositions de commandes qui sont analysées par le responsable en vue d'une validation ou de modification. Ces opérations doivent être faites le matin.
- RO₃ : les commandes validées sont éditées :
- a) dans l'ordre des fournisseurs concernés pour être expédiées à ceux-ci
 - b) dans l'ordre des magasins concernés pour être transmises à ceux-ci
- RO₄ : à chaque livraison fournisseur, le magasinier, contrôle la marchandise livrée en la composant à la marchandise commandée figurant sur la commande au fournisseur.
- RO₅ : la mise à jour du stock s'effectue :
- a) le matin à 9 heures pour les sorties de stock. Celles-ci proviennent du processus gestion des commandes clients et sont transmises au magasin concerné sur une disquette.
 - b) en temps réel (transaction en mode conversationnel) à tout autre moment de la journée de travail pour les autres mouvements. Les anomalies sont immédiatement recyclées.
- RO₆ : le courrier est expédié à 12 heures
- RO₇ : l'inventaire est annuel. Le vendredi soir précédent, il y a édition de l'état
- RO₈ : dans un magasin, tout produit doit pouvoir être rangé dans un seul casier et tout casier ne doit contenir qu'un seul produit.

Processus appro

Ce schéma se déduit du M.C.T en intégrant les contraintes liées à l'organisation

Processus tenue stock

Légende :

AC : automatisé conversationnel

AB: automatisé batch

M : manuel

CHAPITRE 4

MODELES EXTERNES ET VALIDATION

I/ MODELES EXTERNES

1°) PRINCIPE

Un modèle externe sera lié à un ensemble de traitements destinés à exécuter une et une seule des 2 fonctions :

- soit une mise à jour
- soit une consultation

Chaque traitement possède son **modèle externe** ou **vue externe**.

Il s'agit, d'une sorte de M.C.D qui n'aurait été construit que dans l'optique d'un seul traitement.

Ce modèle externe doit être établi sans se préoccuper du M.C.D qui constitue la représentation de la perception du réel au moyen d'une modélisation et ne préjuge pas des actions qu'on fera subir à cette représentation.

Cette séparation des données et des traitements est indispensable si on désire mettre en place un système cohérent, intégré, intégré e adaptable.

La validation permettra d'ajuster M.C.D et vues externes, c'est-à-dire de mettre en accord les données et les traitements.

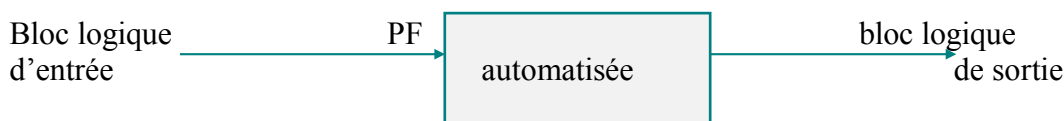
L'utilisateur ne « voit » que des flux de données (mouvements ou lots de données véhiculés par des événements conceptuels ou organisationnels).

Ces flux de données se transportent sous forme de **blocs logiques**. A ce stade, on ne s'intéresse plus qu'aux traitements automatisés.

Les données qui appartiennent à un flux entrant constituent des blocs logiques d'entrée et celles qui appartiennent à un flux sortant des blocs de logiques de sortie.

Les blocs logiques d'entrée (sortie correspondent à des PF automatisées dans lesquelles il y a communication avec l'environnement extérieur au SAI « système automatisé d'information » (saisir ou accès avec les écrans, des claviers, des imprimantes, etc...).

Pour les P.F batch sans communication avec l'extérieur, les données consultées ou émises sont relatives à des événements passés dont on a gardé la trace dans la base d'information ou destinées à produire ultérieurement des événements. On parle alors de blocs logiques internes, issus d'anciens blocs logiques d'entrée ou destinés à confectionner de futurs blocs logiques de sortie.



2° CONSTRUCTION DES MODELES EXTERNES A PARTIRE DES BLOCS LOGIQUES

A propos de chaque PF automatisé, on construira une vue externe pour chaque traitement de consultation et une vue externe pour chaque traitement de mise à jour.

Une P.F aura donc plusieurs modèles externes. En consultation, la vue externe sera construite à partir des données produites (bloc logique de sortie ou bloc logique interne). En mise à jour, elle sera établie à partir des données entrantes (bloc logique d'entrée ou bloc interne).

Toute combinaison d'événements porteurs d'information qui active une synchronisation et qui ne constitue pas une demande de consultation véhicule un bloc logique d'entrée (ou un bloc interne) et toute combinaison d'événements produits par une règle d'émission véhicule un bloc logique de sortie (ou un bloc interne).

3° REGLES DE CONSTRUCTION DES M.E

On construira une vue externe pour chaque mise à jour ou chaque consultation d'une même P.F.

On établira chaque modèle externe à partir d'un bloc logique, en utilisant le formalisme entités/relations étudié.

On appliquera en particulier les règles de normalisation, vérification et décomposition

La seule différence avec les règles du M.C.D est qu'une entité externe pourra ne pas avoir d'identifiant.

Les propriétés, entités ou relations externes pourront ou non être des propriétés, entités ou relations conceptuelles.

Si une propriété externe figure dans le dictionnaire des données, il faudra lui donner le même nom que la propriété conceptuelle correspondante (mais certaines propriétés externes ne se trouveront pas dans le dictionnaire des données, soit parce qu'elles sont de nature organisationnelle, soit tout simplement en raison d'oubli dans le dictionnaire).

Nous retiendrons donc les points suivants :

- une vue externe pour chaque consultation et chaque mise à jour de chaque PF automatisée
- utilisation des blocs logique
- utilisation du formalisme entités/reliations
- identifiant non obligatoire pour une entité externe
- utilisation des noms figurant sur le DD pour les propriétés externes qui correspondraient à des propriétés conceptuelles répertoriées.
- ignorance du modèle conceptuel des données.

4° APPLICATION

Soit la fonction consistant à édifier, en autant d'exemplaires qu'il y a d'élèves, l'emploi du temps d'une classe.

Les données recensés dans le MOT sont :

- classe
- matière
- jour
- nombres d'élèves
- tranche horaire
- salle

Le bloc logique de sortie est alors

supposons que le dictionnaire nous fournisse les données suivantes :

- classe
- n° de salle
- nom de l'élève
- vacation

- matière

REGLE 1 :

Le traitement d'édition, homogène correspond à une unique fonction de consultation. Le niveau ainsi défini est donc justifiable d'un modèle externe.

REGLE 2 : Référence au dictionnaire

- * Les données « classes » et « matière » sont conformes.
- * La « salle » s'appelle « n° de salle » dans le dictionnaire, on gardera donc « n° de salle »
- * Le « jour » est la « tranche horaire » sont absents du dictionnaire où figure la donnée « vacation ». Le traitement ne manipulant pas ces deux données séparément, on peut ne conserver que la « vacation » qui est, en fait le couple (jour, tranche horaire)
- * Le « nombre d'élèves » ne figure pas dans le dictionnaire. Il est calculable à partir des données « nom de l'élèves » et « classe, si l'on introduit une relation d'appartenance.

On gardera la donnée « nombre d'élèves » qui a un caractère élémentaire.

Les données à formaliser sont :

- classe
- matière
- vacation
- n° de salle
- nombre d'élèves

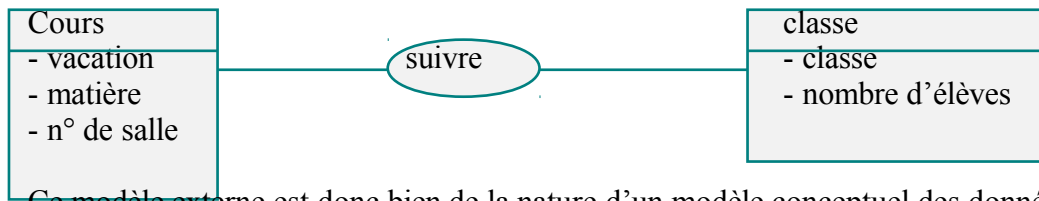
REGLE 3 & 4 : Formalisation

On créera un objet « emploi du temps », identifié par la « classe »

emploi du temps
- classe - matière - vacation - n° de salle - nombre d'élèves

La non répétitivité des propriétés impose de sortir de ces objet « vacation », « matière » et « n° de salle ». Seuls subsistent « classe » et « nombre d'élèves » et dès lors, il est plus judicieux d'appeler « classe » cet objet. Une « matière » et un « numéro de salle » se trouvent dans plusieurs vacations. Par conséquent, aucune de ces 2 données ne pourra identifier un objet portant la « vacation ».

La solution est alors de créer un objet que l'on pourrait appeler « cours » identifié par la « vacation », portant la « matière » et la « numéro de salle » et rattaché à la « classe ».



Ce modèle externe est donc bien de la nature d'un modèle conceptuel des données.

Remarquons que, si la fonction avait été l'édition de l'emploi du temps de chaque classe, le modèle eût été différents. Le bloc logique de sortie aurait alors une dimension supplémentaire, due aux diverses classe, selon le dessin suivant :

Dès lors, l'objet « cours » n'aurait pas été conforme au formalisme, puisqu'à une « vacation » pouvait correspondre plusieurs « matière » et « n° de salle ». La solution aurait été la création d'une relation cours, porteuse des propriétés « matières » et « n° de salle ».

Dans ce cas, le modèle externe aurait été :



Remarque :

A aucun moment, le M.C.D global n'a été utilisé. Seul son dictionnaire et son formalisme ont permis de parler le même langage. Ceci est essentiel et il faut bien se représenter la conception des vues externes comme étant effectuée par une équipe totalement ignorante des données.

L'établissement des vues externes ouvre la voie à l'étape suivante où elles seront confrontées au modèle conceptuel de données : la validation.

II/ LA VALIDATION

Chaque vue externe doit être mise en accord avec le M.C.D en mise à jour, on doit s'assurer qu'on donne bien au système tous les éléments nécessaires à cette mise à jour.

En consultation, on doit s'assurer que le système est capable de sortir les données désirées. Chaque modèle externe doit être déductible du M.C.D.

1° DEFINITION & DEMARCHE

a- Définition

* validation d'un modèle externe.

Valider un modèle externe : s'assurer qu'il est déductible du M.C.D.

* validation du MCD

valider le MCD : ne garer de celui-ci que ce qui est strictement nécessaires aux modèles externes validés

b- Démarche

* validation de chaque modèle externe par rapport au modèle conceptuel brut

- corrections éventuelles du modèle conceptuel avec, dans ce cas, revalidation des modèles externes déjà validées

* validation du modèle conceptuel brut par rapport à l'ensemble des modèles externes valides

- corrections éventuelles du MCD brut.

C- Définition pour chaque vue externe d'un sous modèle conceptuel, extrait du modèle conceptuel validé, dont elle soit déductible

la validation aura donc fourni un modèle conceptuel et des sous modèles validés ainsi qu'un ensemble de vues externes, garantie de la faisabilité des traitements

2° REGLES DE VALIDATION

La validation se procédera par étape, correspondant aux différents concepts du formalisme :

- validation des propriétés externes
- validation des objets (entités) externes
- validation des relations externes
- validation des cardinalités externes

A VALIDATION D'UN MODELE EXTERNE EN MISE A JOUR

Un modèle externe en mise à jour a pour rôle de créer, de modifier ou de supprimer certaines occurrences, définies par l'utilisation, d'un ou plusieurs propriété.

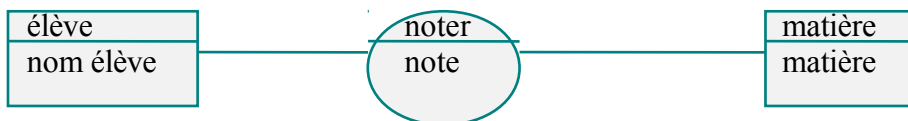
1° LES PROPRIETES

L'accès à l'occurrence d'une propriété que l'on veut mettre à jour nécessite 2 actions successives qui doivent pouvoir s'accomplir dans le cadre du MCD

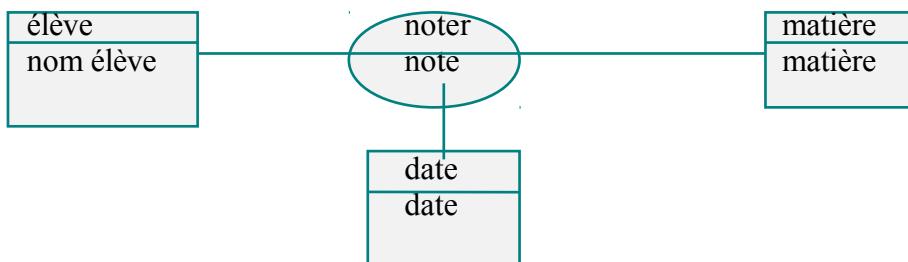
- identifier l'occurrence concernée par la mise à jour
- charger la nouvelle valeur de la propriété sur l'occurrence sélectionnée.
- Pour chaque propriété externe on rencontrera donc 3 cas de non validation.

a) La propriété externe a une fonction de chargement mais son identification est impossible

Exemple : si la vue externe de saisie des notes des élèves est la suivante :

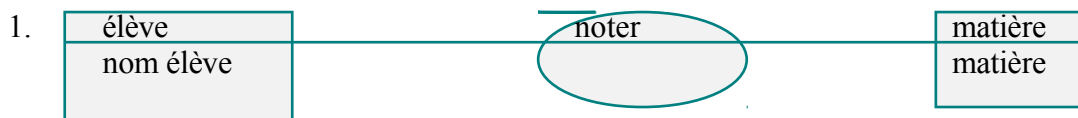


Le chargement d'une nouvelle note doit pouvoir se faire dès que l'on connaît « nom d'élèves » et « matière ». Or le MCD est :

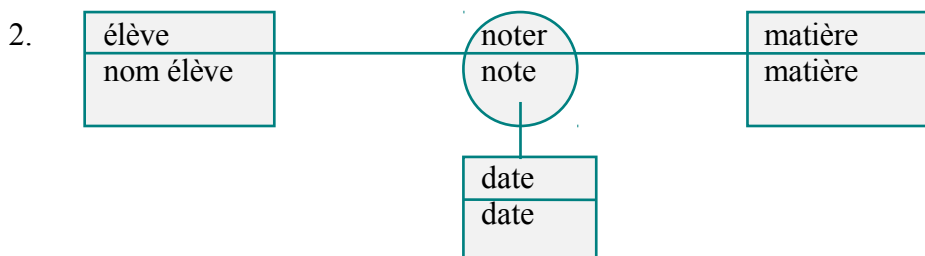


Il n'est donc possible d'accéder à la propriété « note » que si l'on connaît les 3 identifiants des objets associés dans la relation : « nom d'élèves », « matière » et « date ».

On a : 2 solutions



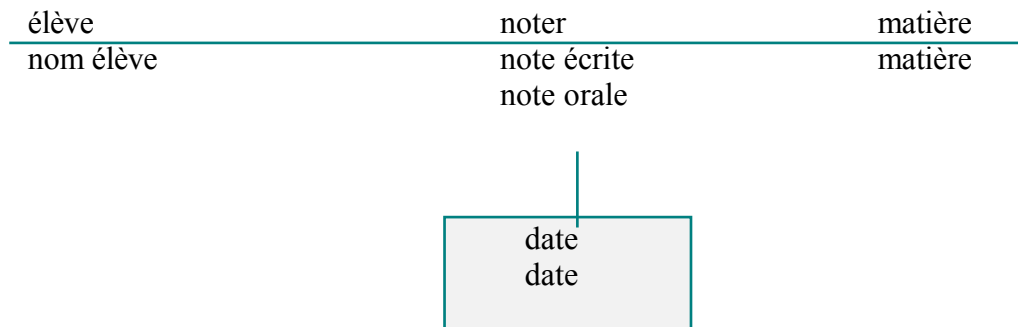
Pour un couple (élève, matière) n'existent qu'une note et modèle. Ceci correspondant à une saisie des notes au fil de leur attribution, puisqu'alors un élève n'a effectivement qu'une note par matière.



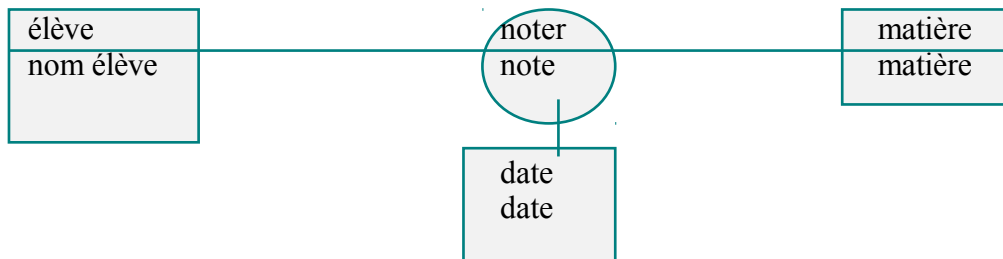
Cette fois, la date introduit une dimension supplémentaire qui permet de saisir plusieurs notes pour un élève, dans une matière, à des dates différentes. Ceci correspond à une saisie des notes regroupées, par exemple tous les mois.

- b) La propriété externe a une fonction d'identification pour une propriété dont le chargement est impossible

La vue externe est :



Le MCD étant :



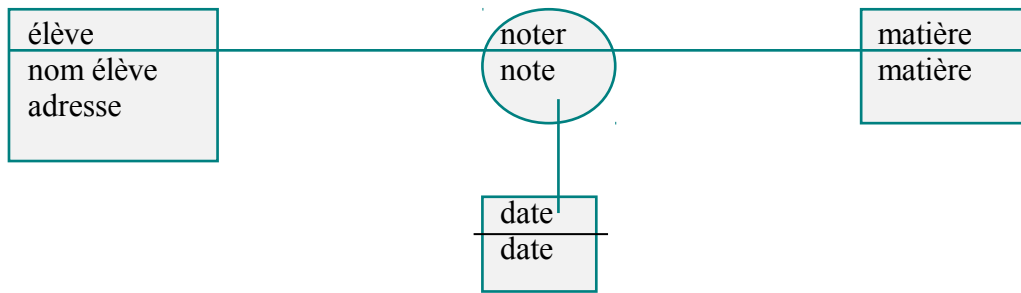
Les propriétés externes permettent d'identifier sans ambiguïté l'occurrence de la relation sur laquelle on veut charger les notes (exemple, Ouattara en maths le 20 janvier). Mais le chargement est impossible car les propriétés « note orales » n'appartiennent pas au MCD. On devra alors :

- soit modifier le schéma conceptuel et remplacer « note » par « note orale » et « note écrite ».
- soit abandonner les 2 notes pour ne garder de la vue externe qu'une propriété « notes » égale, par exemple, à la moyenne des 2 notes.

Le choix sera guidé par l'usage auquel est destiné le système d'information c'est à dire par le niveau de finesse des consultations que l'on souhaite en faire et qu'expriment les règles de gestion.

- c) La propriété externe n'a une fonction ni de chargement ni d'identification

Dans ce cas la propriété externe est superflue et doit être supprimée de la vue externe. Soit par exemple, dans la vue externe précédente, la propriété « adresse de l'élève » portée par l'objet « élèves »

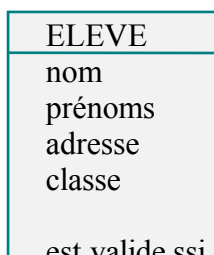


Les propriétés « nom élèves », « matières » et « date » ont une fonction d'identification pour permettre le chargement de la propriété « note ». « adresse élève » n'ayant aucun rôle de cette vue externe doit être éliminée.

2° LES OBJETS ENTITES ET LES RELATIONS

- a) une entité externe (objet externe) est reconnue valide lorsque l'ensemble des propriétés est valide au sens de 3 règles précédentes.

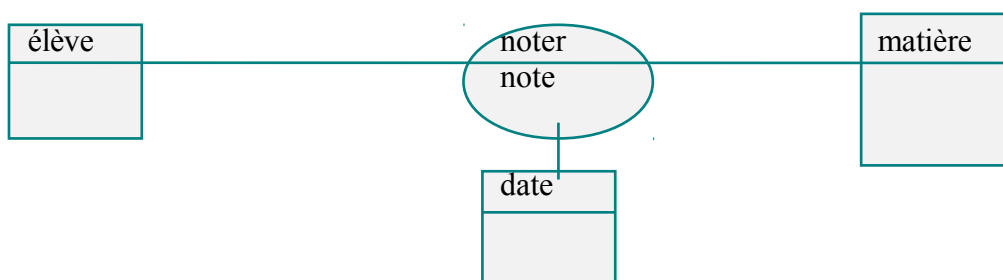
Exemple : l'entité « élève »



est valide ssi les propriétés : nom élève, prénom élève, adresse élève, classe sont valides.

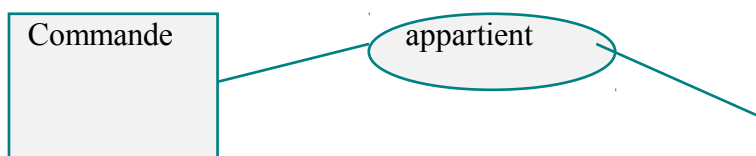
- b) une relation porteuse de propriétés est valide, lorsque l'ensemble des objets qu'elle associe et des propriétés qu'elle porte le sont.

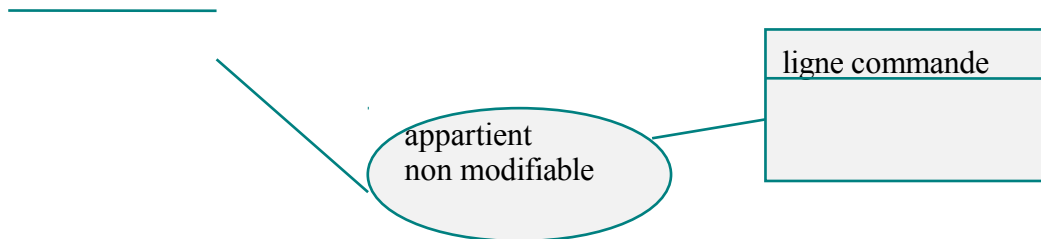
Exemple : la relation « noter »



est valide ssi : - les entités « élèves », « matière », et « date »
- la propriété « note »
sont valides

Une relation non porteuse de propriété est valide lorsqu'elle existe, identifiée par le même nom dans le modèle conceptuel, et que les objets qu'elle associe sont valides.





La relation « appartient non modifiable » sera valide ssi :

- les objets « commande » et « ligne commande » le sont,
- la relation « appartient non modifiable » figure dans le modèle des données

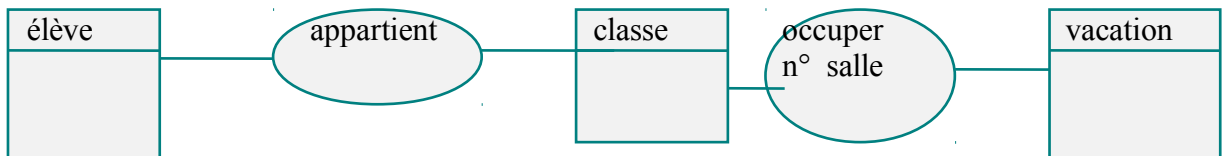
Le modèle est vraisemblablement le suivant :



On devra donc, d'une part, reconnaître que la relation « appartient » du MCD est différente de la relation « appartient non modifiable » et, d'autre part ajouter celle-ci dans le modèle.

3° LES CARDINALITES

Exemple : soit le MCD



Pour comparer les cardinalités externe de élève et vacation dans « se situer à » aux cardinalités conceptuelles, il est nécessaire de composer les 2 relations « appartient » et « occuper ». On obtient alors pour élève la cardinalité conceptuelle :

$$(1,n) \ ((1,n) = (1,1)u(1,n))$$

et pour vacation

$$(1,n) \ ((1,n) = (1,1)u(1,n))$$

On a ainsi le tableau comparatif

	cardinalité conceptuelle	cardinalité externe	conclusion
élève	1, n	1, n	valide
vacation	1, n	0, n	non valide

B° VALIDATION D'UN MODELE EXTERNE EN CONSULTATION

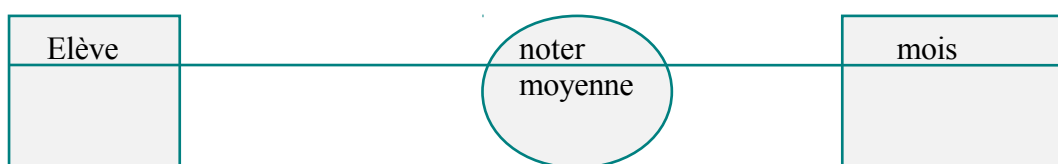
En consultation, l'utilisateur fournit seulement des critères et leurs valeurs. C'est à la machine de sélectionner les occurrences y répondant. Par exemple : « quelles sont les notes de mathématiques de Bernard depuis la rentrée ». La validation consistera à s'assurer que le système fournit toutes les occurrences, et rien que les occurrences, répondant aux critères de sélection. Ici on devra obtenir toutes les occurrences de la relation « note » pour « nom élève » = Bernard et « matière » = mathématiques, quelle que soit l'occurrence « date ».

1) PROPRIETES

Alors que précédemment la première question que la validation conduisait à poser était : « la machine peut elle identifier la propriété que l'on veut mettre à jour il en va différemment en consultation.

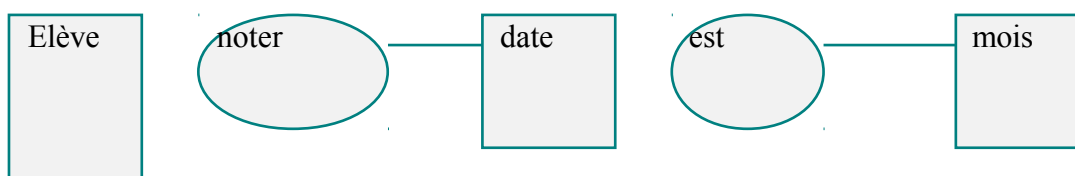
En effet, la consultation se présente sous forme d'une interrogation posée au système (« quels sont les ... ») et le premier souci est donc « la machine peut-elle comprendre cette interrogation ?

La validation imposera alors de façon à être certain que la demande de consultation soit comprise, l'identité des propriétés externes et conceptuelles. On devra, s'attacher à analyser les propriétés relatives ou obtenues par concaténation. Les propriétés calculées seront remplacées par les propriétés conceptuelles soit permettant de les obtenir par exemple, une consultation des moyennes mensuelles des élèves (toutes matières confondues).



Si le MCD est le suivant :

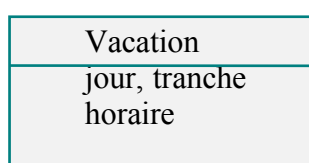
On remplacera alors « moyenne » par l'introduction de « matière » et de « note » qui en permettront le calcul et l'on obtiendra un modèle externe, sous structure du modèle des données.



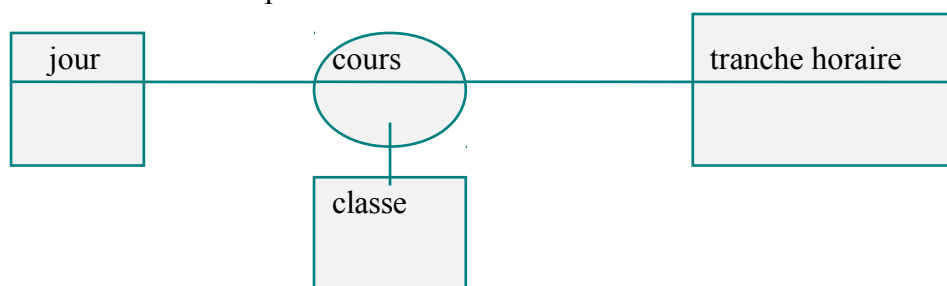


Les propriétés obtenues par concaténation de propriétés conceptuelles élémentaires sont la traduction de relations au sein du modèle externe. On s'assurera donc que celles-ci sont toujours exprimées dans le modèle des données.

Soit, par exemple, une vue externe de consultation des vacances existantes.



Le modèle conceptuel est le suivant :



Si l'on entend par vacation les couples (jour, tranche horaire) d'ouverture de l'établissement, même si aucun cours n'est dispensé (par exemple, lundi de 12 h à 14 heures) le modèle des données ne permettra de connaître que les vacation où existe un cours. La relation cours n'exprime donc pas la même information que la propriété « vacation ».

La validation impliquera alors :

- soit d'abandonner la notion de vacation sans cours et aucune modification n'est nécessaire
- soit, pour conserver cette notion, de créer une relation « vacation » dans le modèle des données entre jour et tranche horaire.

L'usage que l'on souhaite faire du S.I guidera notre choix.

2° ENTITES & RELATIONS

Une fonction de consultation comporte 2 niveaux d'expression :

- * l'énoncé des propriétés à travers lesquelles se fera la consultation (les critères)
- * les occurrences de ces propriétés qui sont concernés (la valeur des critères)

La validation vérifiera la faisabilité de ces 2 niveaux soit :

- peut-on accéder aux propriétés que l'on veut consulter ?
- peut-on ne garder que les seules occurrences qui nous intéressent ?

La validation des entités fournira la 1^{ère} réponse et celles des relations la seconde.

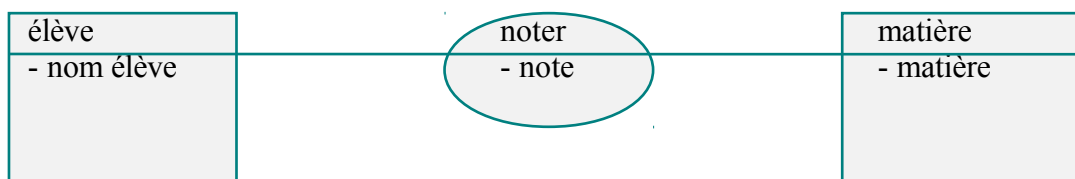
a. Accès aux propriétés

Pour chaque propriété participant à la consultation, on cherchera le ou les entités qu'elle met en jeu. On en déduira les identifiants conceptuels et l'on examinera alors si un traitement permet d'accéder à travers ces identifiants aux propriétés de la vue externe.

Exemple : la recherche de la note d'un élève dans une matière.

La question posée est : « quelle est la note de Bernard en Maths ? »

La vue externe correspondante est alors :



Cette vue exprime que pour un élève et une matière on ne s'intéresse qu'à une note (formellement correcte).

Le modèle conceptuel est le suivant :

Pour atteindre la note d'un élève dans une matière, il est donc nécessaire d'en connaître la date, et la vue externe ne la mentionne pas.

Supposons qu'un objectif impose la gestion d'un historique des notes, ce qui exclut de supprimer la date du modèle conceptuel ; il reste alors 2 hypothèses.

- 1) L'utilisateur ne souhaite pas être contraint de définir la date. Il accepte donc que lui soient communiquées toutes les notes de l'élève. Dans ce cas la vue externe n'est pas valide car, ainsi formalisée, elle ne permet qu'une note. L'entité, précisément la date, conduit à retenir comme vue validée la partie du modèle conceptuel concernée.

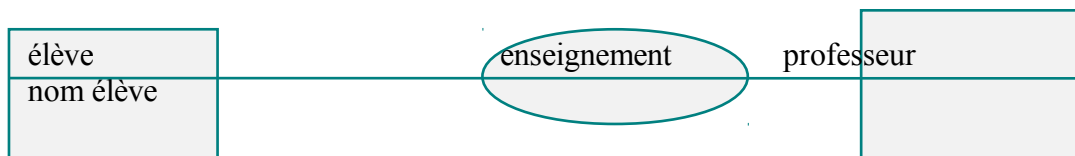
- 2) L'utilisateur ne souhaite obtenir qu'une note. Dans ce cas il doit accepter de préciser la date, soit directement, soit comme une donnée calculée (par exemple, la note la plus récente) ; donnée élémentaire ou calculée, la date doit appartenir à la vue externe. La validation de celle-ci conduit alors à la remplacer aussi, par la partie du modèle conceptuel présentée.

b. Accès aux occurrences

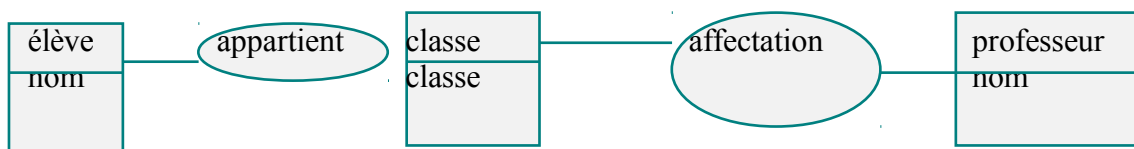
La bonne accessibilité aux propriétés ne suffit pas toujours à conclure à la validité du modèle externe.

Supposons que l'on veuille éditer, pour chaque professeur, la liste des élèves.

La vue externe est alors :



La partie du modèle conceptuel est la suivante :



Les entités externes, identiques aux entités conceptuelles sont donc parfaitement accessibles. Cependant, supposons qu'il existe des cours facultatifs que suivent certains élèves. Le lien entre les professeurs de cours facultatifs et leurs élèves possède des occurrences qui doivent être éditées, puisqu'il correspond parfaitement à la relation « enseignement » du modèle externe.

La validation du modèle externe, si l'on souhaite maintenir cette consultation, impose la création d'une relation « enseignement » entre « élève » et « professeurs » sur le modèle conceptuel.

II/ METHODE DE VALIDATION

On déroulera les étapes de la validation selon la chronologie suivante :

A) Validation des Modèles Externes en Consultation

C'est au cours de cette étape que seront introduites les données organisationnelles dans le modèle conceptuel. Si celui-ci impose aux modèles externes l'ajout d'entités ou de relations, pour permettre des consultations, il sera fréquent que l'on soit aussi emmené, dans le modèle des données, à réparer des oublis ou des erreurs révélées par les traitements. On s'assurera alors que chaque correction ne détruit pas la cohérence avec les autres consultations.

B) Validation des Modèles Externes en Mise en Jour

Le modèle des données étant maintenant stabilisée, cette étape pourra dérouler linéairement, modèle après modèle. Là aussi, on sera parfois emmené à revoir entièrement des phases de saisie ou même à en créer si elles ont été oubliées.

C) Quantification des Modèles Externes

La quantification doit être établie, modèle par modèle. On précisera donc :

- fréquence probable d'utilisation de la fonction
- volume moyen des informations saisies (pour une fonction de mise à jour) ou affichées (pour une fonction de consultation)
- temps estimé pour une utilisation de la fonction par un poste de travail.

On estimera ce temps en tablant sur des valeurs moyennes.

Saisie : 1,5 caractère par seconde

affichage et réflexion devant un écran : 10 secondes

volume d'in écran : 1600 caractères

volume d'une transmission : nombre de caractères utiles

Ces paramètres seront présentés dans un tableau suivant :

D) Validation du Modèle Conceptuel des Données

La validation se poursuivra par la mise à jour du dictionnaire des données du S.I.

- pour chaque propriété
 - * nombre de caractères et structure de l'information
- pour chaque entité
 - * nombre total de caractères
 - * nombre d'occurrences
- pour chaque relation
 - * nombre total de caractères
 - * nombre d'occurrences
- pour chaque entité participant à une relation
 - * cardinalités et minimum et maximum

IV/ CONCLUSION

1) LE POINT SUR LES CONCEPTS UTILISES

- MCD brut : Modèle Conceptuel des Données établi dans l'absolu, sans préjuger des traitements (perception du réel).
- Modèle externe ou vue externe : modèle des données « ou » par l'utilisateur à travers un traitement d'une procédure fonctionnelle.
- Validation : confrontation du MCD et des vues externes dans le but de les faire concorder en rendant chaque vue externe déductible du MCD.
- Modèle externe validé ou vue externe validée : vue externe après validation.
- Sous modèle conceptuel : extrait du modèle conceptuel correspondant à une P.F

2) PRINCIPE DE LA DEMARCHE

La démarche consiste à étudier séparément sur le plan conceptuel les données et les traitements. Il en résulte le MCD et le MCT.

Sur le plan organisationnel, on étudie les traitements (MOT

Les données sont alors validées par les traitements (MCD validé)

On pourra alors passer à l'étude organisationnelle des données (Modèle logique des données qui sera étudié au prochain chapitre).

CHAPITRE 5

LE MODELE LOGIQUE DES DONNEES (M.L.D)

Le modèle logique des données (M.L.D) a pour objet de décrire les enregistrements logiques et permet d'entrevoir la structuration physique des données grâce notamment aux estimations que l'on peut faire sur les volumes des enregistrements à mémoriser. Le modèle logique des données (M.L.D) est construit à partir du MC.D. en tenant compte des éléments suivants :

- Le niveau et le type d'automatisation : il s'agit de ne retenir dans le MLD que la partie du MCD qui sera automatisée
- L'orientation des choix techniques concernant le système de gestion des données :
 - * orientation base de données de type Coda syl
 - * orientation base de données de type relationnel
 - * orientation fichier classique

Bien que chacun des modèles possède un vocabulaire spécifique, ils découlent tous d'algorithmes proches permettant d'obtenir l'ensemble des constituants de chaque enregistrement logique.

I/ LE MODELE CODASYL (conférence ou data systems, languages)

Ce modèle appelé aussi « Réseau » est celui qui a été retenu initialement dans Merise comme support du modèle logique des données. Il est parfaitement adapté lorsque l'on utilise un système de gestion de bases de données de type réseau comme ID5II, Total, Clio.

1) LES CONCEPTS

Le champ (item) : plus petite quantité d'information manipulée. Il correspond à la donnée élémentaire et, comme elle, possède un certain nombre d'occurrence.

Exemple : le champ « matière » a pour occurrences : Français, Maths etc...

Le record (enregistrement, en anglais) est un ensemble de données élémentaires (de un ou plusieurs champs) qui représente un enregistrement logique.

Un champ permet de l'identifier de façon unique. On l'appellera la clé du record. (Il peut exister plusieurs clés).

L'accès à une occurrence d'un record se fera par l'intermédiaire d'un lien qui matérialise une relation avec un autre record.

Les deux principes énoncés ci-dessus sont les 2 principaux mode d'accès à un record.

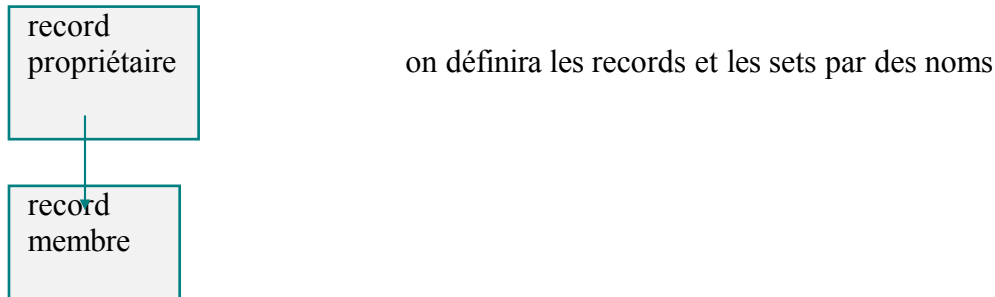
Le SET (le lien, en anglais) : relation entre 2 records, l'un étant déclaré propriétaire (ou "ouvrier") et l'autre membre (ou « member »). Il correspond à la relation binaire, non porteuse de propriété, de type père - fils.

Une occurrence du Set sera composée d'une et une seule occurrence du record propriétaire et d'une ou plusieurs occurrences du record membre.

Exemple : le set « être élève » a pour record propriétaire « classe » et pour record membre ou « élève ». Une occurrence de ce set est : (IG2, Ouattara, Nadège)

Formalisme

Le formalisme représente alors les records dans des rectangles et de sets par les flèches pointant vers le record membre.



2°) Règles d'élaboration du modèle Codasyl (ou règles de passage du MCD au MLD type codasyl)

- Les propriétés

Chaque propriété devient un champ

- Les entités

Chaque entité devient un record et son identifiant sa clé

- Des relations

On distinguera trois cas selon le type de la relation

a- Relation binaire de type père fils

Ce sont donc les relations binaires pour lesquelles les cardinalités d'au moins une des entités sont (0,1) ou (1,1).

La relation est alors transformée en un set. Si elle porte des propriétés celles-ci sont rattachés au record membre.

Exemple : (si un professeur n'enseigne qu'une seule matière)

Selon que la cardinalité de l'entité « fils » est (1,1) ou (0,1) le set.

b- Relation binaire de type autre que père fils

Ce sont des relations binaires pour lesquelles les cardinalités des 2 entités sont (0,n) ou (1,n) ($n > 2$).

La relation donne alors naissance à un record et à 2 sets. Ceux-ci lient les records propriétaires déduits des objets initiaux au record membre ainsi créé.

Si la relation porte des propriétés celles-ci sont attachées au record membre. Sinon, celui-ci est un record vide de propriété simplement destiné à matérialiser la relation. On parlera alors de pseu - record.

Exemple :

C RELATIONS DE DIMENSIONS SUPERIEURE A 2

La règle précédente est généralisée et une relation de dimension n ($n > 3$) créera un record - membre et n sets pointant vers lui.

Là aussi, si la relation est non porteuse de propriétés, le record créé sera un pseu - record.

Exemple :

I/ CAS DES FICHIERS CLASSIQUES

1) Définitions

a- Article d'un fichier : c'est une collection de propriétés se rapportant à un même sujet

b- Un fichier : est un ensemble d'informations regroupées en articles . L'identifiant d'un fichier est une propriété choisie de telle manière qu'à chaque valeur de cette propriété corresponde une et une seule occurrence d'article du fichier.

Exemple :

fichier client

- * num client (identifiant)
- * nom
- * rue
- * ville

2°) Règles de passage du MCD au MLD de type fichiers classiques

- Cas des entités

- * l'entité  1 fichier (logique)
- * l'identifiant  l'identifiant de l'article du fichier
- * les propriétés  les propriétés (rubriques)

- Cas des relations de type « père fils » (avant l'exemple)

exemple M.C.D

MLD « fichiers classiques »

```
fichier commande
* numcde (identifiant)
* date
    * numcli
```

objet « père »	1 fichier « père »
objet « fils »	1 fichier « fils »
identifiant « père »	propriété du fichier « fils »
propriétés de la relation	propriétés du fichier « fils »

* 1 relation $\longrightarrow$ 1 fichier
* identifiant relation $\longrightarrow$ identifiant du fichier

transformer le MCD suivant en MLD de type fichier classique.

The diagram consists of four rows, each representing a mapping from a French term to an English term and its identifier. Each row has a blue arrow pointing from the French term to the English term.

- Row 1: French term "clients" maps to English term "client". The identifier is "* client (identifiant)".
- Row 2: French term "contrat" maps to English term "F - contrat". The identifier is "* contrat (identifiant)".
- Row 3: French term "personnel" maps to English term "F- personnel". The identifier is "* (identifiant)".
- Row 4: French term "qualification" maps to English term "F - qualification". The identifier is "* code qualification (identifiant)".

- * rajout de numclient dans F-contrat (passer contrat)
- * rajout de code qualif dans F -personnel (qualifier)

69

- * relation Requérir se transforme en 1 fichier F - Requérir avec des rubriques : (num cont-code qual (identifiant). N.B jour hommes
- * relation intervenir se transforme en un fichier F - qualif - intervenir avec les rubriques : num cont -codequalif - (identifiant).

III/ LE MODELE RELATIONNEL

Il ne faut pas confondre les relations du modèle relationnel avec les relations du modèle entités / relations à partir duquel le MCD est construit.

Dans le modèle entités / relations, une relation exprime une association entre entités.

Dans le mode relationnel, il s'agit d'associations de propriétés.

1) Les Concepts du Modèle Relationnel

a- Domaine

C'est l'ensemble des valeurs que peut prendre une donnée.

Exemple :

- * La donnée marque sera définie sur le domaine : D1 Renault, Peugeot, Volswagen
- * La donnée couleur sera définie sur le domaine : D2 Bleue, blanche, rouge
- * La donnée salaire sera définie sur le domaine : D3 15 000 à 350 000 F.

b- Relation (appelée couramment Table)

C'est le sous ensemble du produit cartésien de domaines. Ce sous ensemble sera désignée par un nom qui sera le nom de la relation (ou table).

Exemple : considérons le produit cartésien $D1 * D2$:

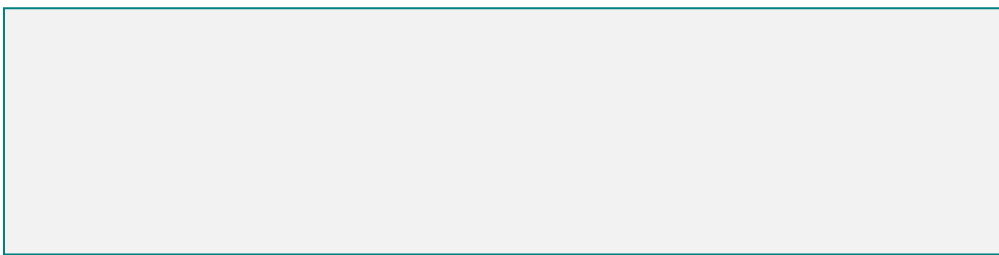
$D1 * D2 =$ Renault, bleue) , (Renault, blanche), (Renault, rouge) (Peugeot, bleue, (Peugeot, blanche) (Peugeot, rouge)

chaque couple est appelé Tuple. Au total, une relation est un ensemble de n - uplets (ou tuples), n étant fini.

c- Attribut

Il représente une colonne d'une relation caractérisé par son nom.

Exemple : produit



numéro	libellé	rayon	prix
001	bière	boisson	300
002	parfum	toilette	1000
003	lait	laitage	350
004	poulet	volaille	1200

La table suivante est constituée de 4 colonnes

d- Clé d'une relation

Une clé est constitué d'un ou plusieurs attributs dont les valeurs définissent de manière unique les triples de la relation. S'il existe plusieurs clés pour une relation, on choisira l'une d'entre elles comme clé primaire, les autres seront appelées clés alternatives (ou clés candidates).

2°) Contraintes référentielles et contraintes d'entité

a) Les contraintes d'entité

Elles sont respectées lorsque :

- * chaque relation possède une clé
- * les différentes valeurs prises par cette clé sont toutes non nulles.

b) La contrainte référentielle

La contrainte référentielle est une contrainte entre 2 relations qui possèdent un attribut (ou plusieurs) commun(s) c'est à dire un attribut ayant la même signification.

Par exemple, la relation Vol (numéro de vol , numéro de pilote) de clé numéro de vol et dont d'un des constituants est l'attribut numéro de pilote qui désigne aussi la clé de la relation pilote (numéro pilote). On a alors une contrainte référentielle entre vol et pilote.

3°) Règles de Passage d'une MCD à un MLD relationnel

3.1 Règles pour les entités du MCD

- * l'entité se transforme en 1 table
- * l'identifiant de l'entité devient la clé primaire de la table
- * les propriétés de l'entité deviennent des attributs de la table.

3.2 Règles pour les relations du MCD (règles algorithmiques)

Les 3 cas de liaisons binaires entre entités sont respectivement :

- a) Cas de la relation père fils (liaison un à plusieurs) ; (maître, esclave)
liaison un (0,1 ou 1,1) à plusieurs (0,n ou 1,n)
FILS PERE

- * l'entité « père » devient la table « père »
- * l'entité « fils » devient la table « fils »

- * l'identifiant de l'entité « père » devient attribut de la table « fils ». Cet attribut est aussi appelé clé étrangère
- * les propriétés de la relation deviennent les attributs de la table « fils »

Exemple :

MCD

Tables MLD relationnel

b) Cas de la relation autre que père fils

liaisons plusieurs $[0,n \text{ ou } 1,n]$ à plusieurs $[0,n \text{ ou } 1,n]$

Les entités deviennent des relations (tables)

« L'association » devient une relation (table R dont la clé est la concaténation des identifiants des deux (2) entités. Si l'association est porteuse de propriété, celles-ci deviennent des attributs de R.

* Liaison un $[1,1]$ à un $[0,1]$

Les deux entités deviennent des relations (des tables). La migration se fait selon la contrainte d'intégrité fonctionnelle. C'est l'entité émettrice de la C.I.F qui reçoit l'identifiant de l'autre.

4°) APPLICATION

Soit le MCD de l'exemple précédent, le passage au MLD de type relationnel s'effectue comme suit :

Les entités se transforment en tables

client $\longrightarrow$ T - client (Numclient,)

contrat	→	T- contrat	(Num contrat,.....)
personnel	→	T - personnel	(Num pers.....)
qualification	→	T - qualification	(codequalif.....)

- Les relations de type « père fils » complètent les tables « fils » déjà créées de la manière suivante :

- La relation passer - contrat complète la table T - contrat par ajout du numéro de client. T - contrat (Num contrat, Numclient,).

- La relation qualifier complète la table T - personnel par ajout du code qualification. T - personnel (numéro emploi, nom code qualification)

Les autres relations se transforment en de nouvelles tables

la relation Requérir	T - Requérir
T - Requérir (num contrat	code qual, nb

la relation Intervenir	T qualif - Int
T - qualif - INT (num contrat - code qualif, num empl)	

MLD relationnel (6 tables)

T - client (numclient,.....)
T - contrat (num contrat,.....)
T - personnel (numpers, nom, code qualif)
T- qualification (code qualif,.....)
T - requérir (num contrat, code qualif, NBJH)
T - qualif - INT (num contrat, code qualif, num pers.....)

4°) FORMES NORMALES

4.1 Concepts

Les 3 premières formes normales ont pour objectif de permettre la décomposition de relations (tables) sans perdre d'informations, à partir de la notion de dépendance fonctionnelle. Plus le degré augmente, moins il y a risques d'anomalies lors des activités de mise à jour des relations. En général, une relation en 3^{ème} forme normale (3 NF) ne présente pas d'anomalies liées aux mises à jour, ni d'anomalies liées aux redondances.

Il est important de noter que la troisième forme normale de BOYCE -Codd -KENT (3NFBC), la quatrième forme normale (4NF) et la cinquième forme normale (5NF) permettent de réduire les anomalies liées à l'utilisation des relations.

4.2 Les Trois Premières Formes Normales

4.2.1 Première forme normale (1NF)

Une relation est en première forme normale si tous ses attributs sont non concaténés et si elle possède au moins une clé. En d'autres termes une relation est en 1NF si tout attribut contient une valeur atomique (simple) c'est à dire aucun attribut n'est décomposable en relation.

Exemple : soit la relation R1 = client (num clien, adresse client) elle n'est pas en 1NF car adresse client peut se décomposer en plusieurs rubriques (attributs) :
rue client, code postclient, télclient etc...

Pour la normaliser il suffit de décomposer

R₂ = client (numcli, nomclient, rueclie, code clie, ville clien telclient.....)

4.2.2 Deuxième forme normale (2NF)

Une relation R est en deuxième forme normale ssi :

1. elle est en première forme normale
2. tout attribut n'appartenant pas à une clé ne dépend pas que d'une partie de cette clé ; (tout attribut autre que les clé ne doit pas dépendre d'une partie seulement de la clé)

exemple : soit la relation R1 = fournisseur (code fournisseur, nom fournisseur, rue fournisseur, ville fournisseur tél fournisseur) prix fournisseur)

Prix de l'article fournisseur (prix fournisseur) dépend uniquement du code article fournisseur (code fournisseur)

Pour la normaliser nous allons en faire 2 relations

R₂ = fournisseur (code fournisseur, nom fournisseur, rue fournisseur, code poste fournisseur, ville fournisseur, téléphone fournisseur)

R₃ = article (code article, prix article, nom fournisseur.

N.B : La 2NF permet d'assurer l'élimination de certaines redondances en garantissant qu'aucun attribut n'est déterminé seulement par une de la clé.

4.2.3 Troisième forme normale (3NF)

La relation R est en 3NF ssi :

- 1 Elle est en 2^{ème} forme normale et
- 2 tout attribut dépend de la clé par une dépendance fonctionnelle élémentaire directe (tout attribut n'appartenant pas à la clé ne dépend pas transitivement de la clé (dans ce cas où la relation possède une seule clé primaire)

Exemple :

$R_1 = \text{voiture (NV, type, puissance, couleur, marque)}$

Règle : le type de voiture détermine la puissance de celle-ci. R_1 n'est pas en 3^{ème} NF, en effet l'attribut non clé type détermine marque et aussi puissance.

En effet, cette relation peut être décomposée en 2 relations pour la normaliser

$R_1 = \text{voiture (NV, type, couleur)}$

$R_2 = \text{modèle (type, marque, puissance)}$

N.B : la 3NF permet d'assurer l'élimination des redondances dues aux dépendances transitives.

La 3^{ème} forme normale est importante. En effet, toute relation a au moins une décomposition en 3NF telle que :

1 la décomposition préserve les D.F

la décomposition est sans perte

Cette décomposition peut ne pas être unique

4.3 Forme normale de Boyce- Codd

Une relation est en BCNF ssi les seules dépendances fonctionnelles élémentaires sont celles dans lesquelles une clé détermine un attribut. Considérons la relation vins (cru, pays, région) dont le clé est le couple (cru pays avec les dépendances fonctionnelles supposées :

région	—————	pays
(cru, pays)	—————→	région

Cette relation est en 3NF car aucun attribut non clé ne dépend d'une partie de la clé ou d'un attribut non clé. Cependant, l'extension de cette relation représentée ci-dessus présente de nombreuses redondances.

Afin d'éliminer ces types de redondances, Boyce et Codd ont introduit la forme normale qui porte leur nom (BCNF). Il a été montré que toute relation a une décomposition en BCNF ne préserve en général pas les DF. Par exemple, la relation vins pourra être décomposée en 2 relations.

Vin cru (cru, région) la D.F (crue, pays) région est perdue
 vin région (région, pays)
 cependant il est possible de recomposer la relation initiale par jointure des 2 relations sur l'attribut région.

4.4 Dépendances multivaluées et quatrième forme normale

4.4.1 Dépendance multivaluées (D.M)

Soit $R(A_1, A_2, \dots, A_n)$ un schéma de relation et X et Y des sous ensemble de $A_1, A_2, \dots, A_n$. On dit que $X \twoheadrightarrow Y$ (X multidétermine Y ou il y a une dépendance multivaluée de Y sur X) si étant donné des valeurs de X il y a un ensemble de valeurs de Y associées et cet ensemble est indépendant des autres attributs $Z = R - X - Y$ de la relation R . Une dépendance multivaluée caractérise donc une indépendance entre 2 ensembles d'attributs (Y, Z) caractérisé par un 3^{ème} X plus formellement on a :
 $(X \twoheadrightarrow Y) \iff \{ (xyz) \text{ et } (xy'z) \text{ appartient } R \implies (xy'z) \text{ et } xyz' \text{ appartient } R \}$ où x, y, z, y', z' désignent des occurrences des attributs x, y, z, y', z' .

Les DF sont des cas particuliers de D.M

Exemple :

1) soit la relation

règle : $R_1 = \text{vol}(\text{num vol}, \text{avion}, \text{pilote})$

On suppose disposer d'un ensemble d'avions et de pilotes tout pilote est conduit à piloter tout avion sur n'importe quel vol. Ainsi, avion et pilote sont indépendants d'où les 2 DM élémentaires :

NV	Avion
NV	pilote

2) Soit la relation

$R_2 = \text{personnes}(\text{n° SS}, \text{prénom enfant}, \text{n° véhicule})$ on a les DM élémentaires suivants.

4.4.2 Quatrième forme normale (4NF)

Une relation est 4NF ssi les seules dépendances multivaluées élémentaires sont celles dans lesquelles une clé détermine un attribut.

Exemple :

Soit la relation $R_1 = \text{étudiant}(\text{num étudiant}, \text{cours}, \text{sport})$ n'est pas en 4NF : la clé est l'ensemble des attributs et il existe des D.M élémentaires.

Num étudiant	cours
--------------	-------

numétudiant sport entre des attributs participant à la clé . La relation R_1 peut-être décomposée en 2 relations

$R_2 = \text{étudiant (numétudiant, cours)}$

$R_3 = \text{étudiant (numétudiant, sport)}$

N.B : la 4NF est une généralisation de la forme normale de Boyce Codd afin de décomposer les relations ayant des D.M élémentaires. Une relation en 4NF est en forme normale de Boyce Codd et donc en 3NF.

4.5 Dépendances de jointures et cinquième forme normale

4.5.1 Dépendance de jointures

Soit la relation ci-dessous

Cette relation modélise des vins bus par des buveurs, d'un cru donné et commandés à un producteur produisant ce cru. Cette relation est 4NF. En effet, il n'existe pas de dépendance multivaluée

Buveur cru est faux car par exemple le triple (Ronald, volnaychaude) n'existe pas.

Cru producteur est faux car par exemple la triple Jeffrey Chablis Claude) n'existe pas

Producteur buveur est aussi faux car par exemple le triple (Jeffrey, volnay nicolas) n'existe pas

La relation ci-dessus présente des redondances : on apprend 2 fois que Ronald boit du chablis et que Nicolas produit du chablis. Elle n'est cependant pas décomposable en 2 relations.

A titre d'exemple de relation décomposable en trois (3) relations en nom décomposable en 2, supposons que la relation vin obéisse à la contrainte d'intégrité suivante :

« tout buveur ayant vu et ayant commandé à un producteur produisant ce vin, a aussi commandé ce cru à ce producteur » . Cela s'écrit formellement :

$(b, c) \text{ appartient } R_1 \text{ et } (b, p) \text{ appartient } R_2 \implies (c, p) \in R_3 \quad (b, c, p) \in R$

Dans ce cas, R sera la jointure de R_1 R_2 et R_3 $R = R_1 \Join R_2 \Join R_3$

Définition

soit $R(A_1 A_2 \dots A_n)$ un schéma de relation et $X_1, X_2, \dots, X_m$ des sous ensembles de $\{A_1, A_2, \dots, A_n\}$. On dit qu'il existe une dépendance de jointure $*\{X_1, X_2, \dots, X_m\}$ si R est la jointure de ses projections sur $X_1, X_2, \dots, X_m$ c'est à dire si :

Exemple :

La relation vins obéit à la dépendance de jointure $*\{\text{buveur cru}, \text{buveur producteur cru producteur}\}$. Elle est donc décomposable en 3 relation R_1, R_2, R_3 . Les dépendances multivaluées sont des cas particuliers de dépendances de jointure.

En effet, une relation $R(X, Y, Z)$ vérifiant la dépendance multivaluée $X \twoheadrightarrow Y$ (et donc $X \twoheadrightarrow Z$) satisfait la dépendance de jointure : $*\{XY, XZ\}$.

4.5.2 Cinquième forme normale

Une relation R est en 5NF ssi toute dépendance de jointure est impliquée par des clés candidates de R .

4.6 L'optimisation du modèle logique

Pour bien le préparer à s'adapter au système de gestion de bases de données (SGBF) qui le supportera, le modèle logique des données doit être optimisé. L'optimisation, consiste à minimiser l'espace de stockage et les temps d'activité. Ces deux (2) objectifs sont contradictoires car on ne peut pas à la fois réduire l'espace de stockage et le temps de traitement, on cherche à trouver un compromis acceptable.

4.6.1 Valorisation des volumes de stockage

On cherche à valoriser les volumes (au sens de tailles) d'informations stockées :

On applique

$$T_1 : = N \times L$$

$$T_2 : N \times P$$

$$T = T_1 + T_2 \text{ avec}$$

T_1 : taille du fichier hors index ou pointeurs

L : longueur logique d'un enregistrement

T_2 : taille due aux index ou jointeurs

P : taille totale de tous les index ou pointeurs pour un enregistrement
N : nombre maximum d'enregistrements
T : taille totale

4.6.2 Trois cas d'optimisation du modèle relationnel

a) Création nécessaire de redondance

exemple : soient les tables suivantes

respectivement à partir des relations :

. Agence (code agence, nom agence, rue agence, ville agence) et
Intérimaire (code intérimaire, code agence, nom intérimaire, rue intérimaire, ville intérimaire)

supposons que lors de la manipulation de la base de données, l'évocation d'une intérimaire implique toujours la connaissance du nom de l'intérimaire, d'où la nouvelle relation :

Intérimaire (code intérimaire, code agence, nom intérimaire, rue intérimaire, ville intérimaire, nom agence)

Elle fera partie du MLD optimisé. Si une telle optimisation autorise un gain en temps d'accès, elle entraîne néanmoins par la redondance de l'attribut un encombrement plus important de la base de données.

L'avantage d'une optimisation avec création de la redondance est qu'elle permet de limiter et simplifier certains accès et d'éviter l'opération très coûteuse de jointures. La relation intérimaire du MLD optimisé n'est plus en 3NF.

b) Création d'index

Un index est un attribut ou ensemble d'attribut par lequel (lesquels) on peut accéder aux enregistrements.

Soit une partie du MCD suivant

On en tire les relations suivantes :

PV (n°, PV, date PV, n° infraction)

Infraction (n° infraction, date infraction, lieu infraction).

En admettant que l'évocation d'une infraction à partir de laquelle le PV a été établi fasse toujours référence à tous les attributs du PV, la relation infraction pourrait prendre la forme suivante :

Infraction (n° infraction, n° PV, date infraction, lieu infraction) ou le n° PV est l'identifiant de PV donc le second index.

c) Suppression de relations

Soit l'exemple : cas gestion hôtelière

l'analyse des traitements montre que tous les accès à la relation classe se font toujours par l'intermédiaire de la relation hôtel ; On peut alors pour ainsi dire identifier les hôtels par un code combinant le n° d'ordre de l'hôtel et le nombre d'étoile cet hôtel. La relation classe viendrait à disparaître. D'une manière générale, il est judicieux de supprimer une relation à laquelle on accède toujours indirectement en faisant migrer les attributs dans la relation par laquelle les accès se font directement.

4.7 Algèbre relationnelle et requêtes

1) Les opérateurs de l'algèbre relationnelle

a- Union

Soient 2 relations R et S d'attributs identiques et définis sur les mêmes domaines l'union de R et S correspond à la relation $R \cup S$ qui contient l'ensemble des n - uplets de R et de S.

Exemple :

Client (code, nom) fournisseur (code, nom)

L'union tiers = client $\cup$ fournisseur est l'ensemble des n - uples de client et de fournisseur

client	fournisseur	client $\cup$ Fournisseur
A01 Dupont	AO1 Dupont	AO1 Dupont
A25 Durand	A32 Dubois	A25 Durand
		A32 Dubois

b- Intersection

L'intersection $R = R_1 \cap R_2$ de 2 relations R_1 et R_2 définies sur les mêmes domaines est constituée des n - uples communs à R_1 et R_2 .

C- La différence

La différence $R = R_1 - R_2$ de 2 relations R_1 et R_2 sur les mêmes domaines est constituée des n-uples de R_1 qui ne figurent pas dans R_2

Exemple :

d- La projection

La projection d'une relation sur certains constituants consiste à ne retenir que ces constituants, tout en éliminant les n-uples dupliqués.

Notation :

Projection nom de la relation (liste des constituants retenus)

Exemple : ouvrage (auteur, titre, collection)

Projection ouvrage (auteur, titre)

e- Sélection (ou restriction)

La section consiste à construire une relation en sélectionnant les n-uples de la relation manipulée qui vérifient une certaine propriété.

On notera : $R_2 = \text{select. } R (P)$

Où R_1 est la relation résultat, R la relation manipulée et P une proposition logique faisant intervenir des constituants de R , des opérateurs relationnels : $<$, $<=$, $>=$, $>$, $=$, $<>$ et éventuellement des opérateurs booléens et (AND), ou (OR), non (NOT).

Exemple : $R = \text{client (code, nom, CAM, CAM-1)}$

$R_1 = \text{select. Client (CAM > CAM-1)}$

$R_2 = \text{Select.client (code < »B13 » et CAM-1 > 5000) :$

f- Composition (ou join ou jointure)

Elle fournit une relation résultat R à partir de 2 relations initiales R_1 & R_2 en concaténant 2 à 2 mes , -uples de R_1 et R_2 qui vérifient entre eux une proposition logique.

On note :

$R = R_1 \times R_2 (P)$

Où P représente une préposition logique portant sur 2 constituants C_1 de R_1 & C_2 de R_2 définis sur le même domaine, au moyen des opérateurs relationnels de comparaison.

Exemple :

$R_1 = \text{produit (réf, désign, pu)}$

La notation produit. Réf signifie réf. dans la relation produit de même client. Réf ; dans la relation client.

g- Division

Soit la relation appro (fournisseur, produit) indiquant quels sont les fournisseurs qui approvisionnent les produits :

et soit la relation liste (produit) des produits à réapprovisionner

La relation $R = \text{appro/ Liste}$ donne l'ensemble des fournisseurs qui approvisionnent au moins les produits de la liste.

La division $R = R_1/R_2$ d'une relation binaire $R_1 (A, B)$ par une relation unaire R_2/B est une relation unaire $R(A)$ telle que tout 1-uple () de R concaténé à tout 1 uple de $R_2 (B$ redonne un 2-uple de R_1 dans notre exemple, la division $R = \text{appro/liste}$ de la relation binaire appro (fournisseur, produit) par la relation unaire liste (produit est une relation unaire R (fournisseur) telle que tout 1-uple ($F0_2$) ou $F0_4$ de R concaténé à tout 1-uple de liste (produit) c'est à dire à (X_{18}) ou à XO_2 , redonne un 2-uple de appro c'est à dire ($FO_2, x X_{18}$) ($F0_2, X0_2$), ($F0_4, X_{18}$) ou ($F0_4, X0_2$).

2° LES PROPRIETES

Les requêtes sont les traitements (manipulation de données ou interrogation) que l'on peut effectuer sur les bases relationnelles.

Exemple : une base de données prenant en compte les situations quotidiennes est constitué des relations :

Pilote (nopil, nompil, prépil, ruepil, villepil,

Avion (noav, nomav, capav)

Vol (novo, nopil, noav, ville)

Requête on veut disposer de la liste des pilotes (nom et prénoms) partant entre 6 heures et 10 heures du matin d' les avions pouvant transporter plus de 300 personnes.

1^{ère} solution

$R_0 = \text{select. avion (capacité} > 300)$

$R_1 = \text{select. Vol (heure départ} > 6\text{h et heure arrivée} < 10\text{h})$

$R_2 = R_0 \times R_1 \{n^\circ \text{ AV}\}$

$R_3 = \text{proj (nopil [R}_3])$

$R_4 = R_3 \times \text{pilote } \{n^\circ \text{ pil}\}$

Résultat = proj(nom, prénom) [R4]

2^{ème} solution

$R_0 = \text{Avion} \times \text{vol } \{\text{NoAV}\}$

$R_1 = \text{select. R}_0 \text{ (capacité} > 300 \text{ et heure départ} > 6\text{h et heure d'arrivée} < 10\text{h})$

$R_2 = \text{proj (n}^\circ \text{ pil) [R}_1]$

$R_3 = R_1 \times \text{pilote } \{n^\circ \text{ pil}\}$

Résultat = proj (nom, prénom) [R3]

Exemple 2

On suppose que la base contient les relations

Client (code, nom, CA)

CA = compte achat

Produit (réf, désign, p.u)

licom = livraison commande

Commande (n° commande, code, date)

Livraison commande (n° commande, réf, quantité)

Requête 1 : quels sont les noms des clients dont le Compte achat est supérieur à 5 000 F ?

Solution

$R = \text{select ; client (CA} \geq 5000)$

Résultat =proj R (nom)

Requête 2 : Quels sont les désignations et les quantités commandées par le client AO1 depuis le 15 janvier 1986 ?

Solution

$R_1 = \text{select commande (code} = \text{« AO1 » et date} \geq \text{« 150186 »)}$

$R_2 = R_1 * \text{licom (n}^\circ \text{ commande)}$

$R_3 = R_2 * \text{produit (réf)}$

Résultat = proj R_3 (quantité, désignation)

CHAPITRE 6

LE MODELE PHYSIQUE DES DONNEES ET LE MODELE OPERATIONNEL DES TRAITEMENTS

I/ LE MODELE PHYSIQUE DES DONNEES

1) Généralités

Le modèle physique des données (MPD) est le dernier modèle réalisé avant la programmation. Il prend en compte les ressources physiques (SGBD, matériel, support etc...) Il va permettre d'implanter en machine l'ensemble des données de MLD.

En effet, la description d'un MPD est étroitement liée aux choix techniques informatiques concernant le système de gestion des données. Il existe Principalement 3 types de solution technique :

- Utilisation d'un système de gestion de base de données de type Codasyl,
Exemple : IDS (de Bull), Total (de cincom systems Inc), Clio (syseca) mais aussi IMS - DM de IBM qui est un système type hiérarchique tendant à disparaître.
- -Utilisation d'un système de gestion de base de données de type relationnel tel que Ingres (Interactive Graphics And Retrieodl System) de relational software), oracle (de oracle cop) et DB2 (d'IBM)
- Utilisation d'un système de gestion de fichiers classiques avec comme principale méthode d'accès le séquentiel indexé.

De plus, l'environnement technique de développement influera aussi largement dans la description du niveau physique.

Exemple d'outils influant sur l'environnement de développement : outils de génération d'écran.

La description du MPD se fera donc dans le langage du système de gestion de données correspondant à la solution choisie. La figure ci-dessous montre les principales caractéristiques de la gestion de physique de relations pour les 3 types de solution évoqués ci-dessus.

2) Exemple de Modèle Physique des Données (type relationnel)

Soit le M.L.D suivant issu d'un M.C.D

T. domiciliation (n° dc, n°cptectlt , code
T client (numclt, nomclt, adr-clt)
T. produit (code prod, lib-prod)
T. devise (code-dev, lib-dev)
T. T. compte n°cptc-clt, numclt , nom-cpte, adr-cpte...)

Modèle physique issu du M.L.D

a- description du fichier compte

nom :	compte	clé, n° cpte -cli	
organisation	séquentielle indexée		
support	disque		
nature fichier	permanent		
longueur enregistrement	270		
rubrique	libellé	type	longueur
n° cpte-clt	numéro de cpte client	AN	12
nom cpte	nom du cpte	AN	20
adresse du cpte	adresse du cpte	AN	32

b- Fichier devise

nom :	devise	clé : code -dév	
organisation :	séquentielle		
support :	disque		
nature fichier :	permanent		
longueur enregistrement :	28		
rubrique	libellé	type	longueur
code-dev	code de la devise	AN	3
lib-dev	libellé de la devise	AN	25
AN :	Alphanumérique		

C- Fichier produit

nom :	produit	clé : code-prod	
organisation :	séquentielle indexée		
support	disque		
nature fichier :	permanent		
longueur enregistrement	9		
rubrique	libellé	type	longueur
code-prod	code du produit	AN	3
lib-prod	libellé du produit	A	6
AN : alphabétique			
A :alphabétique			

3) Démarche générale d'élaboration d'un M.P.D optimisé :

Un M.P.D « optimisé » s'obtient selon le schéma suivant :

M.L.D optimisé M.P.D « brut » M.P.D « optimisé »

et en recherchant le moins mauvais des compromis entre :

- le minimum d'accès physique pour un accès logique
- le minimum d'accès physique pour les traitements les plus fréquents
- la plus grande indépendance du stockage physique des données par rapport aux traitements.

Le respect de cet objectif est vital dans le cas de M.PD comportant des volumes importants de données (à partir de plusieurs dizaines de millions de caractères).

La démarche eut se décomposer en trois (3) étapes :

Etape 1 : étude de tous les paramètres ayant une influence sur les performances

du MPD et constitution d'une maquette ou d'un prototype

Etape 2 : simulation sur des volumes et des traitements représentatifs

Etape 3 : évaluation et appréciation

Si les résultats ne sont pas satisfaisants, il faudra agir sur paramètres susceptibles d'être concernés et retourner à l'étape 1. Dans le cas contraire, les résultats obtenus auront été jugés « satisfaisants » et l'on pourra procéder aux opérations de chargement des données.

II/ LE MODELE OPERATIONNEL DES TRAITEMENTS (M. op T ou MPT)

1) INTRODUCTION

Le modèle opérationnel des traitements (M op T) a pour but de décrire l'architecture des logiciels qui devront être réalisés à partir du modèle organisationnel des traitements.

MOT a permis de décrire :

- Les phases « temps réel » avec la description :
 - * des écrans et de leur enchaînement
 - * des traitements associés à chaque écran
- Des phases « temps différé » avec la description :
 - * des états
 - * des traitements regroupés en tâche

Le M.P.T devra permettre de décrire l'architecture des unités de traitement correspondant aux phases décrites dans le M.O.T. Ces unités de traitement seront selon le cas :

- des transactions (pour les phases « temps réel »)
- des programmes « batch » (pour les phases « temps différé »)

Le M.O.T sera fait selon ces 2 cas.

3) Structuration des transactions

L'environnement de développement interviendra dans la description et la structuration des transactions. Le générateur de grilles d'écran et le moniteur de gestion des transactions seront 2 éléments déterminant dans la production des transactions.

Démarche d'élaboration des transactions :

Règle 1

A chaque écran correspondra une transaction

Règle 2

Chaque transaction sera structurée et représentée de la manière suivante (normalisation I.S.O)

Règle 3 :

- * établir des recommandations ergonomiques pour l’affichage écran et la saisie des données
- * définir des normes d’affichage et de saisie.

Exemple : normaliser l’utilisation des touches de fonction

Règle 4

standardiser la description des traitements en distinguant 4 types de traitement :

- * les contrôles (avec ou sans saisie de données)
- * les mises à jour des données stockées
- * les calculs
- * les éditions

Autres règles ou recommandations :

- prévoir la réalisation de modules de service accessibles par toutes les transactions.
Exemple : * module de contrôle
 - * module de contrôle spécifique
 - * module de transformation de code
- Bien gérer la confidentialité d’accès aux informations : gestion des droits par utilisateur, par information, par type de traitement etc...
- penser à la mise en œuvre de procédure de sécurité journalisation des mouvements de mise en jour en vue d’une reprise ultérieure (suite à un incident)
- gérer l’enchaînement des transactions

3) Structuration des programmes « Batch »

- règles de découpage des programmes « batch »

* découpage selon la périodicité des traitements

Exemple : distinguer les programmes journaliers des programmes mensuels.

- Découpage selon le type de programme :

4 types de programme peuvent être distingués :

- * programme de contrôle
- * programme de calcul

- * programme de mise à jour
- * programme d'édition

- constitution des enchaînements de programmes
des chaînes de traitement seront constituées. Ainsi une phase « temps différé »
du MOT pourra se transformer en une chaîne de programme « batch »
des fichiers intermédiaires devront être définis lorsque 2 programmes d'une
même chaîne devront échanger des données.

4 Exemple de modèle opérationnel des traitements (ou unité de traitement)