

Université Abdelmalek Essaâdi  
Ecole Nationale de Commerce et de Gestion  
www.encgt.ma

## LE MODELE CONCEPTUEL DES DONNEES (MCD)

M. I. EL KHALKHALI

### 1. Introduction

L'importance du système d'information dans la vie des entreprises n'est à ce jour plus à démontrer. Il est ainsi devenu évident pour ces dernières, que leurs performances, voire leur survie dans un contexte de concurrence croissante, dépend du bon fonctionnement de leur système d'information.

De cette prise de conscience est né le besoin d'élaborer des méthodes permettant de concevoir correctement un système d'information et de mettre en place un modèle sur lequel s'appuyer.

La modélisation consiste à créer une représentation virtuelle d'une réalité de telle façon à faire ressortir les points auxquels on s'intéresse.

Ce type de méthode est appelé analyse. Il existe plusieurs méthodes d'analyse, la méthode la plus utilisée étant la méthode **MERISE** (*Méthode d'Étude et de Réalisation Informatique pour les Systèmes d'Entreprise*).

### 2. Historique de la méthode Merise

La méthode d'analyse Merise a été créée en 1979 au Centre Technique Informatique du ministère français de l'industrie. L'objectif était de doter les administrations et les entreprises publiques d'une méthodologie rigoureuse tout en intégrant les aspects nouveaux pour l'époque: informatique répartie, bases de données.

Merise a réellement été introduite dans les entreprises entre 1983 et 1985.

Depuis, MERISE a connu des évolutions en fonction des avancées technologiques avec dernièrement **MERISE 2** tournée vers l'objet. MERISE reste encore une méthode très utilisée même si **UML** et **OMT** sont en train d'inverser la tendance.

### **3. Les niveaux d'abstractions**

La méthode *Merise* propose trois niveaux de représentation d'un système d'information :

- a) Le niveau conceptuel
- b) Le niveau organisationnel ou logique
- c) Le niveau physique

#### **3.1) Le niveau conceptuel**

Ce niveau correspond aux préoccupations du gestionnaire et de l'utilisateur. Il consiste à répondre à la question **QUOI?**. Le but est de comprendre la nature du problème.

Ce niveau est totalement indépendant de toute considération technologique, quelle soit logicielle ou matérielle.

Les deux modèles utilisés sont le *Modèle Conceptuel des Données (MCD)* et le *Modèle Conceptuel des Traitements (MCT)*.

#### **3.2) Le niveau Organisationnel ou Logique**

Ce niveau correspond aussi aux préoccupations du gestionnaire et de l'utilisateur. Il consiste à répondre aux questions : ***QUI?***, ***OU?*** et ***QUAND ?***.

Ce niveau décrit les contraintes *organisationnelles*, *spatiales* et *temporelles*.

Les modèles étudiés à ce niveau sont le *Modèle Logiques des Données (MLD)* et le *Modèle Organisationnel des Traitements (MOT)*.

#### **3.3) Le niveau physique**

Ce dernier niveau correspond aux préoccupations de l'informaticien. Il permet de répondre à la question ***COMMENT ?***.

C'est une représentation des moyens informatiques qui vont effectivement être mis en œuvre pour gérer les données ou activer les traitements. Le niveau physique apporte les solutions techniques. Les modèles étudiés à ce niveau sont le *Modèle Physique des Données (MPD)* et le *Modèle Physique des Traitements (MPT)*.

### **Résumé**

Le tableau suivant résume les modèles que nous allons aborder tout au long de ce cours :

NIVEAU	DONNEES	TRAITEMENTS	CHOIX PRIS EN COMPTE
<b>Conceptuel</b>	Modèle Conceptuel des Données (MCD)	Modèle Conceptuel des Traitements (MCT)	Choix de gestion Quoi ?
<b>Organisationnel ou logique</b>	Modèle Logique des Données (MLD)	Modèle Organisationnel des Traitements (MOT)	Choix d'organisation Qui ?, Où ?, Quand ?
<b>Physique</b>	Modèle Physique des Données (MPD)	Modèle Physique des traitements (MPT)	Choix techniques Comment ?

#### **4) Le Modèle Conceptuel des Données ou le modèle Entité-Association**

Le Modèle Conceptuel des Données (MCD) est une représentation graphique du système d'information. Il a pour but de décrire de façon formelle les données qui seront utilisées par le système d'information. Cet aspect recouvre les mots qui décrivent le système ainsi que les liens existants entre ces mots. Le formalisme utilisé pour décrire un MCD est celui du *modèle entité-association*.

La représentation de ce formalisme s'appuie sur 6 concepts de base :

##### **4.1) L'entité ou l'individu type**

Une entité est la représentation d'un élément matériel ou immatériel ayant un rôle dans le système que l'on désire décrire. C'est un regroupement d'informations bien pensé.

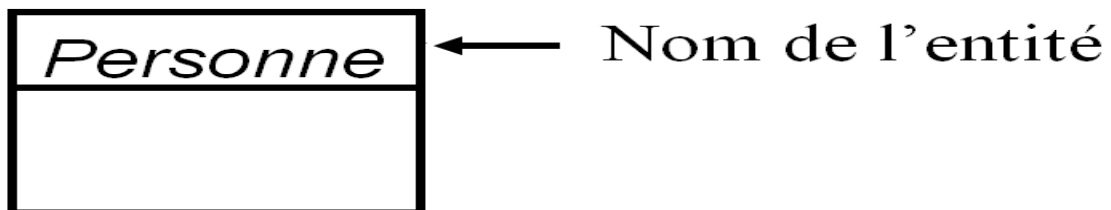
Par exemple, si l'on considère l'entité "Personne" les informations communes aux personnes peuvent être :

- le nom
- le prénom
- la date de naissance
- le lieu de naissance
- le sexe

- l'adresse
- etc...

On schématise une entité par un rectangle.

Exemple :

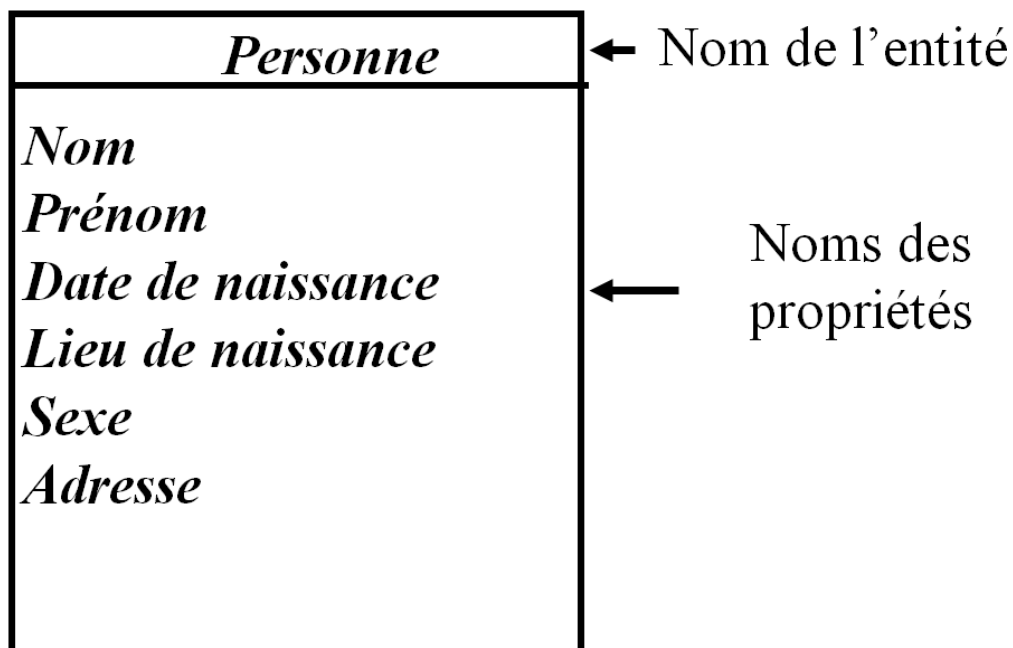


#### 4.2) Propriétés ou Attributs

Les *propriétés* ou les *attributs* sont les caractéristiques décrivant les entités et doivent être représentés comme une liste de mots, la plus simple possible, dans le cadre de l'entité correspondante. On devra préciser le type des données attendues pour chaque attribut ou propriété.

Les propriétés de l'entité s'indiquent dans le rectangle du bas, sous le nom de l'entité :

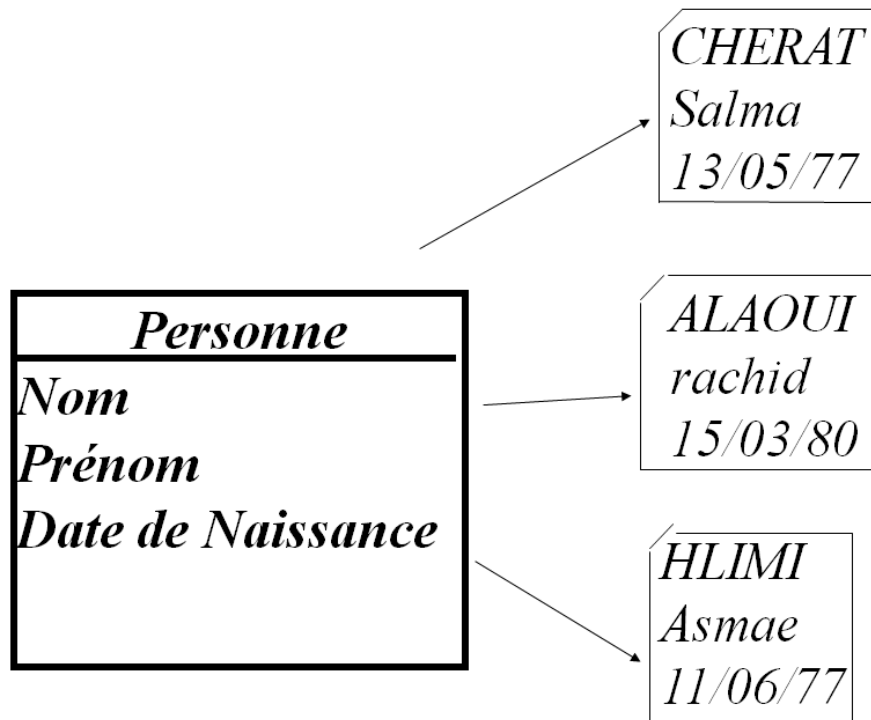
Exemple:



### 4.3) Occurrence

L'occurrence d'une propriété ou d'un attribut est l'une des valeurs que peut prendre cette propriété.

Le tableau suivant présente des exemples d'occurrences de l'entité Personne.



### 4.4) Identifiant

Appelé aussi *clef primaire*, est un attribut (ou un ensemble d'attributs) qui permet (tent) d'identifier de façon unique une occurrence de l'entité.

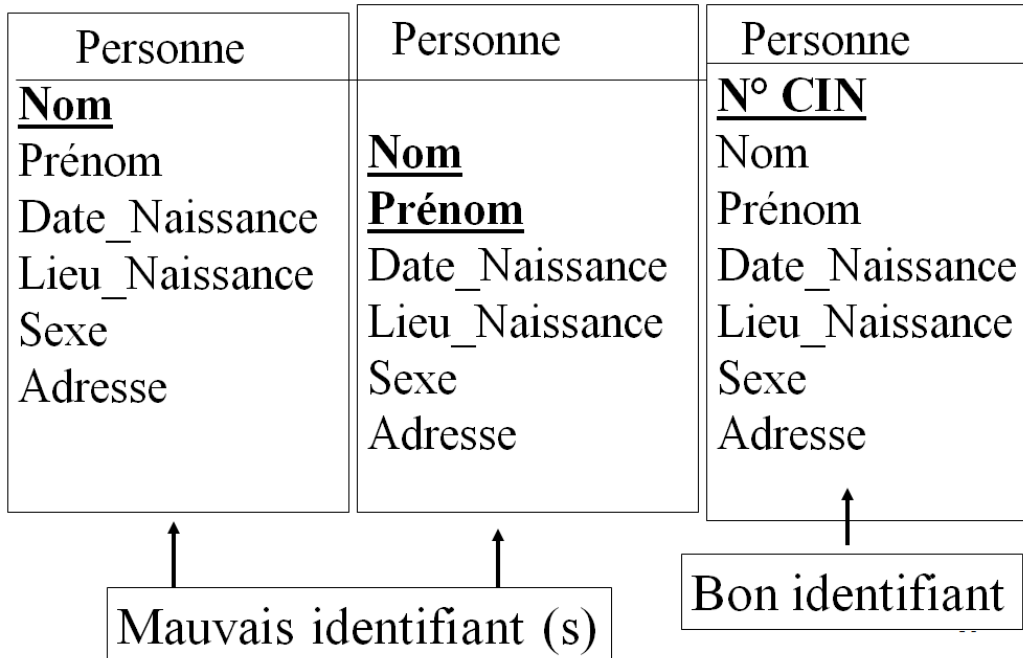
Pour repérer l'identifiant dans la représentation graphique d'une entité, on le souligne. Si c'est une clef composée, alors plusieurs propriétés seront soulignées.

Par exemple l'identifiant de l'entité "Personne" pourrait être le nom. Mais comme le cas d'homonymie est assez fréquent, alors cet attribut constituerait une mauvaise clef.

En revanche, il n'est pas impossible que la clef d'une entité soit composée de plusieurs attributs. Par exemple, la clef de l'entité "Personne" pourrait être le nom et le prénom.

Cependant il n'est toujours pas impossible d'avoir deux personnes dont le nom et le prénom soient identiques...

### Exemples:

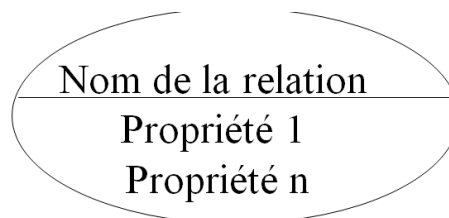


L'identifiant idéal pour l'entité personne serait par exemple le n° de la CIN.

#### 4.5) Relations

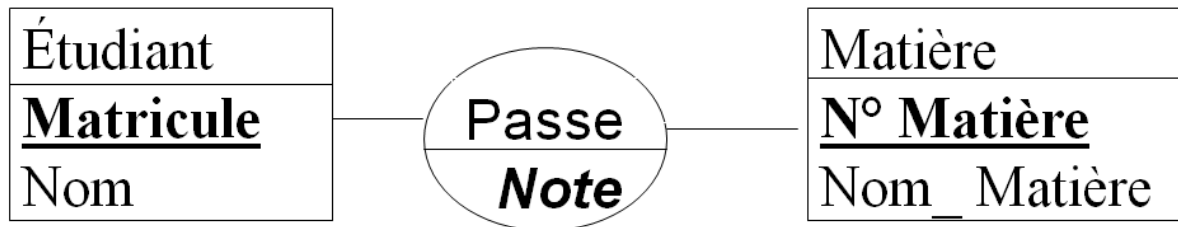
Une relation (appelée aussi parfois *association*) représente les liens sémantiques qui peuvent exister entre plusieurs entités.

Les relations sont représentées par des ellipses dont l'intitulé décrit le type de relation qui relie les entités (généralement un verbe). Une relation peut avoir des propriétés : on la dit porteuse



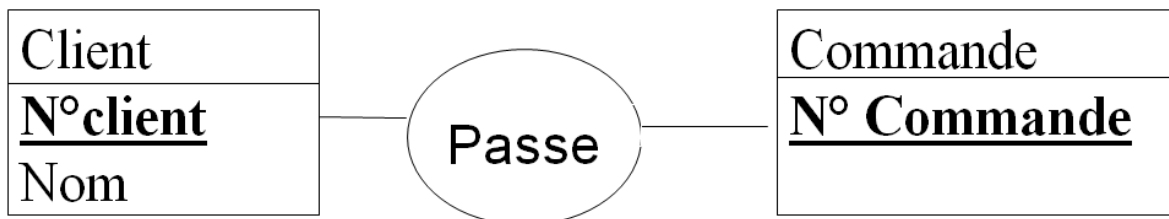
### Exemple d'une relation porteuse

- Un étudiant a obtenu une telle note dans une telle matière donne naissance à la relation « passer » dont la propriété est « note » entre l'entité étudiant et l'entité matière.



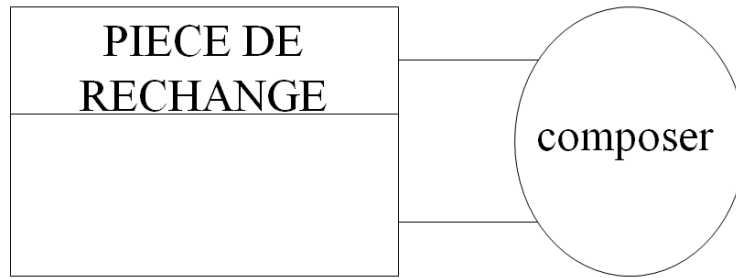
### Exemple d'une relation non porteuse

Les clients passent des commandes donne naissance à la relation « passer » entre l'entité client et l'entité commande.



- Une relation entre deux entités est une relation binaire.
- Une relation entre trois entités est une relation ternaire.
- Une relation entre n entité est une relation n-aire.
- Une relation réflexive est une relation de l'entité sur elle même.
- **Exemple d'une relation réflexive :**

Une pièce de rechange peut entrer dans la composition d'autres pièces de rechange donnera naissance à la relation « composer »



#### 4.6) Cardinalités

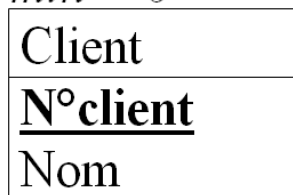
Elles permettent de quantifier le nombre d'occurrences d'une entité qui participent à une relation. Elle s'exprime par deux nombres :

- **Cardinalité minimale ( 0 ou 1 ):** Représente le nombre de fois minimum qu'une occurrence d'un objet participe aux occurrences de la relation.
- **Cardinalité maximale ( 1 ou n ):** Représente le nombre maximum de fois qu'une occurrence de l'objet participe aux occurrences de la relation.

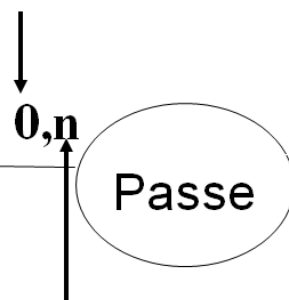
Ces deux nombres se positionnent à côté de l'entité concernée

#### **Exemple:**

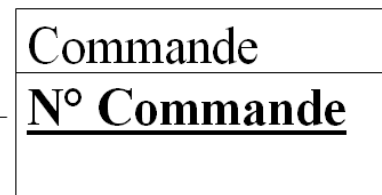
*Un client peut ne pas avoir passé de commandes : cardinalité min = 0*



*Un client peut passer n commandes : cardinalité max. = n*



*Une commande est toujours passée par un client : cardinalité min. = 1*

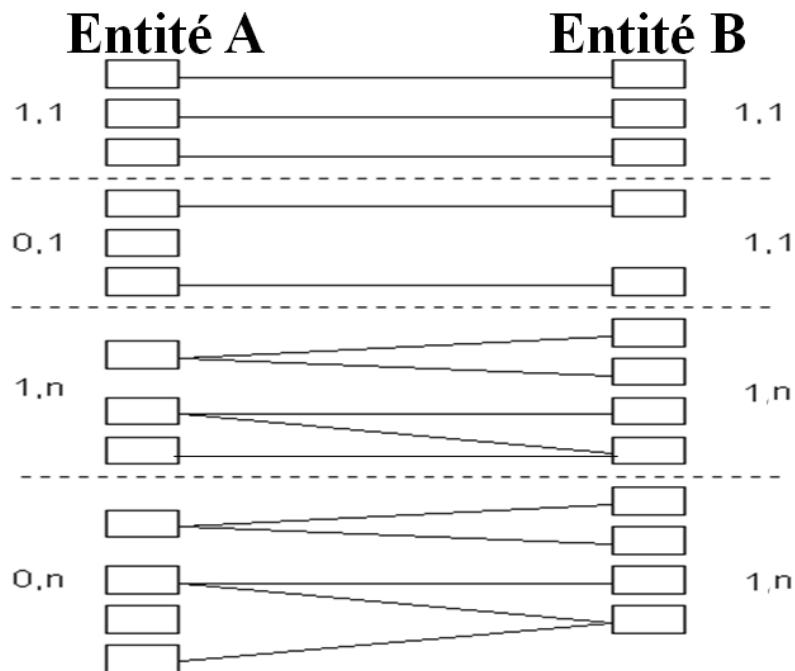


*Une même commande est passée par au plus un seul client : cardinalité max. = 1*



- Utilisation d'un tableau de valeurs :

Pour aider à découvrir les cardinalités, on peut faire appel à ce tableau de valeur:



Le premier cas en haut constitue un cas particulier qui n'est pas valide. En effet dans ce cas chaque occurrence de l'entité **A** correspond une occurrence de l'entité **B** et inversement. Cette bijection exige que les propriétés des deux entités soient regroupées en une seule entité.

#### 4.7) Construction d'un MCD

La démarche à suivre pour la construction d'un MCD est la suivante :

- Recherche des entités*
- Recherche des propriétés à gérer (dictionnaire des données)*
- Recherche des relations entre entités*
- Recherche des cardinalités (règles de gestion)*
- Vérification et validation du modèle conceptuel des données*

### a) Recherche des entités

On commence par identifier les différentes entités du système étudié. Le nom de l'entité doit signifier un critère d'appartenance permettant d'affirmer qu'un acteur du système à étudier peut ou ne peut pas appartenir à cette entité.

Ainsi, un client X sera une occurrence de l'entité client et non de l'entité commande.

### b) Recherche des propriétés à gérer (dictionnaire des données)

Il faudrait lister toutes les propriétés utiles au système étudié. Chacune de ces propriétés sera définie par :

- Un *nom*,
- Une *description* (pour éviter toute ambiguïté sur la compréhension de la donnée),
- Le *type de données* (numérique, texte, booléen, date, etc).

Les entités ainsi que leurs propriétés respectives seront représentées dans le *dictionnaire des données*. Le dictionnaire des données est représenté sous forme de tableau. Rappelons que l'identifiant de chaque entité doit être précisé.

#### Exemple :

<b>Nom entité</b>	<b>Nom Propriété</b>	<b>Description</b>	<b>Type</b>
Client	<u>numcli</u>	Numéro du client	Numérique
	nomcli	Nom du client	Texte
	adcli	Adresse du client	Texte
	villecli	Ville du client	Texte
Commande	<u>numcom</u>	Numéro de la commande	Numérique
	datecom	Date de la commande	Date

### **c) Recherche des associations entre entités**

Il s'agit d'écrire des phrases en français décrivant le modèle. Ces phrases permettront d'établir des liens entre les entités.

Une relation est caractérisée par :

- Son *nom* (en général un verbe)
- Sa *dimension* (nombre d'entités qu'elle unit)
- Sa *collection* (noms des entités qu'elle relie)
- Ses *cardinalités*

### **c) Recherche des cardinalités (règles de gestion)**

Pour définir les associations et les cardinalités, il faut connaître les *règles de gestion*. Ainsi, dans l'exemple étudié, les cardinalités s'expliquent par les règles de gestion suivante :

- R1 : un client peut ne passer aucune commande.
- R2 : un client peut passer autant de commandes qu'il veut.
- R3 : il suffit d'un client pour qu'une commande
- R4 : une commande, n'est passée que par 1 et un seul client.

Les règles de gestion ne sont pas toujours explicites et souvent même mal définies. Il convient donc, dans la construction du modèle, de les expliciter avec clarté.

### **e) Vérification et validation du modèle conceptuel des données**

Le modèle conceptuel des données doit suivre des règles de bases pour être correct.

#### **Règles sur les entités:**

- Toute entité doit posséder un identifiant et un seul.
- Cet identifiant peut être une propriété ou un groupe de propriétés.
- Cette propriété ou ce groupe de propriétés doit répondre à la règle suivant laquelle la valeur de l'identifiant doit être différente à chaque occurrence de l'entité. Elle ne doit

jamais être « nulle » (non renseignée) et toujours être stable ( non sujette à des mises à jour).

**Règles sur les propriétés:**

- Toute propriété ne peut être présente qu'une seule fois sur le MCD.
- Aucune propriété calculée ne doit apparaître sur le MCD. (ex : Montant total de la commande).



## LE MODELE LOGIQUE DES DONNEES (MLD)

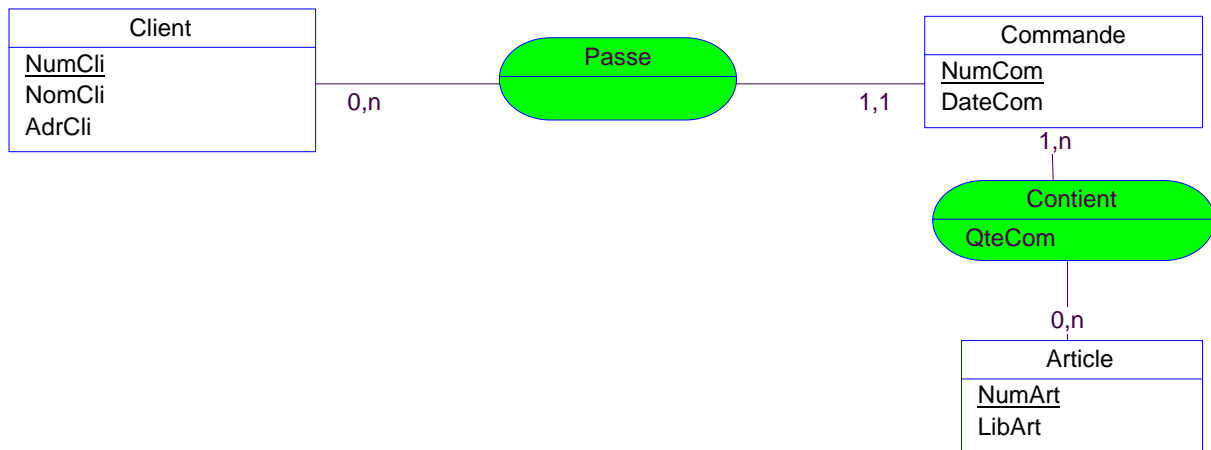
M. I.ELKHALHALI

La description conceptuelle a permis de représenter le plus fidèlement possible les réalités de l'univers à informatiser. Mais cette représentation ne peut pas être directement manipulée et acceptée par un système informatique. Il est donc nécessaire de passer du niveau conceptuel à un niveau plus proche des capacités des systèmes informatiques. Ce niveau, appelé niveau logique,

Le Modèle Conceptuel des Données est, comme on l'a vu, est exprimé dans le formalisme entité-association. Le Modèle Logique des Données, quant à lui, s'exprime usuellement sous la forme **d'un modèle relationnel**.

Le passage d'un Modèle Conceptuel de Données (MCD) au Modèle Logique des Données (MLD) se fait selon trois règles de passage.

Nous allons appliquer ces trois règles en s'appuyant sur le MCD suivant:



**Règle 1** : Les entités deviennent des *tables*, l'identifiant de l'entité devient la *clé* de la table. Les propriétés de l'entité deviennent des *attributs*.

On obtient donc immédiatement les tables :

**CLIENT** (numcli, nomcli, adrcli)  
**COMMANDE** (numcom, datcom)  
**ARTICLE** (numart, libart)

**Règle 2** : Quand on trouve une cardinalité **1,1** on ajoute l'identifiant cible à la table source.

Ainsi, l'entité COMMANDE possède une association avec des cardinalités 1,1. La table *Commande* devient :

**COMMANDE** (numcom, #numcli , datcom)

La propriété dupliquée **numcli** devient clé étrangère dans la table source (Commande) et est marquée par un #. Une clé étrangère dans une table est une propriété qui est clé primaire dans une autre table.

**Règle 3** : Une association de la forme x,n -----y,n devient une table à part entière dont la clé est la concaténation des deux identifiants des entités concernées et dont les attributs sont les propriétés (si elles existent) de l'association.

On obtient donc la table supplémentaire suivante :

**ARTICLE\_COMMANDE** (numcom, numart, qtecom)

En définitive on aboutit au MLD suivant:

**CLIENT** (numcli, nomcli, adrccli)

**COMMANDE** (numcom, #numcli , datcom)

**ARTICLE** (numart, libart)

**ARTICLE\_COMMANDE** (numcom, numart, qtecom)



Université Abdelmalek Essaâdi  
Ecole Nationale de Commerce et de Gestion  
www.encgt.ma

## LE MODELE PHYSIQUE DES DONNEES (MPD)

**M. I.ELKHALKHALI**

Le Modèle Physique des données (**MPD**) est le dernier modèle réalisé pour les données avant la programmation. Le MPD va permettre d'implanter sur machine l'ensemble des données issues du **MLD**.

le modèle physique des données est exprimé dans le langage fourni par le SGBD relationnel. Or ce langage n'est autre que **SQL**.

En effet, **SQL** est à la fois un langage de manipulation de données permettant de formuler des requêtes et aussi un langage de description de données par le biais de la création, de la mise à jour et de la destruction de données ou de tables.

**Exemple :** supposons que l'on veuille créer la table **CLIENT** ( **numcli**, **nomcli**, **adcli**, **telcli**)

On utilise le mot clé **CREATE** pour créer la structure de la table :

```
CREATE TABLE CLIENT (  
numcli NUMBER(3) NOT NULL,  
nomcli CHAR(40),  
adcli CHAR(60),  
telcli NUMBER(10));
```

On constate que chaque attribut est nommé et qu'on lui assigne un type de données : **NUMBER** pour une donnée numérique, **CHAR** pour une donnée texte avec, entre parenthèses, le nombre de chiffres ou de caractères maximaux. **NOT NULL** signifie que l'on doit impérativement renseigner l'attribut concerné car il s'agit d'une clé.

Toutes les tables d'une base relationnelle sont ainsi définies dans ce langage ce qui permet de définir le modèle physique des données.



Université Abdelmalek Essaâdi

## *Ecole Nationale de Commerce et de Gestion*

*National School of Management*

*Tél. :039-31-34-87/88/89, Fax : 039-31-34-93, Adresse: B.P 1255 Tanger-Moroc*

### **LE MODELE CONCEPTUEL DES TRAITEMENTS-MCT**

**M. I. EL KHALKHALI**

#### *Le Modèle Conceptuel des Traitements*

Le Modèle Conceptuel des Traitements (MCT) est un schéma qui a pour but de représenter formellement toutes les activités exercées par une entreprise. Les traitements à effectuer et leur enchaînement apparaissent dans le *diagramme de flux*.

Mais attention, le MCT exprime ce qu'il faut faire mais n'indique pas qui doit le faire, ni quand il faut le faire, ni où il faut le faire, ni comment il faut le faire. Il exprime le quoi mais ni le qui, ni le quand, ni le où, ni le comment.

#### *1) Diagramme des flux*

Il s'agit de décrire par un diagramme l'ensemble des flux d'information existant entre les différents acteurs du S.I ou du domaine étudié.

Le diagramme des flux peut être représenté sous 2 formes : *matrice* ou *graphe*.

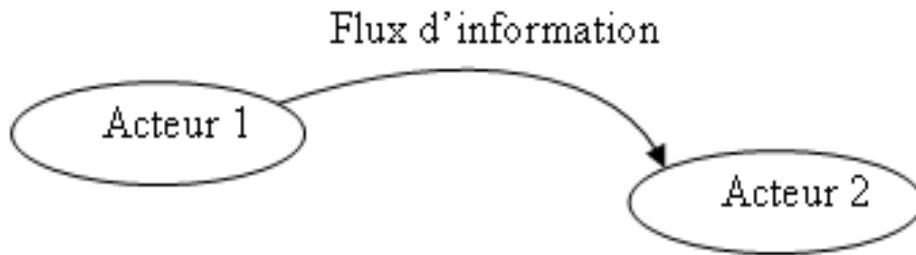
##### a) *Matrice des flux* :

	Acteur 1	Acteur 2		Acteur j
Acteur 1				
Acteur 2				
Acteur i				Aij

*Aij* : représente l'ensemble d'information envoyé par l'acteur i à j.



b) Graphe des flux :



Exemple :

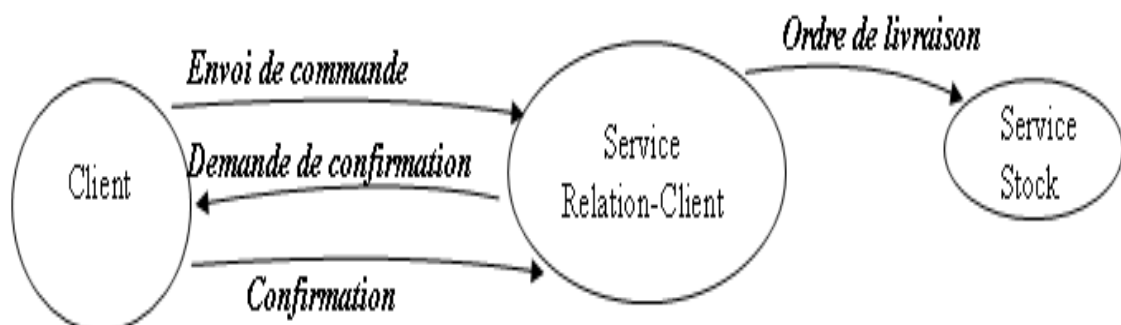
Des clients passent des commandes pouvant concerner plusieurs produits. Si le stock est suffisant pour tous les produits commandés, la commande est mise à livrer et sinon le client est interrogé pour savoir s'il maintient sa commande.

Suivant la réponse du client, la commande est annulée ou confirmée, dans ce dernier cas, certaines parties de la commande sont livrables et les autres mises en attente.

Matrice des Flux :

	Client	Service Relation-Client	Service Stock
Client		<ul style="list-style-type: none"> <li>• Envoi de commande</li> <li>• Confirmation de commande</li> </ul>	
Service Relation-Client	<ul style="list-style-type: none"> <li>• Demande de confirmation</li> </ul>		<ul style="list-style-type: none"> <li>• Ordre de livraison</li> </ul>
Service Stock			

Diagramme des flux :



Le diagramme des flux reste un excellent outil de synthèse pour montrer, les échanges d'information entre les différents acteurs.

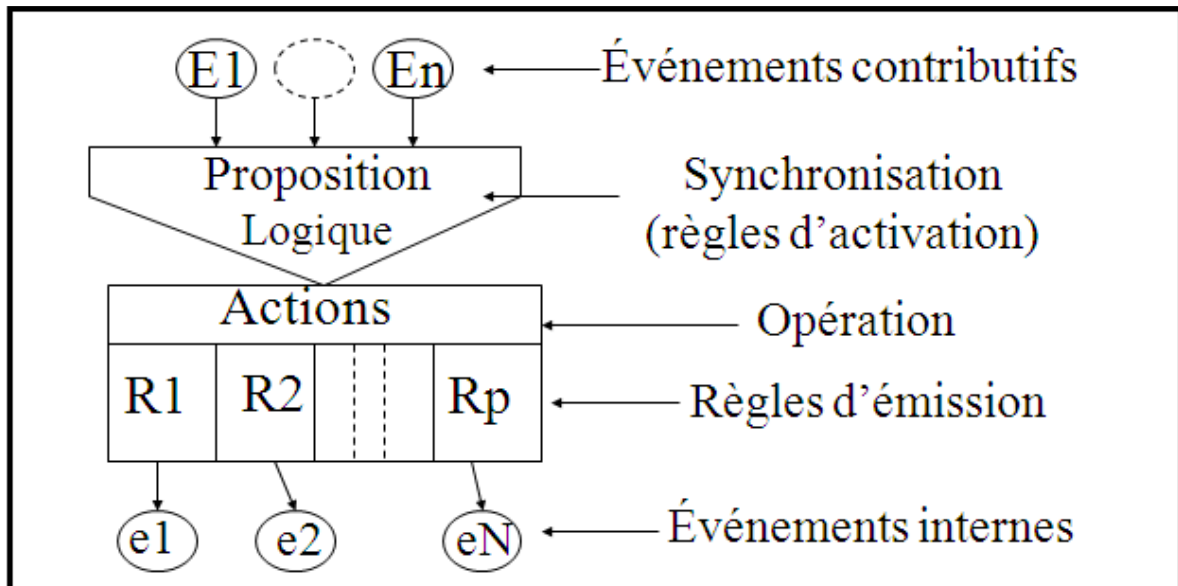
Mais présente l'inconvénient de ne pas mettre en évidence l'ordre du cheminement ainsi que les contraintes liées au système étudié.

## 2) Les concepts de base du MCT

Un MCT est une succession d'**opérations** déclenchées par des **événements** et qui donnent naissance à de nouveaux événements (résultats). Ces résultats associés à d'autres événements ou résultats sont **synchronisés** pour déclencher une nouvelle opération.

Cette formulation fait apparaître les trois concepts du modèle, à savoir :

- a) Événement
- b) Opération
- c) Synchronisation



Un événement est le compte rendu au SI du fait que quelque chose s'est produit dans l'univers extérieur ou dans le SI lui-même :

- Un événement est externe s'il provient de l'univers extérieur, il est interne lorsqu'il est produit par le SI.
- Un événement externe provoque une réaction du SI sous la forme d'une **opération**.
- Un événement interne peut soit provoquer une nouvelle réaction du SI, soit constituer un résultat pour l'univers extérieur.

Une **opération** représente la réaction du système à l'arrivée d'un ou plusieurs événements. Une fois démarrée, une opération s'exécute sans interruptions, elle produit selon les **règles d'émission** (les différents cas possibles de l'opération) des événements internes (résultats).

Une **synchronisation** est associée à une opération dont elle définit la condition de déclenchement, elle associe une liste d'événements, et une condition booléenne (*OU, ET*) qui établit de quelle manière les événements participent à la synchronisation (règles d'activation).

L'arrivée du premier événement met la synchronisation en attente jusqu'à l'arrivée du dernier événement. Si la condition est vraie, l'opération peut être déclenchée.

Si nous reprenons l'exemple précédent, le MCT serait :

