



MODELISATION DES SYSTEMES D'INFORMATION

Mr Mohamed EL KASSIMI

Contenu

1. INTRODUCTION A MERISE	4
1.1. LES MOTIVATIONS	4
1.2. TERMES ET ABREVIATIONS	4
1.3. HISTORIQUE.....	5
1.3.1. Les origines	5
1.3.2. Merise.....	6
1.3.3. Merise/2.....	7
1.3.3.1. Le schéma des flux.....	9
1.3.3.2. Règles de découpage en domaines/activités	12
1.3.3.3. Le Cycle de Vie d'un Objet	12
1.3.3.4. Le Modèle Conceptuel des Traitements Analytiques.....	15
1.3.3.5. Le Modèle Organisationnel des Données.....	16
1.3.4. La modélisation des données aujourd'hui.....	18
1.3.4.1. Merise aujourd'hui.....	18
1.3.4.2. Méthode OOM: Orientation Objet dans MERISE.....	18
1.3.4.3. La méthode Mega.....	19
1.3.4.4. La méthode Axial.....	19
1.3.4.5. Bilan.....	19
1.4. VOCABULAIRE	19
1.4.1. Organisation et système.....	19
1.4.2. Le système d'informations.....	20
1.4.3. Modèle, méthode et analyse.....	21
1.5. LES PRINCIPES DE BASES DE MERISE.....	21
1.5.1. Le cycle de vie	21
1.5.2. Le cycle d'abstraction	24
1.5.3. Le cycle de décision.....	25
1.5.4. Présentation des 6 (ou 7) modèles Merise.....	26
1.6. LA DEMARCHE MERISE	27
1.6.1. La technique	27
1.6.2. La démarche.....	27
1.6.3. Les étapes	28
1.6.4. Périmètre du cours	29
2. LE MCD.....	30
2.1. INTRODUCTION	30
2.1.1. M.C.D.	30
2.1.2. Entité	30
2.1.3. Association	30
2.1.4. Associations particulières.....	31
2.2. COMPLETER LE MODELE	32
2.2.1. Cardinalités	32
2.2.2. Dimensions	35
2.2.3. Les formes normales.....	36
2.3. CONTRAINTES INTER-ASSOCIATIONS	37
2.3.1. Différentes situations.....	37
2.3.2. Partition.....	37
2.3.3. Exclusion	38
2.3.4. Totalité.....	38
2.3.5. Inclusion.....	38
2.3.6. Contraintes sur les propriétés	39
2.3.7. Spécialisation/Généralisation d'entités types.....	39
2.3.8. Contraintes sur spécialisations	40
3. PASSAGE DU MCD AU MODELE PHYSIQUE.....	41
3.1. INTRODUCTION	41
3.1.1. Le MLD, Le MPD.....	41
3.1.2. Démarche	41

3.1.3. Une table exemple	41
3.1.4. Formalisme.....	41
3.2. PASSAGE DU MCD AU MLD	42
3.2.1. Règle 1	42
3.2.2. Règle 2	42
3.2.3. Règle 3.....	42
3.2.4. Règle 4.....	42
3.2.5. Règle 5.....	43
3.2.6. Règle 5 bis	43
3.2.7. Règle 6.....	44
3.2.8. Conclusion.....	44
3.3. LE MPD.....	45
3.3.1. Pourquoi une étape supplémentaire ?	45
3.3.2. Le reverse-engineering.....	47
3.3.3. Les différents outils de modélisation et de génération.....	48
3.3.4. Migration de bases	50

1. Introduction à Merise

1.1. Les motivations

Les responsables d'entreprises, les décideurs, les utilisateurs sont tous d'accord pour dire que les méthodes d'organisation facilitées, la ré-utilisation, la présentation et le stockage des informations. Les méthodes comme MERISE s'imposent d'eux-mêmes. Les spécialistes préconisent qu'un standard pour l'organisation d'information est indispensable.

1.2. Termes et abréviations

Association : L'association représente la mise en relation d'une ou plusieurs entités. Les associations sont énoncées par des règles de gestion. Elles se nomment avec un verbe et peuvent avoir, ou non, des propriétés.

Entité : Élément concret ou abstrait qui présente un intérêt pour le système d'information étudié. Concrètement, l'entité s'apparente à une collection de propriétés.

Évènement : Fait qui déclenche une opération (l'opération est en attente d'évènement). Un événement correspond à un flux conceptuels. Il peut être interne, externe et temporels.

CVO (Cycle de Vie d'un Objet) : Mettre en évidence l'ensemble des états remarquables d'un individu de gestion au cours de sa vie. Le cycle de vie essaye d'introduire une vue dynamique sur un modèle de données essentiellement statique. Le CVO est le lien entre le MCD et le MCTA

MCD (Modèle Conceptuel des Données) : Représentation statique, sous forme schématique, de la situation respective des données d'un domaine de gestion. Ce schéma est conçu pour être très stable dans le temps.

MCT (Modèle Conceptuel des Traitements) : Représentation abstraite des traitements des données, en relation avec les fonctions d'événements qui peuvent se produire sur les données.

MCTA (Modèle Conceptuel des Traitements Analytiques) : Il permet la représentation, des individus de gestion et des actions à effectuer, qui sont nécessaires au déroulement de l'activité. Il permet de distinguer et de prévoir les événements et ressources.

MERISE : Le nom "Merise" n'est pas un acronyme : C'est une métaphore du merisier qui doit être greffé avec le cerisier pour porter des fruits. MERISE est un langage de référence pour lire comprendre et organiser des données. Un langage permet de communiquer avec un plus large public pour se faire comprendre de nos intentions.

MLD (Modèle Logique des Données) : C'est la description de la base qui va être créée, c'est le dernier pas vers le MPD.

MOT (Modèle Organisationnel des Traitements appelé aussi MLT) : Aidé par le MCD, décrit les propriétés des traitements non traités. La représentation sous la forme de table reprend les fonctionnalités de temps, de données et de ressources des actions à réaliser.

MPD (Modèle Physique des Données) : Au contraire du modèle logique (MLD) ou conceptuel (MCD), ce modèle dépend de la base de données et des détails de l'implémentation. Il représente toutes les entités et les données associées entre entités.

MPT (Modèle Physique des Traitements) : C'est le modèle II permet l'étape de migration sur le matériel.

Propriété : Plus petit élément logique d'information manipulé par l'organisation (dit "insécable").

Opération : C'est une suite de traitements non interruptibles. Une opération correspond à une activité élémentaire figurant dans un diagramme de flux conceptuel.

SGBD (Système de Gestion de Bases de Données) : Systèmes de consultations, d'ajout, de modifications, etc. de base de données. Les bases de données constituent des ensembles structurés et organisés permettant le stockage de grandes quantités de données dans la perspective de leur exploitation.

SGBDO (Systèmes de Gestion de Bases de Données orientés Objet) : Systèmes de consultations, d'ajout, de modifications, etc. de base de données pour des langages Orientés Objet.

Synchronisation : Expression logique pour définir le déclenchement d'une opération.

1.3. Historique

1.3.1. Les origines

En 1977 le ministère de l'industrie constitue un pôle méthode en association avec certaines grandes S.S.I.I. dont Cap SOGETI, SESA, STERIA, Eurosoft, et le CETE (centre de traitement du ministère de l'industrie, Hubert TARDIEU en est l'expert en bases de données).

De ce pôle méthode vont naître 2 guides :

- RACINES (établissement du schéma directeur d'informatisation),
- ACTIF (informatisation des activités sur les lieux de travail).

A la suite de la fusion des 2 précédents projets, la méthode MERISE voit le jour en 1978 sur l'initiative des ministères de l'industrie, et de l'équipement (dans le cadre de la mission informatique), associés aux S.S.I.I. Cap SOGETI, SESA, STERIA, Eurosoft, SEMA-METRA, et CGI.

En 1985 le cercle MERISE est créé par la SEMA-METRA, suite au travail effectué en son sein par Hubert TARDIEU, et René COLETTI (les principaux moteurs de la méthode), associé à ce cercle la société CGI (dont le directeur Robert MALLETT est l'inventeur de la méthode CORIG).

La méthode MERISE est devenue la méthode d'analyse des systèmes d'informations la plus utilisée par les sociétés et les administrations françaises, l'estimation actuelle est de 60% des entreprises. Elle s'appuie à la fois sur une méthode, et sur un nombre grandissant d'outils logiciels d'aide à la conception de systèmes informatiques (SECSI, AMC-Designor, Silverun,...).

La méthode MERISE évolue actuellement vers la modélisation des concepts objets dans les bases de données sur l'initiative d'Hubert TARDIEU.

Cette méthode apporte une aide considérable à la formalisation des concepts d'étude des systèmes d'informations; une aide reconnue par une majorité de professionnels, de formateurs, et d'universitaires; ce qui peut répondre aux quelques détracteurs qui lui reprochent un certain manque de rigueur.

MERISE est l'intersection de bonnes idées qui étaient dans l'air du temps des années 1970 (principalement face au développement de l'informatisation associé à un manque évident de méthodologie).

De la méthode CORIG, elle utilise le concept d'étude préalable (CORIG-A) et d'étude détaillée (CORIG-B).

Du guide RACINES, elle tire le concept de schéma directeur, et des groupes d'étude attachés (comité directeur, groupe de projet, et groupe utilisateur).

La méthode AMS (Analyse Modulaire des Systèmes) apporte l'approche systémique (notion de systèmes de pilotage, opérant, et d'informations)

L'approche relative aux données en trois niveaux (conceptuel, externe, interne) provient du rapport ANSI/SPARC de 1975.

Les travaux de TARDIEU, de CHEN (modèle ENTITE-ASSOCIATION) et de CODD (formalisme relationnel) ont amené l'approche bases de données MERISE.

Les recherches de l'équipe de TARDIEU du CETE d'Aix-en-Provence ont permis le développement du modèle Evénement-Résultat.

Des travaux de la société SEMA-METRA sont apparus les idées relatives à l'informatique transactionnelle et temps réel.

MERISE a été la base d'un très gros effort d'intégration et de mise en forme d'idées, et de formalismes récents.

La conception d'un système d'information n'est pas évidente car il faut réfléchir à l'ensemble de l'organisation que nous devons mettre en place. La phase de conception nécessite des méthodes permettant de mettre en place un modèle sur lequel nous allons nous appuyer. La modélisation consiste à créer une représentation virtuelle d'une réalité de telle façon à faire ressortir les points auxquels nous nous intéressons. Ce type de méthode est appelé analyse. Il existe plusieurs méthodes d'analyse, la méthode la plus utilisée en France étant la méthode MERISE.

Le but de cette méthode est d'arriver à concevoir un système d'information. La méthode MERISE est basée sur la séparation des données et des traitements à effectuer en plusieurs modèles conceptuels et physiques. La séparation des données et des traitements assure une longévité au modèle. En effet, l'agencement des données n'a pas à être souvent remanié, tandis que les traitements le sont plus fréquemment.

L'entreprise est un système complexe dans lequel transitent de très nombreux flux d'informations. Sans un dispositif de maîtrise de ces flux, l'entreprise peut très vite être dépassée et ne plus fonctionner avec une qualité de service satisfaisante. L'enjeu de toute entreprise qu'elle soit de négoce, industrielle ou de services consiste donc à mettre en place un système destiné à collecter, mémoriser, traiter et distribuer l'information (avec un temps de réponse suffisamment bref). Ce système d'information assurera le lien entre deux autres systèmes de l'entreprise : le système opérant et le système de pilotage.

1.3.2. Merise

Les points nouveaux de la première version de cette méthode, consignés dans l'ouvrage "La méthode MERISE", étaient alors:

- Une approche systémique qui a ses origines dans la théorie des systèmes de Boulding introduite en France par J.-L. Lemoigne : elle montre les relations existant entre le système d'information et le système opérant d'une part, et le système d'information et le système de pilotage d'autre part. Le

système d'information fournit aux acteurs (système opérant) et aux décideurs (système de pilotage) les informations dont ils ont besoin pour agir et décider.

- Une couverture de tout le cycle de vie du logiciel: Schéma directeur, étude préalable, étude détaillée, étude technique, production de logiciels, mise en œuvre, maintenance;
- Un cycle d'abstraction reposant sur 3 niveaux :
 - Conceptuel (réponse à la question "quoi ?"),
 - Organisationnel ou logique (réponse aux questions "qui ?", "quand ?", "où ?")
 - Physique (réponse à la question "comment ?").
- La séparation entre les modèles de données, analysés avec une approche entité-association, et les modèles des traitements, présentés avec un formalisme proche de celui des réseaux de Pétri.

La méthode conduit à réaliser les 6 modèles suivants :

- **MCD** : Modèle Conceptuel des Données,
- **MCT** : Modèle Conceptuel des Traitements,
- **MLD** : Modèle Logique des Données,
- **MOT** : Modèle Organisationnel des Traitements,
- **MPD** : Modèle Physique des Données,
- **MPT** : Modèle Physique des Traitements.

Une phase de validation permet de vérifier que toutes les données sont présentes pour réaliser les traitements et que tous les traitements utiles pour obtenir les données sont prévus.

1.3.3. Merise/2

A partir des 7 modèles de base (Diagramme des flux, MCD, MCT, MLD, MOT, MPD, MPT) :

Diagramme des flux	
MCD	MCT
MLD	MOT
MPD	MPT

La volonté de répartition par rapport aux sites géographiques et informatiques, conduit à un modèle qui décrira la répartition aussi bien des données que des traitements.

Diagramme des flux	
MCD	MCT
MLD	MOT
MLD R	MLT R
MPD	MPT

Une meilleure définition des niveaux amène à la séparation de l'organisationnel et du logique en deux niveaux distincts ; ce qui amène le découpage suivant :

Merise			Merise / 2		
Concept	Quoi		Concept	Quoi	
			Orga	Où Qui Quand macro organisation	
O/L	Comment	Où Qui Quand	Logi	Comment micro orga poste travail	
Phys	Comment		Phys		

Le niveau organisation : C'est d'où utiliser, qui utilise, quand utiliser ?

Le niveau logique : Comment représenter comment stocker ?

Le niveau logique réparti : Où et comment stocker, comment communiquer ?

Le premier niveau est un niveau qualifié de système d'information organisationnel, il est donc conçu avec la participation des utilisateurs finaux de l'application.

Le second niveau est le système d'information informatisé, il est du ressort de l'équipe informatique uniquement.

La nouvelle approche doit également permettre une meilleure adéquation données/traitements et aussi une validation entre les deux.

Une colonne a été créée entre les données et les traitements afin de mettre en place les Modèles de traitements analytiques, ainsi que le cycle de vie des objets.

Les modèles analytiques essayent de rapprocher les données des traitements et permettent une sorte de validation.

Données	Données traitements	Traitements
MCD	CVO MCTA	MCT
MOD	CVO organisationnel MOT A	MOT
MLD		MLT

Le cycle de vie essaye d'introduire une vue dynamique sur un modèle de données essentiellement statique.

Les flux doivent être appréhendés à différents niveaux ce qui conduit à faire de l'analyse des flux une colonne à part entière pour chaque niveau.

Flux
MC MFC
MFO
MFL

Nouveauté dans la démarche :

- La méthode propose une démarche de composition et de décomposition.
- Décomposition pour analyser et spécifier.
- Composition pour synthétiser, réaliser et intégrer.

Exemple :

Dans un MCT nous nous contentons de l'événement Commande au niveau global, alors que nous retenons l'événement « commande nouveau client » et « commande client référencé » dans un niveau détaillé.

Les modèles doivent donc être vus à 2 niveaux :

1. un niveau global
2. un autre détaillé.

Nous utiliserons comme un Zoom entre les 2. (Démarche utilisée pour les MCD MFC MCT MOT).

Dans le MCD nous ne conservons que les objets généraux dans un premier temps ; nous spécialiserons dans la deuxième phase.

La méthode propose de ne plus bâtir les modèles de manière indépendante et séquentielle, mais de mener l'ensemble des modèles conceptuels en même temps. Nous faisons rarement le MCT avec le MCT A :

↓			
↙		↘	
↔	MFC	↔	↘
MCD ↔	MCT A	↔	MCT
↔	CVO	↔	↗

Nous dirons que Merise/2 contient :

- des modèles de communication, de flux (ceux destinés vers l'utilisateur, le niveau conceptuel)
- des modèles de découpages en domaines et activités (ceux destinés vers les concepteurs, le niveau conceptuel)
- des modèles de comportement (les cycles de vie)
- des modèles de conception (ceux qui prennent en compte la validation, les modèles analytiques)
- des modèles de structure et organisation (ils définissent la structure de l'application, le niveau logique)

1.3.3.1. Le schéma des flux

Nous devons répondre à la question du "quoi" ? Il faut être indépendant de l'organisation. Nous pourrions dans cette partie élaborer 2 modèles :

- **Le modèle de contexte** : Pour la détermination du domaine en montrant les flux dans leur généralité.
- **Le modèle des flux conceptuels** : Pour la détermination des activités (processus) du domaine avec un graphe des flux, détaillé.

1. Le modèle de contexte

Objectifs :

- Illustrer les interactions entre le domaine et l'extérieur (intervenants externes ou autres domaines).
- Définir le domaine d'étude.
- Déterminer les échanges avec l'extérieur.

Concepts :

Le domaine fonctionnel :

Il est orienté gestion de l'entreprise et recouvre une finalité majeure de celle-ci. C'est un domaine pilotable, autonome, ayant une certaine organisation : la vente, la gestion de production, le personnel.

Le domaine d'étude :

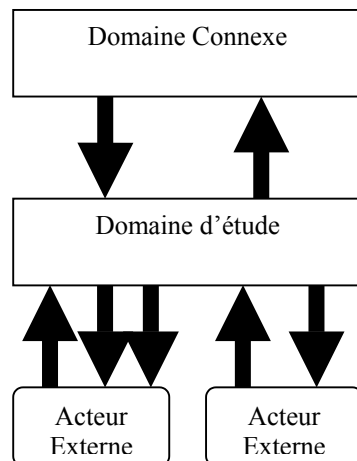
- Soit un domaine fonctionnel,
- Soit un sous-ensemble du domaine fonctionnel,
- Soit plusieurs domaines fonctionnels.
- C'est un sous-ensemble cohérent (grâce au modèle de contexte), bien délimité et formant le contenu du sujet à étudier. C'est un projet de taille réaliste.

Le flux : C'est un message conceptuel.

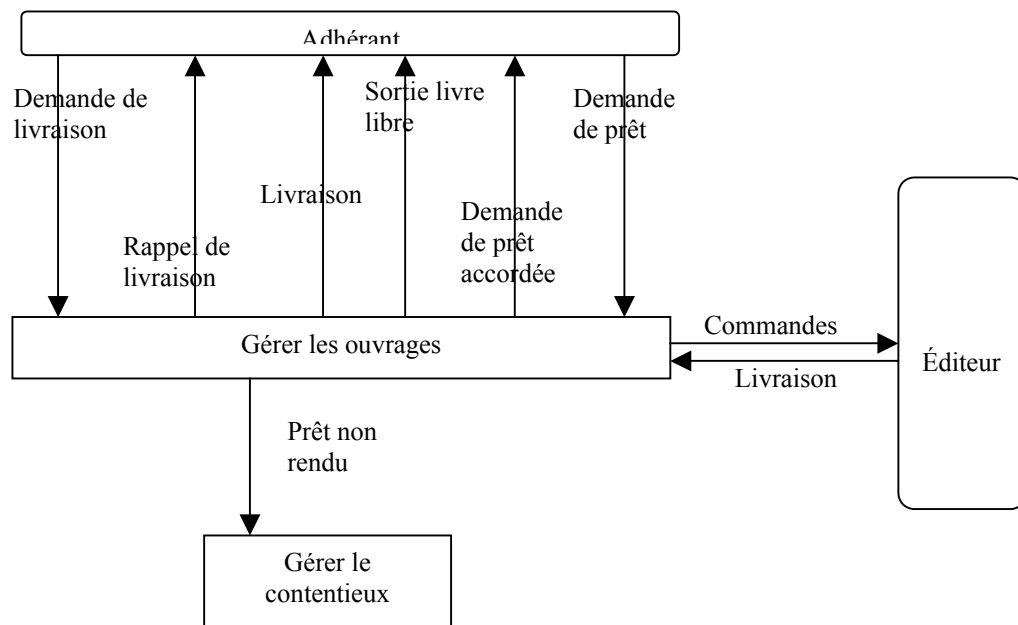
Domaine connexe et acteur externe :

- **domaine connexe** : extérieur au domaine d'étude,
- **acteur externe** : extérieur à l'entreprise.

Le formalisme :



Exemple :



LE MODELE DE CONTEXTE

2. Le modèle de flux conceptuels

Objectifs :

Décomposer le domaine d'étude en activités en utilisant les techniques de composition/décomposition.

Concept :

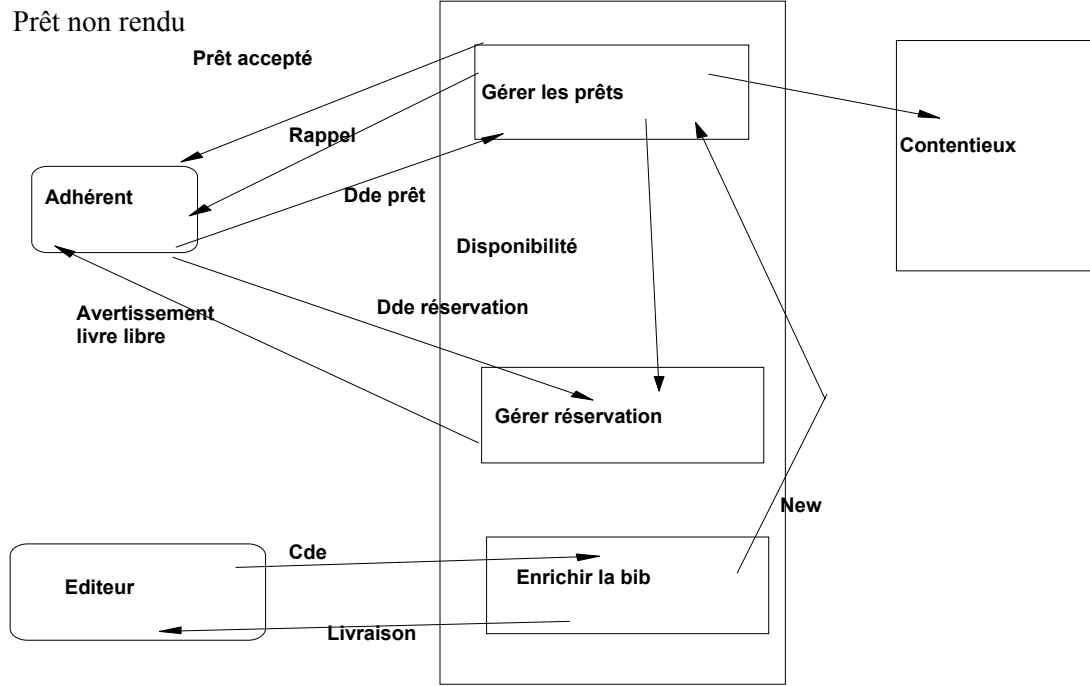
Activité : C'est un ensemble cohérent de traitements qui transforme ou manipule des données.

Caractéristiques d'une activité :

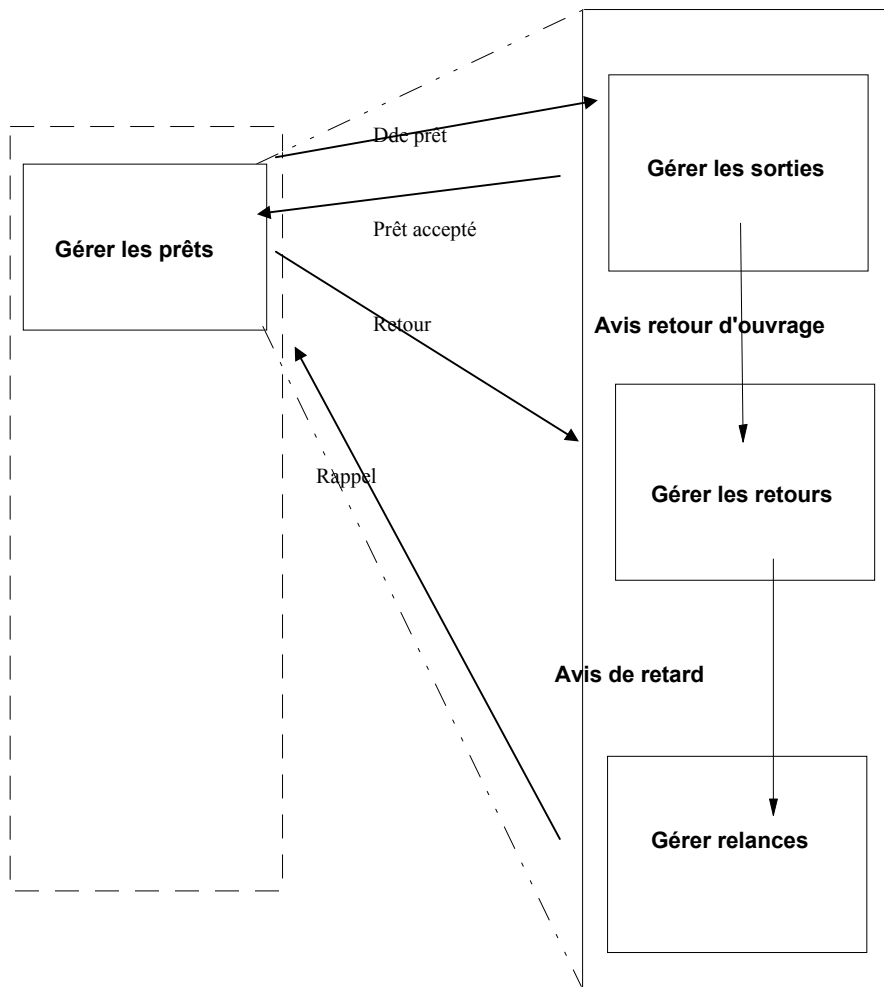
1. A une certaine autonomie.
2. Est pilotable.
3. Assure une finalité majeure du domaine d'étude.

Nous pouvons décomposer une activité jusqu'à 4 niveaux. Une sous-activité ne doit jamais être redondante.

Exemples : Découpage de premier niveau



De second niveau :



1.3.3.2. Règles de découpage en domaines/activités

Objectifs :

- Permettre de répartir l'étude entre plusieurs équipes.
- Permettre de limiter l'étude à un domaine particulier.
- Aider à la définition d'objectifs mesurables.
- Permettre de réfléchir aux états stables du système d'information.
- Essayer de donner des objectifs à chaque niveau de décomposition. Ceux-ci doivent être mesurables.

Les flux qui sont importants :

	Acteur externe	Activité	Domaine connexe
Acteur externe		Oui	
Activité	oui	Oui	oui
Domaine connexe		Oui	

Règles de découpage en domaine/activités :

- Le domaine doit être stable par rapport à l'organisation.
- Le domaine doit être aussi autonome que possible.
- Un domaine est décomposé en 8 activités au maximum.

Technique de décomposition :

- Identifier des critères d'arrêt. Si la sous-activité devient une action organisationnelle, nous avons trop décomposé (tamponner la carte de lecteur).
- Décomposition jusqu'à 4 niveaux. Ce découpage est très proche des objectifs.

Nous pouvons avoir des sous-activités sans lien avec l'extérieur, ce qui est rare au niveau de l'activité. L'activité est la notion de base de l'étude conceptuelle des traitements.

Exercice : Elaborer le modèle de contexte du domaine de Gestion des séminaires d'une société de formation.

Hypothèse, les fonctions couvertes par le système sont :

- Gérer la planification des séminaires (un lieu, une date, un matériel)
- Gérer les demandes de séminaires, les inscriptions, les convocations, les feuilles de signatures
- Gérer la qualité

LE MCD étendu :

- Prise en compte de nouvelles contraintes
- Sous-type d'entités et d'associations
- La notion d'agrégation
- Le nouveau formalisme
- Construction par spécialisation

1.3.3.3. Le Cycle de Vie d'un Objet

Objectifs :

- Mettre en évidence l'ensemble des états remarquables d'un individu de gestion au cours de son cycle de vie.
- Identifier les événements qui transforment les états.
- Illustrer les interactions des changements d'états.
- Affiner les règles de gestion du MCD.
- Préparer la construction du MCT A.

Nous parlerons de :

- Etat de l'objet
- Evénement
- Transition

L'état de l'objet permettra de mieux concevoir le MCD.

Le MCT A permettra de prévoir les événements

Dans le CVO seuls les individus de gestion nous intéressent (pas les associations). Nous ne devons traiter que les objets principaux du MCD.

Exemple :

	enregistrée
	en attente
Commande	livrable
	facturée
	archivée

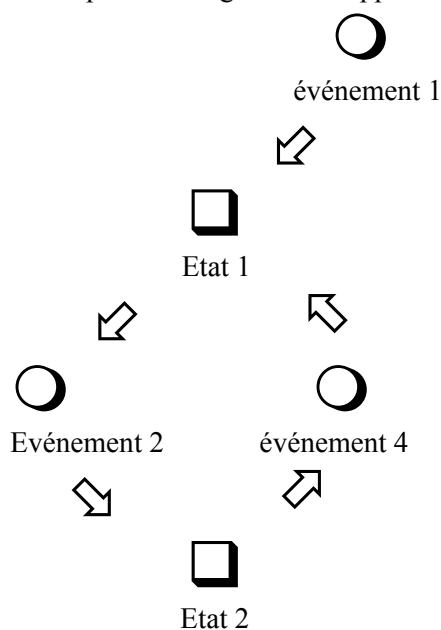
Ce sont des états stables ou remarquables (ils ne peuvent évoluer sans événements)

Le MCD est donc un modèle purement statique. Le CVO est lui un modèle dynamique

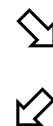
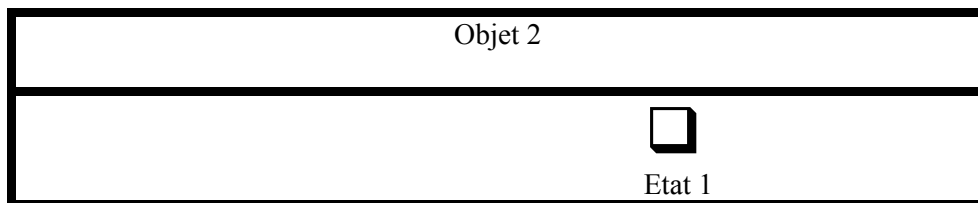
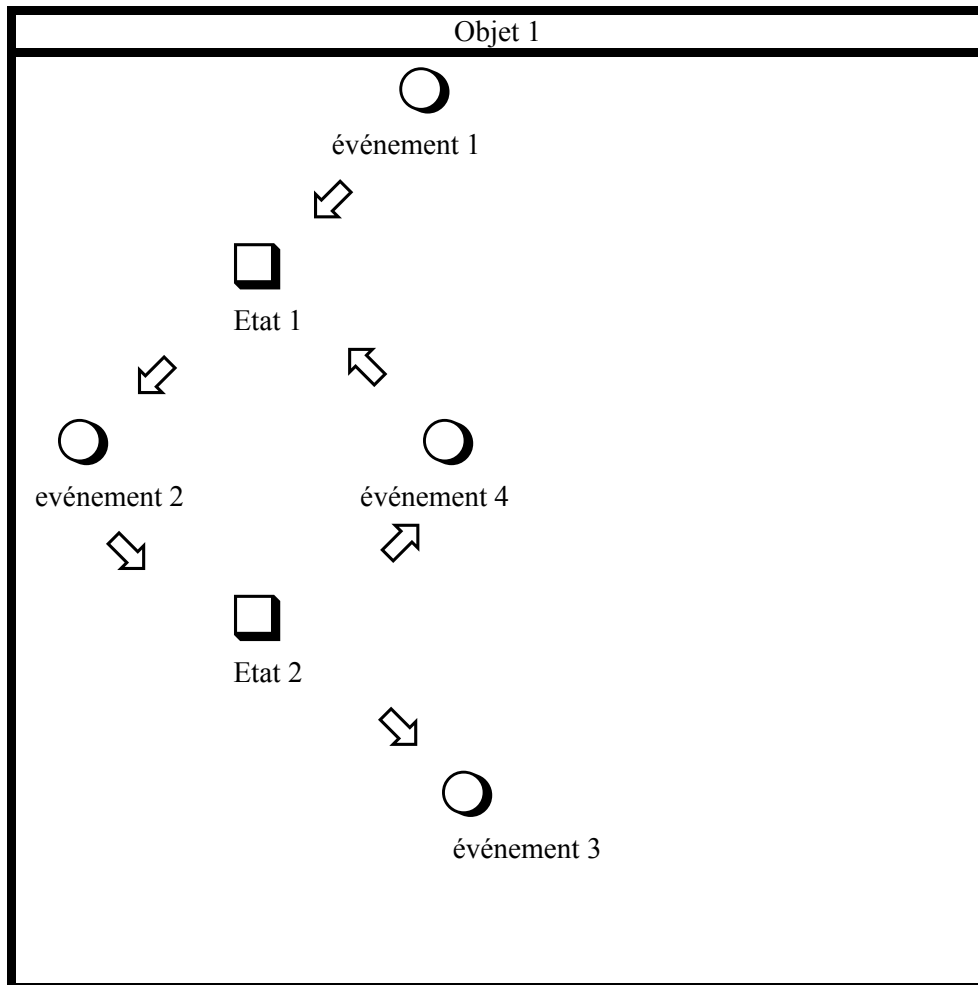
Etat d'un individu :

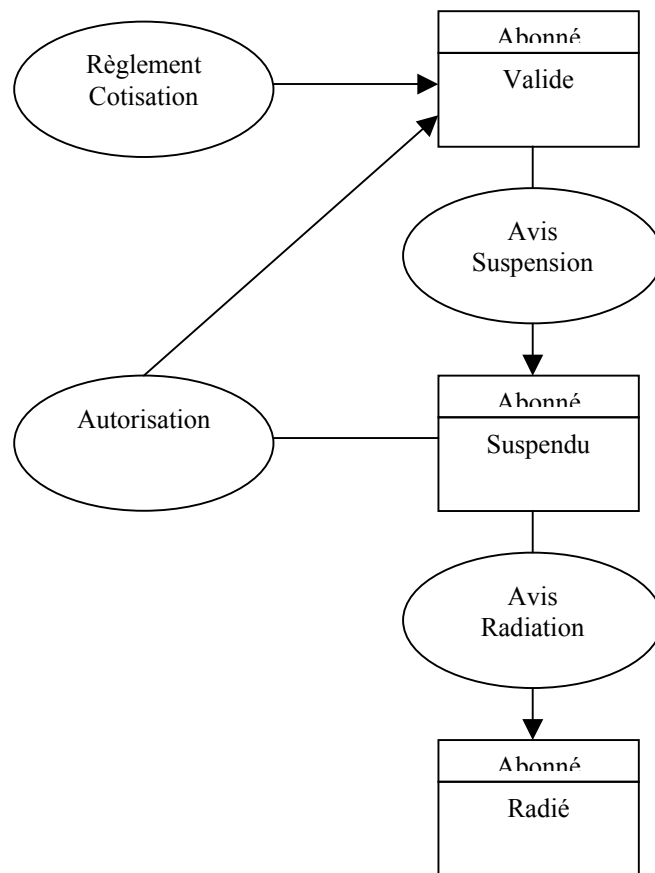
- Un état est un stade transitoire par lequel passe un objet au cours de son cycle de vie.
- C'est le côté transitoire qui permet de différencier un état d'un sous-type (au sens de la généralisation ou de la spécialisation).
- Il correspond à des choix de gestion.
- L'événement est le temps ou une action.
- Une occurrence de l'individu ne peut être dans plusieurs états à la fois.
- Dans le MCD nous représentons le CVO par une propriété (code état) ou une entité (état).

Le CVO est le lien entre le MCD et le MCTA. Il permet d'affiner les règles de gestion. L'étude du CVO permet de confirmer le découpage en domaines ou en activités. Le CVO est une première tentative pour se diriger vers l'Approche Objet



Un événement peut être le temps ou une action. Nous devons établir un cycle de vie par objet (entité) significatifs Il est possible de montrer les interactions entre les différents objets par rapport au CVO



Exemple d'un CVO :**1.3.3.4. Le Modèle Conceptuel des Traitements Analytiques****Objectifs :**

- Décrire le fonctionnement du système indépendamment de l'organisation.
- Préparer le passage à un niveau organisationnel.
- Rapprocher les données et les traitements.

Le MCT A permet de :

- Représenter les individus de gestion qui sont nécessaires au déroulement de l'activité.
- Représenter les actions effectuées sur ces derniers.
- Distinguer entre événements et ressources

Distinction entre MCT/ MCT A :

- MCTA est un modèle de conception.
- MCT est un modèle de communication.

Points commun entre MCT/ MCT A :

- Le formalisme est identique.
- Les concepts d'opération, d'événement, de résultat et de synchronisation sont identiques.
- Les règles de construction reposent sur la non redondance, et la non interruptibilité.

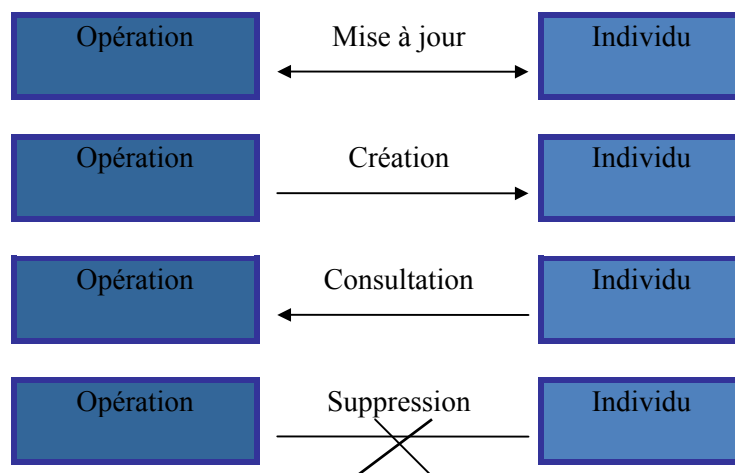
Les Actions : Une action sur un individu peut être de 4 types :

1. Consultation
2. Création
3. Mise à jour
4. Suppression

Une action occasionne un changement d'état



L'état 1 est l'état de départ le 2 et 3 est celui d'arrivée.



Nous pouvons également préciser les différents types d'actions, à savoir :

- Les actions itératives
- Les actions suivant conditions
- Les actions et les contraintes d'intégrité.

Les règles de gestion : Une règle de gestion précise la façon dont s'effectue la transformation des propriétés consultées en entrée, pour produire un individu modifié en sortie. Elles s'expriment sous formes structurées en tenant compte du découpage en activités.

Exemple : Si un exemplaire de l'ouvrage est disponible, créer Prêt.

Nous pouvons également créer une matrice des traitements par rapport aux données où nous indiquons les différents types d'actions

- Lecture
- Ecriture
- Suppression
- Mise à jour

	Client	Produit	Compte	Opération	Devise	Gestionnaire	Signature
Opération 1	L		L			L	
Opération 2	L	L	E	E			L
Opération 3			E	E	L		L
Opération 4	E		E	E		E	E

1.3.3.5. Le Modèle Organisationnel des Données

Objectifs :

- Distinguer les données à informatiser.
- Quantifier les volumes et définir la durée de vie des données.

- Localiser les données par site.
- Définir les niveaux de sécurité requis.
- Définir des individus et des relations de type organisationnel.

Nous pouvons représenter le MOD de 3 manières avec :

- **Le MOD général :** C'est la structure organisationnelle. Nous nous attachons à la nature des objets et des relations, au système d'autorisation et nous ajoutons des objets ou relation organisationnelle.
- **La vue d'un type de site :** Nous précisons les accès aux données à partir d'un type de site ainsi que les besoins locaux d'archivage.
- **La vue d'un type de poste :** Nous précisons les accès aux données par poste et la responsabilité des personnes.

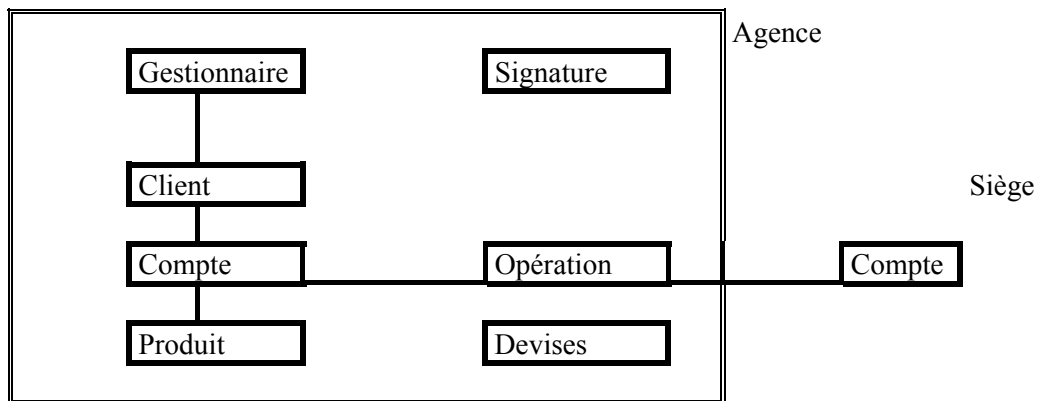
Le choix des informations à mémoriser et l'historique des informations :

Lors de l'étude du MCD il a été décidé de ne pas conserver les informations calculées. En effet il était trop tôt pour connaître les durées de conservation des informations et pour être capable de savoir si le calcul était techniquement envisageable et même possible pour des problèmes d'historique de données.

Pour le MOD nous devons déterminer exactement quelles sont les informations exactes à conserver. Nous allons donc ajouter des entités ou des relations.

La typologie des groupes de données :

- **Privées :** Données mises à jour par le type de site et inaccessibles aux autres sites
- **Protégées :** Données mises à jour par le type de site et consultables par les autres sites.
- **Partagées :** Données mises à jour et consultables par les autres sites.
- **Consultables :** Données uniquement consultables par le type de site.

Exemple de vue par type de site

Il n'est pas nécessaire de montrer tout le détail des données, nous pouvons procéder à des regroupements.

1.3.4. La modélisation des données aujourd'hui

1.3.4.1. Merise aujourd'hui

Les points forts :

- La méthode s'appuie sur une approche systémique : C'est donc une approche globale.
- Les concepts sont peu nombreux et simples.
- Elle est dans le domaine public donc gratuite (rare aux USA).
- Elle est assez indépendante vis à vis de la technologie.
- Elle est la plus utilisée en France dans les domaines de gestion.
- Elle sert de référence aux enseignements sur les méthodes.

Les critiques :

- La méthode s'arrête au niveau organisationnel pour les traitements et les données.
- Elle ne s'occupe pas de l'interface utilisateur.
- Elle est très adaptée à un contexte de création d'application mais pas forcément à un problème de maintenance ou de seconde informatisation.
- Elle ne permet pas réellement une validation rapide de la part des utilisateurs.
- Elle est davantage destinée à des sites centraux plus qu'à des développements en temps réel, ou sur micro.
- Il est très difficile de valider les traitements par rapport aux données et cela au niveau conceptuel ou organisationnel.

1.3.4.2. Méthode OOM: Orientation Objet dans MERISE

La troisième version de MERISE, OOM, date de 1992; elle est totalement marquée par l'orientation objet, et nous retrouvons : la dimension statique, la dimension dynamique et la dimension fonctionnelle.

C'est la dimension fonctionnelle qu'il est conseillé de développer en premier : elle permet de représenter les frontières du système à modéliser par rapport à son environnement en utilisant un diagramme de contexte. Elle permet également de définir les besoins par des diagrammes de flots de données.

La dimension statique reste la plus importante des trois car elle est la plus stable dans le temps; la description des données repose sur un modèle entité-association étendu. A chaque objet nous associons son comportement, par l'intermédiaire des méthodes apparues dans les diagrammes de flots de données.

La dimension dynamique permet de représenter l'enchaînement des opérations dans différents scénarios, les opérations pouvant rassembler les services de plusieurs objets: cela se traduit par des modèles de traitements "classiques" de MERISE auxquels sont ajoutés les objets sur lesquels portent les traitements.

1.3.4.3. La méthode Mega

Yves Tabourier (départ 1997 - Méga se porte vers UML)

3 découpages sur 4 niveaux de description :

	Communication	Données	Traitements
Conceptuel	MCC	MCD	MCT
Organisationnel	MOC	MOD	MOT
Logique	MLC	MLD	MLT
Physique	MPC	MPD	MPT

L'analyse de la composante communication à différents niveaux permet la prise en compte d'une architecture client/serveur:

- Le MCC permet de montrer l'analyse des besoins
- Le MOC montre les flux entre les différents services avec leur support physique ainsi que la répartition entre les différents sites géographiques
- Le MLC permet de caractériser les types de lignes utilisées.

1.3.4.4. La méthode Axial

3 découpages sur 3 niveaux :

Données	Données / Traitement	Traitement
D.O.G.	D.B.A. / D.G.A.	D.F.G.
S.R.N.	S.T.U. / D.D.T.	D.U.M.
E.T.S.	D.R.A.N. / D.E.M.T.	M.T. (I.P.T.)

Axial dans ces différents modèles met beaucoup l'accent sur le rapprochement des données et des traitements.

1.3.4.5. Bilan

Une nouvelle version de Merise doit être capable d'intégrer les nouvelles techniques informatiques comme :

- Les langages objets, la programmation objet, la conception objet.
- Les applications coopératives de type Client/serveur.
- Le développement des applications sur micro-ordinateurs couplées avec les sites centraux.
- L'évolution des SGBD (prise en compte des contraintes au niveau des données).
- Ne pas jeter les modules Merise existants mais compléter la méthode.
- Le problème est donc de faire évoluer Merise afin de s'appuyer sur les personnes déjà formées à la méthode, plutôt que de proposer une nouvelle méthode trop longue à mettre en place.

1.4. Vocabulaire

1.4.1. Organisation et système

Organisation :

Nous englobons sous ce terme l'ensemble des structures économiques qui existent :

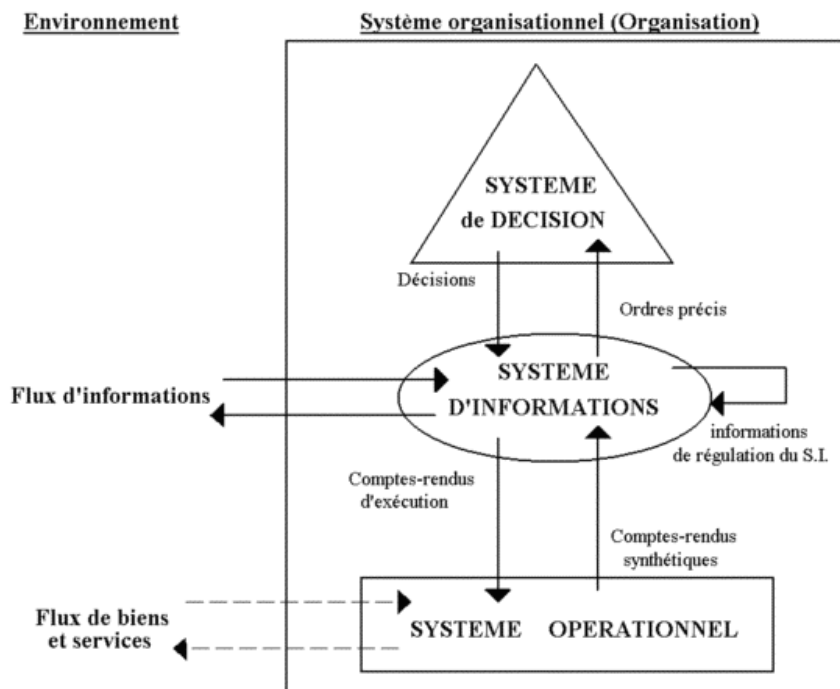
- les entreprises publiques,

- les entreprises privées,
- les administrations,
- les associations...

Système : Un système est un ensemble d'éléments organisés autour d'un but, et dont la modification d'un constituant entraîne la modification d'une partie, ou de l'ensemble des constituants du système. Nous voyons apparaître la notion de cohérence du système, et donc d'interactivité entre les éléments qui le constituent. Nous parlons de constituants en interaction dynamique.

1.4.2. Le système d'informations

Le système d'information doit décrire (nous disons encore représenter) le plus fidèlement possible le fonctionnement du système opérant. Pour ce faire, il doit intégrer une base d'information dans laquelle sera mémorisé la description des objets, des règles et des contraintes du système opérant. Cette base étant sujette à des évolutions, le système d'information doit être doté d'un mécanisme (appelé processeur d'information) destiné à piloter et à contrôler ces changements. Le schéma suivant synthétise l'architecture d'un système d'information.



Le processeur d'information produit des changements dans la base d'information à la réception d'un message. Un message contient des informations et exprime une commande décrivant l'action à entreprendre dans la base d'information. Le processeur d'information interprète la commande et effectue le changement en respectant les contraintes et les règles.

Si le message exprime une recherche sur le contenu de la base d'information, le processeur interprète la commande et émet un message rendant compte du contenu actuel de la base d'information. Dans tous les cas, l'environnement a besoin de connaître si la commande a été acceptée ou refusée. Le processeur émet, à cet effet, un message vers l'environnement.

Relativement à la conception d'un système d'information, l'architecture présentée ci-dessus induit une double conception :

- celle de la base d'information (aspect statique)
- celle du processeur de traitement (aspect dynamique)

Pour aider le concepteur dans ces deux tâches, la méthode Merise propose un ensemble de formalismes et de règles destinées à modéliser de manière indépendante les données et les traitements du système d'information. Ces modèles ne sont qu'une base de réflexion pour le concepteur et un moyen de communication entre les divers acteurs du système d'information dans l'entreprise. Seule la validation de l'ensemble se fera en commun.

1.4.3. Modèle, méthode et analyse

Un modèle est une représentation de la réalité. Le modèle permet de tester un système d'informations, en permettant de tester sa réaction aux divers événements de la vie de l'organisation. La conception d'un ensemble de modèles au sein d'une organisation revient à concevoir l'état futur du système d'informations.

La méthode de METIS raisonnement rusé ou ruses de l'intelligence et HODOS chemin à suivre. Une méthode est la mise en œuvre d'un certain nombre d'étapes (méthodologiques) :

- une démarche,
- des principes,
- des outils
- à traces papiers standardisées,
- à informatiques,
- un vocabulaire

Au sens informatique, l'analyse consiste d'une part à comprendre et modéliser le fonctionnement d'un domaine de gestion d'une organisation, et d'autre part à concevoir la solution informatique adéquate. Nous distinguons deux types d'analyse :

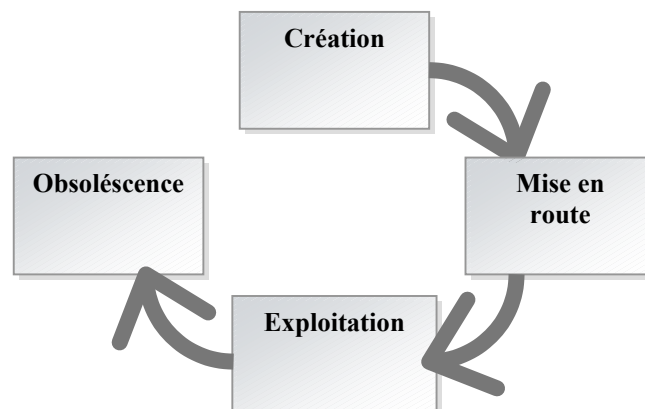
- l'analyse de l'existant,
- l'analyse de conception.

1.5. Les principes de bases de Merise

1.5.1. Le cycle de vie

(La vision que peut avoir l'utilisateur de l'évolution du logiciel)

Ce cycle décrit les diverses phases du système d'informatisation, donc du développement du logiciel de la décision du développement de l'application, jusqu'à sa mort de celle ci.



Le cycle de vie est une préconisation de la manière de conduire le projet de développement d'un système informatique.

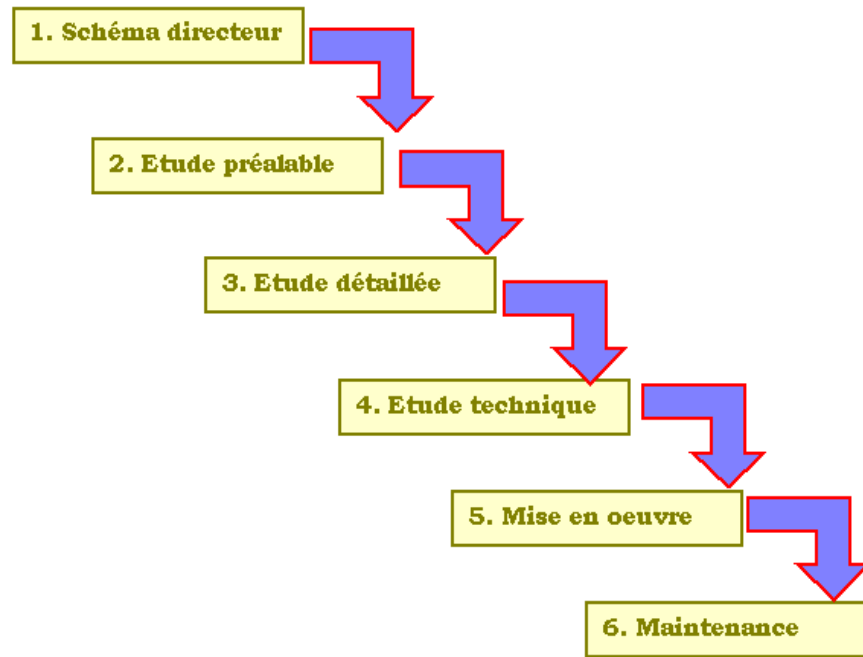
A ce titre, il fournit un procédé (au sens où nous parlons de génie des procédés dans une industrie) et représente un savoir-faire.

Il est remarquable que cette préconisation (empruntée aux méthodes d'informatisation qui ont précédé Merise) se rencontre sous une forme semblable dans la plupart des disciplines de l'ingénieur (par ex. en Génie Civil).

Le projet est découpé en 6 étapes séparées et qui se succèdent dans le temps :

1. **Le schéma directeur** consiste à définir les finalités du système d'information en fonction de la stratégie de l'organisation.
Il définit des priorités qui s'inscrivent dans un découpage en domaines (que certains ont aussi appelé une cartographie) du système d'information.
2. **L'étude préalable** modélise les solutions techniques et organisationnelles susceptible de répondre au problème posé. Elle doit donc passer en revue tous les modèles du cycle d'abstraction, mais selon une portée particulière : un sous-ensemble représentatif du domaine.
Par exemple : s'il s'agit de développer un système de commerce électronique, l'étude préalable pourra consister à étudier de manière approfondie un modèle physique des communications sous l'angle de la sécurité mais se contenter d'un MCD sommaire.
Nous avons déjà rencontré cette idée que la modélisation opère souvent en sélectionnant le noyau dur du problème, nous la retrouverons aussi dans la méthode Jackson. Il ne faut donc pas confondre une étude préalable avec un dégrossissement plus ou moins vague du niveau conceptuel. Elle doit produire les éléments qui permettront de choisir entre différentes solutions organisationnelles, mais aussi techniques.
3. **L'étude détaillée** consiste à fournir les spécifications fonctionnelles du futur système. Soit quelque chose qui ressemble à un Cahier des Charges.
4. **L'étude technique** consiste à produire le logiciel : codage des programmes, leur test et leur mise au point.
5. **La mise en œuvre** consiste à déployer l'application auprès des utilisateurs, mettre en place les moyens humains (formation, recrutement), organisationnels et techniques.
6. **La maintenance** consiste à corriger les erreurs qui n'auraient pas été détectées par les tests et à faire évoluer l'application pour répondre à d'éventuels nouveaux besoins des utilisateurs.
Remarque : que cette étape soit explicitement préconisée par la méthode revient à reconnaître que malgré l'effort important de modélisation et d'abstraction, tout n'est pas définitivement réglé pour autant.

La forme générale de ce cycle de vie est celui de la cascade que nous pouvons résumer ainsi :



Caractéristiques : Le projet est découpé en étapes qui se succèdent dans le temps : nous passons à l'étape suivante quand la précédente est terminée. Selon nous, c'est probablement ce cycle de vie qui a fait de Merise une méthode aujourd'hui obsolète.

Il ne faut cependant pas en oublier les avantages :

- Chaque étape étant clairement définie, nous pouvons lui associer une production documentable et vérifiable : l'étude préalable doit produire un dossier d'avant projet, l'étude détaillée doit produire un cahier des charges, etc.
- Nous pouvons associer à chaque étape une responsabilité humaine (division du travail)
- Nous pouvons planifier et suivre l'état d'avancement du travail, mesurer des écarts

L'inconvénient reconnu est l'extrême difficulté de faire "marche arrière". S'il est certes possible à l'intérieur d'une étape de corriger une erreur, lorsque cette étape est terminée elle l'est définitivement.

Par conséquent si nous détectons une erreur dans les étapes terminales et que celle-ci conduit à remettre en cause une étape amont, il faut refaire tout le travail depuis cette étape amont à travers toutes les étapes intermédiaires qui ont été construites sur la base de la précédente.

C'est peut être ce qui a valu à Merise le sobriquet (injuste cependant) de Méthode Eprouvée pour Retarder Indéfiniment la Sortie des Etudes.

La raison de l'obsolescence de cette manière de faire n'est probablement pas spécifiquement informatique à trois points de vue :

1. Nous nous sommes aperçus dans nombre de secteurs de l'ingénierie qu'il était possible d'être aussi efficace et rapide en adoptant une ingénierie concourante. **Par exemple :** pour construire une usine, commencer l'installation des machines avant que les murs ne soient terminés. Donc ce qui est en cause ici, est un mode d'organisation du travail en équipe : passer du travail à la chaîne aux îlots de production.
2. le cycle de vie en cascade s'appuie sur l'hypothèse que l'avenir est prévisible avec certitude. C'est en particulier la fonction du schéma directeur de Merise lorsqu'il définit des priorités sur une période de l'ordre de trois à cinq ans.

Il faut aujourd'hui raisonner en avenir incertain : que devient une étude en cours de développement pour des besoins internes lorsqu'elle doit tout d'un coup répondre aussi aux besoins d'une filiale qui vient d'être absorbée par l'entreprise ?

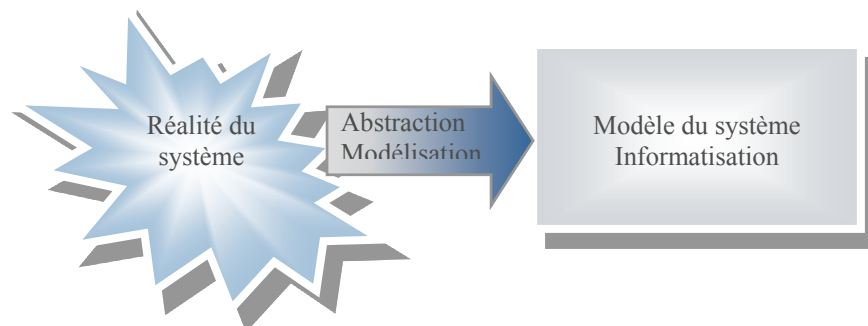
- Le cycle de vie de Merise est en phase avec une époque où l'informatique était propriété exclusive des informaticiens, représentés au plus haut niveau de l'organisation dans une direction informatique. Ils pouvaient s'engager sur des (gros) projets à financement spécifique. L'informatique est devenue propriété des utilisateurs et de leurs services. Les informaticiens doivent développer des (plus petits) projets financés au même titre que d'autres au sein de ces services.

La question devient donc plutôt de faire coopérer un grand nombre de projets diversifiés, et de moins en moins d'en planifier un petit nombre.

1.5.2. Le cycle d'abstraction

(« La vision que peuvent avoir **les informaticiens** de l'évolution du logiciel »)

Ce cycle représente la démarche de spécification du système (*Organisation étudiée*); de l'abstraction de la réalité de l'existant du système, pour en développer un modèle, et l'informatiser.

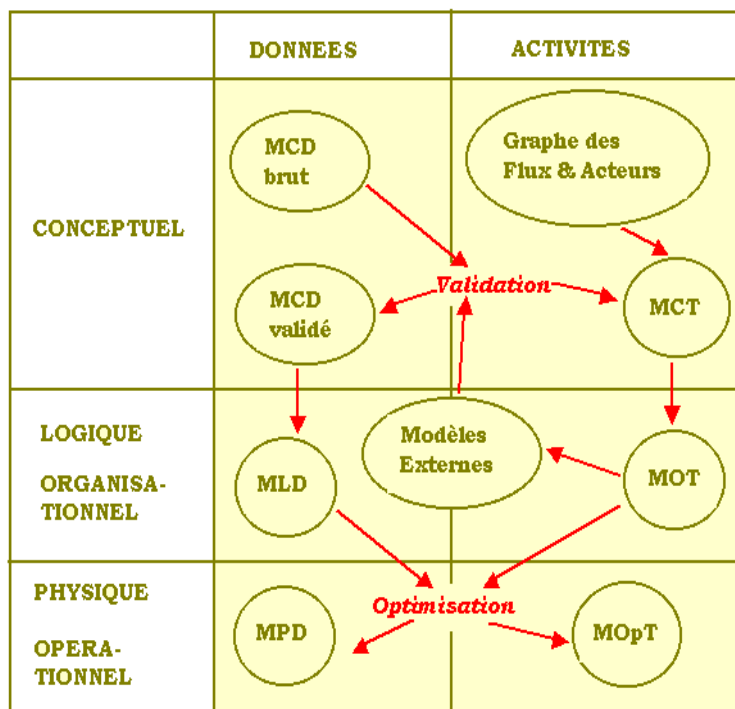


Le cycle d'abstraction : concevoir c'est distinguer des **couches** ou "**niveaux** d'abstraction" au moyen de modèles. L'objectif est **d'isoler l'invariant** du système d'information (sa part la plus stable, celle du niveau conceptuel)

NIVEAU	PORTE SUR	UTILISE	MODELISÉ PAR
CONCEPTUEL	De QUOI s'agit-il ? L'UNIVERS du DISCOURS ou le MONDE REEL	Des règles de GESTION	Un Graphe des Flux et des Acteur Un Modèle Conceptuel des Données (MCD) Un Modèle Conceptuel des Traitements (MCT)
ORGANISATIONNEL ou LOGIQUE	QUI fait quoi ? OÙ ? QUAND ?	Des règles d'ORGANISATION	Un Modèle Organisationnel des Traitements (MOT) Un Modèle Logique des Données (MLD)
OPERATIONNEL ou PHYSIQUE	COMMENT ?	La technologie du moment	Un Modèle Opérationnel des Traitements (MOpT) Un modèle Physique des Données (MPD)

Dans une démarche de construction **partiellement itérative** :

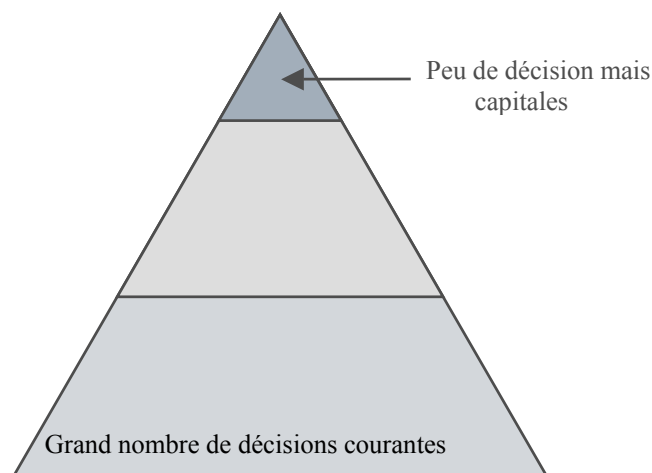
La description séparée des données et des traitements permet de confronter les vues externes issues des traitements pour corriger le modèle de données conçu antérieurement :



1.5.3. Le cycle de décision

- (« la vision que peuvent avoir **les décideurs** de l'évolution du logiciel »)
- Ce cycle représente le point de vue de tous les décideurs de l'organisation étudiée quelques soient leur niveau de décision et d'action (*point de vue d'un décideur au sens large*).

Hierarchie des décisions



Le cycle de décision a pour objectif de hiérarchiser les niveaux de décision en fonction de la nature des choix à faire (du haut vers le bas) :

Choix stratégiques : Décisions qui concernent la stratégie et politique d'entreprise. Elles sont matérialisées dans le schéma directeur par le découpage en domaines.

Choix de gestion : **Par exemple** : quelles sont les règles et qui décide de la répartition des produits dans les entrepôts, des modalités de livraison des commandes clients, de facturation de ces commandes, etc.

Choix d'organisation : Décisions en matière de définition des postes de travail, des procédures à respecter, de leur contrôle

Choix techniques : Windows contre Linux, C++ contre Java, Oracle contre MySQL, etc.

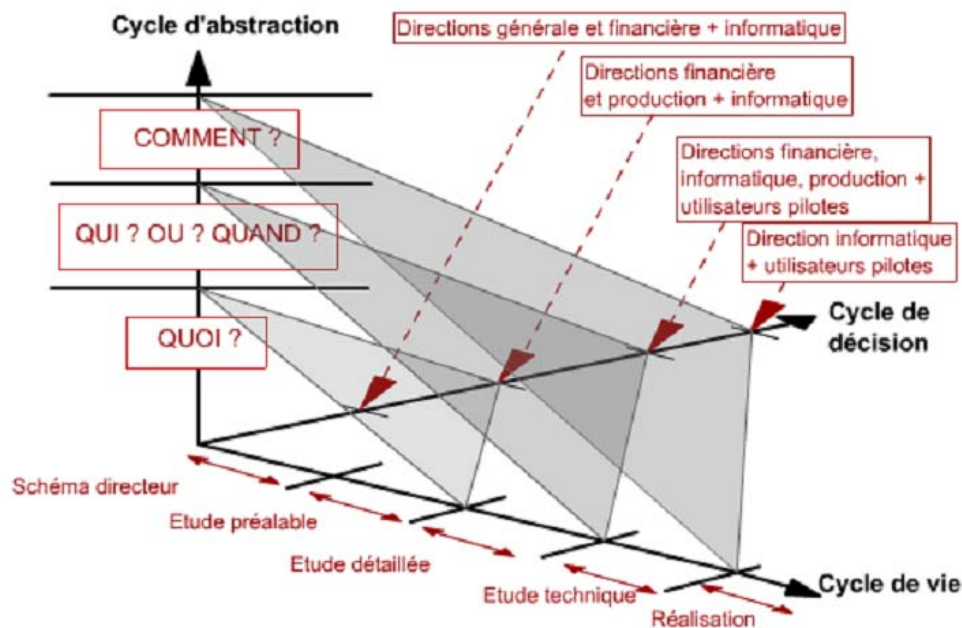
Remarque : L'explicitation de ces règles est essentielle au succès d'un projet informatique. La mise en place d'une application dans un contexte où les rapports humains dans le travail ne sont pas clairement définis dégrade généralement encore davantage les situations (nous savons de moins en moins qui est responsable de quoi).

De ce point de vue l'informatique actuelle est doublement concernée par la question de la qualité :

- d'un point de vue **interne** : garantir la qualité de ses propres productions
- d'un point de vue **externe** : améliorer la qualité de la production de ses utilisateurs.

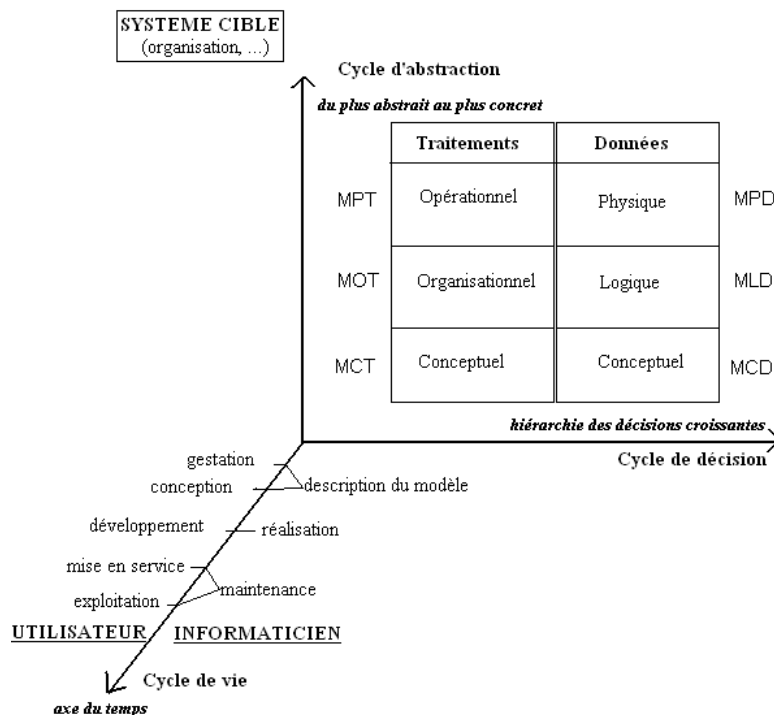
1.5.4. Présentation des 6 (ou 7) modèles Merise

Nous pouvons résumer les différents cycles par une représentation en trois dimensions, qui laisse apparaître les différentes phases de la méthodologie d'analyse MERISE.



Nous notons que dans la méthodologie d'analyse MERISE la description des traitements, et des données est séparée.

L'idée est de permettre la modification d'un niveau sans affecter les niveaux qui lui sont supérieurs. L'objectif est de permettre l'évolution du système d'informations sans obtenir en résultat un système informatique dépendant instable.



1.6. La démarche Merise

1.6.1. La technique

MERISE est une méthode décomposée en phase hiérarchique, qui partent de la notion la plus abstraite de système d'informations, pour atteindre le niveau de spécification le plus concret. Le découpage du système d'informations en sous système que nous nommons les domaines.

Nous effectuons les choix majeurs de gestion, d'organisation, et les choix technologiques.

Nous adoptons une planification de développement en précisant les priorités.

Nous différencions les procédures qui resteront manuelles, de celles qui seront automatisées.

Pour les procédures automatisées, nous discernons celles en mode conversationnelles (temps réel), de celles en temps différé (batch). Définir les postes de travail et les tâches qui y seront accomplies.

Enfin, nous spécifions les aspects techniques finaux (les grilles écran, les formats d'états imprimés, les procédures de contrôle, les contenus conversationnels, ...)

1.6.2. La démarche

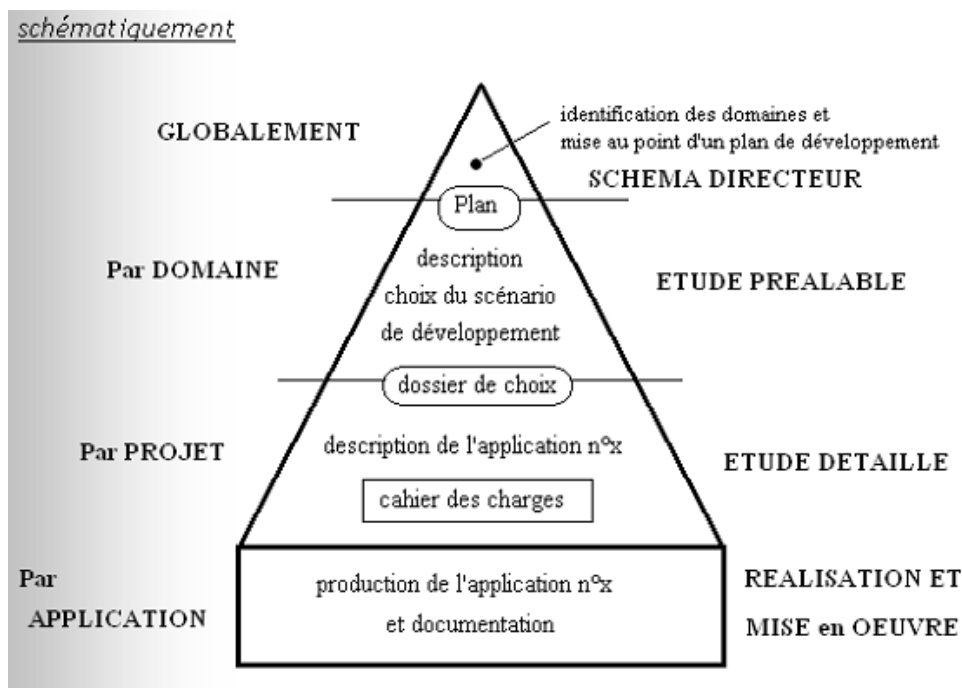
MERISE respecte les concepts définis par le 'Génie Logiciel', ce qui nous permet de définir les étapes suivantes : L'étude préalable à partir ou non d'un schéma directeur.

L'étude détaillée.

La réalisation.

La mise en œuvre.

La maintenance



1.6.3. Les étapes

Il s'agit d'un déroulement données / traitements, selon trois "niveaux" correspondant à trois groupes de questions :

Le "niveau conceptuel" (le "Quoi ?"), aboutissant aux M.C.D. ("Modèle conceptuel des données") et M.C.T. ("Modèle conceptuel des traitements"). A ce stade, données et traitements sont étudiés de manière parallèle, dissociée (en principe par des équipes différentes).

Le "niveau logique" (pour les données) et le "niveau organisationnel" (pour les traitements) (le "Qui?", le "Quand ?", le "Où ?") correspondant aux M.O.T ("modèle organisationnel des traitements") et M.L.D. ("Modèle logique des données"). En fait, l'équipe qui a travaillé sur le M.C.T. passe à un degré de détails beaucoup plus fouillé des traitements (le M.O.T.), donnant, à son tour, une vision "en puzzle" de l'organisation des données. Cette vision est alors confrontée au schéma global qu'est le M.C.D. et de ce rapprochement (parfois houleux), résulte une version corrigée de l'organisation des données, permettant de la finaliser sous la forme du M.L.D.

Enfin, le "niveau physique" (pour les données) aboutissant à la création des tables, et le "niveau opérationnel" (pour les traitements) enclenchant analyse détaillée de chaque traitement, et développements. Ce niveau, dépendant étroitement de chaque contexte concret, ne fait en principe pas l'objet d'exposé dans les ouvrages consacrés à Merise.

L'une des difficultés majeure posées par Merise réside dans ce qu'il est convenu d'appeler la "référence en avant", qui consiste à mentionner des concepts nouveaux ("M.C.D." etc.) qui ne seront expliqués qu'ultérieurement dans l'exposé. En fait, l'ensemble ne prend sa cohérence qu'à la fin !

En fait, les "cycle de vie" et "de décision" avaient pour objet de souligner l'importance: du caractère nécessairement pluridisciplinaire des équipes en charge de la démarche (pour être concret, informaticiens et non informaticiens, à "poids égal") d'une formalisation écrite soignée (et lisible !) des travaux relatifs au cycle d'abstraction ... afin que chacun s'implique, prenne ses responsabilités, par écrit également, lors des étapes de rédaction et de validation.

Il peut être surprenant de qualifier de "cycles" une progression symbolisée par des axes. L'idée était en fait prémonitoire : L'expérience a prouvé que, comme dans tout système, des phénomènes de rétroaction (en clair, la douloureuse prise de conscience d'erreurs révélées par les étapes ultérieures) amenaient à décrire cet enchaînement plus comme des successions de spirales que comme des itinéraires rectilignes.

Par ailleurs, l'idée d'une approche données / traitements d'abord dissociée puis confrontée constitue un des intérêts majeurs de cette méthode.

1.6.4. Périmètre du cours

Même si la méthode MERISE étant, avant tout, une méthode de conception de systèmes d'information, et non de systèmes informatiques, il apparaît aujourd'hui que les systèmes d'information sont largement gérés par des applications informatiques. Les modèles MERISE doivent donc être utilisés pour faciliter le développement de ces applications en s'appuyant sur les technologies logicielles actuelles telles que les bases de données relationnelles et/ou l'architecture client-serveur.

Bien que le MCT est sensé validé le MCD, dans la pratique les concepteurs profitent de la séparation des données et des traitements pour se focaliser sur les données, les traitements étant explicités de façon textuelle dans le cahier des charges.

Nous allons donc nous focaliser sur les données en nous concentrant sur le MCD, MLD et survolerons le MPD.

L'acquisition de compétences au sein des laboratoires Oracle et Microsoft vous permettront d'affiner votre vision du MLD.

2. Le MCD

2.1. Introduction

2.1.1. M.C.D.

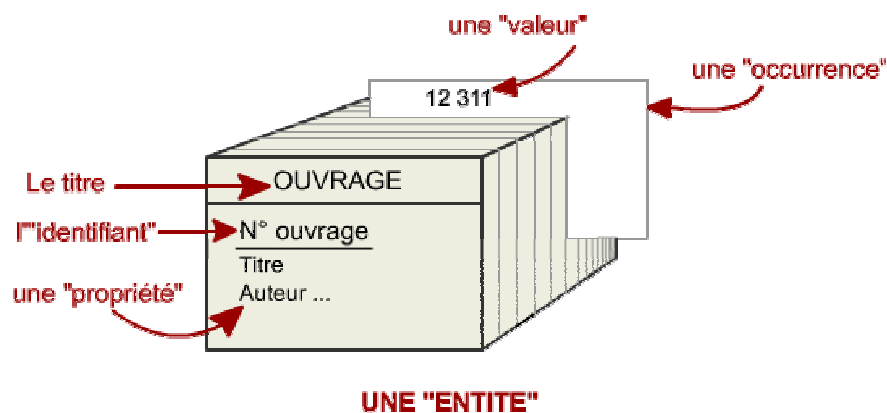
Le MCD ("Modèle conceptuel des données") : Représentation **statique**, sous forme schématique, de la situation respective des **données** d'un domaine de gestion. Ce schéma est conçu pour être **très stable** dans le temps.

Propriété : Le niveau le plus fin en matière de données (dit "insécable"). Ce qui suppose que la base a été épurée des polysèmes et des synonymes.

L'identification des données est parfois loin d'être évidente : Que faut-il choisir : "Adresse" ou "Numéro", "nom de rue" etc. ?

En fait l'usage ultérieur peut guider : Il est certain que si des traitements de sélection, de tri et autres, par rues sont prévus, il est plus rationnel d'isoler d'emblée les "nom de rue" : C'est là, typiquement, un des intérêts de la confrontation ultérieure qui est censée avoir lieu à la fin du M.O.T, et qui aboutira au M.L.D.

2.1.2. Entité



Entité : "Collection de propriétés, dont nous pouvons identifier sans ambiguïté chaque occurrence, grâce à une propriété particulière : l'"identifiant' " : A chaque valeur de cet identifiant correspond **une seule** occurrence. "Pour chaque occurrence, il faut que **toutes** les propriétés prennent **une valeur et une seule**".

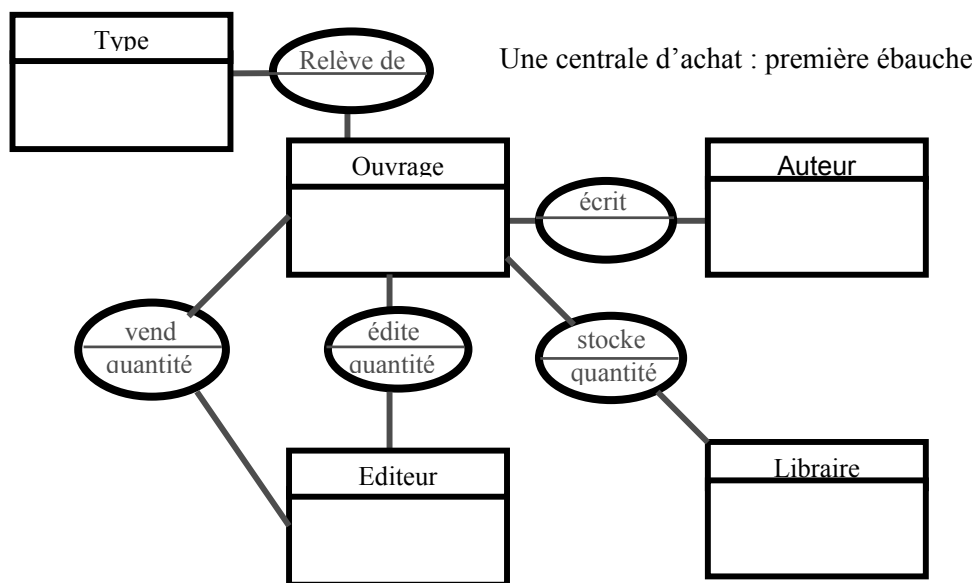
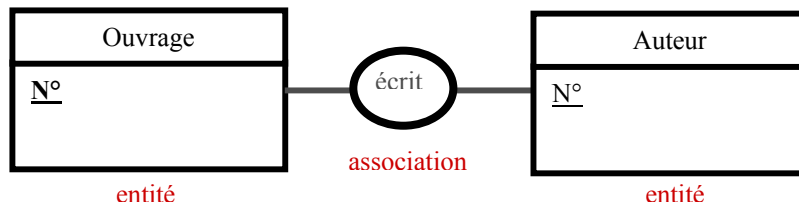
Identifiant : L'identifiant est une propriété de l'entité qui permet l'identification de chacune des occurrences de l'entité. Cette propriété est soulignée dans le MCD.

2.1.3. Association

Association (N.B. : Dans les anciens ouvrages, c'est le terme de "relation" qui est employé) : Objet permettant d'associer deux ou plusieurs entités, et dont chaque occurrence est identifiée par la concaténation des identifiants des entités concernées. L'association n'a donc pas d'existence propre : Elle n'existe qu'en fonction des entités qu'elle relie. Comme pour l'entité, à chaque valeur de cet

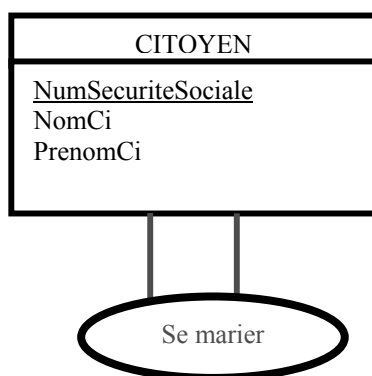
identifiant concaténé, correspond une occurrence et une seule, de l'association. Chaque occurrence de l'association relie une et une seule occurrence de chaque entité participant à cette association.

Le fait de symboliser l'association par un verbe (généralement conjugué) et non par un substantif correspond à une analogie avec la grammaire : L'association joue, en fait, le rôle de "rotule". L'image sera plus claire lors de l'exposé du M.L.D.



2.1.4. Associations particulières

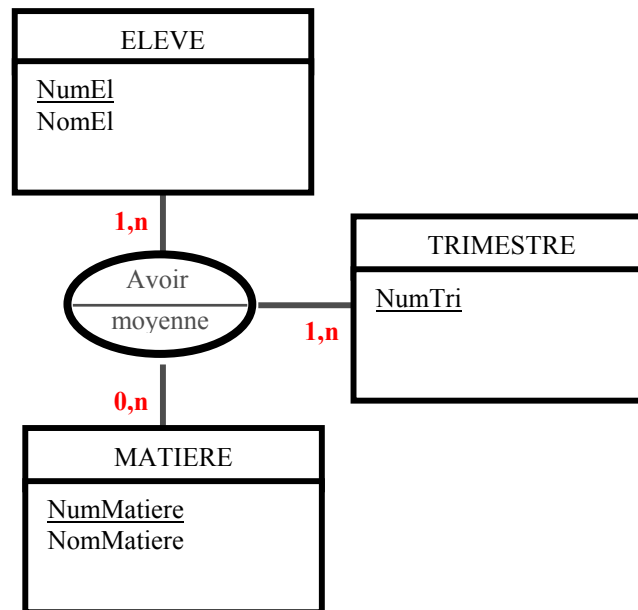
Les associations réflexives : Elle relie une association vers elle-même.



Les associations binaires : Elle relie deux associations. C'est la plus courante.



Les associations ternaires : Elle relie trois associations. Nous pouvons la généraliser aux associations n-aires. Nous pouvons trouver des associations reliant 5 ou 6 entités mais elles sont très rares. En général, essayer de simplifier le modèle en associations binaires ou réflexives.



2.2. Compléter le modèle

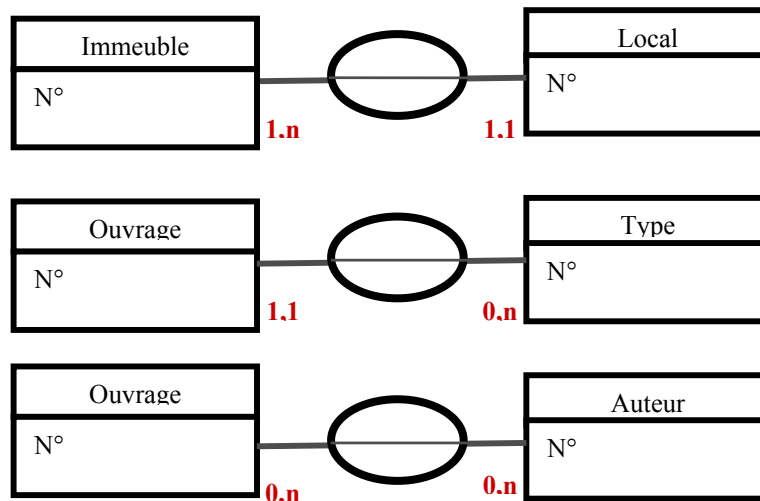
2.2.1. Cardinalités

Cardinalités : C'est le nombre d'occurrences, minimal et maximal, d'une association par rapport à chaque occurrence d'une entité donnée. D'une entité donnée vers une association donnée :

- La cardinalité minimale peut être 0 ou 1.
- La cardinalité maximale peut être 1 ou n.

Il s'agit d'une étape essentielle de la démarche. Des réponses qui y seront apportées dépendront le nombre des tables de la base, l'étendue des fonctionnalités possibles, et certaines contraintes concernant des traitements de première importance pour l'intégrité de la base.

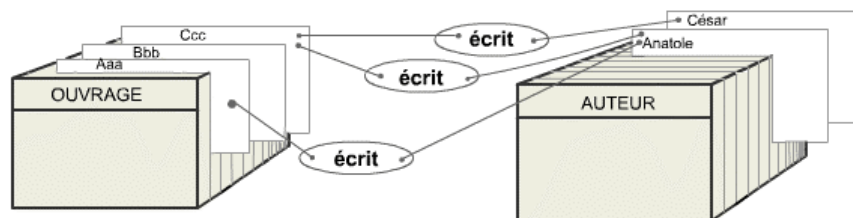
- Une cardinalité minimale à "0" signifie que nous autorisons le cas d'occurrences de l'entité considérée qui ne soient pas reliée à l'association. Par conséquent, une cardinalité minimale à "1" exprimera l'obligation de relier **toutes** les occurrences de l'entité à l'association (ce qui se matérialisera ultérieurement par une contrainte "traitement").
- Une cardinalité maximale à "n" signifie que nous autorisons le cas d'occurrences de l'entité considérée qui soient éventuellement reliées, chacune, à plusieurs occurrences de l'association. Par conséquent, une cardinalité maximale à "1" exprimera l'interdiction du "pluriel".



Immeuble / Local : les deux cardinalités minimales à "1" signifient que nous optons pour une création simultanée d'un immeuble et d'au moins un de ses locaux constitutifs. Les cardinalités maximales expriment le fait qu'un local ne peut appartenir qu'à un leur immeuble (relation d'"appartenance" souvent représentée par un cercle).

Ouvrage / Type : Les minimales expriment le fait que nous pourrions créer des types ne comportant pas encore d'ouvrages rattachés, mais qu'en revanche, lors de la création d'un ouvrage, la relation avec un type sera obligatoire. La cardinalité maximale à "1" de "ouvrage" vers type signifie que nous nous interdisons de rattacher un même ouvrage à plusieurs types (et que dans un cas, il faudra créer des types "mixtes", "historico-policier" par exemple).

Ouvrage / Auteur : Les minimales à "0" expriment, d'une part, que nous prévoyons les ouvrages anonymes, et de l'autre part, que nous admettons la création d'auteurs avant tout enregistrement de ses œuvres.



	Aaa	Bbb	Ccc
Anatole	X		X
César			X

L'ouvrage "Ccc" a été co-écrit par César et Anatole, qui a également écrit "Aaa"
Le détail d'une telle association peut être représenté par un tableau à double entrée

Il ne s'agit pas de déterminer les cardinalités dans l'absolu, d'une manière universelle, mais de fixer les **bornes** du type de gestion que nous voulons pouvoir prendre en charge dans la base de données qui est en construction, reflet d'un **contexte précis**, soit actuel soit futur.

De l'imagination et du savoir-faire que nous aurons dépendra l'**adaptabilité** de la base. Nous pourrions être tenté de prévoir des "pluriels" partout, afin d'ouvrir au maximum le champ des possibilités. Mais, en contrepartie, la gestion des pluriels alourdit la base et les traitements. L'art du concepteur consiste

ici à trouver le meilleur compromis entre la restriction des possibilités et la complexité de gestion de la base.

Cardinalité minimale : "1" Exprime l'obligation de créer simultanément les deux (ou plus) occurrences des entités et l'association (ex. : Une facture comporte un "entête" (comportant les coordonnées du client, la date etc.) et la ou les lignes de facture. L'ensemble représenté par cette facture sera créé au cours d'une même opération, avec toutes précautions pour éviter les enregistrements "orphelins").

"0" peut exprimer des situations très diverses :

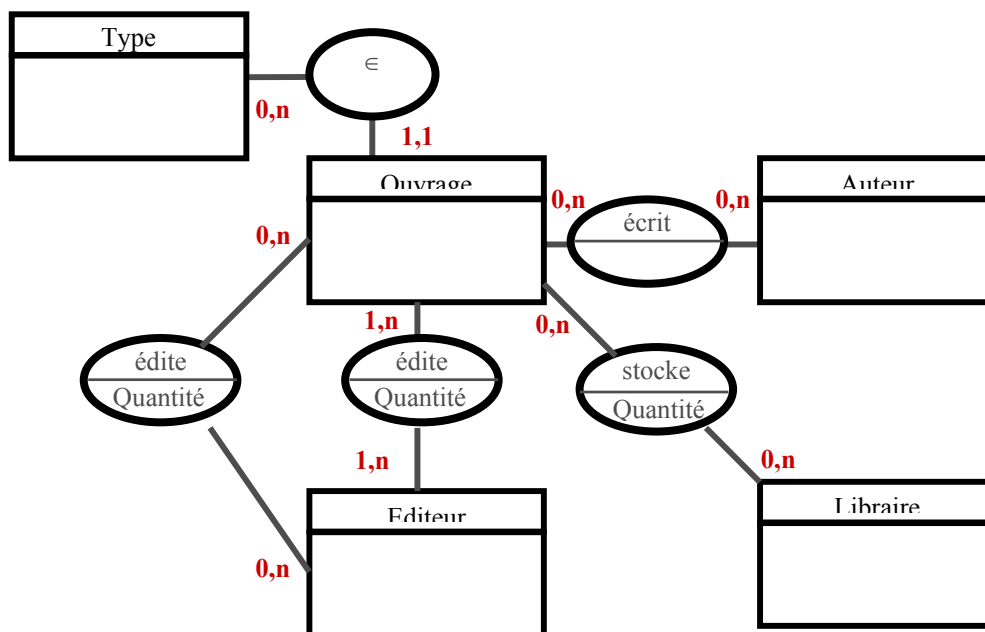
- Soit l'absence définitive de relation (ex. : Un ouvrage anonyme)
- Soit l'absence temporaire de relation : (ex. : Un éditeur veut pouvoir créer dans sa base un auteur, pour disposer de ses coordonnées, alors qu'il n'a pas encore accepté de manuscrit)
- Soit des situations d'alternative ("ou" exclusif)

Ces exemples soulignent l'importance d'un commentaire joint. La détermination des cardinalités minimales donne souvent lieu à des discussions byzantines. Rappelons que ce point ne joue pas de rôle au niveau du nombre des tables ou de leur contenu, mais au niveau des contrôles, lors de l'élaboration des traitements, et que ces questions remonteront à la surface, nécessairement, au moment de la confrontation et de la mise au point du M.L.D. Mais cet exercice a au moins le mérite de discuter, le plus tôt possible, de points qui se révéleront ultérieurement importants.

Les cardinalités maximales :

- "1" : Si la cardinalité maximale de l'autre entité, pour la même association est "n", signifie que les deux entités se trouvent dans une situation "feuille/nœud" (pour reprendre le vocabulaire des arbres de données), dite aussi "hiérarchique", ou encore "d'appartenance" (d'où l'usage de certains auteurs de représenter ce type d'association non par un ovale avec la mention d'un verbe, mais par un cercle contenant le signe "appartient à" ce qui facilite une lecture rapide du M.C.D.).
Si la cardinalité maximale de l'autre entité, pour la même association est "1", signifie, en principe, que la première entité regroupe des propriétés qui, dans certains cas, ne sont pas significatives pour toutes les occurrences de la deuxième entité (ce qui suppose donc que la cardinalité minimale de la deuxième entité est "0")
- "n" : Si la cardinalité maximale de l'autre entité, pour la même association est "n", signifie que les deux entités se trouvent dans une situation de réseau.

Sinon, nous nous retrouvons dans la situation "1 / n" mentionnée ci-dessus.

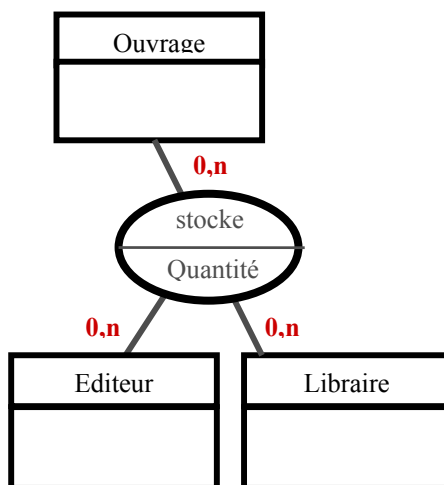


Une centrale d'achat : les cardinalités

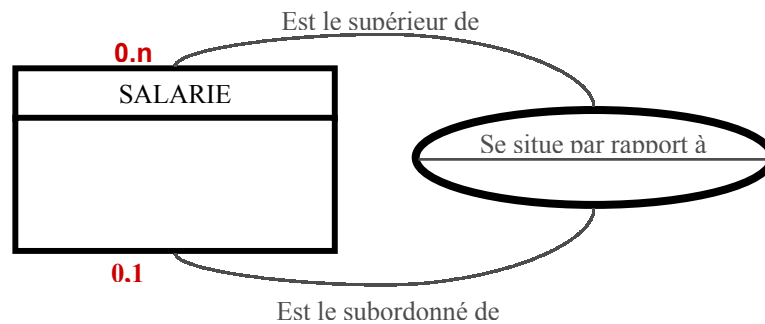
2.2.2. Dimensions

Dimension : La "dimension" d'une association correspond au nombre des entités reliées. Certains auteurs utilisent le terme de "niveau".

Association de dimension 3 : Ainsi, dans l'exemple de la centrale d'achat de libraires, La "quantité", de l'association "Stocké" (ouvrage/libraire) exprime le nombre d'ouvrages d'un même titre, stockés chez un même libraire, tous éditeurs confondus. Si nous optons pour une association "Stocké" de dimension "3" (dont l'identifiant est cette fois : N° d'ouvrage/N° d'éditeur/N° de libraire), la "quantité" devient une information plus précise, puisqu'elle est différenciée, cette fois, en fonction de l'éditeur

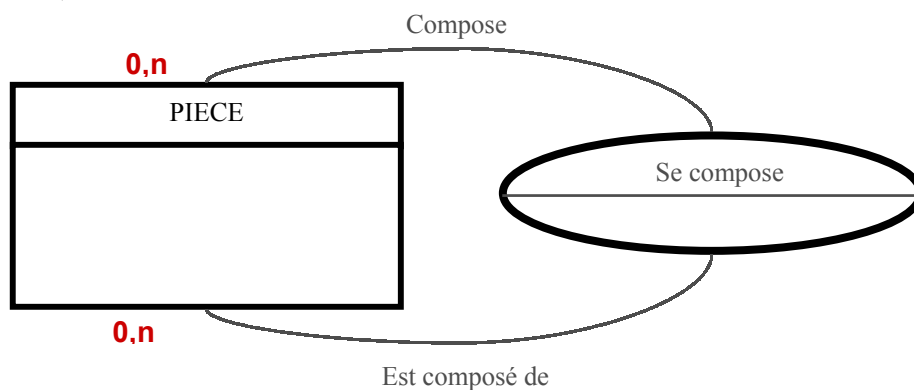


Association de dimension 1 : Les cardinalités "0,n / 0,1" (ce sont les maximales qui comptent) expriment une "hiérarchie" (c'est-à-dire un arbre).



Une «hiérarchie» ('0,n' / '0,1')

Les cardinalités "0,n / 0,n" (ce sont les maximales qui comptent) expriment une "nomenclature" (c'est-à-dire un réseau).



Une «nomenclature » ('0,n' / '0,n')

2.2.3. Les formes normales

Des travaux très antérieurs aux différentes méthodologies (E. Codd 1969) ont permis d'élaborer des règles permettant de rationaliser les bases. Elles peuvent représenter une aide précieuse au stade de la détermination des entités et des associations. Elles peuvent paraître parfois très rigoureuses, mais il faut garder à l'esprit qu'il s'agit d'un premier jet, qui peut être revu ultérieurement pour des raisons d'opportunité, mais en connaissance de cause.

1ère forme normale

Elle est susceptible de concerner toute entité et toute association. Une entité ou une association est dite en "première Forme Normale" si toutes ses propriétés sont :

- "Élémentaires" (non subdivisables, au vu du contexte en question).
Exemple : "adresse" ou "Nom de voie", "code postal" etc.
- "Non répétitives"
Exemple : dans une entité Salarié, il **ne faut pas** prévoir des propriétés "enfant N°1", "enfant N°2" etc. Dès qu'une information est "plurielle", elle doit être "expulsée" et constituer une entité à part (ici une table "enfant"), reliée par une association. Cette règle vaut également pour les informations déductibles les unes des autres. (Ex. le montant TTC, alors que le montant HT est déjà présent et que le taux de TVA est accessible)
- "Significative pour toutes les occurrences"
Exemple : dans une entité Salarié, "voiture de fonction (oui/non)" est significatif pour toutes les occurrences. Mais pas "type de la voiture de fonction".

Commentaire : Dans la pratique, cette 1ère Forme Normale joue surtout un rôle au niveau des entités. Elle permet d'alléger les objets soit par suppression pure et simple de propriétés inutiles (car déductibles), soit par création de nouvelles entités. Elle aboutit donc à fractionner les entités.

2^{de} forme normale

Elle ne concerne que les objets ayant un identifiant concaténé (c'est-à-dire : les associations). Une association est dite en 2^{ème} Forme Normale si :

- Elle est en 1^{ère} Forme Normale
- **Toutes** ses propriétés sont en dépendances fonctionnelle avec **tout** l'identifiant de cette association

Exemple : d'une association "Fournisseur / Produit" :

- identifiant : "N° Fournisseur / N° Produit"
- propriétés : Quantité, Nom du fournisseur

"Nom du fournisseur" est en dépendance fonctionnelle avec la première moitié de l'identifiant. Elle doit donc être supprimée, et figurer dans une entité "Fournisseur".

3^{ème} forme normale

Elle est susceptible de concerner toute entité et toute association.

Une association est dite en 3^{ème} Forme Normale si :

- Elle est en 2^{ème} Forme Normale
- Il n'existe pas de dépendance fonctionnelle entre les propriétés non-identifiantes. (une propriété ne doit dépendre que de l'identifiant, qu'il soit concaténé ou non).

Exemple : Une entité "véhicule des salariés" :

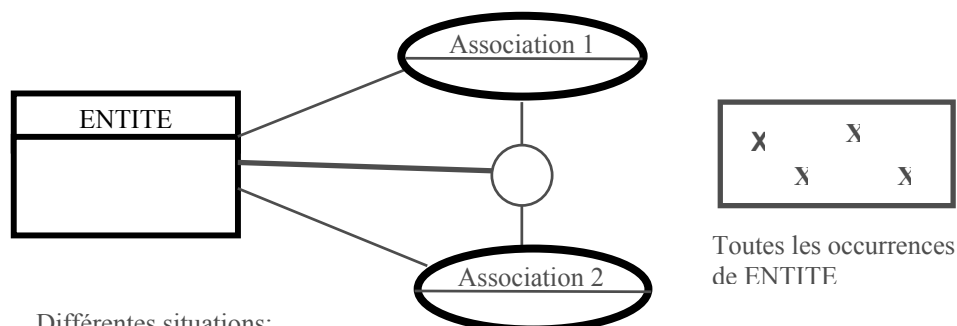
- Identifiant : N° de salarié
- Propriétés : Type de véhicule, puissance du véhicule.

Or "puissance du véhicule" est en dépendance fonctionnelle avec "type du véhicule". Elle doit donc migrer vers une entité à part.

2.3. Contraintes inter-associations

2.3.1. Différentes situations

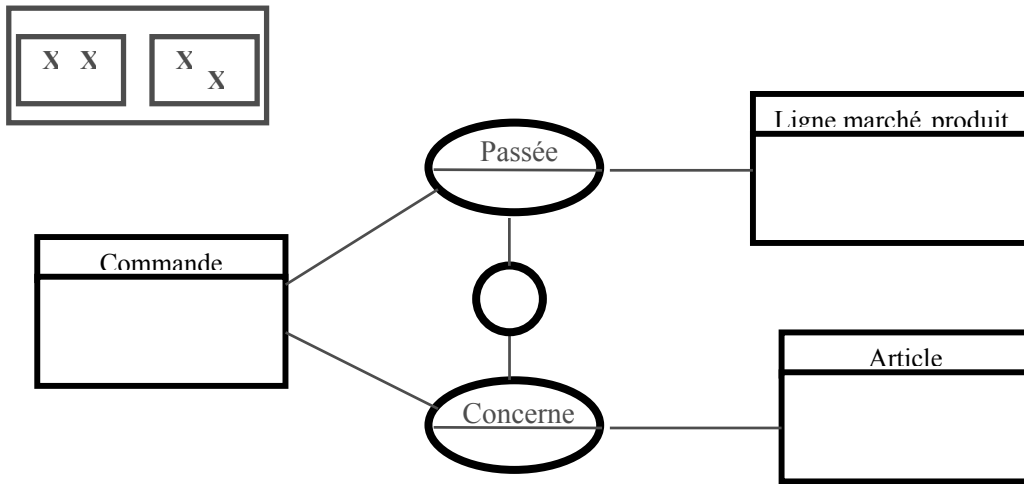
C'est une contrainte exposée lors de la participation d'une entité à plusieurs associations. Nous distinguons trois cas: La partition, l'exclusion, la totalité.



Différentes situations:
Partition
Exclusion
Totalité

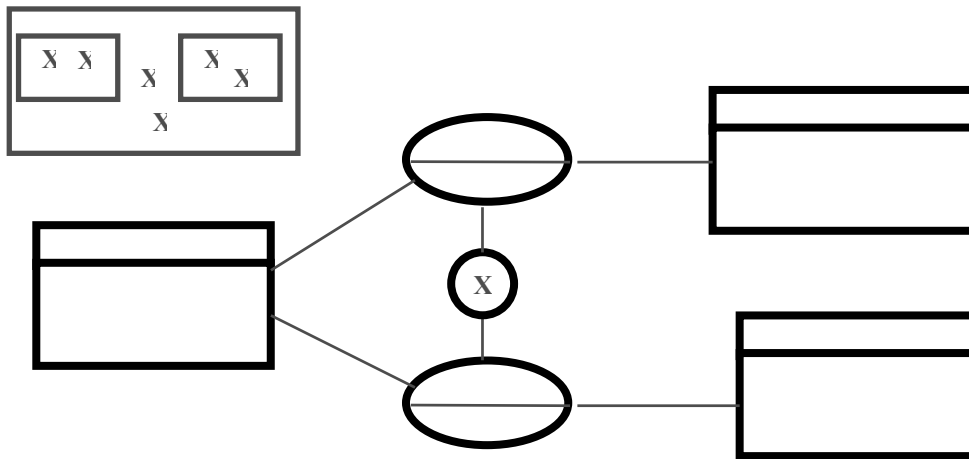
2.3.2. Partition

Chaque occurrence de l'entité participe à l'une ou l'autre des associations.



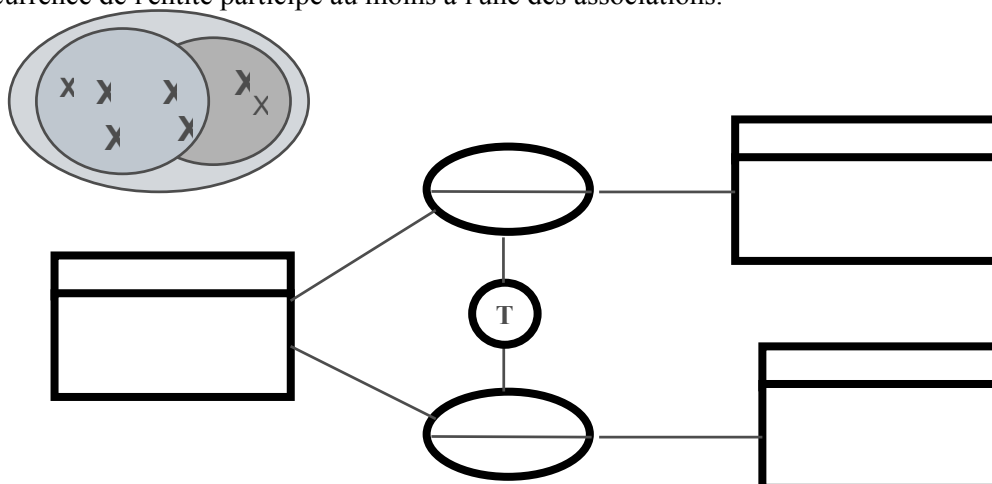
2.3.3. Exclusion

Si une occurrence de l'entité participe à l'une des associations, elle ne participe pas à l'autre.



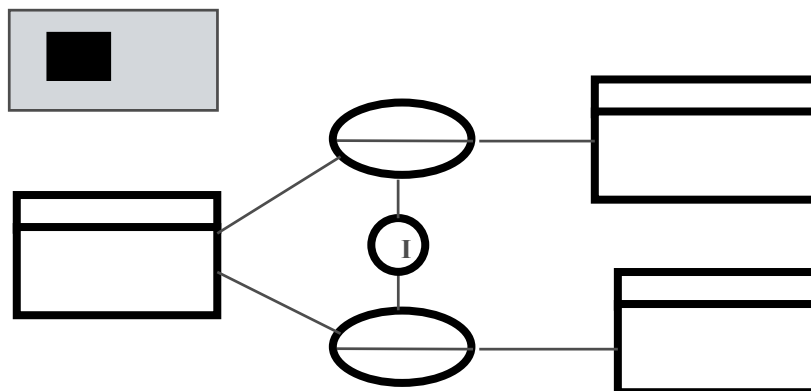
2.3.4. Totalité

Une occurrence de l'entité participe au moins à l'une des associations.



2.3.5. Inclusion

Si une occurrence de l'entité participe à l'une des associations, elle participe obligatoirement à l'autre.



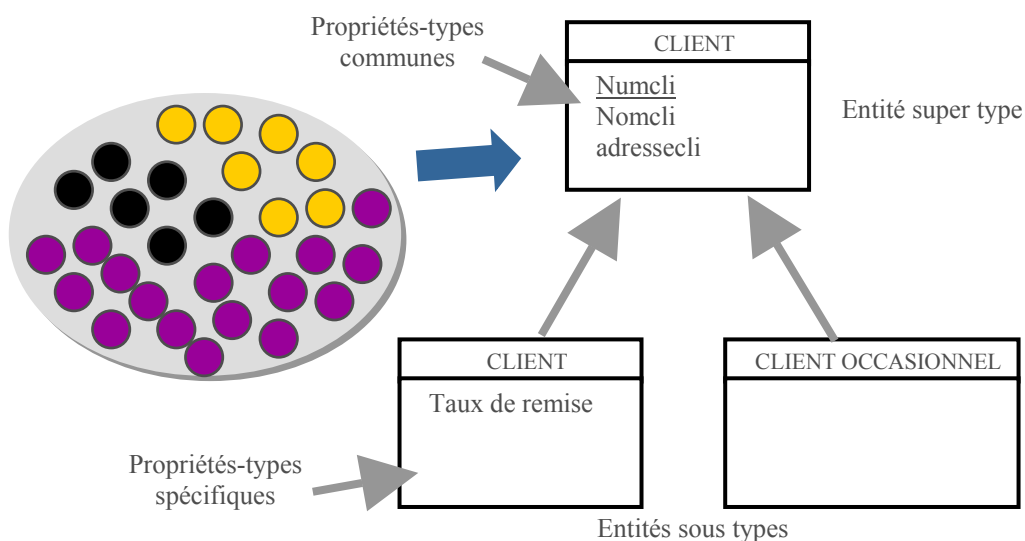
2.3.6. Contraintes sur les propriétés

Sur une propriété :

- forme, liste de valeurs, fourchette de valeurs possibles
- Stabilité (la valeur de la propriété ne change pas au cours du temps)
- Sur plusieurs propriétés d'une même entité ou d'une même association
- $heuredep < heurearr$
- Sur des propriétés d'entités ou d'associations différentes
- Montant d'une commande est égal la somme des montants des lignes de cette commande

2.3.7. Spécialisation/Généralisation d'entités types

Nous retrouvons la notion d'héritage en POO

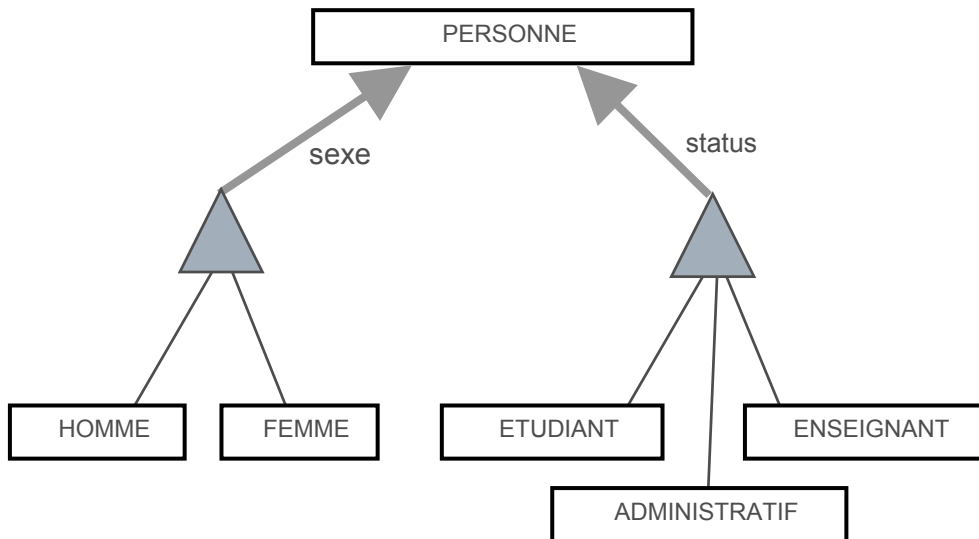


* Les sous-types peuvent avoir leur propre identifiant.

Une spécialisation d'entités-types peut comporter un nombre quelconque de sous-types.

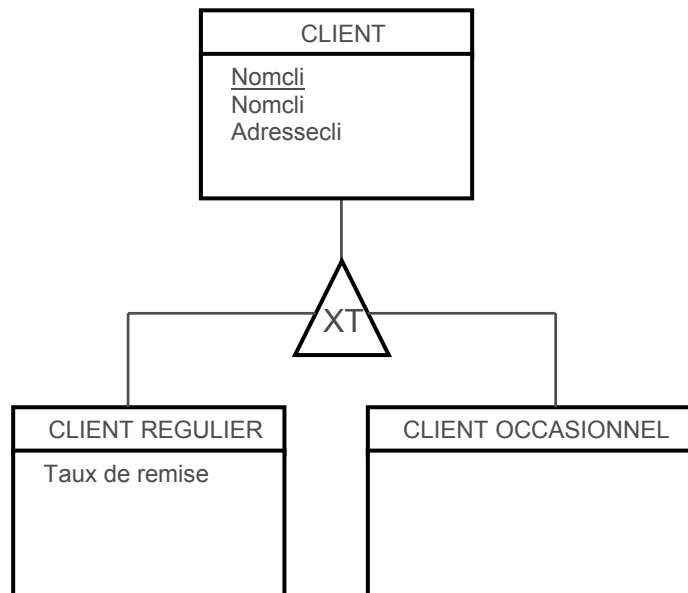
Une entité sous-type d'une spécialisation peut à son tour être super-type d'une autre spécialisation (hiérarchie de spécialisations).

Le découpage d'une population en sous-populations peut s'effectuer sur plusieurs critères; chaque critère produisant une spécialisation en différents sous-types.



2.3.8. Contraintes sur spécialisations

- Exclusivité (X)
- Totalité (T)
- Exclusivité et totalité (XT)



3. Passage du MCD au modèle physique

3.1. Introduction

3.1.1. Le MLD, Le MPD

Après avoir conçu le Modèle Conceptuel de Donnée (MCD), il est maintenant temps de le transposer en Modèle Logique de Données (MLD).

Ce MLD est en fait le dernier pas vers le Modèle Physique de Données (MPD), c'est-à-dire la description de la base qui va être créée.

3.1.2. Démarche

Il s'agit du passage entre le Modèle Conceptuel de Donnée et l'implémentation physique de la base. Le MLD est lui aussi indépendant du matériel et du logiciel, il ne fait que prendre en compte l'organisation des données. Nous parlons aussi de schéma relationnel.

3.1.3. Une table exemple

ID_newsletter	Sujet	DateEnvoie	Contenu	ID_rubrique
25	news 25	11/01/2006	Texte 25	10
26	news 26	12/01/2006	Texte 26	20

3.1.4. Formalisme

```
NEWSLETTER (id_newsletter, Sujet, DateEnvoie, Contenu, #id_rubrique)
```

Chaque enregistrement doit être identifié de manière unique (L'identifiant de l'entité). L'attribut qui permet d'identifier de façon unique chaque ligne est appelé la **Clé Primaire**. Elle peut être composée, c'est-à-dire comprendre plusieurs **attributs**. Ici, il s'agit de l'attribut `id_newsletter`.

La table Newsletter comprend un attribut provenant de la table RUBRIQUES, l'attribut `id_rubrique`. Cet attribut est appelé **Clé Étrangère**.

Dans le formalisme, la clé primaire est soulignée, et la clé étrangère est précédée du signe **#**. D'où l'écriture définitive :

```
MATABLE (Cle Primaire, Colonne1, Colonne2, #Cle_Etrangere)
```

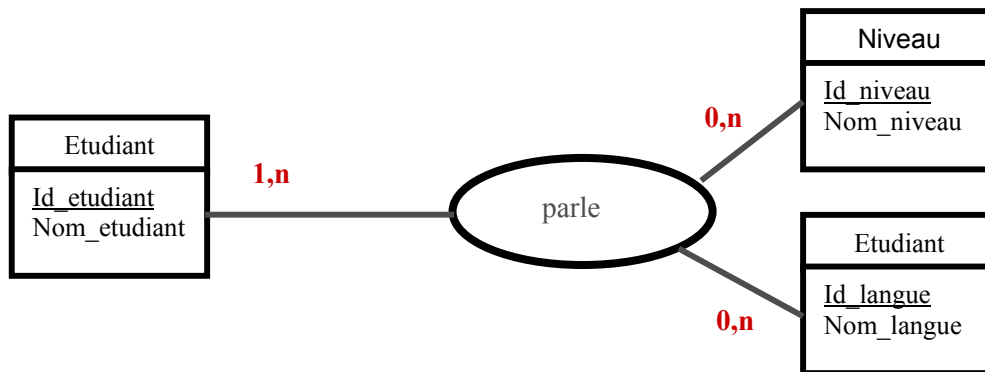
Dans notre exemple :

```
Rubrique (id_rubrique, ...)
```

```
Newsletter (id_newsletter, Sujet, DateEnvoie, Contenu, #id_rubrique)
```

Ici, `id_rubrique` est la Clé Primaire de la table RUBRIQUE, et est une Clé Étrangère dans la table NEWSLETTER.

Il y a création d'une table supplémentaire ayant comme Clé Primaire la concaténation des identifiants des entités participant à la relation. Si la relation est porteuse de donnée, celles ci deviennent des attributs pour la nouvelle table.



```

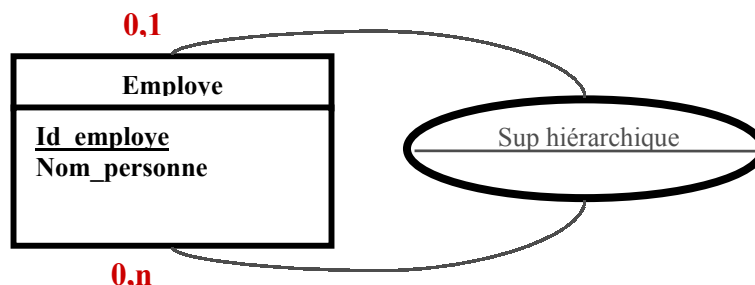
ETUDIANT (id_Etudiant, Nom_Etudiant)
NIVEAU (id_Niveau, Nom_Niveau)
LANGUE (id_Langue, Nom_Langue)
PARLE (#id_Etudiant, #id_Niveau, #id_Langue)
  
```

3.2.5. Règle 5

Association Réflexive.

Premier cas : cardinalité (X,1) - (X,n), avec X=0 ou X=1.

La Clé Primaire de l'entité se dédouble et devient une Clé Etrangère dans la relation ou nouvelle table. Exactement comme si l'entité se dédoublait et était reliée par une relation binaire (X,1) - (X,n) (Cf règle 2).



```

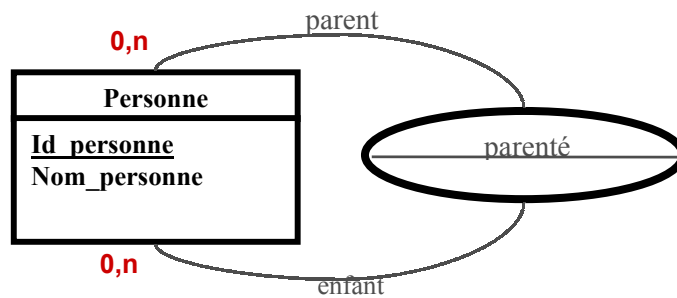
EMPLOYE (id_Employe, Nom_Employe, #id_Sup_Hierarchique)
#id_Sup_Hierarchique est l'identifiant (id_Employe) du supérieur hiérarchique direct de l'employé considéré.
  
```

3.2.6. Règle 5 bis

Association Réflexive.

Deuxième cas : cardinalité (X,n) - (X,n), avec X=0 ou X=1.

De même, tout se passe exactement comme si l'entité se dédoublait et était reliée par une relation binaire (X,n) - (X,n) (Cf règle 3). Il y a donc création d'une nouvelle table.

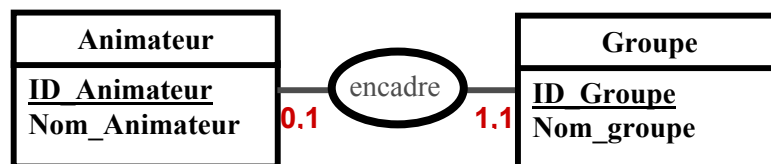


```
PERSONNE (id Personne, Nom_Personne)
PARENTE (#id Parent, #id Enfant)
```

3.2.7. Règle 6

Relation binaire aux cardinalités (0,1) - (1,1).

La Clé Primaire de la table à la cardinalité (0,1) devient une Clé Etrangère dans la table à la cardinalité (1,1)



```
ANIMATEUR (id Animateur, Nom_Animateur)
GROUPE (id Groupe, Nom_Groupe, #id_animateur)
```

3.2.8. Conclusion

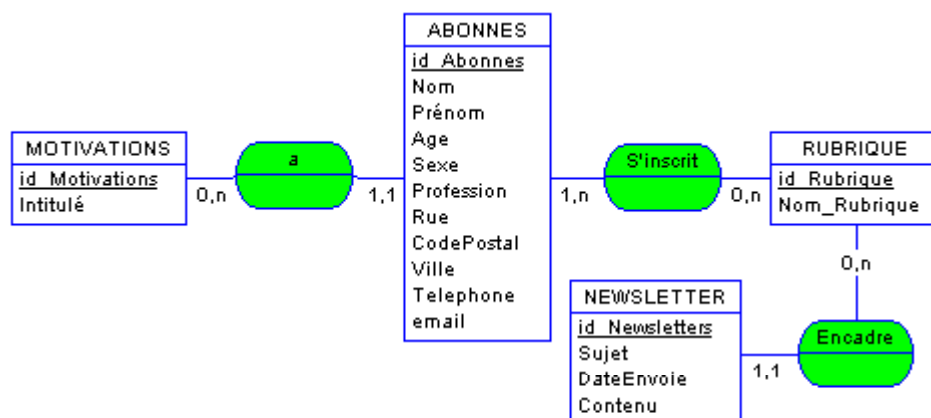
Ces 6 règles représentent TOUS les cas que vous pourrez rencontrer. Il ne faut surtout pas se laisser impressionner par le nombre de schémas, ni se laisser intimider par le côté inhabituel du processus de modélisation. Il est très simple à acquérir. En fait, au bout de quelques modélisations et d'un ou deux développements, vous vous rendrez compte que finalement tout ceci est très logique et d'une évidence rare... Et surtout, votre base de donnée correspondra EXACTEMENT au système d'information décrits dans le cahier des charges. De plus, écrire le MCD, le valider avec votre client, puis en déduire le MLD et donc le Modèle Physique vous fera rentrer complètement dans le projet à réaliser. Vous irez ensuite beaucoup plus vite, avec très peu de risque d'être hors sujet. Après, la majorité du travail restant ne sera plus qu'une question de requêtes, de mise en forme et d'ergonomie, avec une bonne gestion d'Entrée/Sortie de l'information...

Voici les 6 règles du passage du MCD ou MLD :

Règle	Table1	Table2	Table3	Règle
1	x,y			Chaque propriété de l'entité devient un attribut de cette relation, et dont une colonne de la table correspondante. L'identifiant de l'entité devient la <u>Clé Primaire</u> de la relation (elle est donc soulignée)
2	x,1	X,n		La <u>Clé Primaire</u> de la table à la cardinalité (X,n) devient une Clé Etrangère dans la table à la cardinalité (X,1)

3	x,n	x,n		Il y a création d'une table supplémentaire ayant comme <u>Clé Primaire</u> une clé composée des identifiants des 2 entités. Nous disons que la <u>Clé Primaire</u> de la nouvelle table est la concaténation des <u>Clés Primaires</u> des deux autres tables. Si la relation est porteuse de donnée, celles ci deviennent des attributs pour la nouvelle table.
4	x,y	x,y	x,y	Il y a création d'une table supplémentaire ayant comme <u>Clé Primaire</u> la concaténation des identifiants des entités participant à la relation. Si la relation est porteuse de donnée, celles ci deviennent des attributs pour la nouvelle table.
5.1	x,1 – x,n			La <u>Clé Primaire</u> de l'entité se dédouble et devient une #Clé Etrangère dans la relation ou nouvelle table. Exactement comme si l'entité se dédoublait et était reliée par une relation binaire (X,1) - (X,n) (Cf règle 2).
5.2	x,n – x,n			De même, tout se passe exactement comme si l'entité se dédoublait et était reliée par une relation binaire (X,n) - (X,n) (Cf règle 3). Il y a donc création d'une nouvelle table.
6	0,1	1,1		La <u>Clé Primaire</u> de la table à la cardinalité (0,1) devient une Clé Etrangère dans la table à la cardinalité (1,1).

x peut prendre la valeur 0 ou 1
y peut prendre la valeur 1 ou n



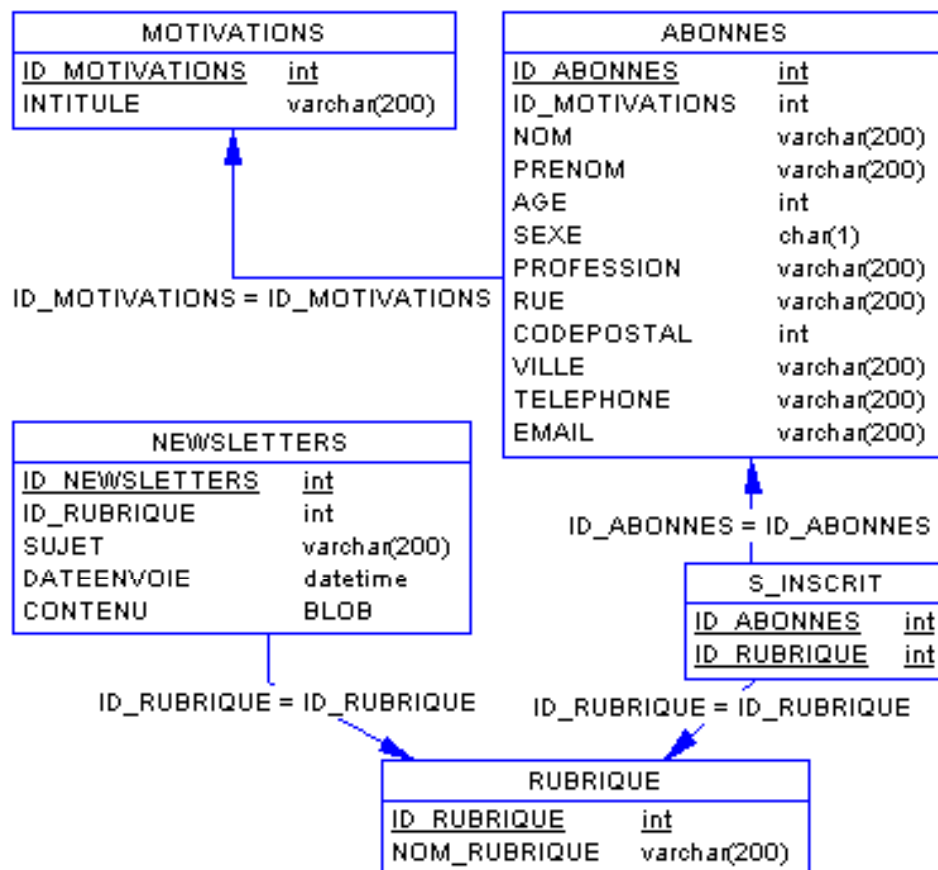
MOTIVATIONS (id Motivation, Intitule)
 ABONNES (id Abonne, #id Motivation, Nom, Prenom, Age, Sexe, Profession, Rue, CodePostal, Ville, Telephone, Email)
 S_INSCRIT (id Abonne, id Rubrique)
 RUBRIQUES (id Rubrique, Nom_Rubrique)
 NEWSLETTERS (id Newsletters, #id Rubrique, Sujet, DateEnvoie, Contenu)

3.3. Le MPD

3.3.1. Pourquoi une étape supplémentaire ?

La dépendance aux bases

Le MPD définit par l'analyse MERISE relatif à la conception des bases de données permettant de définir la mise en œuvre de structures physiques et de requêtes portant sur des données. Le MPD, au contraire du modèle logique (MLD) ou conceptuel (MCD) dépend de la base de données et des détails de l'implémentation.



En effet suivant les bases de données le type de champ n'est pas toujours le même.

Types alphanumériques :

CHARACTER (ou **CHAR**) : valeurs alpha de longueur fixe.

CHARACTER VARYING (ou **VARCHAR** ou **CHAR VARYING**) : valeur alpha de longueur maximale fixée.

Ces types de données sont codés sur 2 octets (EBCDIC ou ASCII) et nous devons spécifier la longueur de la chaîne.

Exemple :

```
NOM_CLIENT CHAR(32)
OBSERVATIONS VARCHAR(32000)
```

NATIONAL CHARACTER (ou **NCHAR** ou **NATIONAL CHAR**) : valeurs alpha de longueur fixe.

NATIONAL CHARACTER VARYING (ou **NCHAR VARYING** ou **NATIONAL CHAR VARYING**) : valeur alpha de longueur maximale fixée sur le jeu de caractère du pays.

Ces types de données sont codés sur 4 octets (UNICODE) et nous devons spécifier la longueur de la chaîne.

Exemple :

```
NOM_CLIENT NCHAR(32)
OBSERVATIONS NCHAR VARYING(32000)
```

Note : La valeur maximale de la longueur est fonction du SGBDR.

Types numériques :



NUMERIC (ou **DECIMAL** ou **DEC**) : nombre décimal à représentation exacte à échelle et précision facultatives.

INTEGER (ou **INT**): entier long.

SMALLINT : entier court.

FLOAT : réel à virgule flottante dont la représentation est binaire à échelle et précision obligatoire.

REAL : réel à virgule flottante dont la représentation est binaire, de faible précision.

DOUBLE PRECISION : réel à virgule flottante dont la représentation est binaire, de grande précision.

BIT : chaîne de bit de longueur fixe.

BIT VARYING : chaîne de bit de longueur maximale.

La stratégie de déploiement :

Table de décision pour le choix l'implantation physique des données par site.

Besoin de données à jour et partagées	Partitionnement horizontal possible	Volatilité importante	Décision à prendre
OUI	OUI		Distribuer
OUI	NON	OUI	Centraliser
OUI	NON	NON	Centraliser ou Copies multiples
NON			Copies multiples Partitionnement vertical si nécessaire

3.3.2. Le reverse-engineering

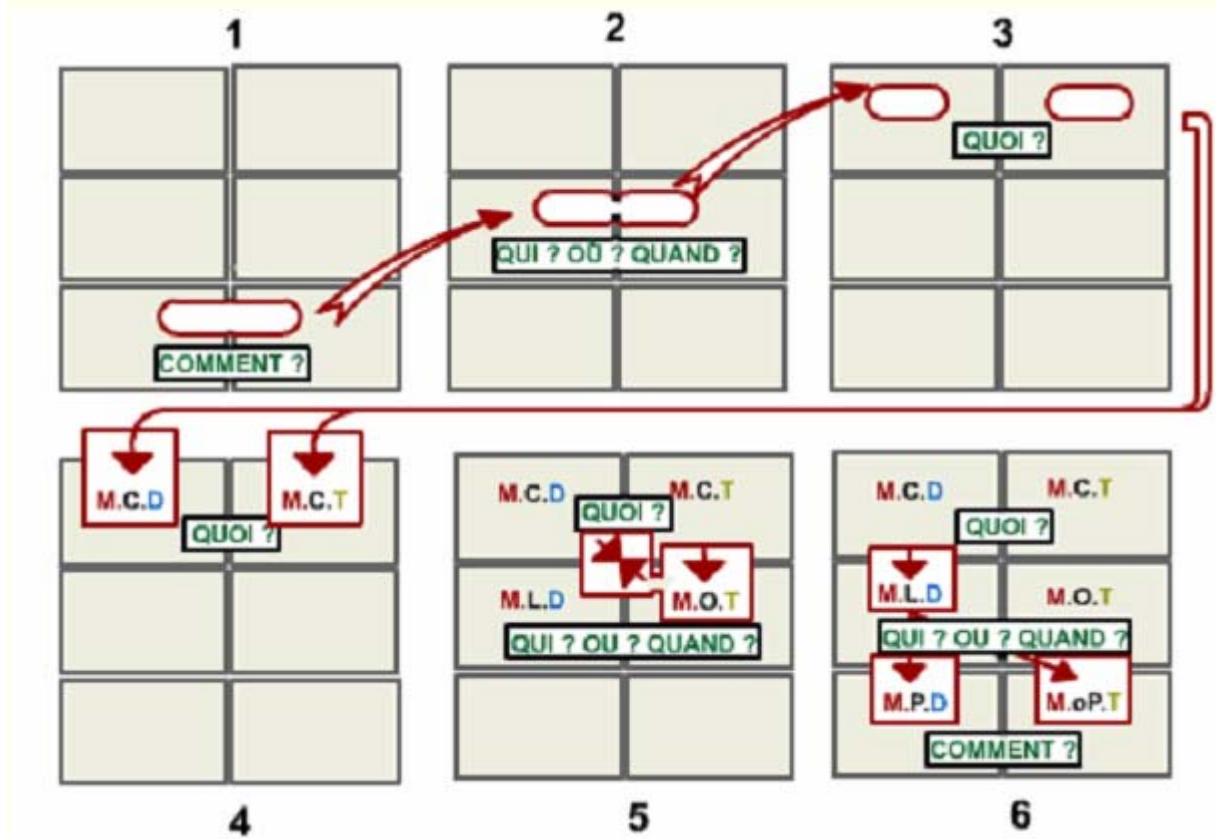
La rétro-ingénierie (traduction littérale de l'anglais Reverse engineering), également appelée rétro-conception, est l'activité qui consiste à étudier un objet pour en déterminer le fonctionnement. L'objectif peut être par exemple de créer un objet différent avec des fonctionnalités identiques à l'objet de départ sans contrefaire de brevet. Ou encore de modifier le comportement d'un objet dont nous ne connaissons pas explicitement le fonctionnement.

La démarche utilisée peut être celle de l'étude d'une boîte noire : nous isolons l'objet à étudier, nous déterminons les entrées et les sorties actives. Nous essayons ensuite de déterminer la réponse du système en fonction du signal d'entrée. Mais il est également possible de démonter le système jusqu'à un certain point pour en analyser les constituants.

La rétro-ingénierie s'applique aussi au logiciel. Ceci peut être réalisé en utilisant des outils d'analyse comme le décompilateur. Les méthodes employées sont similaires à celle du débogage.

Dans le cas de la base de données le Reverse engineering consiste à retrouver la modélisation de départ à partir du modèle physique. Nous passons du MPD au MLD et nous en déduisons le MCD.

Nous appliquons alors les modifications nécessaires sur le MCD et nous recréons le MLD puis le MPD. La plupart des outils de modélisation intègre du reverse engineering.



3.3.3. Les différents outils de modélisation et de génération

PowerAMC est un logiciel de modélisation. Il permet de modéliser les traitements informatiques et leurs bases de données associées. Créé par Powersoft, ce logiciel est produit par Sybase depuis le rachat par cet éditeur en 1995. Hors de France, la version internationale est commercialisée par Sybase sous la marque PowerDesigner.

Power AMC permet de réaliser tous les types de modèles informatiques. A noter qu'il reste un des seuls qui permet de travailler avec la méthode MERISE. Selon Riff News, cela permet d'améliorer la modélisation, les processus, le coût et la production d'applications.

Les différents types de modèles

Modèle	Description
Modèle Conceptuel de Données (MCD)	Fournit une représentation formelle des données nécessaires à la gestion d'une entreprise ou à l'exercice d'une activité professionnelle. Il est ensuite <i>dérivé</i> en modèle physique directement exploitable par un SGBDR.
Modèle Libre (MLB)	Permet de modéliser graphiquement selon les besoins : architecture du système et des applications, scénarios de cas d'utilisation des applications, organigrammes, ou même utiliser sa propre méthode
Modèle Orienté Objet (MOO)	Permet d'utiliser des objets dont l'interaction produit certaines opérations et qui constituent dans leur ensemble un système d'informations. Dans Power AMC, le standard UML est utilisé.
Modèle Physique de Données (MPD)	Spécifie les modalités de mise en œuvre physique d'une base de données. Le MPD est directement dépendant du SGBD cible et permet ainsi de traiter les contraintes relatives à l'accès et au stockage des données.

Modèle XML (MSX)	A l'aide de son diagramme et de son arborescence, une vue globale et schématique de tous les éléments composant le fichier XML est obtenue. Cette vue est très utile pour comprendre, vérifier et modifier la structure complexe d'un fichier XML
Modèle de Fluidité de l'Information (MFI)	Permet de répliquer des objets depuis une base de données source vers un ou plusieurs moteurs de réplication ou bases de données. Ce modèle a pour but de fournir une représentation globale de l'ensemble des réplifications
Modèle de Gestion des Exigences (MGX)	Modèle documentaire utilisé pour répertorier et décrire les besoins du client qui doivent être satisfaits lors d'un processus de développement
Modèle de Processus Métiers (MPM)	Peut être utilisé comme un document de base pour toute analyse orienté-objet, pour décrire les collaborations entre partenaires de même niveau et pour modéliser le processus interne à une organisation
Modèle de Traitements Merise (MTM)	Spécifie la façon dont les données sont traitées dans le système d'information. ces modèles peuvent être importés depuis le logiciel spécialisé Merise <u>Mega</u>

Le référentiel est un outil de Power AMC permettant de gérer le travail en commun, en groupe dans un environnement client/serveur.

Cet outil permet de faciliter ce qu'implique le travail en équipe :

- partage de l'information entre les concepteurs,
- préservation de l'intégrité des données à l'aide de restrictions d'ordre administratif sur le contenu du référentiel.

Il est possible de suivre toute la traçabilité et l'évolution des objets placés dans le référentiel grâce à la gestion par configurations, branches et versions.

DBDesigner 4 est un système de conception de base de données sous forme graphique qui intègre tout les niveaux de conception, de modification, de création et de maintenance sur la structure d'une base de données. Il combine une interface très conviviale avec des outils puissants qui permettent de générer rapidement des scripts SQL ou XML pour créer les bases conçues ou bien permet le reverse engineering sur des bases existantes pour en extraire la structure et en donner une interprétation graphique. Y sont également intégrés des Plugin permettant de générer des reports de structures des bases, ainsi qu'une mini application en PHP de visualisation/modification basée sur des vues de la base (views).

De part ses capacités, il est donc comparable à des applications telles que *Oracle's Designer*®, *IBM's Rational Rose*®, ou encore *theKompany's DataArchitect*®, à ceci prêt qu'il s'agit d'un projet Open Source qui fonctionne sur de nombreux systèmes. Il est distribué sous la licence GPL (General Public Licence).

DBDesigner 4 a été développé et optimisé pour être utilisé avec le SGBD relationnel MySQL, lui aussi disponible gratuitement, pour permettre à tous de développer de puissantes bases de données avec des outils performants.

Le concepteur de DBDesigner4, Michael G. Zinner ayant intégré la société MYSQL AB, le successeur de DBDesigner 4, MYSQL WORKBENCH est en cours de développement.

Lien : <http://www.fabforce.net/dbdesigner4/>

AnalyseSI est l'équivalent libre (licence GPL) du logiciel PowerDesigner (feu PowerAMC) c'est à dire un logiciel d'analyse MERISE permettant de modéliser facilement des bases de données.

Autres outils :

Pour les outils Merise, les principaux, sont : **Mega Development**, **Windev**, **Adelia**, etc.

Un outil RAD comme **Delphi** inclut dans ses éditions haut de gamme des outils de modélisation UML comme **ModelMaker** ou **Together** selon les versions.

D'autres outils existent, orientés plutôt en UML comme **ArgoUML** en OpenSource, ou **Poseidon** qui est gratuit mais pas libre. Citons aussi Devaki écrit en Java/Swing et sous licence GPL.

3.3.4. Migration de bases

La migration de données désigne le processus de transfert de volumes (souvent très vastes) de données des systèmes existants vers de nouveaux systèmes. Les systèmes existants peuvent être très divers, depuis les infrastructures informatiques personnalisées jusqu'aux bases de données autonomes, en passant par les feuilles de calcul.

La migration de données comporte toutes les étapes nécessaires pour nettoyer, corriger et déplacer les données vers le nouveau système. Les évolutions technologiques, les changements de fournisseurs, les mises à jour logicielles ou les projets d'exploration de données (data mining) ou d'entrepôts de données (data warehousing) rendent nécessaires ces opérations délicates et cruciales.

Réduire les risques : Les données étant l'une des ressources cruciales d'une entreprise, il est primordial qu'aucune manipulation n'entraîne de perturbation. Des méthodes et des outils éprouvés sur le terrain, alliés à des plans de migration soigneusement préparés qui exploitent les données existantes et décomposent la complexité de la migration en projets simples à gérer, garantissent la maîtrise des risques.

Améliorer la qualité des données : Des solutions de nettoyage et de correction garantissent une intégrité parfaite des données après qu'elles aient été migrées. Du point de vue de l'utilisateur et de celui du développement, les données migrées sont intégralement optimisées.

Migrer une base sans interruption de service : Garder la couche de communication entre les applications et la nouvelle base de données fait réussir une migration. Reste alors à gérer la compatibilité.

« Chaque seconde d'indisponibilité de SQL Server peut nous faire perdre jusqu'à 45 000 clients », annonce Christophe Laporte, administrateur des bases de données de 123 Multimédia.

Alors forcément, quand il s'agit de faire évoluer son infrastructure vers une version plus moderne de SQL Server, la société, qui crée et commercialise en ligne des contenus (logos, sonnerie, etc.) pour le Minitel, l'Audiotel, le Web, la télévision interactive ou les téléphones portables, n'a pas envisagé un instant d'opter pour une procédure classique de mise à jour.

Dupliquer les bases est moins risqué : En fait, seules les entreprises disposant de versions récentes d'Oracle et de MySQL en cluster sont capables d'effectuer une migration de manière transparente. Grâce à un mécanisme assimilable au grid, ces deux bases savent rediriger automatiquement les utilisateurs vers le nœud d'un cluster pendant qu'un autre est mis à jour. « Dans la pratique, il est beaucoup plus simple de travailler sur une seconde architecture, précise toutefois Sherly Brothier, directeur technique de la société de services IdealX. Ne serait-ce que pour des raisons de sécurité. »

Bien souvent, les entreprises rechignent à effectuer des opérations de maintenance sur des bases en production. De plus, un tel scénario n'est valable que lors du passage à une version plus récente. Il ne l'est pas dans les changements de bases car les mécanismes sont propres à chaque éditeur.

En pratique : Faire migrer une base de données sans interruption de service impose le plus souvent de dupliquer l'architecture. Le principe est simple : les utilisateurs sont maintenus sur l'architecture existante, et basculés vers la nouvelle à un moment donné. La réussite de l'opération dépend moins de la migration des données elles-mêmes que de la maîtrise de la couche de communication entre les

applications et les bases. Autrement dit, c'est une affaire de middleware. Ce qui rend le projet plus ardu.

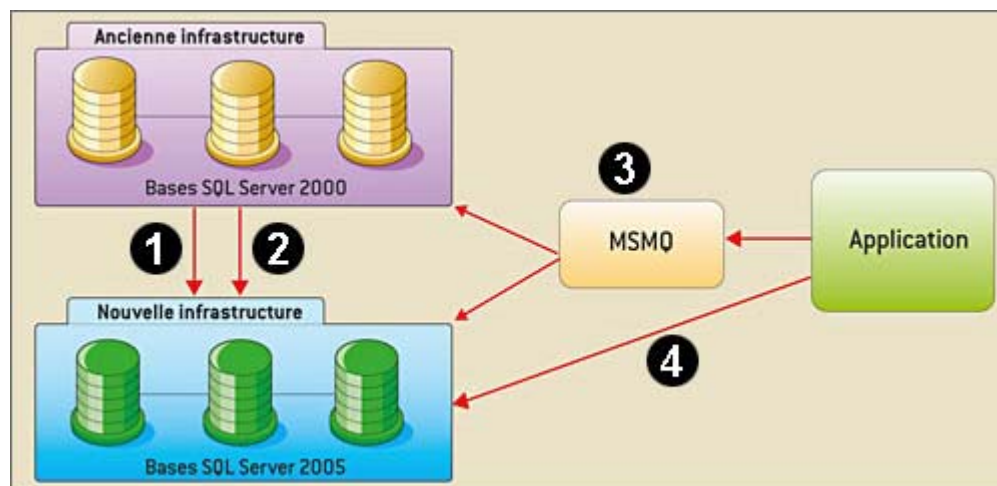
En outre, il se complique souvent en raison des adhérences entre la base et les applications (les appels directs à la base codés directement dans l'application). « La plupart des entreprises sous-estiment l'impact de la migration d'une base de données sur les applications. Pourtant, 70 % du travail ne porte que sur ces dernières », souligne Aomar Bariz, avant-vente pour les produits DB2 et Informix d'IBM. Au moment voulu, les applications critiques et centralisées (PGI, GRC) doivent arrêter de pointer vers l'ancienne architecture et adresser les requêtes à la nouvelle base. Ce qui implique de remplacer l'adresse de l'ancienne architecture de données par la nouvelle dans tous les fichiers de configuration des applications. Une opération qui prend du temps.

6 logiciels de migration de bases

Editeur	Description
Oracle	Migration Workbench Commentaire : Pour récupérer des bases SQL Server, Access, Sybase, Informix Dynamic Server, DB2, et MySQL.
IBM	DB2 Migration Toolkit Commentaire : Pour récupérer des bases Oracle, SQL Server, Sybase, et Informix
Microsoft	SSMA (SQL Server Migration Assistant) Commentaire : Pour récupérer des bases Oracle et, prochainement, DB2 et Sybase.
MySQL AB	MySQL Migration Toolkit Commentaire : Pour récupérer des bases SQL Server, Oracle et Access
Ispirer	SQLWays Commentaire : Conversion entre DB2, Oracle, SQL Server, Sybase, Infomix, MySQL, Progress, SAP, Pervasive, Interbase, etc.
EMS	Data Pump Commentaire : Conversion entre MySQL, PostgreSQL, Interbase, SQL Server, DB2, et DBISAM.

2 Exemples de migration :

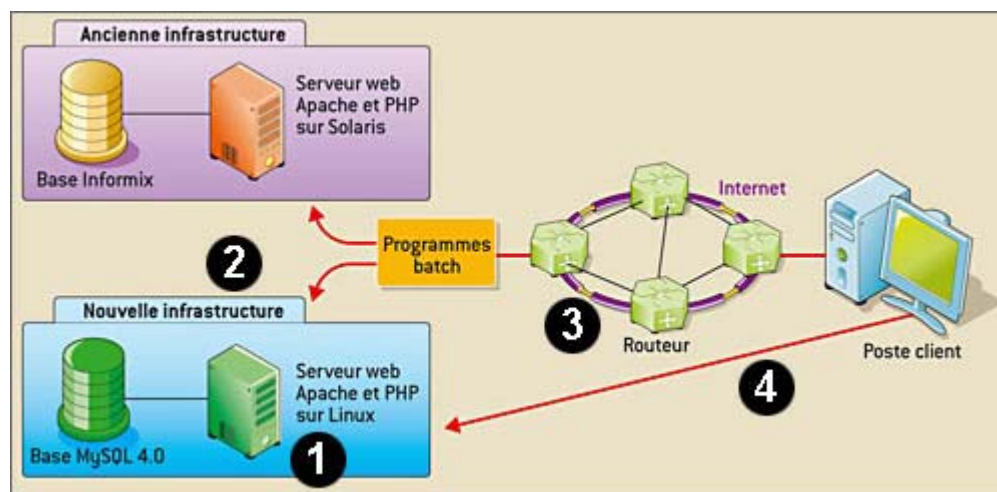
123Multimédia



Etape	Description
1. Les données sont sauvegardées	123 Multimédia a construit une architecture constituée de plusieurs bases cibles avec la nouvelle version de sa base de données SQL Server 2005. Elle a ensuite sauvegardé toutes les données présentes sous l'architecture existante (plus de 600 Go) et transféré la sauvegarde sur la nouvelle architecture
2. La base cible est	Des modifications sont intervenues sur la base en version 2000 pendant la

mise à jour	phase de sauvegarde et de transfert, qui a nécessité entre cinq et six heures. La société procède donc à une mise à jour ne transférant que la différence. Une opération réalisée en moins d'une minute.
3. Les transactions en attente sont suspendues et stockées	Les applications qui écrivent dans la base sont aussitôt coupées de l'ancienne architecture, les transactions internes ou issues du web étant stockées en file d'attente par l'infrastructure <i>middleware</i> MSMQ (messagerie asynchrone). Celles en lecture seule restent branchées sur l'ancienne architecture.
4. Les fichiers sont transférés vers la nouvelle architecture	Remplacement de l'adresse IP de l'ancienne architecture par la nouvelle dans les fichiers de configuration des applications. Les transactions en attente dans MSMQ ont été adressées à la nouvelle architecture, le basculement n'ayant provoqué en fait qu'un léger et très court ralentissement.

Meteo Consult



Etape	Description
1. Le nouveau <i>back office</i> est installé	Météo Consult a redéveloppé son application et changé d'environnement d'exploitation pour tirer pleinement profit de sa nouvelle base MySQL. Elle a construit une architecture en parallèle.
2. Une double alimentation des données est instaurée	Les bases sont alimentées plusieurs fois par jour par des données externes, qui, avant d'être injectées dans la base, subissent traitements de contrôle corrections effectués par les prévisionnistes météo. Les programmes batch d'alimentation ont été reprogrammés pour alimenter les deux architectures.
3. Les routeurs sont mis à jour	En temps normal, les routeurs conservent en mémoire l'adresse IP d'un nom de domaine pendant 24 ou 48 heures. En prévision du changement d'adresse, Météo Consult a pris les devants et reconfiguré son nom de domaine une semaine avant le basculement.
4. L'ancienne base est basculée vers la nouvelle	Remplacement de l'ancienne adresse IP par la nouvelle dans les tables de routage. Pendant les dix minutes nécessaires aux modifications, les utilisateurs étaient orientés vers la nouvelle ou l'ancienne architecture. Un léger ralentissement : cinq minutes après la fin des modifications, tous les routeurs disposaient de la nouvelle adresse.