

Introduction

Répondre mieux et plus vite aux besoins des utilisateurs finaux en assurant la maîtrise des coûts, tout en garantissant la pérennité des investissements, voilà les impératifs majeurs auxquels doivent répondre les décideurs dans tous les domaines.¹

L'informatique se caractérise de plus par une forte évolutivité des besoins et des techniques qui conduisent à une visibilité restreinte sur l'avenir. Il faut donc absolument prendre en compte les points clés suivants :

- Les demandes des utilisateurs se renouvellent et s'accroissent, en quantité comme en qualité.
- Les matériels et les logiciels de base sont en mutation permanente et rapide.
- Les ressources pour satisfaire les demandes diminuent plus qu'elles n'augmentent.

La mise en place d'un système informatisé, c'est à dire l'informatisation du système d'information (SI) de l'entreprise ne peut être efficace que si celle ci est dirigée du début à la fin par une méthode rationnelle et compréhensible par les différents acteurs qui interviennent tout au long de la vie du projet.

Par la définition d'un certain nombre de règles strictes, la modélisation du système à l'aide de la méthodologie Merise garantie d'une part la cohérence des données et traitements et donc la validité d'une application et d'autre part la pérennité des développements réalisés grâce une "maintenabilité" accrue.

L'utilisation d'une telle méthode est donc primordiale pour tout développement informatique actuel afin d'assurer la cohérence entre le cahier des charges initial, issu de la volonté des dirigeants et l'application livrée à l'utilisateur final.

1 « Les méthodes d'analyse et de conception des systèmes de production », Henri Pierraval .P :36

I- **Les modèles de conception des SI :**²

La conception d'un système d'information n'est pas évidente car il faut réfléchir à l'ensemble de l'organisation que l'on doit mettre en place. La phase de conception nécessite des méthodes permettant de mettre en place un modèle sur lequel on va s'appuyer. La modélisation consiste à créer une représentation virtuelle d'une réalité de telle façon à faire ressortir les points auxquels on s'intéresse.

Ce type de méthode est appelé *analyse*. En effet il existe plusieurs méthodes de conception des systèmes d'information qu'on peut présenter comme suit :

1. **La méthode cartésienne :**

Issues des méthodes d'analyse [années 60]. Elles consistent à diviser le problème en éléments simples, étudiant chaque élément puis les réunissent à nouveau. Ces méthodes associent en général la démarche par étapes à une approche fonctionnelle du SI. Le SI est abordé par les fonctions qu'il doit assurer. Exemple de méthodes [60-70] Minos, Ariane, Corgi. Sous l'influence de la programmation structurée, certaines méthodes anglo-saxonnes ont adopté [80] une démarche de décomposition fonctionnelle descendante telle que SADT et Yourdon.

2. **Méthodes à approche par les données :**

L'approche par les données, dans laquelle les premiers concepts analysés sont ceux de rubrique, de relation et d'entité, aboutit à un modèle initial du système qui est le modèle conceptuel de données, véritable carte de l'entreprise sur laquelle on va spécifier les applications en parcourant les mêmes chemins d'accès que les fonctions. La suite de la phase de conceptions est consacrée à étudier les opérations capables de modifier l'état de chacune des rubriques, relations et entités ou de les consulter.

3. **Méthodes à approche par les traitements :**

L'approche par les traitements, dans laquelle on doit commencer par étudier les concepts d'activité et de règles d'exécution, conduit à déterminer les constituants de chaque traitement de la façon dont il va s'exécuter (avec quelles données, selon quel algorithmes, pour quels résultats) à partir d'un modèle initial de traitement établi sur la base d'une liste des fonctionnalités attendues.

4. **Méthodes à approche par les résultats :**³

2 MERISE - Initiation à la conception de systèmes d'information. D. Murey

3 merise.developpez.com

Cette approche consiste à débiter l'analyse par le recensement des lots résultats, c.à.d. des ensemble de données à produire. A partir de la structure hiérarchique des données d'un état ou d'un écran que l'on détermine le traitement à effectuer, l'algorithme qui le compose et les fichiers ou base de données, autrement dit les rubriques et les entités nécessaires à l'exécution du dit algorithme.

5. **Méthodes systémiques :**

Nées en 70. L'essentiel réside dans la compréhension du SI, l'importance de la décomposition du système en éléments et sur la mise en évidence des relations entre éléments.

Le SI couplé d'une part au système opérant dont il est la représentation cohérente, complète, actuelle, non redondante et structurée, d'autre part, au système de pilotage dont il est le support. L'approche retenue est conceptuelle. Concevoir le SI, c'est construire un modèle de la réalité organisationnelle qui est à la fois une image et une vue abstraite des collections de données et des programmes qui seront implantés sur l'ordinateur.

II- **La méthode Merise :**

1. définition

S'inscrivant dans les méthodes systémiques, MERISE est une méthode d'analyse destinée à concevoir et à développer des systèmes d'information. Elle conduit à recenser et à décrire toutes les informations nécessaires au bon fonctionnement de l'entreprise, que ces informations soient utilisées manuellement ou par des machines.

Cette méthode date de 1978-1979, et fait suite à une consultation nationale lancée en 1977 par le ministère de l'Industrie dans le but de choisir des sociétés de conseil en informatique afin de définir une méthode de conception de systèmes d'information. Les deux principales sociétés ayant mis au point cette méthode sont le CTI (Centre Technique d'Informatique) chargé de gérer le projet, et le CETE (Centre d'Etudes Techniques de l'Equipement) implanté à Aix-en-Provence.

Dès lors, MERISE est reconnue en tant que standard de l'ingénierie informatique car la norme AFNOR Z67-101 dite "Recommandations pour la conduite des projets informatiques" en reprend la trame

2. Les objectifs⁴

Au plan de la conception et de la réalisation, les objectifs de MERISE sont les suivants:

1. Etre utilisable dans tous les organismes, publics ou privés, quelles que soient leur taille et leur automatisation antérieure.
2. Etre applicable à toutes les solutions techniques connues : traitement par lots ou transactionnel avec base de données ou fichiers classiques, informatique centralisée ou répartie, langage de quatrième génération (L4G) ou non...
3. Prévoir dès le départ l'évolutivité des traitements.

⁴ Introduction aux Systèmes d'Information (cours 1), Jean-Louis Guénégo

4. Eviter les retours en arrière, qui sont toujours coûteux.
5. Permettre une utilisation partielle, évitant ainsi toute lourdeur pour résoudre un problème particulier.
6. Définir des invariants à différents niveaux afin d'assurer de solides points d'ancrage lors de l'évolution des systèmes et d'éviter ainsi le syndrome du château de cartes.

3. Les principes

Merise se caractérise par une double démarche : **par niveaux** et **par étapes**.

La démarche par niveaux a pour objectif de formaliser le futur système tant sous ses aspects techniques et organisationnels que sur le plan des règles de gestion et de la stratégie d'entreprise.

Celui de la démarche par étapes est de hiérarchiser les décisions au cours de la conception, du développement, de la mise en œuvre et de la généralisation d'un nouveau système d'information mais aussi lors de son évolution.

Cette double démarche permet de maîtriser les risques (coûts, délais, effets sur le personnel) et les enjeux (efficacité, productivité). Elle favorise également l'introduction de nouvelles technologies (bases de données relationnelles ou objets, systèmes experts...) et apporte une aide au règlement des problèmes sociaux, économiques ou techniques (réarrangement des postes de travail, responsabilisation...). Enfin, elle facilite l'évolution du système d'information en redéfinissant les modalités de pilotage et en prenant en compte les besoins nouveaux.

I. III LES DEMARCHES DE MERISE

1. LA DEMARCHE PAR NIVEAUX

C'est un des points forts de MERISE par rapport aux méthodes antérieures. Cette démarche intègre les **propositions** du comité de normalisation ANSI SPARC⁵ qui a défini les différents niveaux ainsi que les techniques de représentation utilisées dans les réseaux de PETRI pour ce qui concerne la dynamique des traitements.

Les trois niveaux sont, comme pour l'ANSI SPARC, les niveaux conceptuel, logique et physique. Mais, à la différence de l'ANSI SPARC qui privilégie les données, MERISE propose une approche parallèle pour les données et les traitements ce qui conduit à disposer en fin de conception des modèles suivants:

NIVEAU	DONNEES	TRAITEMENTS
CONCEPTUEL	Modèle Conceptuel des Données (MCD)	Modèle Conceptuel des Traitements (MCT)

5 ANSI SPARC : L'architecture ANSI-Sparc est divisée en trois niveaux, celui du schéma interne (SI), celui du schéma conceptuel (SC) et celui des schémas externes (SE).

LOGIQUE	Modèle Logique des Données (MLD)	Modèle Organisationnel des Traitements (MOT)
PHYSIQUE	Modèle Physique des Données (MPD)	Modèle Opérationnel des Traitements (MOPT)

1.1. Le niveau conceptuel

A ce niveau sont représentées les informations et leurs relations d'une part ainsi que les utilisations qui en sont faites et les contraintes associées d'autre part. Elles sont définies en faisant abstraction de toutes les contraintes liées à l'organisation de l'entreprise. C'est la réalité perçue par l'entreprise dans son ensemble qui est exprimée et ce sont les finalités de l'organisation et ses modalités de réaction aux sollicitations de son environnement qui dictent les choix. C'est donc la partie la plus stable du système puisqu'il faut une remise en cause de la finalité de l'organisation ou de ses modalités de fonctionnement pour qu'il soit nécessaire de la modifier. Cette partie est celle qui répond à la question "QUOI ?".

➤ 1.1.1/ Les données

Leur description fait appel au formalisme entité-association aussi appelé entité-relation ou individuel. C'est la traduction du monde dans lequel se meut l'entreprise en termes d'individus (ou entités) et de relations. Ainsi, des clients (*entité*) commandent (*relation*) des fournitures (*entité*).

➤ 1.1.2/ Les traitements⁶

Sur le plan des traitements, les entités sont décrites par leurs sollicitations et par les réactions qu'elles déclenchent au sein du système d'information. Ce sont donc les traitements dont elles sont les causes ou les conséquences qui sont abordés. Ceci se fait en termes d'événements, de synchronisation et d'opérations.

Ainsi, le fait qu'un contribuable (*entité*) paye son premier tiers hors délais (*événement*) après avoir été sollicité une seconde fois par le trésor public (*événement*) entraîne dès réception (*synchronisation*) la mise à jour de son dossier (*opération*).

La synchronisation, qui précise la coexistence dans le temps de plusieurs événements, décrit une condition de déclenchement de l'opération. Cette dernière, quant à elle, décrit l'enchaînement d'actions élémentaires à effectuer (consultation ou tri de table, calculs, contrôles, etc.) pour obtenir un résultat (édition d'une facture, d'un rappel, etc.).

⁶ www.misfu.com

1.2. Le niveau organisationnel ou logique

C'est la réalité telle qu'elle est perçue par les acteurs qui est exprimée à ce niveau. C'est celle qui répond aux questions "QUI ?" "OÙ ?" et "QUAND ?". Toutes les situations qui sont décrites à ce niveau sont externes à la référence que constitue la description conceptuelle du système d'information. Aucune différence n'est faite entre l'homme et la machine; on se contente d'affecter des tâches à la machine et aux hommes.

➤ 1.2.1/ Les données

Elles se traduisent par une représentation du système d'information propre à un acteur particulier. C'est à ce niveau que s'effectue le passage de la donnée à l'information nécessaire à un acteur pour qu'il puisse tenir le rôle que lui a dévolu l'organisme. Le modèle logique, fait appel à un mode de représentation différent : relations pour une classe de solutions relationnelle, records pour une classe de solutions navigationnelle etc.

Il y a à ce niveau transformation des données mais pas enrichissement. La création de redondances ou de données calculées à des fins d'optimisation n'est pas une création d'informations nouvelles !

➤ 1.2.2/ Les traitements

Les événements décrits sont à dominante spatiale. Les questions du type "Qui remplit le dossier d'inscription?", "Où le descriptif de l'ouvrage est-il saisi?", "Quand le bureau d'accueil est-il ouvert?" trouvent ici leur réponse.

Au contraire des données, le passage du niveau conceptuel au niveau organisationnel amène pour les traitements un enrichissement par attachement des acteurs aux événements précédemment définis et incorporation d'événements organisationnels (accord chef de bureau, terminal disponible...).

1.3/ Le niveau opérationnel ou physique

C'est une représentation des moyens qui vont effectivement être mis en œuvre pour gérer les données et réaliser les traitements. C'est aussi répondre à la question « COMMENT ? ».

➤ 1.3.1/ Les données

Il y a passage d'une classe de solutions à un élément de cette classe. Un modèle navigationnel entraîne l'utilisation d'un système de gestion de bases de données (SGBD) tel qu'IDS2 ou SOCRATE. Un modèle relationnel, quant à lui, amène l'usage d'un SGBD relationnel comme INGRES ou ORACLE. Les données sont décrites et manipulées à l'aide des langages spécifiques au système choisi (langage de description de données et langage de manipulation de données). Les fonctionnalités du système peuvent permettre d'optimiser le modèle en créant des index et des structures de stockage adaptées.

Cependant, le modèle physique choisi peut éventuellement appartenir à une classe plus pauvre que celle du modèle logique. Ainsi, un modèle logique navigationnel pourrait conduire à un modèle physique constitué d'un simple système de gestion de fichiers.

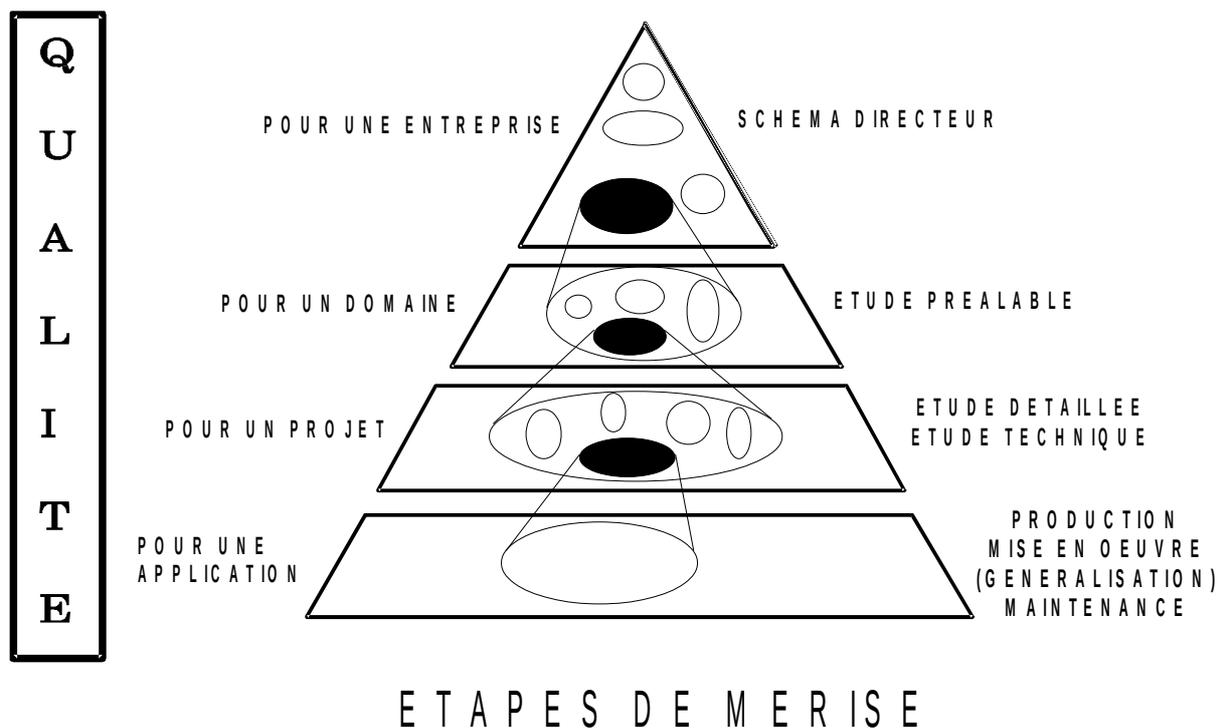
➤ 1.3.2/ Les traitements

On y décrit l'architecture des programmes (structures et algorithmes) qui vont activer les différentes tâches dévolues à l'ordinateur (menu de l'application, sécurité, procédures fonctionnelles, sauvegardes, etc.) à l'aide d'un pseudo langage. La programmation proprement dite dans un L3G⁷ ou un L4G⁸ n'a lieu que lors de la réalisation.

2. LA DEMARCHE PAR ETAPES

Les étapes du processus d'informatisation couvertes par MERISE sont:

- le schéma directeur
- l'étude préalable
- l'étude détaillée
- l'étude technique
- la production
- la mise en œuvre
- la généralisation
- la maintenance et l'évolution



⁷ L3G : dit programme de 3eme génération, tel Cobol, C,C++,java

⁸ L4G : LangVB, Delphi,..

2.1/ Le schéma directeur⁹

Le schéma directeur est une étape de planification du développement du système d'information. C'est une réflexion globale sur le futur système. Il s'agit donc d'une étape prospective devant assurer la cohérence entre le plan d'action général de l'entreprise et la définition des besoins en information. Il doit donc définir les différents domaines de gestion de l'organisme ainsi que les orientations de développement du système d'information en termes :

- de structure générale de l'organisation future des services,
- de structure des réseaux de communication,
- de répartition des postes de travail.

On y définit aussi les priorités entre les différentes applications à développer, les budgets prévisionnels et les niveaux de fiabilité et de confidentialité de chacune d'elles. Il aboutit à l'élaboration du plan directeur.

2.2. L'étude préalable

C'est une réflexion sur une grande fonction de l'entreprise; elle porte sur la totalité des fonctions d'un domaine dont elle affine les frontières. Son objectif est la constitution d'un dossier de choix permettant d'apprécier les diverses solutions possibles d'informatisation du domaine considéré en tenant compte des orientations définies par le schéma directeur.

Pour cela, elle précise les composantes du système actuel en termes de modèles, de circuits d'information, de répartition géographique des données et des traitements, de degré d'automatisation, de fiabilité, de confidentialité, de moyens informatiques existants et de coût de fonctionnement. Un bilan de l'existant fait ressortir les points forts et les points faibles du système actuel.

Un soin tout particulier doit être apporté à la définition des différentes solutions envisageables qui doit préciser les avantages, les inconvénients, les coûts et les délais prévus. Ceci est d'autant plus facilité que les frontières entre les projets sont définies avec rigueur.

Deux écueils sont à éviter :

- privilégier le processus de décision et donc conditionner le développement des moyens d'étude à la connaissance des réponses à toutes les interrogations qui peuvent se poser
- privilégier le processus de conception et faire progresser des études longues, détaillées et coûteuses sans qu'il y ait prise de position.

Pour cela, MERISE préconise de définir des sous-ensembles représentatifs du domaine concerné afin de pouvoir dégager plus facilement des éléments de réponse aux questions que se pose le commanditaire du projet.

9 « Les méthodes d'analyse et de conception des systèmes de production », Henri Pierraval. P40

2.3/ L'étude détaillée

C'est un dégagement des spécifications fonctionnelles (aussi appelées externes) d'un projet particulier qui ne porte que sur les fonctions à automatiser. Elle complète la solution qui a été choisie parmi celles ayant fait l'objet de l'étude préalable. Ces compléments portent sur le mode de fonctionnement du nouveau système (interfaces avec les utilisateurs, moyens humains, équipements nécessaires, fonctionnement en mode dégradé) et la formalisation des niveaux conceptuel et organisationnel des traitements.

Enfin, le modèle logique des données est optimisé.

2.4/ L'étude technique

Elle complète elle aussi l'étude préalable, mais uniquement en ce qui concerne le niveau physique ou opérationnel, en fournissant les spécifications internes du projet. La définition du MPD et l'architecture des transactions et des programmes sont affinées. Les différents sites d'exploitation sont identifiés, les normes de programmation et de documentation définies et les documents contractuels réalisés (ex : dossier de réalisation, dossier d'exploitation, guide utilisateur, normes de tests, procédures de sécurité, protocoles de reprise en cas de panne, etc.).

Les jeux d'essais sur la base desquels la qualité des programmes développés sera appréciée sont précisés.

2.5/ La production

Elle consiste en la fabrication des différents programmes et transactions dont l'architecture a été définie ainsi qu'en la fabrication des structures qui seront remplies par les utilisateurs. Les logiciels sont réalisés de façon modulaire et donnent lieu à des test d'intégration fondés sur les jeux d'essais définis précédemment.

La production peut passer par la fabrication d'un prototype*. Elle peut être réduite si des techniques de traduction des spécifications en instructions (ateliers de génie logiciel) sont utilisées.

- *Prototype : Logiciel intégrant tout ou partie des fonctionnalités à développer mais n'utilisant qu'une partie des données. Il ne faut pas le confondre avec une maquette qui est un logiciel destiné à montrer la faisabilité d'une application.*

2.6/ La mise en œuvre

C'est la mise à la disposition des utilisateurs du système développé. La recette est provisoire et ne deviendra définitive qu'à l'issue d'une période d'essais intensifs durant la montée en charge réelle du système.

Dans le cas d'un prototype, cette période permet de juger l'ergonomie et les fonctionnalités avant extension et passage au système complet.

2.7/ La généralisation

Elle consiste à mettre le système développé à la disposition de l'ensemble des utilisateurs. Elle est utilisée dans le cas de grandes entreprises ayant plusieurs sites informatiques. Si les matériels d'exploitation sont différents, des adaptations sont nécessaires.

2.8/ La maintenance et l'évolution

La maintenance vise à maintenir le système nouvellement développé en ordre de marche à fonctionnalités constantes. Il y a évolution lorsqu'on lui assortit une partie "évolution du système d'information" (fonctionnalités croissantes) par intégration de nouvelles règles.

II. IV/ CYCLES ET PHASES¹⁰

1. LES TROIS CYCLES DE MERISE

Trois cycles concourent à l'étude d'un système d'information et permettent d'en situer les étapes : le cycle de vie, le cycle de décision et le cycle d'abstraction.

1.1/ Le cycle de vie

Il permet de décrire la vie du système d'information. MERISE distingue trois périodes :

- la conception du système d'information qui aboutit à la conception détaillée des spécifications fonctionnelles et techniques,
- la réalisation qui consiste à produire des programmes et des consignes d'utilisation correspondant aux spécifications détaillées,
- la maintenance du système d'information qui a pour objectif son adaptation aux évolutions de son environnement.

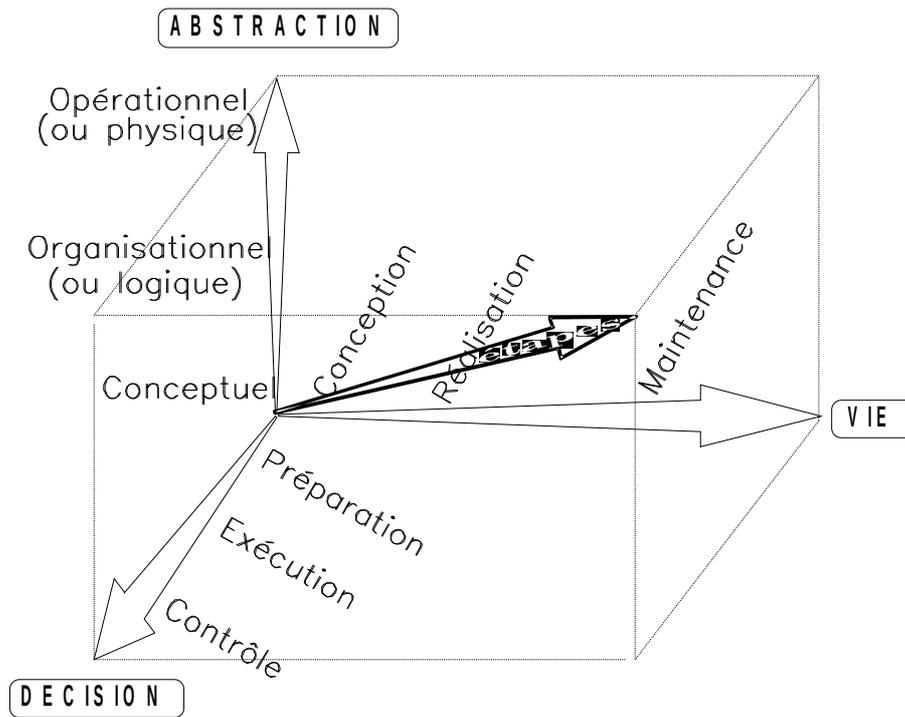
1.2/ Le cycle de décision

Il concerne les différents choix qui sont effectués tout au long du cycle de vie. La plupart de ces décisions marquent la fin d'une étape et le début d'une autre.

1.3/ Le cycle d'abstraction

Il offre les concepts nécessaires à la description du monde réel dans le système d'information. On y trouve les trois niveaux d'abstraction de MERISE (conceptuel, logique et physique).

¹⁰ OKOUMBA-NKOGHE, « la courbe de soleil »



1.4/ La nécessité des trois cycles

➤ 1.4.1/ Cycle de vie

Une méthode sans cycle de vie est inimaginable car il faut nécessairement des démarches de durées limitées pour bâtir des modèles (abstraction) et prendre des décisions.

➤ 1.4.1/ Cycle de décision

S'il n'y a pas de hiérarchie dans les décisions, cela signifie que le cycle d'abstraction choisi est suffisamment naturel et acceptable par les décideurs pour qu'ils se plient à sa logique et prennent leurs décisions dans l'ordre déterminé par le concepteur. Ceci peut conduire à retarder le projet si la suite des décisions n'est pas logique. (ex : prise en compte tardive de l'ergonomie).

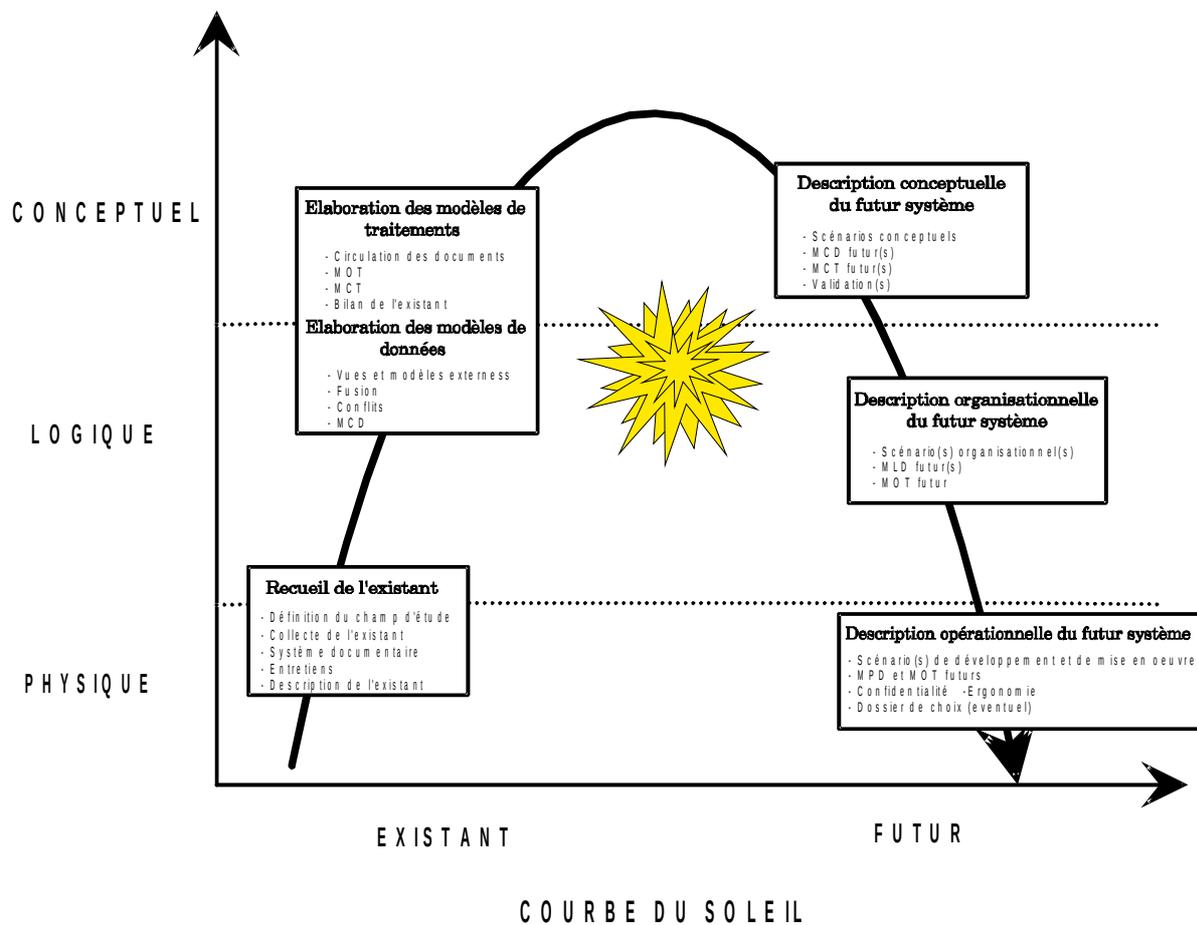
➤ 1.4.1/ Cycle d'abstraction

L'absence d'abstraction consisterait à appuyer les décisions sur l'examen de la réalité. Ceci n'est possible que pour une organisation simple. Une organisation complexe amènerait un blocage du cycle de décision et, par voie de conséquence, une interruption du cycle de vie.

2. LES PHASES D'ETUDE

2.1/ La courbe du soleil

Les phases d'étude sont disposées par rapport à la fameuse "courbe du soleil" qui représente le cheminement du processus de conception et d'étude à mettre en œuvre.



Pour cela, MERISE préconise de partir des niveaux physique et logique actuels. Les éléments recueillis permettent alors de bâtir un modèle conceptuel ne conservant que les grandes finalités de l'entreprise, indépendamment des contraintes organisationnelles ou physiques.

Cette vue synthétique permet de passer au système futur en intégrant de nouveaux éléments de gestion. On se situe alors au niveau conceptuel de l'état futur; La description est ensuite complétée par des composantes organisationnelles pour élaborer le niveau logique du système futur. Enfin, l'introduction des contraintes techniques au niveau physique permet de réaliser le système futur.

Ce processus de description présente maints avantages :

- Il limite les erreurs possibles de la description en partant du concret.
- La description du système existant rend possible une vérification du bien fondé de la représentation proposée en permettant une participation active des non informaticiens.

La prise en compte directe du futur système d'information ne se justifierait que si l'organisme était constamment en train de modifier ses modes de fonctionnement. Ce n'est pas le cas ; les systèmes à construire se situent nécessairement dans la continuité des systèmes antérieurs.

Section 2 : Modèle conceptuel de données : MCD

1-présentation

Le MCD est une représentation stable de l'ensemble des données manipulées par l'entreprise ainsi que des relations entre ces données

Le graphe de flux permet souvent de mettre en évidence des individus du MCD

Le MCD est précisé lors de l'étude préalable et complète lors de l'étude détaillée du SI

2- concepts manipulés¹¹

a- **Entités :**

Une entité ou individu est la représentation dans le SI d'un objet concret ou abstrait matériel ou immatériel de l'univers de l'entreprise

Exemple : client, compte-bancaire

On parle d'entité type c.-à-d. classe d'entité ayant des propriétés analogues

b- **Relation :**

La relation est in lien sémantique de plusieurs entités (indépendamment des traitements) on l'appelle aussi association entre deux ou plusieurs entités

Il est souhaitable de limiter la dimension des relations à deux (relation binaire)

Plus la dimension d'une relation est grande, plus la relation devient complexe et difficile à gérer

(La dimension d'une relation est le nombre d'entité qu'elle relie)

On parle de relation-type c-a d une relation entre plusieurs entités type

Exemple : possède est une relation qui lit l'entité client à l'entité compte bancaire

c- **Propriété (attribut) :**

Est une donnée élémentaire qui caractérise une relation

Exemple :

11 Hubert Tardieu, Arnold Rochfeld, René Colletti, Georges Panet, Gérard Vahée (1985). *La méthode Merise - Tome 2 Démarches et pratiques*. Editions d'organisation (Paris) : 460 p

- ✓ l'entité client est caractérisée par les propriétés numéro client, nom client, adresse client
- ✓ L'entité compte bancaire est caractérisé par les propriétés numéro compte, nom banque, agence
- ✓ La relation possède peut être caractérisé par la propriété date ouverture

Une propriété est caractérisée par une structure (classe et longueur)

d- **Occurrence :**

Les occurrences d'entité sont les valeurs qui appartiennent à l'entité
Exemple : TAZI et ALAOUI sont des occurrences de l'entité client

e- **Identifiant ou clé :**

Constitué d'une ou de plusieurs propriétés permettent de distinguer sans ambiguïté les différentes occurrences de l'entité

Exemple : l'identifiant de l'entité et numéro client car deux clients ne peuvent pas avoir le même numéro

Un identifiant d'une relation est l'ensemble des identifiants de l'entité reliés à cette relation

f- **Cardinalité¹² :**

Est notée sur la forme (X, Y)

X : le nombre minimum d'occurrence de l'entité 1 relié a l'entité 2

Y : le nombre maximum d'occurrence de l'entité 1 relié à l'entité 2

Exemple : un client possède au minimum 1 compte bancaire et au maximum plusieurs représenté par N d'où la cardinalité 1-N

Alors qu'un compte bancaire appartient à un client et un seul d'où la cardinalité 1-1

Dans la pratique, on gère les cardinalités suivantes :

- ✓ 0-1 : Chaque occurrence de l'entité est relié a au plus une occurrence de la relation Exemple : un employé est responsable d'au plus d'un service
- ✓ 1-1 : Chaque occurrence de l'entité est reliée exactement à une occurrence de la relation
Exemple : un compte bancaire appartient a un client et un seul
- ✓ 0-N : chaque occurrence de l'entité est relié au moins une occurrence de la relation
Exemple : un enseignant assure au moins une matière

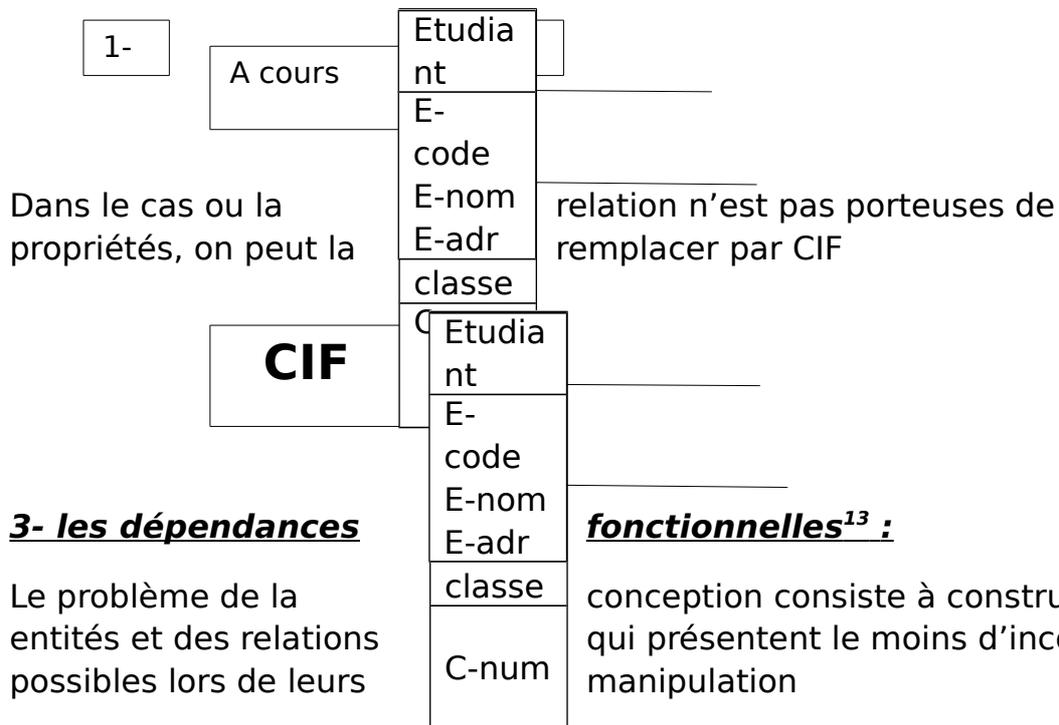
g- **Contraints d'intégrité fonctionnelle (CIF) :**

Une CIF dans le cas d'une relation binaire est identifié par la présence d'une cardinalité (1-1) ou (0-1)

12 Matheron, J.-P. Comprendre Merise. Eyrolles, 1994.

Une CIF porte sur une relation et sert d'indiquer que l'une des entités liées à la relation est entièrement déterminé par la connaissance des autres

Exemple : un étudiant a cours dans une et une seule classe



La méthode employée est basée sur l'étude des DF permettent de décomposer l'entité

Chaque étape de décomposition éliminera un certain nombre d'incohérence, le processus de décomposition est appelé normalisation et le niveau de décomposition sont appelés les formes normales

a- Définition :

Soit X et Y deux propriétés d'une entité

Il existe une DF entre X et Y notée $X \rightarrow Y$ (X détermine Y) si et seulement si pour une valeur de X il existe une seule valeur de Y

Exemple : Num client \rightarrow Nom client c.-à-d. pour un même numéro client, il n'existe pas deux noms de clients différents

Num fournisseurs, code produit \rightarrow prix c.-à-d. connaissant le numéro du fournisseur et le code produit, on connaît le prix. Autrement dit il existe un seul prix pour un fournisseur par produit

Exemple : si code-client \rightarrow adresse client

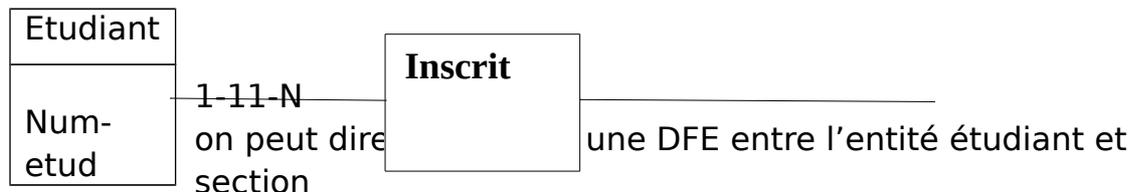
¹³ Nanci, D., Espinasse, B., Cohen, B. et Heckenroth, H. Ingénierie des systèmes d'information

Alors code- client, nom -client
élémentaire

adresse -client n'est pas

On appelle une DFE directe, celui qui n'est pas déduite par transitivité

Exemple : Num-étud → no-section
 No-section → nom-section
 Num-étud → nom-section n'est pas directe



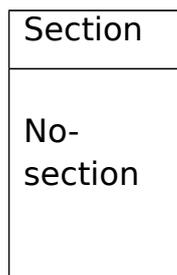
Remarque : la cardinalité maximum 1 correspond à une DF

b- Règles de normalisation :

La décomposition ou normalisation d'une entité s'effectue selon des niveaux qui s'appellent les formes normales

- Première formes

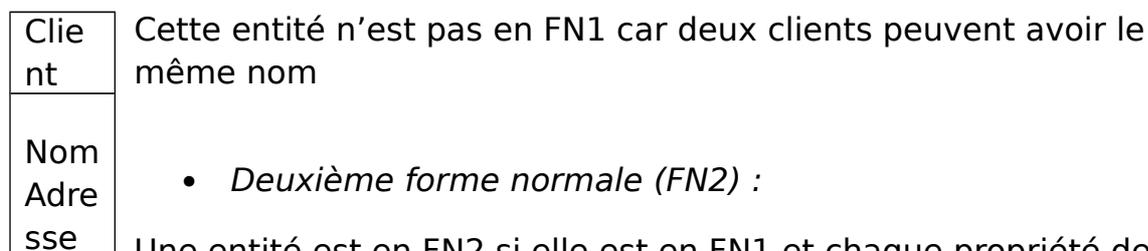
Une entité en FN1 si (non décomposable) et



normale (FN1):

chaque propriété est élémentaire
chaque entité possède une clé

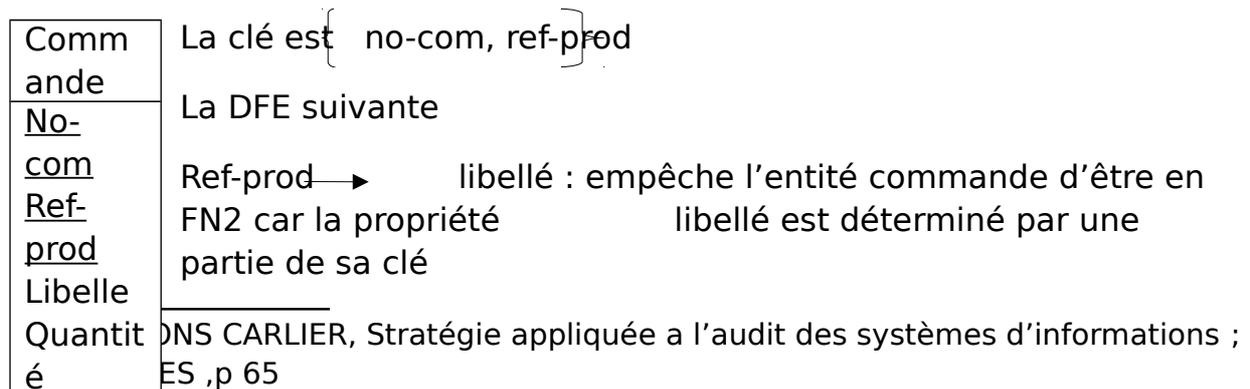
Exemple :



- Deuxième forme normale (FN2) :

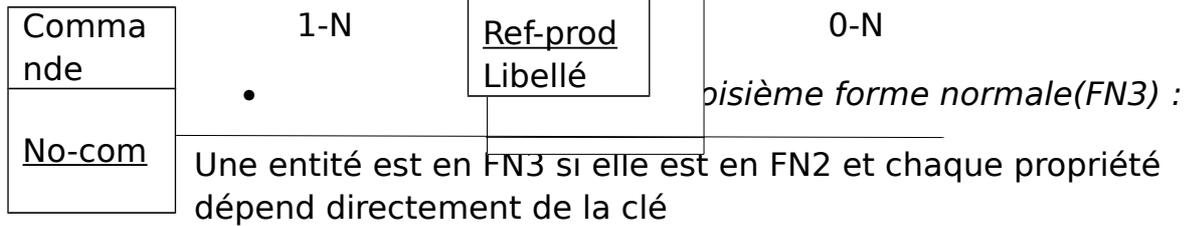
Une entité est en FN2 si elle est en FN1 et chaque propriété de l'entité dépend pleinement de sa clé

Exemple¹⁴ :

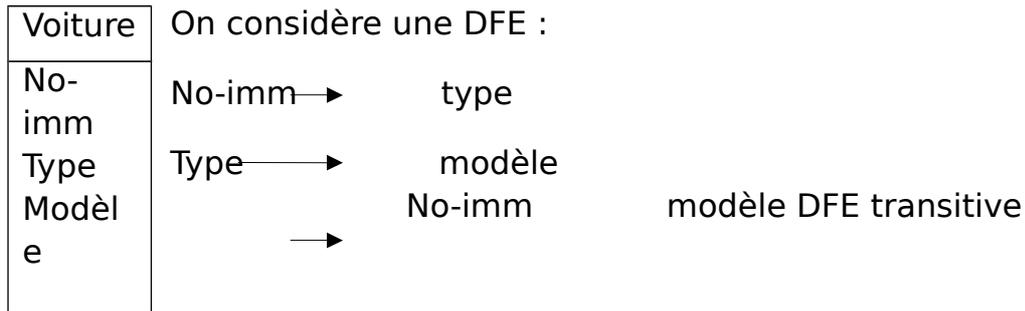


Cette entité doit être
suivante :

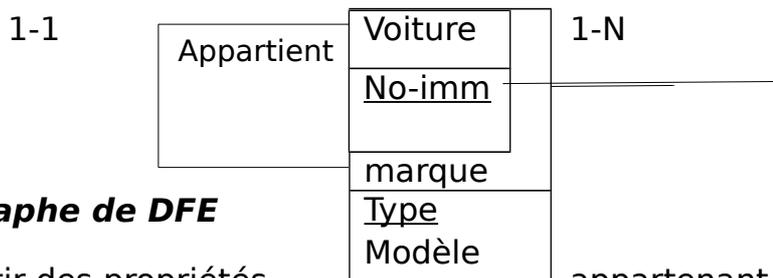
décomposée de la manière



Exemple :



L'entité voiture doit être décomposé de la manière suivante :



c- Graphe de DFE

À partir des propriétés
données (DD) épuré¹⁵

appartenant au dictionnaire des

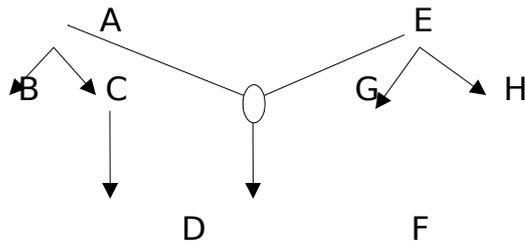
On établit la liste des DFE sous forme de graphe

Exemple : soit l'ensemble des DFE suivantes :



¹⁵ 15A DD épuré est celui qui ne contient que des propriétés élémentaires, élimination des propriétés calculées et des synonymes

Le graphe correspondant :



Section 3 : le modèle logique de données

1 : Méthodologie¹⁶

Face à un problème bien formulé (même si ,ca n'existe pas) procéder ainsi :

- identifier les entités en présence ;
- lister leurs attributs ;
- ajouter les identifiants ;
- établir les relations entre les entités ;
- lister leurs attributs ;
- éliminer les synonymes (plusieurs signifiant pour un signifie) et les polysèmes (plusieurs signifie pour un signifiant) ;
- calculer les cardinalités ;
- vérifier la troisième forme normale et la normalisation des relations ;
- effectuer les corrections nécessaires.

Il faut garder également à l'esprit que le modèle doit être exhaustif (c'est-à-dire contenir toutes les Informations nécessaires) et éviter toute redondance (le prix unitaire d'un article n'a pas besoin de figurer dans l'entité commandes).

Maintenant que le MCD est établi, on peut le traduire en différents systèmes logiques, comme les systèmes par fichiers ou les bases de données.

2 : Les systèmes logiques

Avant l'apparition des systèmes de gestion de base de données (SGBD ou DBMS pour Data Base Management System en anglais), les données étaient stockées dans des fichiers binaires et gérées par des programmes exécutables (Basic, Cobol ou Dbase par exemple). La maintenance des programmes (en cas de modification de la structure des données par exemple) était très problématique.

3 : Schéma relationnel

Concentrons-nous sur le MLDR. Lorsque des données ont la même structure (comme par exemple, les bordereaux de livraison), on peut les

¹⁶ Chantal Morley, Jean Hugues, Bernard Leblanc, Olivier Hugues , Processus métiers et S.I. : évaluation, modélisation, mise en œuvre" , éd (Dunod).

organiser en table dans laquelle les colonnes décrivent les champs en commun et les lignes contiennent les valeurs de ces champs pour chaque enregistrement.

Les lignes d'une table doivent être uniques, cela signifie qu'une colonne (au moins) doit servir de clé primaire.

La clé primaire d'une ligne ne doit pas changer au cours du temps, alors que les autres colonnes le peuvent.

Par ailleurs, il se peut qu'une colonne d'une table ne doive contenir que des valeurs prises par une colonne d'une autre table (par exemple, le numéro du client sur une commande doit correspondre à un vrai numéro de client).

Par convention, on souligne les clés primaires et on fait précéder les clés étrangères d'un dièse # dans la description des colonnes d'une table :

Clients (numéro client, nom client, prénom, adresse client, ...)

Commandes (numéro commande, date, #numéro client, ...)

Remarque :

- une même table peut avoir plusieurs clés étrangères mais une seule clé primaire (éventuellement

Composées de plusieurs colonnes) ;

- une clé étrangère peut aussi être primaire ;

- une clé étrangère peut être composée (c'est le cas si la clé primaire en liaison est composée).

La section suivante contient des exemples.

On peut représenter les tables d'une base de données relationnelle par un schéma relationnel dans

lequel les tables sont appelées relations et les liens entre les clés étrangères et leur clé primaire est symbolisée par un connecteur :

4 : Traduction

Pour traduire un MCD en troisième forme normale en un MLDR, il suffit d'appliquer cinq règles (à

Connaitre par cœur). Mais avant, on dit qu'une association entre deux entités (éventuellement réflexives) est de type :

- 1 : 1 si les deux cardinalités sont 0,1 ou 1,1 ;

- 1 : n si une des deux cardinalités est 0, n ou 1, n ;

- n : m (plusieurs à plusieurs) si les deux cardinalités sont 0, n ou 1, n.

Règle 1 : toute entité devient une table dans laquelle les attributs deviennent des colonnes. L'identifiant de l'entité constitue alors la clé primaire de la table.

Par exemple

- ❖ l'entité articles devient la table : articles (numéro article, nom article, prix unitaire de vente, ...)

Règle 2 : dans le cas de deux entités reliées par une association de type 1 : 1, on ajoute aux deux tables une clé étrangère vers la clé primaire de

l'autre. Les attributs de l'association sont alors repartis vers l'une des deux tables.

Par exemple

- ❖ L'association résider est traduite par : êtres humains (numéro personnel, nom, prénom, #numéro appartement, date d'entrée, ...)
- ❖ Logement (numéro appartement, adresse, # numéro personnel, montant du loyer, ...)

Remarque : d'autres techniques sont parfois proposées pour la règle 2 (fusionner les tables, utiliser une clé primaire identique) mais en pratique elles ne sont pas exploitables dans le cas général.

Règle 3 : dans le cas de deux entités reliées par une association de type 1 : n, l'identifiant de l'entité

Côté 0, n ou 1, n devient une clé étrangère vers la clé primaire de la table coté 0,1 ou 1,1. Les attributs de l'association glissent vers la table coté 0,1 ou 1,1.

Par exemple, l'association concerner est traduite par :

- ❖ Articles (numéro article, nom article, prix unitaire de vente, ...)
- ❖ Livraisons (numéro livraison, date, #numéro article, quantité livrée)

Règle 4 : une association entre deux entités et de type n : m est traduite par une table supplémentaire (parfois appelée table de jointure) dont la clé primaire est composée de deux clés étrangères vers les clés primaires des deux tables en association. Les attributs de l'association deviennent des colonnes de cette table.

Par exemple, l'association concerner est traduite par :

- ❖ Articles (numéro article, nom article, prix unitaire de vente, ...)
- ❖ Lignes de commande (#numéro commande, #numéro article, quantité commandée)
- ❖ Commandes (numéro commande, date, ...)

Règle 5 : une association entre trois entités ou plus est traduite par une table supplémentaire dont la clé primaire est composée d'autant de clés étrangères que d'entités.

Les attributs de l'association deviennent des colonnes de cette table.

Par exemple, l'association vols devient la table :

- ❖ Vols (#numéro avion, #numéro pilote, #numéro aéroport, date et heure, durée, distance)

CAS D'APPLICATION

Application 1 :

Dans un atelier de fabrication, on s'intéresse aux informations suivantes :

1. N° d'équipe ;
2. Nom de l'employé faisant partie de l'équipe ;
3. N° de matricule de l'employé ;
4. Qualification de l'employé ;

5. Nombre de personnes ayant une qualification donnée dans l'entreprise ;
6. Qualifications mises en œuvre dans une équipe ;
7. Nombre de personnes ayant une qualification donnée dans une équipe.

Informations supplémentaires :

- Dans une équipe, il y a toujours un chef unique, et au moins un employé.
- Un employé a une seule qualification et appartient à une seule équipe.

TAF : on vous demande de proposer un modèle conceptuel des données pour la conception de cette application ?

Application 2 :

Il s'agit du système d'information d'une entreprise qui loue du matériel à ses clients.

Les locations se font dans les diverses agences de l'entreprise.

Les interviews ont permis de détecter les règles de gestion suivantes :

RG1 : toute location porte sur une durée exprimée en nombre de semaines et d'au moins une semaine.

RG2 : une location concerne un ou plusieurs matériels.

RG3 : toute location doit donner lieu à un contrat entre l'entreprise et le client.

Les interviews ont également permis le recueil des documents suivants :

Liste des clients :

co de	nom	rue	Ville
A01	BELK BIR	6 RUE HASSAN I	Nom Saâda
A25	TAZI	2 RUE SAHRA	Centre
B03	ALAM I	7 AV. MED	L'habita

Liste des agences :

Catalogue :

Réf	Désignation	Prix de location /semaine
X01	Ampli	200
X02	Magnétoscope	300
X03	Télé	150
X04	Haut-parleur	50
X05	Disque laser	200

Etat du stock disponible :

Contrat

en cours :

Agence		Contrat n° 201		01
Réf	Quantité disponible	Agence 01	Date	
X01	20	Client A 01	15/02/07	Farah
X02	40	Durée	4 semaines	
X03	40	Montant	1800	02
X04	50	Réf	Désignation	
X05	10	Quantité louée		
		X03	Télé	
		X02	Magnétoscope	
		1		02
		X04	Haut-parleur	
		2		

TAF : on de modèle

données pour la conception de cette application ?

vous demande proposer un conceptuel des

Corrigé de l'application 1:**1- Dictionnaire des données :**

Liste des propriétés	Significations
Numéro	Numéro d'équipe
Nom	Nom de l'employé appartenant a l'entreprise
Matricule	N° du matricule de l'employé
Qualif.	Qualification de l'employé
Nombre 1	Nombre de personnes ayant une qualification dans l'entreprise
Qualif. 2	Qualifications mises en œuvre dans l'équipe : redondance car c'est la même que qualif.
Nombre 2	Nombre de personnes ayant une qualification dans l'équipe
Chef-	Chef d'équipe : redondance car le nom du chef et son matricule existent déjà dans modèle

2- Liste des DFE :

Matricule □ Nom

Matricule □ Qualif.

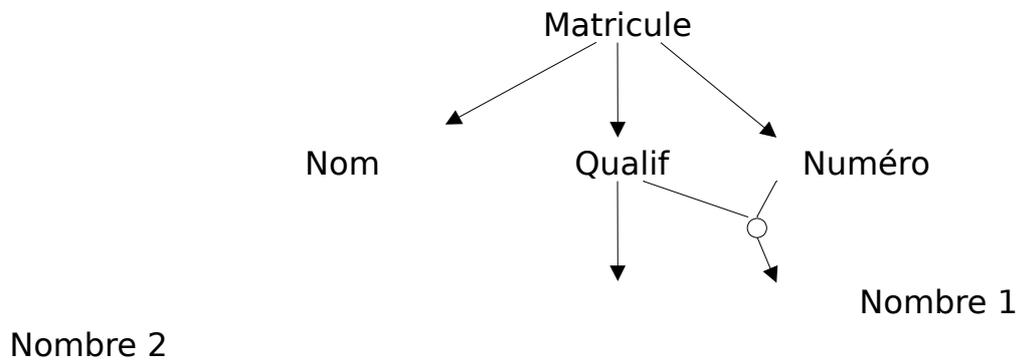
Matricule □ numéro

Qualif. □ Nombre 1

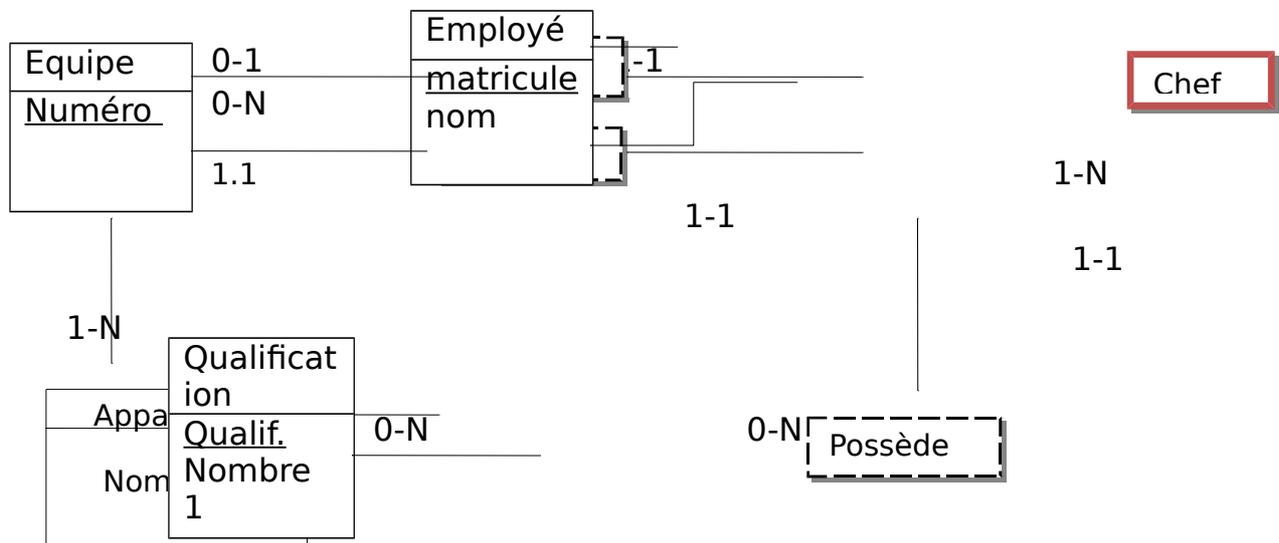
Numéro - Qualif. □ Nombre

Préparé par : S.FAHM, G.DIAZ et S.REDA

3- **Graphe du DFE :**



4- **MCD :**



Q : si on désire avoir le chef d'un employé indépendamment de son équipe ?

R : on va faire une association réflexive entre l'entité elle-même, ce qui est représenté par « **CHEF** »

5- Transformation de MCD en MLD :Equipe (numéro, matricule)Employé (matricule, nom, numéro, qualif.)Qualification (qualif., nombre 1)Appartient (numéro, qualif., nombre 2)**Corrigé de l'application 2 :****1- Dictionnaire de données épurées :**

Propriété	Signification
Durée	Durée de la location
Qté 2	Quantité de matériel loué dans un contrat
Code	Code du client
Nom 1	Nom du client
Rue	Adresse du client
Num 1	Numéro de l'agence
Nom 2	Nom de l'agence
Réf	référence du matériel
Désignation	La désignation du matériel
Prix	Prix de la location par semaine
Ville	Ville du client
Qté 1	Quantité disponible d'un matériel dans une agence
Num 2	Numéro du contrat
Montant	Montant global : c'est calculé donc cette propriété ne doit pas exister car $\text{Montant} = \text{prix} * \text{Q} * \text{durée}$

2- Liste des DF élémentaires :

Code □ Nom 1

Réf □ Prix

Code □ Rue

Num 2 □ Date

Préparé par : S.FAHM, G.DIAZ et S.REDA

Master : MSFE

Méthode de conception de SI : MERISE

Code □ Ville

Num 2 □ Durée, Num 1, code

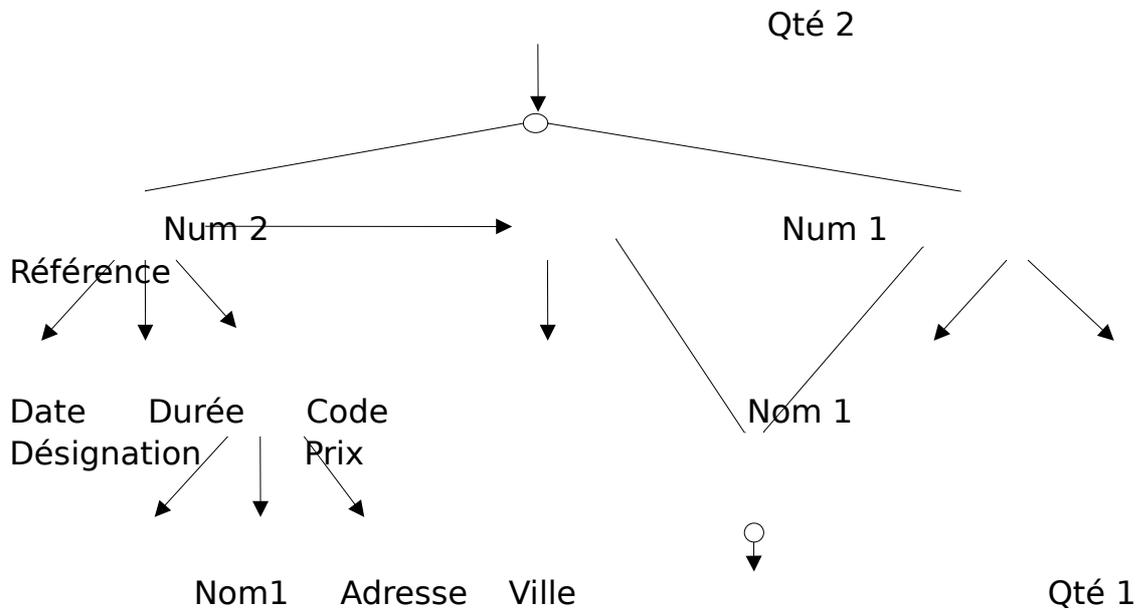
Num 1 □ Nom 2

Num 1, Réf □ Qté 1

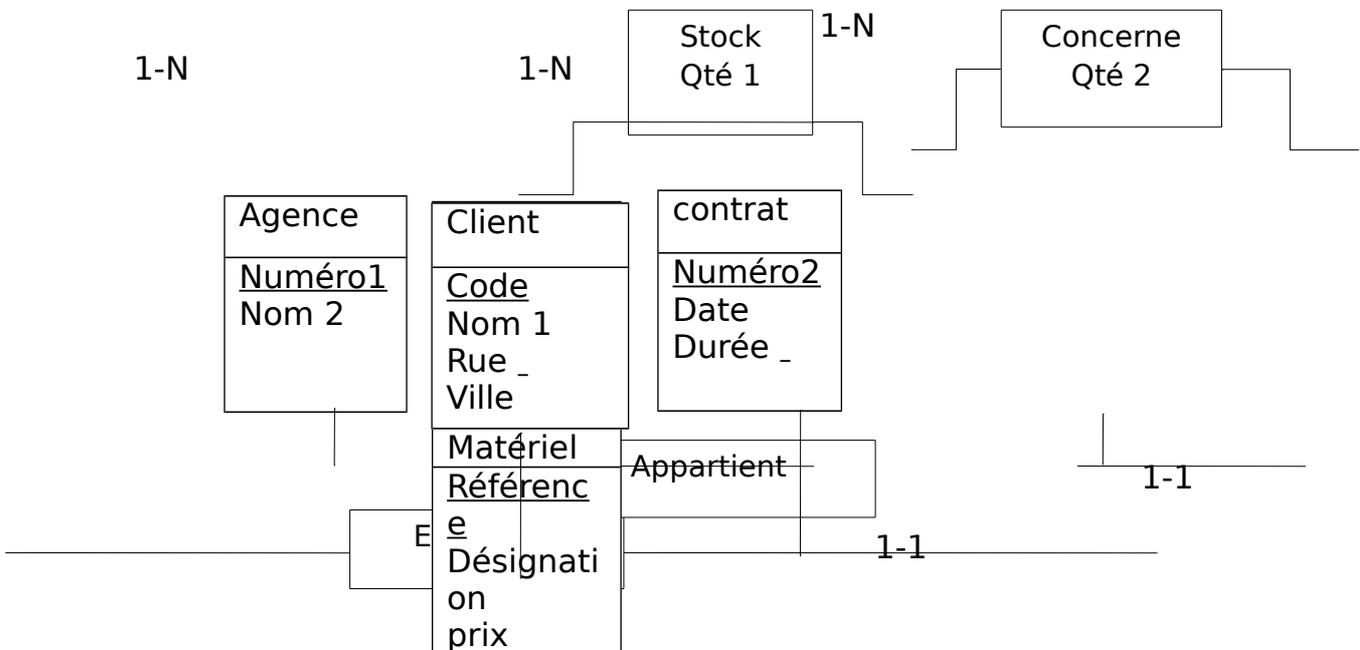
Réf □ Désignation

Num 2, Réf □ Qté 2

3- Grappe du DFE :



4- MCD :



5- Le schéma est normalisé, c'est-à-dire que la Forme Normale 3 (FN3) est vérifiée □ le MCD est correct.

6- **La transformation du MCD au MLD :**

Contrat (numéro 2, date, durée, **numéro 1**, **code**)

Matériel (Référence, désignation, prix)

Agence (Numéro1, Nom 2)

Client (code, nom 1, rue, ville)

Concerne (**numéro 2**, **Référence**, **Qté 2**)

Stock (**numéro 1**, **Référence**, **Qté 1**)