

Chapitre 04 : Méthodologie de Développement d'un SI

Introduction

Avec la complexité croissante des systèmes d'information, et avec l'importance du nombre de participants dans la conception de ces systèmes, la nécessité d'élaborer des méthodes permettant de concevoir correctement un système d'information et de mettre en place un modèle sur lequel s'appuyer est devenue nécessaire.

La modélisation consiste à créer une représentation virtuelle d'une réalité de telle façon à faire ressortir les points auxquels on s'intéresse. Ce type de méthode est appelé analyse. Il existe plusieurs méthodes d'analyse, la méthode la plus utilisée étant la méthode MERISE (Méthode d'Étude et de Réalisation Informatique pour les Systèmes d'Entreprise)

Partie I : Processus de développement d'un SI

Le processus de développement d'un SI est ensemble structuré d'activités à réaliser pour atteindre l'objectif d'un projet SI, dont les activités varient en fonction de l'organisation, du projet, et du type de système à développer. Ce processus doit être explicitement décrit pour être adéquatement géré.

1. Cycle de vie d'un SI

Le « **cycle de vie d'un logiciel** » (en anglais *software life cycle*), désigne toutes les étapes du développement d'un logiciel, de sa conception à sa disparition. L'objectif d'un tel découpage est de permettre de définir des jalons intermédiaires permettant la **validation** du développement logiciel, c'est-à-dire la conformité du logiciel avec les besoins exprimés, et la **vérification** du processus de développement, c'est-à-dire l'adéquation des méthodes mises en œuvre.

L'origine de ce découpage provient du constat que les erreurs ont un coût d'autant plus élevé qu'elles sont détectées tardivement dans le processus de réalisation. Le cycle de vie permet de détecter les erreurs au plus tôt et ainsi de maîtriser la qualité du logiciel, les délais de sa réalisation et les coûts associés.

2. Activités de développement d'un SI :

Le cycle de vie du SI comprend généralement les activités suivantes :

Initiation - Étude de faisabilité

- Analyser le problème
- Définir les objectifs
- Définir les frontières du système
- Identifier les contraintes imposées (politiques, matériel, temps, technologies, etc.)
- **Évaluer la faisabilité technique, économique, opérationnelle.**

Spécification des exigences

- Définir le périmètre du SI
- Analyser le domaine d'application
- Identifier les exigences :
 - fonctionnelles : les activités de l'organisation qui devraient être couvertes par le SI
 - non fonctionnelles : les performances techniques, l'environnement d'utilisateur, système d'exploitation, matériel, etc.
- Conceptualiser les besoins (cas d'utilisation, scénarios, objectifs, etc.)
- Déterminer la priorité des besoins
- Valider les exigences avec les utilisateurs
- Élaborer un glossaire de termes

Analyse -Exploration des solutions possibles

- Analyser les exigences
- Proposer une ou plusieurs solutions conceptuelles
 - Vue structurelle du système (modèle objet)
 - Vue dynamique du système (cycle de vie des objets, diagrammes d'activités, etc.)
- Proposer une architecture d'implémentation
- Valider toutes les propositions avec les utilisateurs

Conception- Spécification détaillée de la solution choisie :

- Définir le modèle physique de la base de données
- Définir les contraintes d'intégrité
- Choisir le SGBD
- Modéliser les interfaces utilisateur
- Définir les standards de sécurité du système
- Définir l'architecture du système

Construction- Développement du code :

- Créer les programmes
- Créer et remplir les bases de données
- Intégrer les sous-systèmes
- Écrire la documentation

Tests

- Tester les programmes
- Tester le système global

Mise en place

- S'assurer que tout le matériel et les réseaux nécessaires pour le nouveau SI sont mis en place
- Installer le système
- Former l'utilisateur

3. Modèles de développement d'un SI

La séquence et la présence de chacune des activités dans le cycle de vie dépend du choix de modèle de cycle de vie entre le client et l'équipe de développement. Il existe différents types de modèles de cycles de vie. Ces modèles prendront en compte toutes les activités/étapes de développement des SI vus dans le point précédent.

3.1 Cycle de vie en cascade

Dans le cycle de vie en cascade, les étapes se succèdent dans le temps. Chaque étape produit des documents qui sont utilisés pour en vérifier la conformité avant de passer à la suivante. Le cycle de vie en cascade est souvent schématisé de la manière suivante:

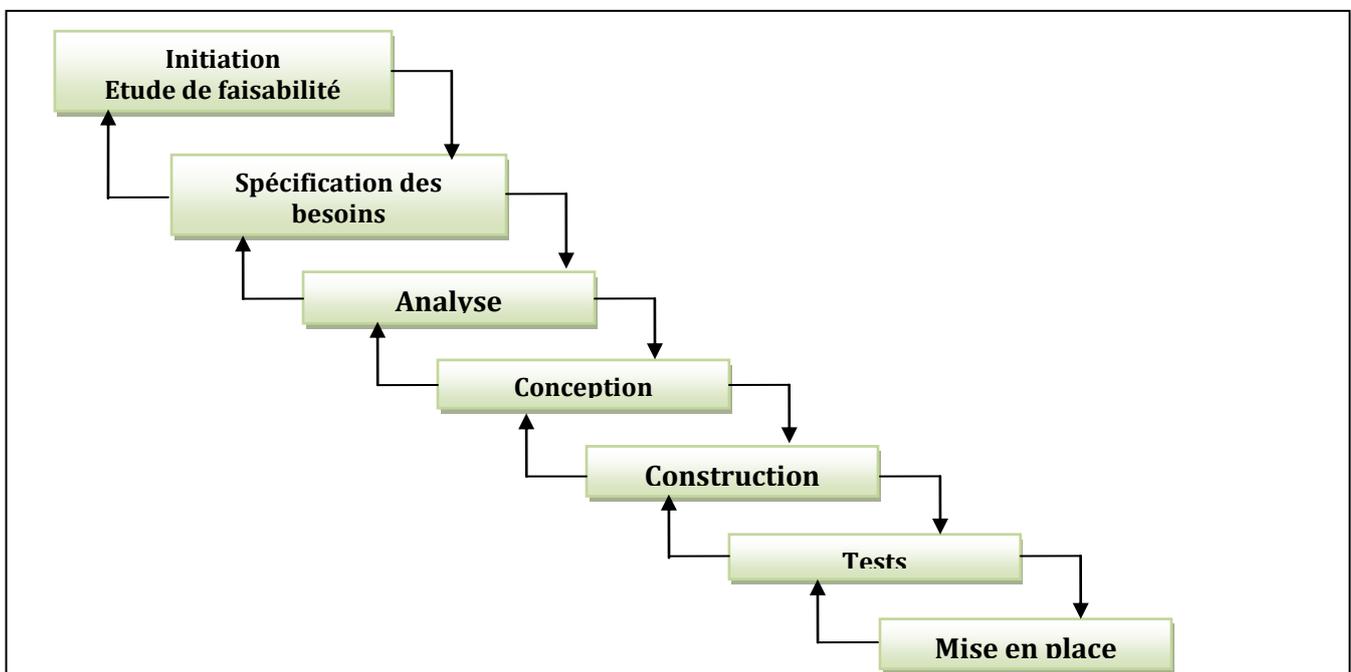


Figure 4.1 : Cycle de vie en cascade.

3.2 Cycle de vie en V

Le principe de base de ce modèle est la décomposition du système en composants qui peuvent être développés de manière indépendante. Ce modèle permet d'avoir une meilleure

réactivité que le modèle en cascade : il permet de limiter les retours en arrière en cas d'anomalie.

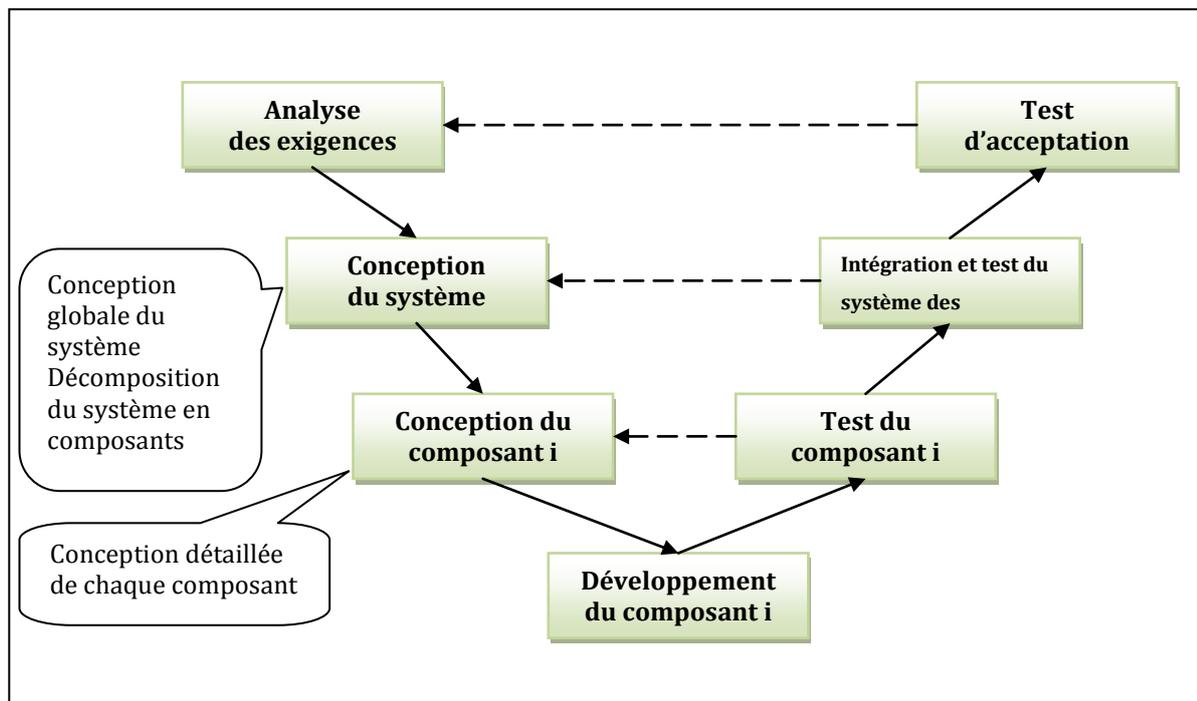


Figure 4.2 : Cycle de vie en V.

3.3 Cycle de vie en spirale

Le cycle de vie en spirale est un modèle générique de cycle de vie évolutif qui a été proposé par Barry W. Boehm en 1984. Ce modèle, axé sur la maîtrise et la réduction des risques, est davantage un cadre de travail guidant la construction d'une démarche spécifique de projet, plutôt qu'une démarche formalisée.

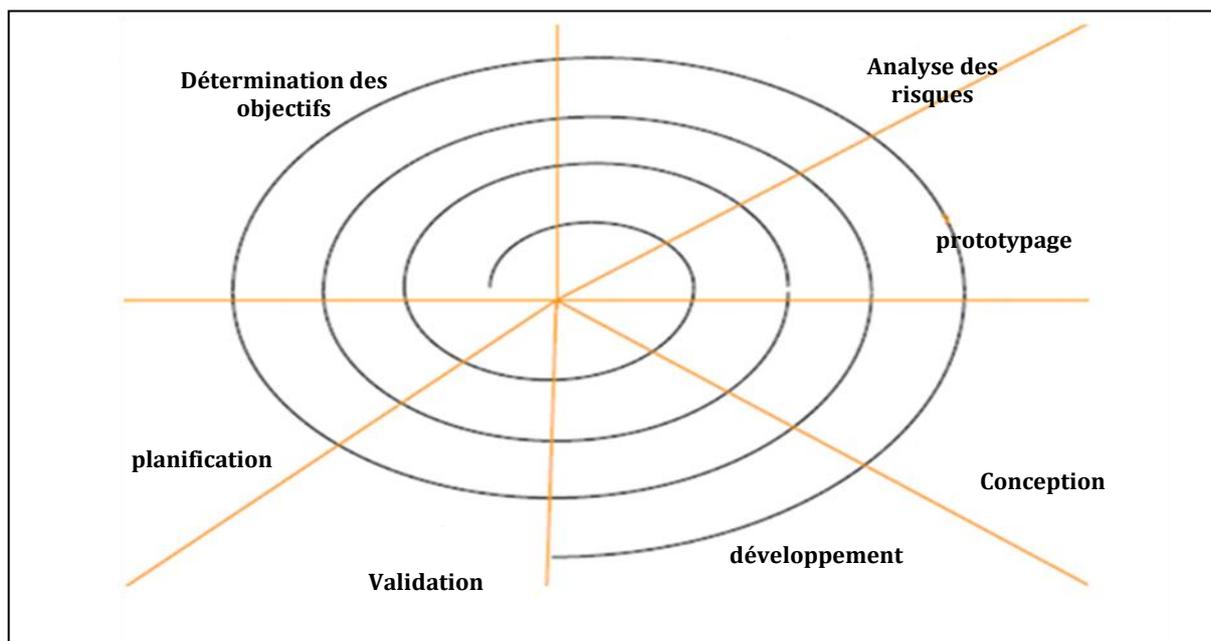


Figure 4.3 : Cycle de vie en Spirale

Chaque boucle de spirale permet d'identifier :

- les objectifs propres de la boucle
- les moyens alternatifs pour atteindre les objectifs
- les contraintes de chaque alternative

Ce modèle reprend les différentes étapes du modèle en V en mettant un accent sur la gestion du risque, qui se retrouve à chaque début d'itération de la spirale.

4. Classification des méthodes de conception d'un SI

Une **méthode d'analyse et de conception** est un procédé qui a pour objectif de permettre de formaliser les étapes préliminaires du développement d'un système afin de rendre ce développement plus conforme aux besoins du client.

Une méthode de conception est composée des:

- **Des modèles** : ensembles de concepts et de règles destinés à expliquer et construire la représentation de phénomènes organisationnels.
- **Des langages** : pour élaborer les spécifications, et faciliter leur communication.
- **Une démarche** : processus pour effectuer les travaux préconisés, étape par étape.
- **Des outils (AGL) ou techniques** : pour aider à la mise en œuvre des trois composantes ci-dessus.

Donc, une méthode est un mode d'emploi particulier d'un modèle. Elle dit comment observer les éléments

Trois générations de méthodes de conception :

Génération	Période	Approche
1ère génération	Années 70	<i>Méthodes analytiques ou cartésiennes</i>
2ème génération	Années 80	<i>Méthodes systémiques</i>
3ème génération	Années 90	<i>Méthodes orientés objet</i>

Tableau 4.1 : Génération des méthodes de conception.

4.1 Méthodes analytiques ou cartésiennes

La démarche de ces méthodes consiste à :

- **Découper** le domaine d'étude en fonctions.
- Prendre chaque fonction et la **décomposer** de façon **hiérarchique** en sous fonctions.
- Arrêter la décomposition lorsqu'on atteint un **niveau** de découpage **suffisamment fin** pour que le codage des sous fonctions soit simple à réaliser.

Ces méthodes privilégient l'approche par **traitements**.

Exemples

- Méthodes de **programmation structurée**
- Méthode **SADT** (Structured Analysis and Design Techniques).
- Méthode de **Jakson**
- Méthode de **Yourdon** (Modern Structured Analysis)

Avantages et inconvénients

Avantages :

- Correspond à la démarche naturelle pour aborder un problème.
- Facilité de recenser les besoins des utilisateurs.
- Facilité de produire des solutions à plusieurs niveaux d'abstraction.

Inconvénients :

- Concentration de l'effort d'analyse sur les traitements et négligence de la cohérence des données (redondance).
- Absence de règles de décomposition produisant des hiérarchies de décompositions différentes selon les analystes.
- Difficultés de tenir compte des interactions non hiérarchiques dans le cas de systèmes complexes.
- L'intégration des différentes applications obtenues est peu conforme à la réalité que l'on a voulu décrire.

4.2 Méthodes systémiques

Ces méthodes sont basées sur les **concepts** suivants :

- Le SI est perçu comme un *objet complexe* actif dont il faut décrire la **structure** et les **objectifs fonctionnels**.
- La modélisation du SI est abordée selon deux points de vue complémentaires :
 - *Modélisation des données* : Aboutit à un modèle de données garantissant la cohérence des données.
 - *Modélisation des traitements* : Aboutit à l'élaboration d'un modèle de traitements décrivant les traitements à réaliser sur les données.

La **démarche** de ces méthodes est la suivante :

- Le domaine d'étude est représenté à l'aide d'un **modèle réduit** (sous-ensemble représentatif).
- Le modèle réduit est **découpé** en sous domaines.
- Chaque sous domaine est ensuite **étendu** à l'ensemble.

Exemples

- La méthode **Merise**
- La méthode **Axial**
- La méthode **Information Engineering (IE)**

Avantages et inconvénients

Avantage :

- Meilleure cohérence des données.
- Respect des niveaux de représentation introduits par le groupe ANSI/SPARC (Niveau conceptuel, externe et interne).

Inconvénients :

- Absence de règles pour assurer la cohérence entre modèle des données et modèle des traitements.
- Les frontières entre les niveaux conceptuel, interne et externe ne sont pas nettes.
- Faiblesse de la modélisation des traitements : mélange des connaissances (règles de gestion) et du contrôle (contraintes d'intégrité).

4.3 Méthodes orientées objet

Les caractéristiques de ces méthodes sont les suivantes :

- Elles constituent une évolution des méthodes systémiques vers une **plus grande cohérence entre les objets et leur dynamique**.
- Basée sur le concept d'**objet**.

- Permet de décrire la **dynamique** du SI comme un ensemble d'**opérations rattachées aux objets** constituant le système.
- Cette représentation permet une meilleure modularité et réutilisation des composants du SI.
- C'est une approche ascendante :
 - Identification des objets de base du SI.
 - Par composition, constitution d'objets de plus en plus complexes.

Exemples

- **OOD** (G. Booch)
- **HOOD** (Hood Technical Group)
- **OOA** (S. Shlear et S. Mellor)
- **OOA / OOD** (T. Coal et E. Yourdon)
- **OMT** (J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, W. Lorensen)
- **OOSE** (I. Jacobson, M. Cristerson, P. Jonson, G. Övergaard)
- **OOM** (M. Bouzeghoub et A. Rochfeld)

Avantages et inconvénients

Avantage :

- Grande capacité à modéliser les objets complexes.
- Réduction des distorsions entre le réel et le système informatique.
- Grande capacité à intégrer la dynamique des objets.
- Possibilité d'encapsuler les parties privées.

Inconvénients :

- Risque d'avoir une perception monolithique des applications.
- Difficulté de l'effort d'abstraction.

Partie II : Méthodologie MERISE

1. Introduction

Merise (**M**éthode d'**E**tudes et de **R**éalisation Informatique pour les **S**ystèmes d'**E**ntreprise) est une méthodologie de modélisation à usage général dans le domaine du développement de systèmes d'information, du génie logiciel et de la gestion de projet. Introduit pour la première fois au début des années 1980, elle était largement utilisée en France. Elle a été développée et perfectionnée à un point tel que la plupart des grandes organisations gouvernementales, commerciales et industrielles françaises l'ont adopté.

2. Caractéristiques

- Fait partie des méthodes *systemiques*.
- MERISE présente dans sa démarche d'analyse trois cycles fondamentaux :

- le cycle d'abstraction,
- le cycle de vie (de développement),
- le cycle de décision.
- Le cycle d'abstraction est basé sur deux axes :
 - Modélisation des données
 - Modélisation des traitements
- Utilisée essentiellement pour la conception de bases de données.
- Elle est basée sur deux approches :
 - Approche par niveaux
 - Approche par étapes.

2. Démarche de Merise

Merise se caractérise par une double démarche : par **étape de construction** et par **niveau d'abstraction**.

3.1 Démarche par étape de construction

- **Schéma directeur** permet :

- L'élaboration d'une stratégie en matière de SI
- La définition des domaines d'étude
- La planification du développement de chaque domaine
- L'évaluation des moyens humains et financiers nécessaires pour chaque domaine

- **Étude préalable** :

Porte sur un **sous-ensemble représentatif** du domaine étudié comportant les activités les plus importantes, les plus critiques et les plus sujettes au changement. Elle se fait selon trois phases :

- **Recueil** : Elle consiste à recadrer la mission et étudier l'existant
- **Conception** : Elle consiste à définir des solutions organisationnelles et techniques
- **Appréciation** : Elle consiste à déterminer des scénarios de réalisation des solutions proposées et d'en quantifier les coûts, les services rendus, les risques et les enjeux.

- **Étude détaillée** : Consiste à généraliser et à détailler le travail effectué pendant la phase d'étude préalable. Elle complète les descriptions effectuées lors de l'étude préalable et respecte les solutions décidées à l'issue de cette étude. Le résultat de cette étape est un dossier de spécifications fonctionnelles ou cahier des charges utilisateurs.

- **Réalisation** : Consiste à produire les programmes selon les spécifications de l'étude détaillée avec test et validation. Elle se fait selon deux phases :

- **Étude technique** : Elle consiste à aboutir à une description physique des données et des traitements.

- **Production des programmes** : Elle consiste à développer les programmes et les mettre au point. Elle se fait en trois étapes :
 - ✓ **Planning de production** : Consiste à planifier la production des programmes et de réception interne.
 - ✓ **Développement** : Il s'agit de produire les programmes et de les mettre au point en interne.
 - ✓ **Mise au point globale** : Il s'agit d'effectuer des tests d'intégration.

- **Maintenance** : Consiste à faire des corrections et des adaptations du logiciel.

- **Contrôle de qualité** : Consiste à vérifier de la conformité des résultats par rapport aux demandes

3.2 Démarche par niveau d'abstraction

La méthode Merise introduit un modèle tridimensionnel, Le développement du système d'information selon ce modèle se fait suivant trois axes appelés **cycles** :

- **Cycle de vie** : décrit les différentes étapes correspondant au cycle de vie du système d'information.
- **Cycle de décision** : décrit le cycle de développement correspondant au cycle de vie du projet.
- **Cycle d'abstraction** : comprend les niveaux de description du système d'information.

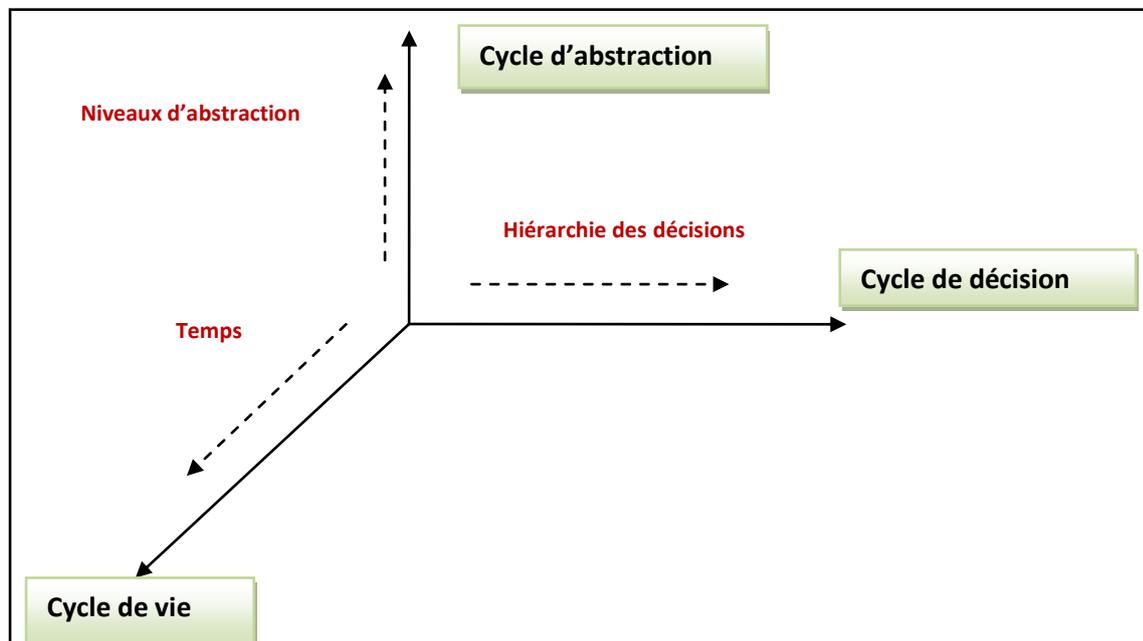


Figure 4.4 : Démarche par niveau d'abstraction.

Le cycle de vie : Correspond à la vie du système d'information, depuis sa conception jusqu'à l'exploitation en passant par sa naissance, sa maturité et sa maintenance. Comprend trois périodes :

- **Conception** : aboutit à la description détaillée des spécifications fonctionnelles et techniques du système.
- **Réalisation** : production des programmes et structures de données correspondant aux spécifications détaillées.
- **Maintenance** : adaptation du système à l'évolution de son environnement.

Le cycle de décision : Correspond aux choix qui doivent être faits durant le cycle de vie du système d'information. A travers ce cycle, l'organisation s'assure que le système correspond aux objectifs.

Les types de décisions possibles dans ce cycle sont:

- **Décisions de gestion** : Objectifs, orientations, règles de gestions, ...
- **Décisions organisationnelles** : Choix d'organisation, répartition des tâches
- **Décisions techniques** : Choix techniques (SE, SGBD, outils de développement, Bureautique, architecture, ...).
- **Orientation de la gestion du projet** : Ressources allouées, priorités de développement, planning d'avancement, ...

Le cycle d'abstraction : ce cycle correspond aux différents niveaux permettant la description et la spécification du système d'information. Trois niveaux d'abstraction :

- **Niveau conceptuel** : C'est le niveau d'abstraction le plus élevé. Il comprend les éléments les plus stables. Il décrit les classes d'objets et les règles significatives en fonction des objectifs fixés par les décideurs.
- **Niveau logique / organisationnel** : Représente les ressources utilisées pour supporter les descriptions du niveau conceptuel.
- **Niveau physique / opérationnel** : Donne une représentation physique des données et opérationnelle des traitements en tenant compte des contraintes et choix techniques.

Niveaux de description	Données	Traitements
Conceptuel (déterminé par des choix de gestion)	Représentation conceptuelle des données sous forme d'entités et d'associations inter entités. Modèle conceptuel de données (MCD)	Représentation conceptuelle des traitements sous forme d'opérations déclenchées par des événements Modèle conceptuel de traitements (MCT)
Logique /organisationnel (déterminé par des choix d'organisation)	Description logique des données en fonction des schémas disponibles (fichiers classiques, schémas hiérarchique, réseau ou relationnel). Modèle logique de données (MLD)	Simulation organisationnelle des traitements intégrant les ressources hommes, machines et interactions. Modèle logique de traitement (MLT)

<p>Physique/ opérationnel (déterminé par des choix techniques)</p>	<p>Représentation physique des données en fonction de la fréquence d'accès, des temps de réponses attendus, des choix de répartition, ...</p> <p>Modèle physique de données (MPD)</p>	<p>Représentation opérationnelle des traitements : applicatifs, et communication en fonction des architectures, configurations, langages et environnements techniques retenus.</p> <p>Modèle physique de traitements (MCD)</p>
---	--	---

Tableau 4.2 : Modèles de MERISE.

4. Concepts pour la modélisation statique (modélisation des données)

4.1 Modèle conceptuel de données (MCD)

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

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

4.1.1 Entités

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

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

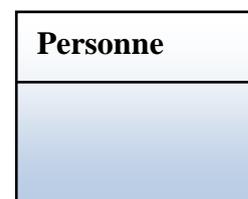
- le nom
- le prénom
- la date de naissance
- le lieu de naissance
- Adresse

Donc, une entité peut être un objet, un événement, un lieu, une personne, ..., une chose, identifiable sans ambiguïté.

- **Classe d'entités** : c'est un regroupement d'entités de même nature. Un ensemble d'entités.
- **Formalisme de classe d'entité** : On schématise une entité par un rectangle.



Exemple de la classe d'entités : **Personne**

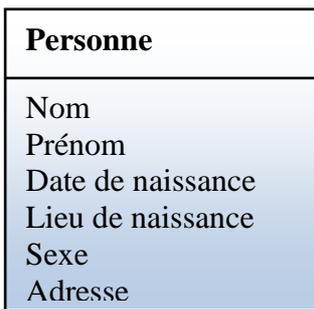


Une entité est une valeur particulière d'une classe d'entités. Nous nous intéressons bien sûr à la classe d'entités

4.1.2 Propriétés ou Attributs

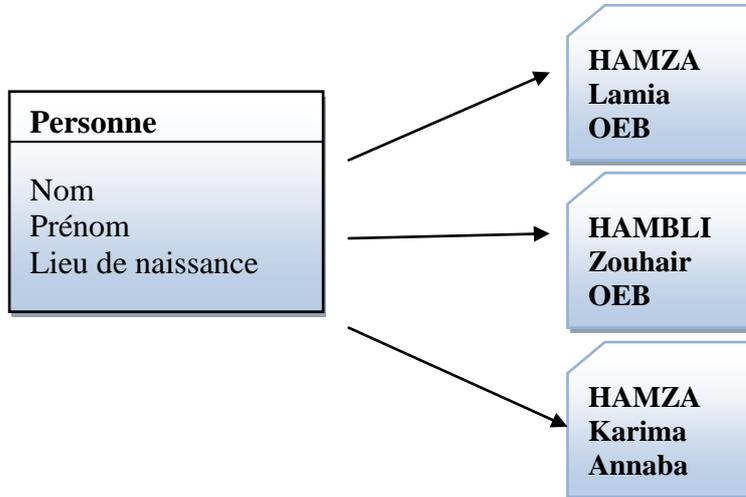
Les *propriétés* ou les *attributs* sont les caractéristiques décrivant les entités et doivent être représentés comme une liste de mots, la plus simple possible, dans le cadre de l'entité correspondante. On devra préciser le type des données attendues pour chaque attribut ou propriété.

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



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

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



4.1.3 Identifiant

- **Objectif** : chaque occurrence doit pouvoir être repérée de manière unique et sans ambiguïté, pour être distinguée de toutes les autres.
- **Identifiant** : propriété ou groupe de propriétés dont la valeur identifie sans ambiguïté une entité ou une liaison d'une classe (**identifiant=clé primaire**).
- **Identifiant d'une classe d'entités**: est un attribut ou groupe d'attributs qui permet de repérer une occurrence de manière unique.
- **Représentation d'un attribut** : l'identifiant est souligné pour le représenter dans une entité.

- Il est préférable de choisir l'identifiant le plus court, le plus naturel, exemple : (n°bon de commande, n° étudiant,...)

4.1.4 Association

Une association décrit un lien entre deux ou plusieurs entités. Chaque association possède un nom, généralement un verbe à l'infinitif.

En général une association relie deux entités ; elle peut toutefois relier une entité avec elle-même (relation réflexive) ou relier trois voire n entités (relation ternaire / n-aire)

- Une association peut avoir des attributs : on parle d'association porteuse de données.



4.1.5 Typologie des associations (cardinalités des rôles)

Une association permet de relier, une ou plusieurs entités. Les cardinalités précisent la participation de l'entité concernée à la relation.

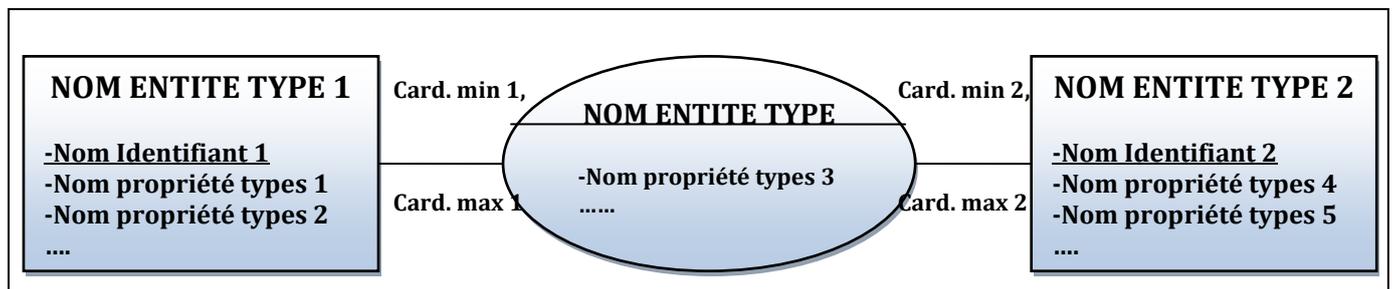


Figure 4.5 : Types d'association

Quatre types d'association :

- De 0,1 : une entité de A peut être reliée à aucune ou à une seule entité de B
- De 1,1 : une entité de A est reliée à une seule entité de B
- De 0 à plusieurs (0,N) : une entité de A peut être reliée à aucune ou à plusieurs entités de B.
- De 1 à plusieurs (1,N) : une entité de A peut être reliée à une ou plusieurs entités de B

En général, Les cardinalités d'une association sont défini par deux nombres (min, max) représentant le nombre de fois minimum et le nombre de fois maximum qu'une entité participe à une association. Les valeurs possibles sont : (0,1), (1,1); (0,N), (1,N)

- Min : Correspond à la réponse à la question : combien de fois au moins une entité de A est reliée à une entité de B.

- Max : correspond à la réponse à la question : combien de fois au plus une entité de A est relié à une entité de B.

Remarque : Il faut poser ces questions dans les deux sens de A vers B puis de B vers A.

4.1.6 Cas particulier d'associations

❖ Cas d'une association ternaire

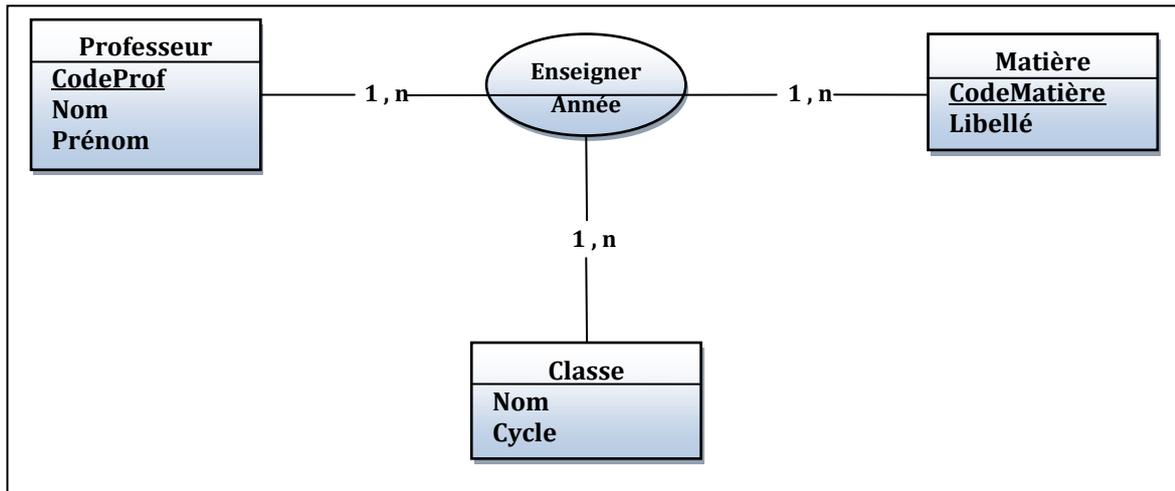


Figure 4.6 : Exemple d'une association ternaire

Chaque occurrence de la relation enseigner associe un professeur à une matière et une classe pour une année donnée. Ou encore, ce modèle nous permet de montrer pour chaque année scolaire quelle matière est enseignée dans quelle classe par quel professeur.

Généralement, si une ou plusieurs des entités liées à une association ternaire possèdent une cardinalité maximale de 1, la modélisation n'est pas optimisée. Il faudrait mieux décomposer l'association ternaire, c.à.d. la représenter par deux associations binaires.

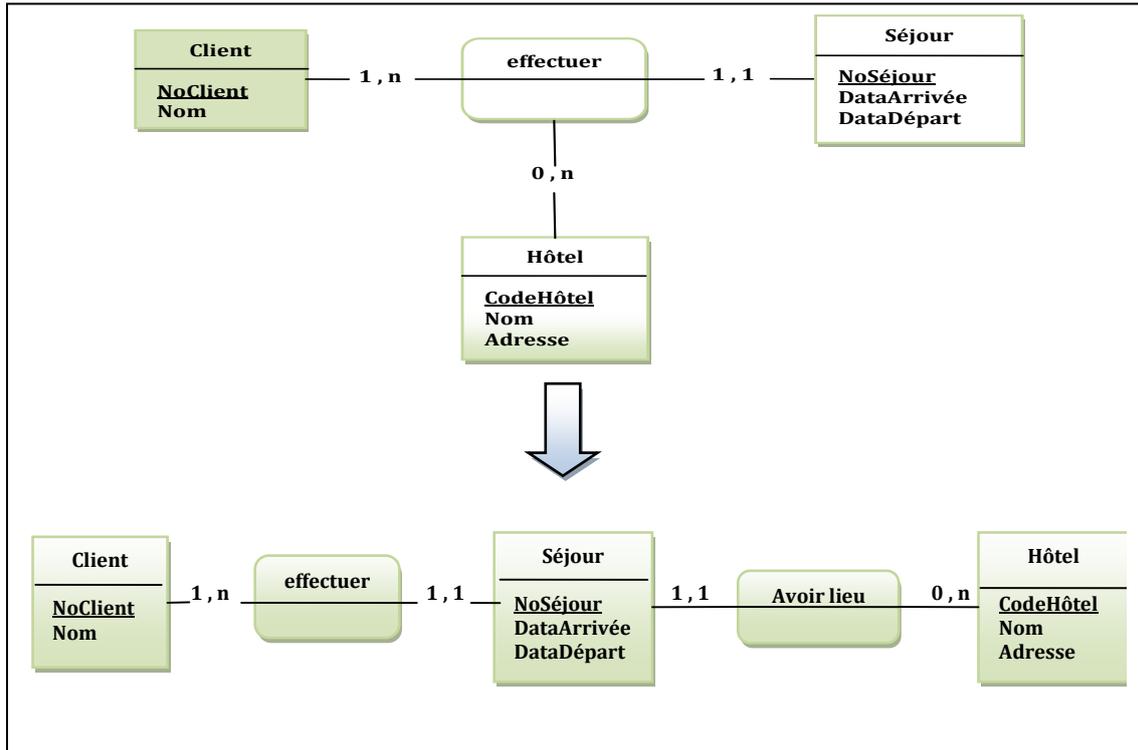


Figure 4.7 : Décomposition de la relation ternaire.

- ❖ **Cas d'association réflexive** : une association réflexive, est une relation, dont les deux pattes sont liées à une même entité. En général, la signification des pattes d'une association réflexive devrait être clarifiée par l'indication d'un rôle.

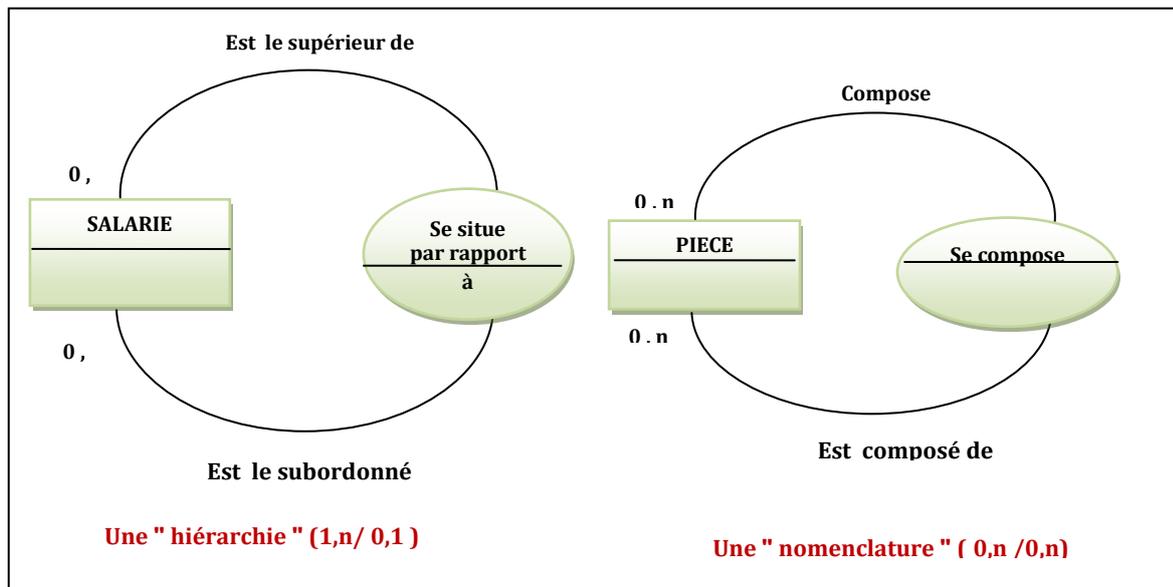


Figure 4.8 : Exemple d'une relation réflexive.

4.2 Modèle logique de données (MLD)

Le MLD constitue une étape intermédiaire entre le modèle conceptuel et le modèle physique de données. C'est le **MCD** auquel on rajoute la définition de l'**organisation logique** des données et en l'**optimisant** compte tenu des traitements à appliquer aux données.

A ce niveau, on doit choisir le mode d'organisation des données :

- modèle hiérarchique
- modèle réseau
- modèle relationnel
- fichiers classiques

Le mode d'organisation des données présenté dans ce cours est le modèle relationnel.

4.2.1 Les concepts de base de modèle relationnel

Le modèle relationnel est un modèle d'organisation des données se forme des relations.

- **Attribut** : un attribut est un identificateur (un nom) décrivant une information stockée dans une base.
Exemples d'attribut : l'âge d'une personne, le nom d'une personne, le numéro de sécurité sociale.
- **Domaine** : Le domaine d'un attribut est l'ensemble, fini ou infini, de ses valeurs possibles.
Par exemple, l'attribut numéro de sécurité sociale a pour domaine l'ensemble des combinaisons de quinze chiffres et nom a pour domaine l'ensemble des combinaisons de lettres (une combinaison comme cette dernière est généralement appelée chaîne de caractères ou, plus simplement, chaîne).
- **Relation** : Une relation est un sous-ensemble du produit cartésien de n domaines d'attributs ($n > 0$).
Une relation est représentée sous la forme d'un tableau à deux dimensions dans lequel les n attributs correspondent aux titres des n colonnes.
- **Schéma de relation** : Un schéma de relation précise le nom de la relation ainsi que la liste des attributs avec leurs domaines. Nom de relation (Clé primaire, att1, att2, ..., attn). *Exemple* : Clients (numclient, nom client, prénom, adresse)

Sur un système de gestion de base de données relationnel, les relations deviennent des tables, les attributs se nomment les champs, Chaque table contient des enregistrements

Le tableau suivant montre un exemple de relation Personne (N° sécu , Nom , Prénom) et précise son schéma.

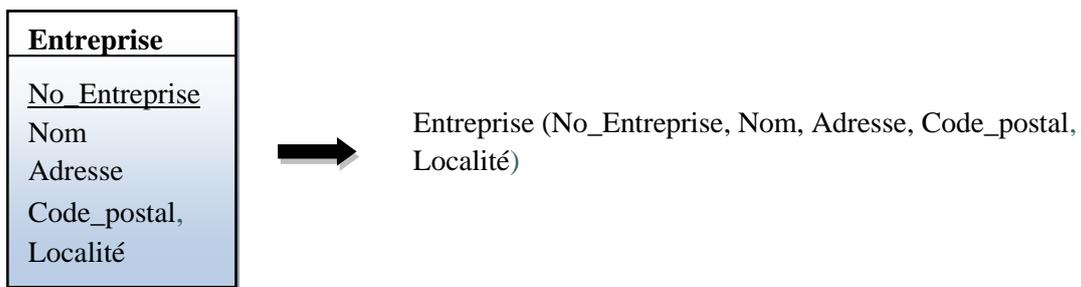
N° Sécu	Nom	Prénom
354338532195874	Hamza	Lamia
345353545435811	Hamza	Said
173354684513546	Hambli	Adem
973564213535435	Hambli	Sarra

Tableau 4.3 : Exemple d'une relation.

- **Clé primaire :** la clé primaire d'une relation est l'attribut qui permet d'identifier de manière unique une relation, la clé primaire est soulignée.
- **Clé étrangère :** une clé étrangère dans une relation est formée d'un ou plusieurs attributs qui constituent une clé primaire dans une autre relation.
- **Schéma relationnel (Modèle relationnel) :** un schéma relationnel est constitué par l'ensemble des schémas de relation.

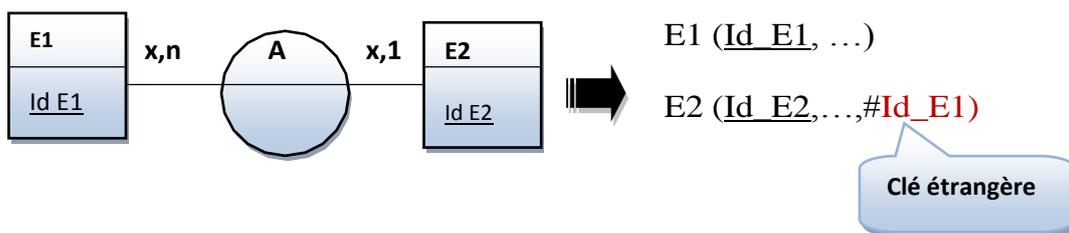
4.2.2 Transformation du modèle MCD au MLD (Règles de passage)

- **Transformation des entités :** Toute entité est transformée en une relation. Les propriétés de l'entité deviennent les attributs de la relation. L'identifiant de l'entité devient la clé primaire de la relation.



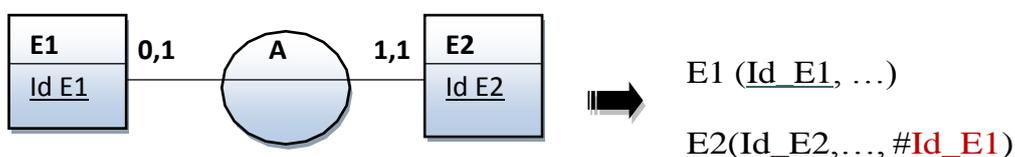
➤ **Transformation des associations**

a) Transformation des associations binaires du type (x,n)-(x,1)

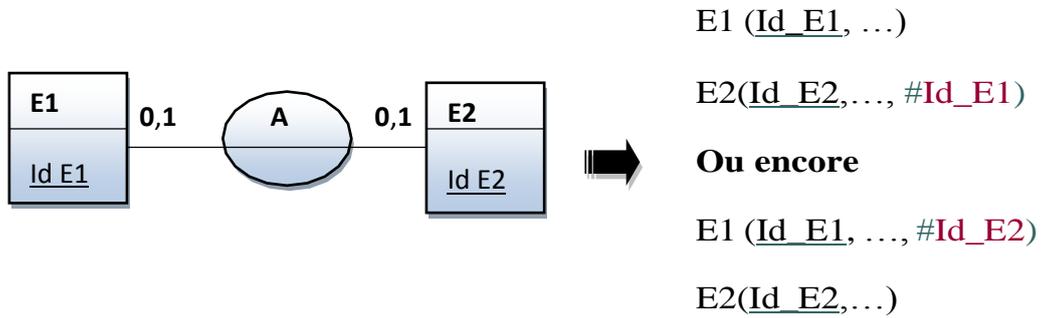


b) Transformation des associations binaires du type (x,1)-(x,1)

❖ **Cas d'une association binaire (0,1)-(1,1)**

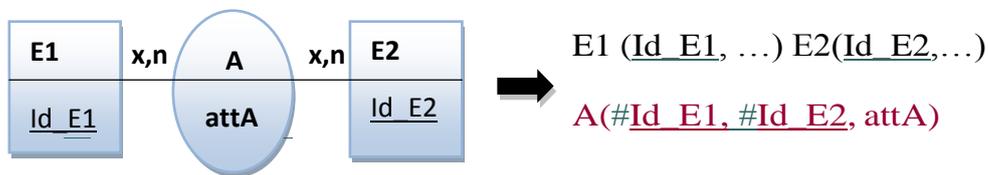


❖ Cas d'une association binaire (0,1) (0,1)

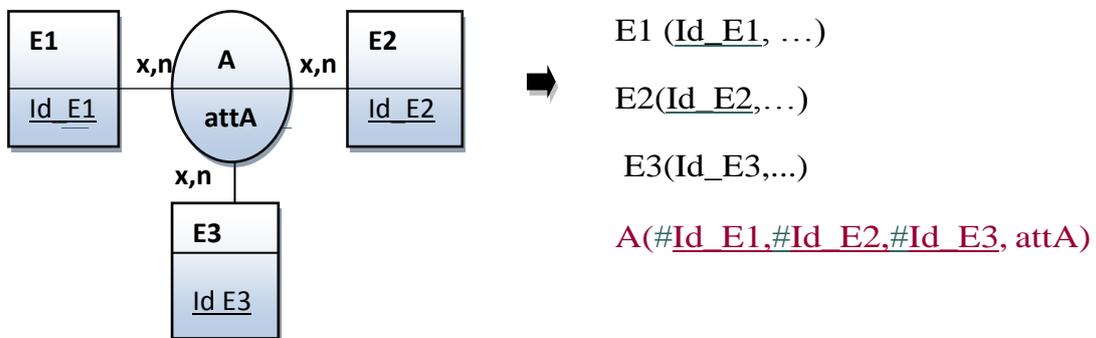


c) Transformation des associations de type (x,n) (x,n)

❖ Transformation des associations binaires :

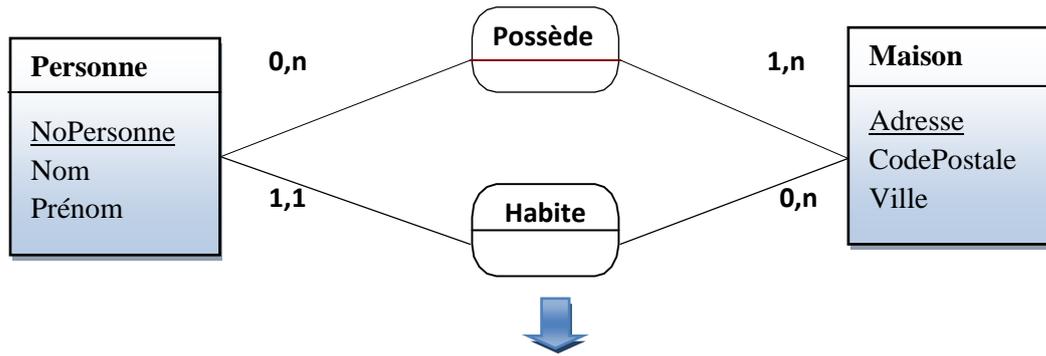


❖ Transformation des relations ternaires :



d) Transformation de plusieurs relations entre deux entités

Les règles générales s'appliquent. On traite chacune des associations indépendamment des autres, ce qui peut donner lieu à l'ajout de plusieurs références.



Personne (NoPersonne, Nom, Prénom, #Adresse)

Maison(Adresse, CodePostale, Ville)

Possède (#NoPersonne, #Adresse)

5 Concepts pour la modélisation dynamique (modélisation des traitements)

5.1 Modèle conceptuel des traitements (MCT)

Le modèle conceptuel des traitements (MCT) permet d'effectuer une représentation conceptuelle des traitements effectués dans l'entreprise. Il s'appuyant sur la spécification des règles de gestion, il représente : les événements, les résultats, les opérations et les synchronisations. Le MCT donne une représentation dynamique du système d'information de l'entreprise.

5.2 Concepts de base de MCT

5.2.1 L'événement

Un **événement** représente un changement dans l'univers extérieur au système d'information, ou dans le système d'information lui-même.

- un événement externe est un changement de l'univers extérieur
- un événement interne est un changement interne au système d'information

Un événement interne est représenté par une ellipse en trait plein, et un événement externe est représenté par une ellipse en trait pointillé.



Exemples : réception commande, absence de réponse

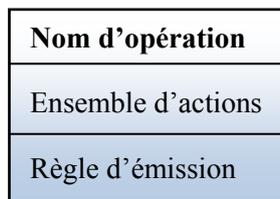
Un événement est généralement désigné par un verbe au participe passé ou par un substantif dérivé.

5.2.2 L'opération

Une opération est un ensemble d'actions exécutées par le système suite à un événement, ou à une conjonction d'événements.

Exemples : traitement de commande, paiement facture

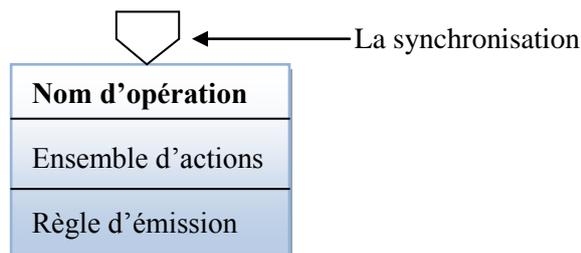
- ❖ Une action est une fonction élémentaire. Entre les actions d'une opération, il n'y a pas d'état d'attente, et leur déroulement est séquentiel.
- ❖ Une règle d'émission définit la condition sous laquelle des événements résultats seront produits par une opération. Une opération peut avoir une ou plusieurs règles d'émission, une règle gérant l'émission de un ou plusieurs événements résultats.
- ❖ Une opération peut ne pas avoir de règle d'émission. Dans ce cas, l'émission des événements est inconditionnelle.



5.2.2 La synchronisation

La synchronisation est l'association de deux ou plusieurs événements pour le déclenchement d'une opération. C'est une expression booléenne formée à partir des opérateurs ET, OU et NON.

Exemples : Facture reçue et marchandise réceptionnée.



5.2.3 Le résultat

C'est la réponse du système aux événements ayant déclenché une opération. Exemples : commande traitée, marchandise livrée.

Remarque:

- Une opération peut produire plusieurs résultats.
- La production des résultats peut être soumise à des conditions de sortie de l'opération.

- Le résultat d'une opération peut participer en tant qu'événement dans une autre opération, c'est l'événement interne.
- Une opération ne peut pas être déclenchée que par des événements internes. Il doit y avoir au moins un événement externe.

5.2.4 Le processus

C'est un enchaînement synchronisé d'opérations représentant une unité homogène de traitement. Un processus est propre à un domaine d'activité. Un domaine peut être représenté par un ensemble de processus. Par exemple : processus de facturation.

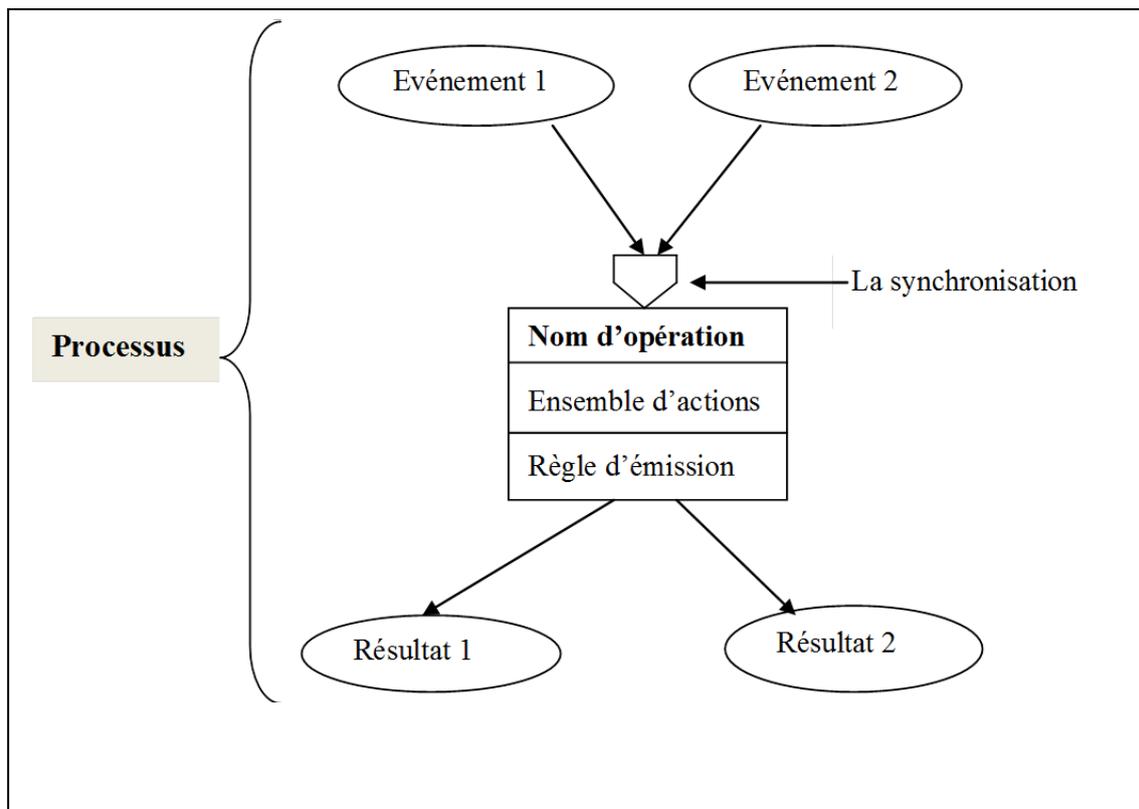


Figure 4.9 : Formalisme d'un processus

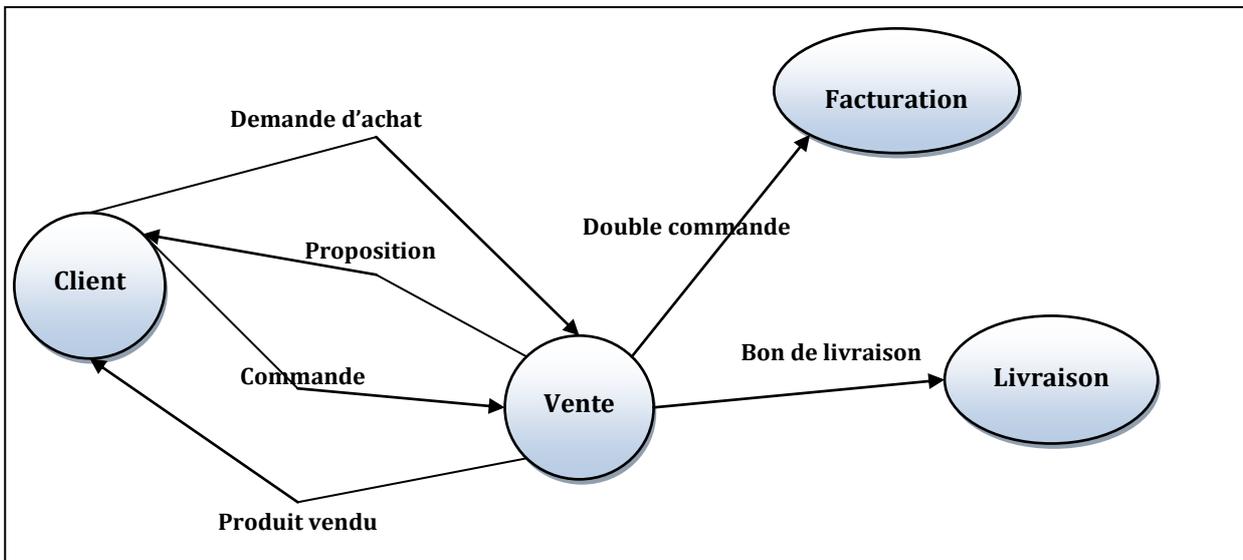
5.3 Construction du MCT

Le modèle conceptuel des traitements permet de représenter schématiquement la gestion des événements. Le MCT se représente par un enchaînement de processus. La construction d'un MCT se fait selon les étapes suivantes :

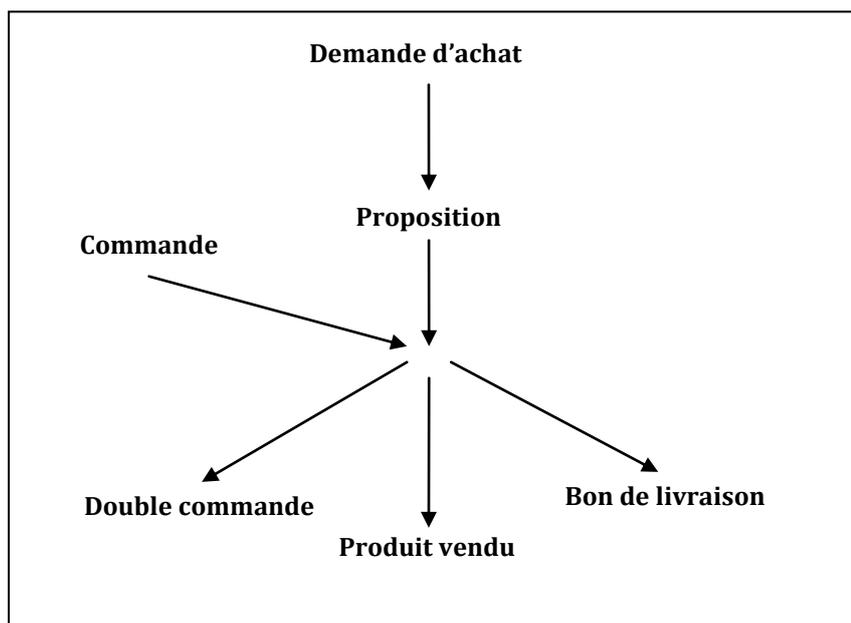
1. Construction d'un diagramme des flux entre les acteurs externes et internes pour chaque domaine.
2. Transformation des flux en graphe d'enchaînement des flux (événements/résultats).
3. Transformer ce graphe d'enchaînement des flux en MCT en remplaçant chaque passage d'un ensemble d'événements à un ensemble de résultats par une opération.

Exemple

1. Diagramme des flux du domaine vente :



2. Graphe d'enchaînement des événements résultats:



3. Construction du MCT

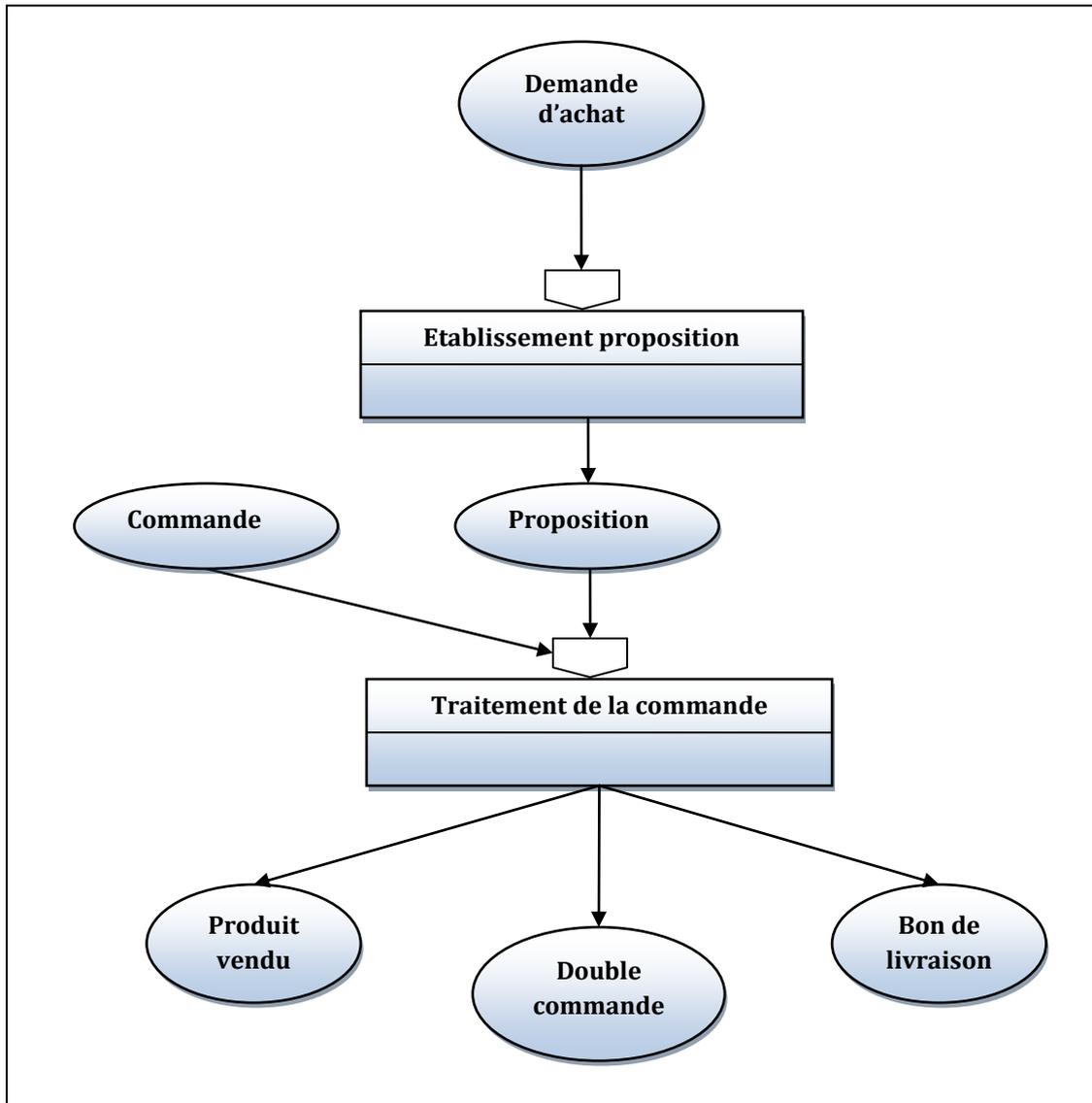


Figure 4.10: Exemple d'un modèle MCT.

Conclusion

Selon les concepts présentés dans ce chapitre : Merise est une démarche de construction de système d'information, elle se caractérise par une double démarche : par niveau d'abstraction et par étape de construction. Le processus de modélisation de Merise basé sur la séparation entre la modélisation des données et la modélisation des traitements. En ce qui concerne la modélisation des données, Merise utilise le modèle conceptuel des données qui est basé sur les concepts de base du modèle Entité-Association. Ensuite elle transforme les résultats obtenus en modèle relationnel. Pour la modélisation des traitements : Merise utilise le modèle conceptuel des traitements pour identifier les fonctionnalités d'un système selon une approche "top / down". Ensuite elle transforme ce modèle pour prendre en compte d'autres critères importants dans le côté organisationnel d'un système.

Références

1. Abdellatif. A , Cours : Conception des systèmes d'information, Département des sciences de l'informatique, Faculté des Sciences de Tunis, 2008.
2. Bertrand LIAUDET, MERISE – Introduction , <http://bliaudet.free.fr/IMG/pdf/MERISE-01-Introduction.pdf>
3. Chaker Kharrat Fatma, Modélisation et Conception des systèmes d'Information, MERISE, École Supérieure de la Statistique et de l'Analyse de l'Information (ESSAI), <https://pdfslide.tips/download/link/mod-elisation-et-conception-des-systemes-d-information-merise-1-fchaker>
4. El khalkhali. M, Cours : Le modèle conceptuel des données (MCD), Université Abdelmalek Essaadi, Ecole nationale de commerce et de gestion. <https://fr.scribd.com/doc/139164297/Mcd>
5. Jean Jacques, Méthode de Conception des Systèmes d'Information Auteur, https://foad-mooc.auf.org/IMG/pdf/cours_mcsi.pdf
6. Vailly. A et André. P, Conception des systèmes d'information – Panorama des méthodes et des techniques, Ellipses, collection TECHNOSUP / Génie Logiciel, 2001.