

Chapitre 4 : Notion de méthodes d'analyse et de conception d'un SI

4.1 Pourquoi une méthode ?

Méthode : une démarche et un formalisme

Démarche : succession d'étapes pour :

- Mieux maîtriser le déroulement d'un projet,
- Meilleure visibilité pour les utilisateurs sur certains résultats intermédiaires et garantir que le résultat final sera celui attendu.

Formalisme défini par:

- Un langage formel,
- Un langage semi formel généralement graphique,
- Un langage naturel.
- Fonction :
 - Représenter le monde réel tel qu'il est perçu par le concepteur,
 - Outil de communication entre informaticiens et utilisateurs,
 - Constitué par un ensemble de modèles permettant d'assurer une bonne compréhension des besoins des utilisateurs.

Modèles

- Représentation abstraite de la réalité qui exclut certains détails du monde réel,
- Permet de réduire la complexité d'un phénomène en éliminant les détails qui n'influencent pas son comportement significatif,
- Représente ce que le concepteur croit important pour la compréhension et la prédiction du phénomène modélisé, les limites du phénomène modélisé dépendent des objectifs du modèle.

4.2 Historique des méthodes

4.3 Les méthodes cartésiennes

4.4 Les méthodes systémiques

4.5 Les méthodes orientées objet

LA METHODE MERISE

5.1 Présentation de la méthode

La méthode d'analyse Merise a été créée à la fin des années 70, par le Ministère de l'Industrie afin de doter les administrations et les entreprises publiques d'une méthodologie rigoureuse et qui tiendrait compte des nouveaux paradigmes tels les bases de données et l'informatique répartie et même dans son extension de l'orienté objet.

La méthode Merise propose une approche de la conception qui sépare la conception des données de celle des traitements, avec un cycle d'abstraction par niveaux.

A chaque niveau deux modèles sont fournis avec un certain nombre de documents (MCD, MLD, MCT, MOT, MPD, MOPT). Ces modèles sont indispensables à l'élaboration et à l'implémentation de tout projet informatique.

L'élaboration des modèles de traitements a non seulement pour but de définir les traitements à effectuer, mais également de valider les modèles de données.

La méthode Merise préconise, non pas d'effectuer l'analyse des données, puis ensuite celle des traitements, mais plutôt de mener en parallèle, à chaque niveau, l'analyse des données et celle des traitements.

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

- Le niveau conceptuel.
- Le niveau organisationnel (logique).
- Le niveau opérationnel (physique).

Ces différents niveaux répondent aux questions suivantes :

Modèles Conceptuels	MCD	Quoi Faire ?
	MCT	Avec Quelles Données ?
Modèles Organisationnels	MLD	Qui ?
	MOT	Où ?
		Quand ?
Modèles Opérationnels	MOPT	Comment ?
	MPD	

- **Le niveau conceptuel**

Le niveau conceptuel représente les informations et leurs relations d'une part, les utilisations qui en sont faites et les contraintes d'autre part. Ces définitions sont établies en faisant abstraction de toute contrainte liée à l'organisation.

En terme de données, cette description fait appel au formalisme *Entité-Association* et se traduit par des entités de base et par des relations avec ces entités.

En terme de traitements, ces mêmes entités vont être décrites par leurs sollicitations ou par les réactions qu'elles déclenchent de la part du système d'information, donc par les traitements dont elles sont les causes et les conséquences. Ceci se fait à l'aide d'événements, de synchronisation et d'opérations.

- **Le niveau organisationnel ou logique**

Alors qu'au niveau conceptuel est exprimé la réalité perçue par l'entreprise dans son ensemble, le niveau organisationnel exprime cette même réalité telle qu'elle est vécue par les acteurs quels qu'ils soient. A ce niveau, aucune différence n'est faite entre les hommes et les machines. On intègre à l'analyse les critères liés à l'organisation.

- En terme de données, les entités et relations suscitent la création de tableaux. La vue logique est nécessairement orientée vers une classe de solutions.

- En terme de traitements, les événements décrits ne sont pas des événements temporels mais des événements à dominante spatiale.

- **Le niveau opérationnel ou physique**

C'est une représentation des moyens qui vont effectivement être mis en œuvre pour gérer les données ou activer les traitements. Le niveau physique apporte des solutions techniques.

- En ce qui concerne les données, il y a passage d'une classe de solutions à un produit de cette classe. Concrètement, cela se traduira par l'utilisation d'un SGBD. On effectue des choix sur les méthodes de stockage et d'accès.

- En terme de traitements, le modèle opérationnel décrira l'architecture des programmes qui vont activer les différentes tâches de l'ordinateur. En aucun cas à ce niveau, il n'y a de programmation effective.

5.2 Les modèles de représentation

- **Le modèle conceptuel de données (MCD)**

Le Modèle Conceptuel des Données (MCD), permet la description statique du système d'information à l'aide des concepts d'entité et d'association.

- **Modèle Logique de données (MLD)**

Le Modèle Logique des Données (MPD) permet une représentation de la structure physique d'une base de données.

- **Modèle organisationnel des traitements (MOT)**

Le Modèle Organisationnel des Traitements (MOT) offre une vision globale du système d'information. C'est à ce stade de l'analyse des traitements que sont mises en évidence les contraintes réelles de l'organisation.

- **Modèle physique de données (MPD)**

- **Modèle opérationnel des traitements (MOPT)**

5.3 Le modèle conceptuel de données

5.3.1 Généralités

5.3.2 Construction d'une liste d'informations

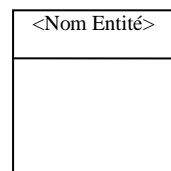
5.3.3 Qu'est ce qu'une donnée ?

5.3.4 Formalisme du M.C.D.

- **Entité**

Une **entité** est la représentation d'un objet matériel ou immatériel, ayant une existence propre, et conforme au choix de gestion de l'entreprise.

Le symbole de représentation d'une entité :

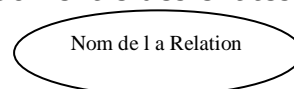


L'**occurrence** d'une entité est un élément individualisé appartenant à cette entité. Cette notion est représentée au niveau du MCD au travers du concept de cardinalité.

- **Association**

Une **association** traduit le fait qu'il existe une relation entre des entités.

Le symbole de représentation d'une association :



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).

- **Propriété**

Une **information** est une donnée élémentaire que l'on peut attacher à une entité ou à une association ; dans ce cas, une information est nommée **attribut** ou **propriété**.

Chaque entité doit être munie d'au moins un **identifiant**, qui est une propriété particulière telle qu'à chaque valeur de cette identifiant corresponde une et une seule occurrence de l'entité considérée. Lors de la génération des tables d'une base de données, l'identifiant prend le rôle de clé primaire. Placée en premier au sein de l'entité, la propriété jouant le rôle d'identifiant doit être soulignée.

- **Lien et cardinalités**

Un **lien** représente une liaison entre une entité et une association. Il est caractérisé par sa **cardinalité**. Cette cardinalité est constituée d'une borne minimale et d'une borne maximale.

La cardinalité minimale est le nombre de fois minimum qu'une occurrence d'une entité participe aux occurrences de l'association.

La cardinalité maximale est le nombre de fois maximum qu'une occurrence d'une entité participe aux occurrences de l'association.

Les cardinalités possibles sont :

0 , 1 aucun ou un seul

1 , 1 un et un seul

0 , n aucun ou plusieurs

1 , n au moins un ou plusieurs

- **Héritage**

L'héritage peut être défini comme la possibilité d'exprimer des caractéristiques communes à plusieurs entités formant une même famille.

Chaque entité "père" et "filles" possède un attribut de génération qui détermine l'existence réelle au niveau physique de la table correspondante.

5.4 Le modèle conceptuel de traitement

- **Graphe des flux**

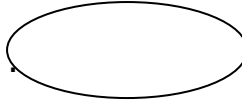
Le graphe des flux permet une description des flux d'information échangés entre acteurs.

- **Acteur**

Un **acteur** est une entité, humaine ou matérielle, intervenant dans le système d'information.

Les acteurs se divisent entre deux catégories, internes et externes, selon qu'ils appartiennent ou non à l'entreprise.

Un acteur est représenté par le symbole :



- **Flux**

Un **flux** est un échange de biens ou d'informations entre un acteur émetteur et un autre récepteur.

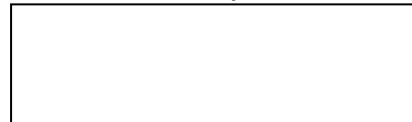
- **MCT**

Le Modèle Conceptuel des Traitements (MCT) permet une description dynamique du système d'information à l'aide des concepts d'opération et d'événement.

- **Opération**

Une **opération** est une production de flux d'information. Une opération est définie "immatériellement", sans contrainte organisationnelle. Elle décrit aussi bien la gestion manuelle, que la gestion automatisée. Une opération se décompose en actions.

On représente une opération par le symbole :



- **Action**

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 action peut faire référence à une ou plusieurs règles de gestion. Elle peut utiliser une ou plusieurs entités et/ou associations pour des actions de création, modification, suppression ou consultation.

- **Règle de gestion**

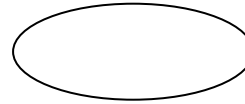
Une **règle de gestion** est une loi qui, à l'échelle de l'entreprise, va s'appliquer systématiquement dans les divers cas qu'elle est censée régir.

Les règles de gestion servent à définir l'ensemble des règles à respecter pour les actions. Une même règle de gestion peut être appliquée à plusieurs actions.

- **Evénement**

Un **événement** est un flux de nature quelconque ou un fait concourant au lancement d'une opération. Un événement est généralement désigné par un verbe au participe passé ou par un substantif dérivé.

On représente un événement par le symbole :



- **Synchronisation**

Une **condition de synchronisation** est représentée par une condition booléenne liant les événements déclencheurs grâce aux opérateurs logiques " Et ", " Ou " et " Non ".

L'opération n'est pas déclenchée si la condition n'est pas réalisée.

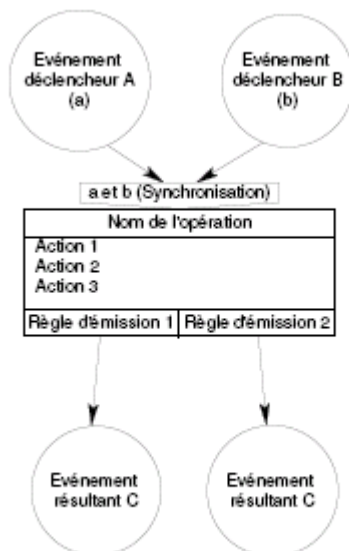
- **Règle d'émission**

Une **règle d'émission** définit la condition sous laquelle des événement résultats seront produits par une opération.

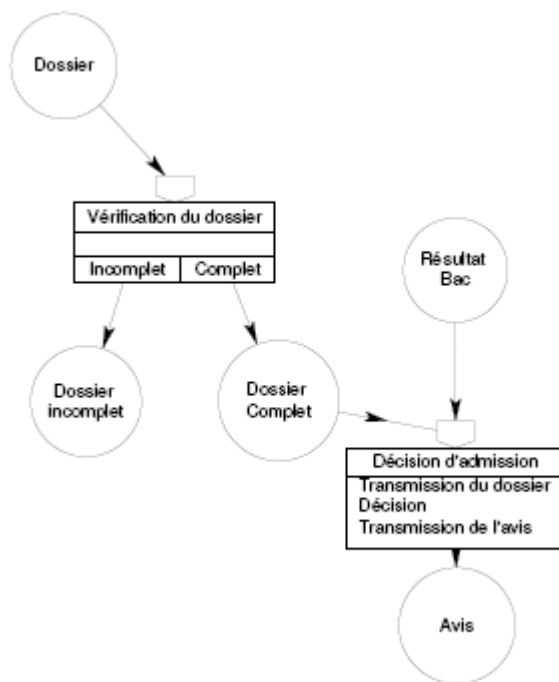
Une opération peut avoir une ou plusieurs règles d'émission, une règle gérant l'émission de un ou plusieurs événements résultats.

Une opération peut ne pas avoir de règle d'émission. Dans ce cas, l'émission des événements est inconditionnelle.

On représente un MCT de la façon suivante :



Exemple



5.4.1 Généralités

5.4.2 Acteur

5.4.3 Événement, Résultat

5.4.4 Opération

5.4.5 Règles d'émissions

5.4.6 Synchronisation

5.4.7 Règles de Syntaxe

5.4.8 La vérification du modèle (vu en séance de cours)

5.5 Le modèle organisationnel de traitement

Le MOT découle à priori du MCT établi préalablement.

- **Acteur**

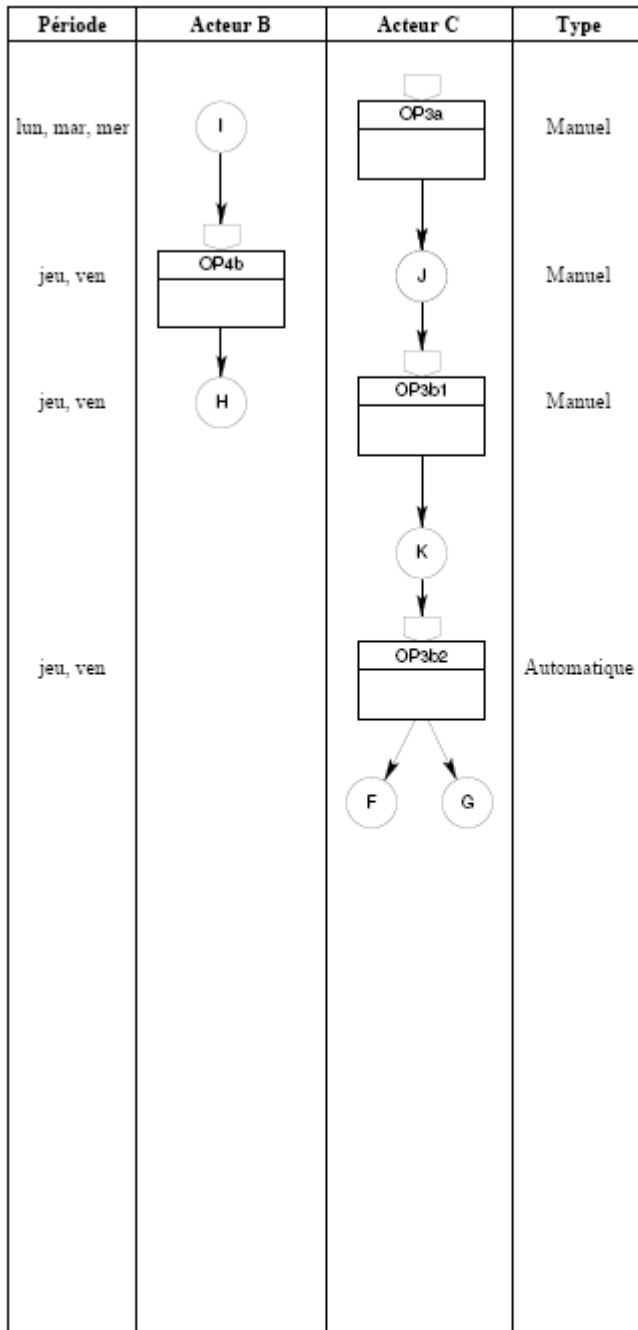
Un **acteur** est une entité organisationnelle chargée d'exécuter un certain nombre de phases.

- **Phase**

Une **phase** est une suite ininterrompue de tâches de même périodicité, exécutées par le même acteur. Une **règle d'organisation** découle d'un choix d'organisation. Elle peut donc être appliquée à une ou plusieurs tâches. Elle correspond souvent à une règle de gestion du niveau conceptuel, à laquelle on ajoute des contraintes organisationnelles.

- **Type et période**

Formellement, un MOT est représenté de la même manière qu'un MCT auquel on ajoute une colonne à gauche, représentant la **période** (date ; durée), et une colonne à droite, représentant le **type** de la tâche (manuel / automatique / interactif).



5.6 Le modèle logique de données

- **Table**

Une **table** est la représentation physique équivalente d'une entité.

Elle est composée de **colonnes** qui sont elles-mêmes les équivalents des propriétés.

Les **lignes** des tables correspondent aux occurrences dans le MCD.

Les identifiants deviennent les **clés primaires**.

Une **référence** est la traduction au niveau physique d'une association entre entités. Elle exprime un lien entre deux colonnes de deux tables pour indiquer que ces deux colonnes représentent la même information.

Ex :

- **Règles de génération**

Le MPD peut être élaboré directement à partir du MCD en suivant un certain nombre de règles prédéfinies :

Une entité devient une table

Une propriété d'entité devient une colonne

L'identifiant d'une entité devient une clé primaire

Une association (0,n) – (0,1) provoque une migration d'une clé étrangère et la naissance d'une référence :

Pour une association portant un lien identifiant (1,1), la clé étrangère migrée fait partie de la clé primaire de la table dans laquelle elle migre.

Une association (0,n) – (0,n) donne naissance à une table. Les identifiants des entités auxquelles l'association est reliée migrent dans cette table. La clé primaire est alors constituée de ces colonnes migrées.

5.7 Le modèle physique de données

5.8 Le modèle opérationnel des traitements

Chapitre 6 : Le Langage de Modélisation Unifié (UML)

6.1 Genèse d'UML

UML (*Unified Modeling Language*), que l'on peut traduire par **langage de modélisation unifié** est une notation permettant de modéliser un problème de façon standard.

La création des logiciels étant fondamentalement une tâche système dont les éléments sont en corrélation complexe. La conception d'un logiciel répondant aux aspirations personnelles du client passe nécessairement par une bonne conception (modélisation)

UML comble une lacune importante des technologies objet car Il permet d'exprimer et d'élaborer des modèles objet, indépendamment de tout langage de programmation. Il a été pensé pour servir de support à une analyse basée sur les concepts objet.

UML peut être utilisé pour **visualiser, spécifier, construire et documenter** les artefacts d'un système à forte composante logicielle. Il est également adapté à la modélisation des systèmes, depuis les systèmes informatiques d'entreprise jusqu'aux applications distribuées basées sur le Web en passant par les systèmes temps réels embarqués. C'est un langage simple à comprendre et à utiliser mais aussi et surtout très expressif qui couvre toutes les perspectives nécessaires au développement puis au déploiement des tels systèmes.

6.2 Les diagrammes d'UML

Un diagramme est la représentation graphique d'un ensemble d'éléments qui constituent un système. La plupart du temps il se représente sous la forme d'un graphe connexe où les sommets correspondent aux éléments et les arcs aux relations.

Un diagramme donne à l'utilisateur un moyen de visualiser et de manipuler les éléments de modélisation. Le Meta modèle UML fournit une panoplie d'outils permettant de représenter l'ensemble des éléments du monde objet (classe, objet) ainsi que les liens qui les relient.

Toute fois étant donné qu'une seule représentation est trop subjective, UML fournit un moyen astucieux permettant de représenter diverses projections d'une même représentation grâce aux vues. Une vue est constituée d'un ou plusieurs diagrammes. UML définit neuf sortes de diagrammes structurels et comportementaux pour représenter respectivement les vues statiques et dynamiques.

Chaque vue est articulée autour de plusieurs diagrammes, chacun d'eux étant dédié à la représentation des concepts particuliers d'un système logiciel.

Les vues statiques

Les vues statiques représentent les diagrammes structurels suivants :

- Diagramme de classe
- Diagramme d'objet
- Diagramme des cas d'utilisation
- Diagramme de composant
- Diagramme de déploiement

Les vues dynamiques

La classification dynamique est un concept qui permet de changer le type d'un objet de façon dynamique.

- Diagramme d'activité
- Diagramme de séquences
- Diagramme d'état de transition
- Diagrammes de collaboration