

Méthode de Conception des Systèmes d'Information

Auteur : NGANG BILOUNGA Jean Jacques

Discipline : Analyse Informatique

Objectifs : L'objectif de cet enseignement est double. Tout d'abord faire comprendre à des étudiants en informatique les notions de base sur les systèmes d'information. Et ensuite étudier et utiliser une méthodologie pour la conception de systèmes informatiques.

A la fin de ce cours, l'étudiant sera capable de :

- comprendre les objectifs de la méthode MERISE et les différents niveaux de sa démarche,
- connaître le vocabulaire spécifique de la méthode,
- discerner la complémentarité des approches de type systémique et analytique,
- savoir formaliser les modèles conceptuel et organisationnel de MERISE,
- identifier les rôles et responsabilités des différents acteurs impliqués dans le processus de conception.

Prérequis : - notions de programmation Description :

Le cours commence par la définition de concepts de base utilisés dans la conception des systèmes d'information. Il replace ceux-ci dans le contexte d'informatisation. Une méthode d'analyse et de conception des systèmes d'information est ensuite étudié : MERISE. D'abord dans ses aspects fondamentaux, puis dans ses aspects les plus pratiques qui permettent de mettre en œuvre la méthode pour des cas concrets.

Introduction Description :

Lorsqu'on parle d'informatisation, les systèmes d'information sont incontournables. Il est normal que dans une filière qui concerne l'informatisation, les systèmes d'informations soient largement étudiés. Étant donné l'importance du sujet, cette étude se divise en deux parties

Ce cours constitue la première partie de l'étude des systèmes d'informations (S.I. ou I.S. : Information System en anglais).

Il a pour principal objet de présenter la méthode MERISE , très largement utilisée dans le monde francophone. Il commence par une brève présentation de quelques notions liées au sujet. Il se poursuit par l'étude proprement dite de Merise. Cette étude concerne essentiellement les modèles utilisés. MERISE étant également une démarche, celle-ci est ensuite présentée. Le cours se termine par une étude de cas qui montre la succession d'étapes dans la mise en œuvre de MERISE.

Sur les systèmes d'information

Description : Nous aborderons dans cette section les notions et concepts utiles nous permettant d'évoquer l'analyse informatique dans les termes appropriés

Concepts de base

Quelques définitions utiles.

Processus

séquence de phénomènes dynamiques (mouvements, réactions chimiques, activités cellulaires, opérations techniques, actions ou comportements, interactions humaines) menant à des résultats déterminables. En analyse des systèmes : tout changement dans le temps de matière, d'énergie ou d'information qui se produit dans le système, traitant ces variables d'entrée et les menant aux variables de sortie.

www.mcxapc.org/static.php

Ensemble d'opérations, logiquement liées, aboutissant à certains résultats. En conception de systèmes d'information, selon la méthode Merise, le processus se situe au niveau du modèle conceptuel de traitement.

revuesim.free.fr/index.php

Étapes ou phases de la méthode d'exécution ou de fonctionnement de quelque chose, que ce soit dans les systèmes commerciaux ou techniques, mettant en cause les actions de plus d'une personne, d'une unité ou d'une division. Système d'opérations dans la production de quelque chose. Série de mesures, de changements ou de fonctions qui produisent un résultat final.

www.cchra-ccarh.ca/fr/phaseII/glossaire.asp

Information : encore appelée donnée. Élément de connaissance susceptible d'être codé pour être conservé, traité ou communiqué.

Systèmes : Un système est un assemblage d'éléments reliés entre eux compris dans un ensemble plus grand. En latin et en grec, le mot « système » veut dire combiner, établir, rassembler. Un sous-système est un système faisant partie d'un autre système. Généralement, un système est constitué de composants (ou d'éléments) organisés ensemble dans le but de faciliter le flux d'informations, de matières ou d'énergie. ...

fr.wikipedia.org/wiki/Système

Système d'information ou (information system) est un ensemble de composants de traitement de l'information et de communication, ainsi que l'environnement dans lequel il opèrent. (en Anglais: MIS ou Management of Information Systems).

Ensemble

structuré:

(1) *de données*, de leurs traitements et de leurs communications, décrit à l'aide de structures, de procédures et de protocoles → dictionnaire de données (en Anglais: DD ou Data Dictionary)

(2) *de moyens techniques*: Le système informatique (matériel et logiciel) et de communication de documents ayant pour but de générer, mémoriser, traiter, transférer et exploiter des informations dans le cadre d'objectifs définis.

Modèle

Le mot modèle synthétise les deux sens symétriques et opposés de la notion de ressemblance, d', de représentation. En effet, il est utilisé * pour un objet dont on cherche à donner une

représentation, qu'on cherche à imiter (exemple : le « modèle » du peintre, le « modèle » que constitue le maître pour le disciple). * pour un concept ou objet qui est la représentation d'une autre (le « modèle réduit », le « modèle » du scientifique).

fr.wikipedia.org/wiki/Modèle

Un modèle est une représentation de la réalité.

Abstraction

L'abstraction consiste à choisir, parmi l'ensemble des propriétés de plusieurs objets du monde réel ou imaginables, un certain nombre d'entre elles pour caractériser un objet-type, ou objet idéal, qui est ensuite plus commode à manier qu'une énumération d'objets réels, surtout si elle est infinie. Ainsi les nombres pairs ou les nombres premiers ont un caractère d'abstraction. Mais à vrai dire, les nombres eux-mêmes ont un caractère d'abstraction. ...

fr.wikipedia.org/wiki/Abstraction

concrétisation : phénomène inverse de l'abstraction: passe du modèle à monde réel:

concept : idée d'un objet conçu par l'esprit, permettant d'organiser les perceptions et les connaissances.

www.unice.fr/BU/lettres/parme/glossaire.html

Un concept est une construction de l'esprit permettant de mieux saisir intellectuellement le réel. Élément de base d'une théorie, le concept se veut une représentation abstraite d'un objet. Personne, par exemple, n'a vu une classe sociale, car elle est un concept.

www.collegeahuntsic.qc.ca/Pagesdept/Sc_Sociales/multi/Glossaire.html

Notion d'information

En informatique, la notion de données est très utilisée. Par exemple un programme a généralement des données sur lesquelles il travaille. On peut définir une donnée comme une informations numériques ou alphanumériques, représentées sous forme codée, compréhensibles par la seule machine, pouvant être enregistrées, traitées, conservées et communiquées.

En réalité, on fait une distinction entre les données et l'information. La données est un fait brut non interprété tandis que l'information est porteuse de sens. On peut dire qu'une information est un ensemble d'informations interprétés dans un contexte particulier.

Une information est une ensemble trois éléments :

- une entité : l'être, l'objet ou le concept concerné
- un attribut : un élément de la description de l'entité
- une mesure : une valeur associée à l'attribut

une information apporte un renseignement au sujet d'une d'une entité. Elle nous permet de représenter une entité et de transformer cette représentation.

Rôle de l'information

Tout acte de la vie d'une organisation s'accompagne ou est conditionné par des informations pour :

- améliorer son fonctionnement
- faciliter la prise de décision

Systèmes d'information

Une entreprise crée de la valeur en traitant de l'information, en particulier dans le cas des sociétés de service. Ainsi, l'information possède une valeur d'autant plus grande qu'elle contribue à l'atteinte des objectifs de l'organisation.

Un **système d'Information** (noté *SI*) représente l'ensemble des éléments participant à la gestion, au traitement, au transport et à la diffusion de l'information au sein de l'organisation.

Une organisation (entreprise par exemple) peut être vue comme un système qui transforme les entrées en sorties. Par exemple une usine de jus de mangue transforme les entrées qui sont des mangues en sorties qui sont des bouteilles de jus de mangue.

Ce système peut être divisé en deux sous-systèmes :

- le système opérant constitué de la partie du système qui s'occupe effectivement de transformer les mangues (les machines, les ouvriers, les techniciens, ...)
- le système de pilotage qui définit les objectifs de l'entreprise et s'efforce de tout mettre en oeuvre pour qu'ils soient atteints. Pour cela, il prend des décisions. Ces décisions sont prises à partir de paramètres venant du système opérant.

C'est le système d'information qui relève ces paramètres, les traite et les transmet au système de pilotage. Il peut être vu comme la partie qui relie les deux systèmes précédents.

Très concrètement le périmètre du terme Système d'Information peut être très différent d'une organisation à une autre et peut recouvrir selon les cas tout ou partie des éléments suivants :

- base de données de l'entreprise,
- progiciel de gestion intégré (ERP),
- Outil de gestion de la relation client (Customer Relationship Management),
- Outil de gestion de la chaîne logistique (SCM - Supply Chain Management),
- Applications métiers,
- Infrastructure réseau,
- Serveurs de données et systèmes de stockage,
- Serveurs d'application,
- Dispositifs de sécurité.

Pour une description plus imagée du système d'information, suivre le lien suivant : <http://fplanque.net/Blog/itTrends/2003/10/15/p403>

Fonctions des systèmes d'information

les systèmes d'information ont trois grandes fonctions :

- la mémorisation (des informations brutes ou résultats de traitement)

- circulation : accès à la mémoire et échange entre les acteurs
- traitement : rapprochement, calcul, comparaison d'informations

A ces trois fonctions on peut en ajouter une quatrième :

- collecte et saisie des informations

Classification des systèmes d'information On doit cette classification à Blumenthal. On distingue les grandes catégories suivantes :

- SICOP : Système d'Information de Contrôle Opérationnel Physique
 - SICOL : Système d'Information de Contrôle Opérationnel Logistique
 - SICOMP : Système d'Information de Contrôle Opérationnel de Matières Premières
 - SICOPR : Système d'Information de Contrôle Opérationnel de Production
 - SICOPC : Système d'Information de Contrôle Opérationnel de Produits Commercialisables
 - SICOAP : Système d'Information de Contrôle Opérationnel des Actifs Physiques
 - SICOIM : Système d'Information de Contrôle Opérationnel des Installations et Matériels
 - SICOPI : Système d'Information de Contrôle Opérationnel de Projet d'Investissement
- SICOA: Système d'Information de Contrôle Opérationnel et Administratif
 - SICOF : Système d'Information de Contrôle Opérationnel Financier
 - SICOC : Système d'Information de Contrôle Opérationnel de Comptabilité
 - SICOT : Système d'Information de Contrôle Opérationnel de Trésorerie
 - SICOM : Système d'Information de Contrôle Opérationnel Monétaire (notamment pour la paie, la gestion des avantages et indemnités, et l'administration du personnel)

Cycle de vie

Le « **cycle de vie d'un logiciel** » (en anglais *software lifecycle*), 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.

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

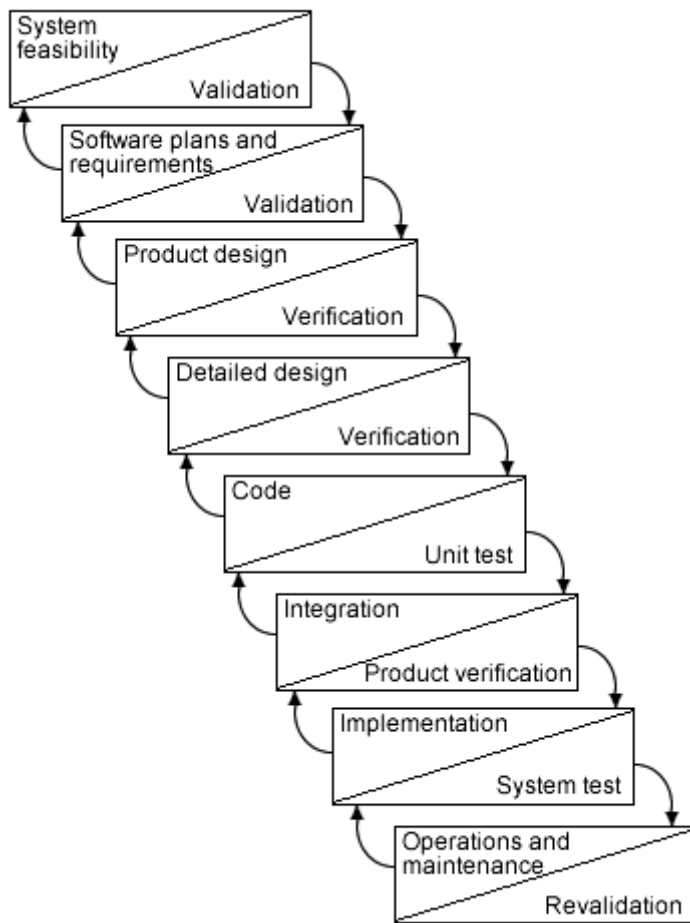
- **Définition des objectifs**, consistant à définir la finalité du projet et son inscription dans une stratégie globale.
- **Analyse des besoins et faisabilité**, c'est-à-dire l'expression, le recueil et la formalisation des besoins du demandeur (le client) et de l'ensemble des contraintes.
- **Conception générale**. Il s'agit de l'élaboration des spécifications de l'architecture générale du logiciel.
- **Conception détaillée**, consistant à définir précisément chaque sous-ensemble du logiciel.
- **Codage** (Implémentation ou programmation), soit la traduction dans un langage de programmation des fonctionnalités définies lors de phases de conception.
- **Tests unitaires**, permettant de vérifier individuellement que chaque sous-ensemble du logiciel est implémenté conformément aux spécifications.
- **Intégration**, dont l'objectif est de s'assurer de l'interfaçage des différents éléments (modules) du logiciel. Elle fait l'objet de *tests d'intégration* consignés dans un document.
- **Qualification** (ou *recette*), c'est-à-dire la vérification de la conformité du logiciel aux spécifications initiales.
- **Documentation**, visant à produire les informations nécessaires pour l'utilisation du logiciel et pour des développements ultérieurs.
- **Mise en production**,
- **Maintenance**, comprenant toutes les actions correctives (maintenance corrective) et évolutives (maintenance évolutive) sur le logiciel.

La séquence et la présence de chacune de ces activités dans le cycle de vie dépend du choix d'un modèle de cycle de vie entre le client et l'équipe de développement.

Cycle de vie en cascade

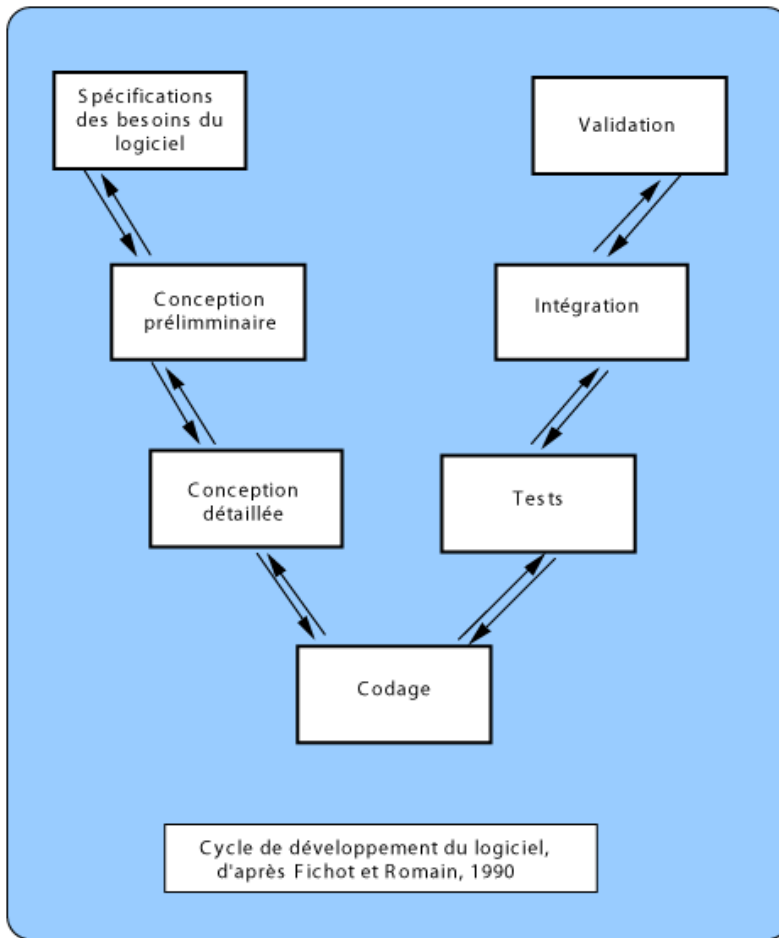
Dans le cycle de vie en cascade, les étapes se succèdent dans le temps. Chaque étape produit des documents qui sont utilisés pour en vérifier la conformité avant de passer à la suivante. Le cycle de vie en cascade est souvent schématisé de la manière suivante :

En utilisant le découpage défini ci-haut, on aurait ainsi le cycle de vie suivant:



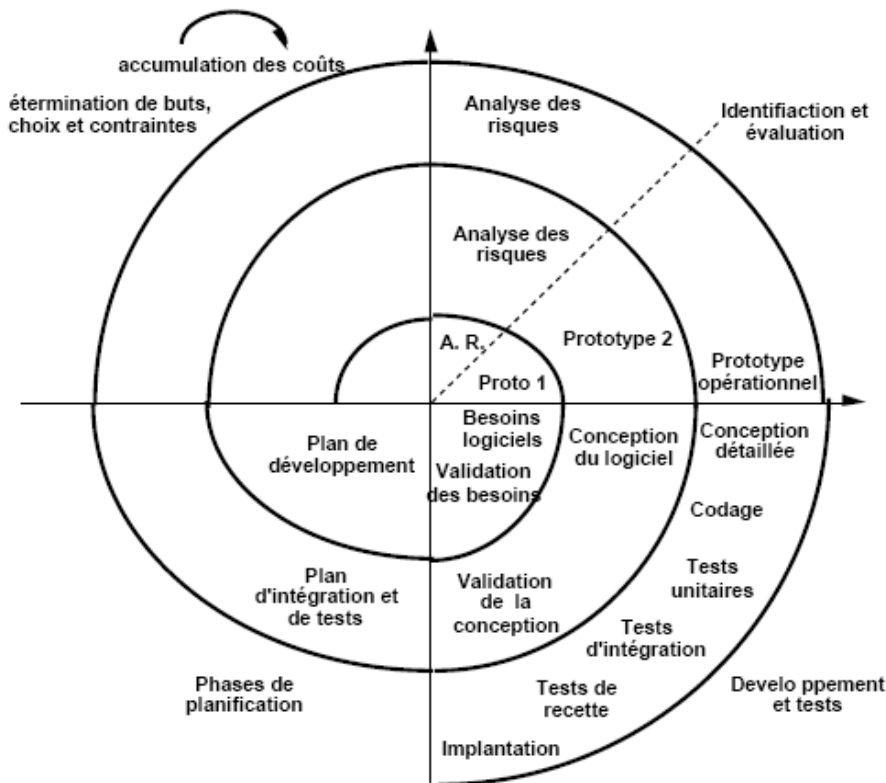
© Barry W. Boehm, A Spiral Model of Software Development and Enhancement, IEE Computer, May 1988.

Cycle de vie en V Le modèle de cycle de vie en V part du principe que les procédures de vérification de la conformité du logiciel aux spécifications doivent être élaborées dès les phases de conception.



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.



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

Elle donne lieu au choix d'une alternative, validée par un prototype le cas échéant, et à l'exécution de l'alternative choisie. A l'issue de la boucle, une revue des produits et des résultats fournit une évaluation qui sert d'entrée pour la boucle suivante.

La dernière boucle est séquencée comme un cycle de vie en cascade.

Méthodes d'informatisation Une méthode d'informatisation est composée :

- **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

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

Les avantages

L'emploi de méthodes doit permettre:

- de réduire la complexité du processus d'informatisation

- d'offrir des moyens de pilotage dans la construction du système
- d'informatisation
- la capitalisation des expériences et des solutions qui ont déjà fait leur preuve dans des situations posant problème
- la coordination des projets (réduire les coûts et d'augmenter la productivité dans l'accomplissement des tâches)
- la qualité de réalisation des étapes (garantie de pérennité) → tâches de d'évolution plus aisées
- une diminution du nombre des anomalies : les corrections et la localisation se font plus facilement ce qui implique que les tâches de maintenance sont moins onéreuses
- une discipline commune au sein des équipes (définitions des rôles et des responsabilités).
- un langage commun entre les intervenants (qualité des documentations, intégration rapide d'une nouvelle personne formée à la méthode)

Typologie des méthodes d'informatisation

On utilise souvent deux critères de classification de méthodes d'informatisation : le type d'approche et les domaines d'application:

1. Type d'approche des problèmes d'informatisation:

- les méthodes classiques
- les méthodes cartésiennes (Exemple : SADT)
- les méthodes systémiques (Exemple : MERISE)
- les méthodes à objets (Exemple : UML)

2. Les domaines d'application :

On a :

- les méthodes d'élaboration de schémas directeurs (Exemple : RACINES)
- les méthodes de rédaction de cahiers des charges (Exemples : AROC, MUSCADE)
- les méthodes de conception des architectures techniques (Exemple : TACT)
- les méthodes de conception des systèmes d'information
- les méthodes de programmation
- les méthodes de sécurité des systèmes (Exemple : MARION)
- les méthodes de conduite des projets (Exemples : GANTT, PERT, MCP)

Classification des types Lorsqu'on évoque les méthodes d'informatisation, il existe plusieurs manières de le faire. On distingue :

- les méthodes classiques ;
- les méthodes cartésiennes ;
- les méthodes systémiques ;
- les méthodes orientée-objet.

Nous vous présentons dans les sections qui suivent la classification des types d'informatisation.

Méthodes classiques • Spécification du système complet

- Caractérisation de la totalité des données et traitements

Plusieurs vues :

- Entité/association → MERISE
- Fonctions → SADT ou SA/RT

INCONVENIENTS:

- Systèmes figés (évolution de l'environnement)
- Perte de la sémantique du système (buts, comportements...)

Méthodes cartésiennes

Discours de la Méthode (1637) de René Descartes

sous-titré "pour bien conduire sa raison, et chercher la vérité dans les sciences"

Quatre règles méthodologiques:

(1) Le principe du doute: tout en se défaisant des opinions toutes faites, ne rien croire sans preuve dûment perçue par soi-même, d'où éviter la précipitation

(2) La règle de division: pour mieux résoudre un problème, il faut le décomposer

(3) La gradation des difficultés: aller du plus simple au plus complexe

(4) L'exhaustivité pour bien connaître un sujet

Anglo-saxonnes → approche fonctionnelle:

- décomposition hiérarchique des processus et des flux de données (ou flux d'information)
- analyse et conception du système d'information à partir de la définition de fonctions
- processus d'analyse et de conception

Un traitement de l'information répond aux règles de procédures de gestion pour produire des sorties:

La démarche de travail est descendante, "top-down": part du général pour aller vers le particulier.

Elle débute par l'identification d'une fonction globale de gestion. Toute fonction est décomposée en sous-fonctions, et ainsi de suite, par raffinements successifs, jusqu'à ce que les ensembles élémentaires soient intelligibles

Une représentation en arbre: les fonctions globalement perçues sont éclatées en processus spécifiques. Les relations entre ces derniers sont également recherchées.

Programmation modulaire et décomposition fonctionnelle: **S.A.D.T., J.S.D.**

Méthodes systémiques

Théorie du système général. Il existe 9 niveaux de complexité

A chaque niveau, le système comporte tous les caractères du niveau inférieur

- l'objet passif et sans nécessité: niveau le plus simple de description
- l'objet actif: connu par son activité, son comportement.
- l'objet actif régulé: le système peut refuser certains comportements possibles en fonction de son comportement précédent
- l'objet s'informe: régulation du système par l'intermédiaire de flux d'information
- l'objet décide son activité: le système a un comportement non inévitable,

c'est-à-dire qui est le résultat d'une décision: 2 types d'information:
l'information-représentation et l'information-décision

- l'objet actif a une mémoire: le système peut avoir besoin de la représentation d'événements passés
- l'objet actif se coordonne: il existe au sein du système des processus de coordination
- l'objet actif s'auto-organise: il est capable de s'adapter et d'inventer de nouvelles organisations
- l'objet actif s'auto-finalise: le système génère ses projets, détermine ses finalités

Méthodes orientée-objet

- Spécification comme un système ouvert
- Interface avec l'existant (communication avec l'homme)
- Caractérisation du comportement ou du but du système
- Décomposition en sous-systèmes ou objets

Plusieurs vues :

Comportement → UML

Buts → KADS

Les modèles de MERISE

A chaque niveau de la conception du système d'information, un modèle est élaboré. Il permet de décrire le système au niveau de conception considéré.

Sachant que la communication, les données et les traitements sont examinés, on a les modèles suivants :

Au niveau conceptuel :

Le Modèle Conceptuel de la Communication

le Modèle Conceptuel des Données (MCD)

le Modèle Conceptuel des Traitements (MCT)

Au niveau Organisationnel :

le Modèle Logique des Données (MLD)

le Modèle Logique des Traitements (MLT) ou Modèle Organisationnel des Traitements (MOpT)

Au niveau physique :

le modèle Physique des Données (MPD ou MPhD)

le Modèle Opérationnel des Traitements (MOT ou MOpT)

Les modèles de la communication Le modèle conceptuel de la communication est généralement le seul modèle utilisé dans ce cadre.

Les modèles de la communication Le modèle conceptuel de la communication est généralement le seul modèle utilisé dans ce cadre.

Le Modèle Conceptuel de la Communication (MCC)

Le MCC (Modèle conceptuel de la communication) définit les flux d'informations à prendre en compte.

La première étape de ce modèle est d'arriver à isoler le système en le délimitant.

Il s'agit donc de définir le système et les éléments externes avec lesquels il échange des flux d'information. Ces éléments extérieurs sont appelés *acteurs externes* (ou partenaires).

La seconde étape consiste à découper l'organisation en entités appelées *acteurs internes* (ou domaines).

Lorsque les domaines d'une organisation sont trop importants, ils peuvent être décomposés eux-mêmes en *sous-domaines*.

La dernière étape est l'analyse des flux d'information, c'est-à-dire la définition des *processus*.

Les modèles de données

C'est la représentation schématique d'un ensemble de données relatives aux phénomènes qui présentent de l'intérêt pour une application.

On retrouve trois niveaux de modèles de données

- le modèle conceptuel de données,
- le modèle logique de données, appelé aussi modèle fonctionnel de données,
- le modèle physique de données

Le modèle logique des données (MLD)

le MLD reprend le contenu du MCD précédent, mais précise la volumétrie, la structure et l'organisation des données telle qu'elles pourront être implémentées. Il s'agit du passage entre le Modèle Conceptuel de Donnée et l'implémentation physique de la base. Par exemple, à ce stade, il est possible de connaître la liste exhaustive des tables qui seront à créer dans une base de données relationnelle.

Le MLD est lui aussi indépendant du matériel et du logiciel, il ne fait que prendre en compte l'organisation des données. C'est d'ailleurs le point primordial de la modélisation : si l'organisation des données est relationnelle (si elles sont "liées" entre elles), alors le MLD est Relationnel et devient le MLDR, ou Modèle Logique de Donnée Relationnel.

Le MLD est une transcription (également appelée dérivation) du MCD dans un formalisme adapté à une implémentation ultérieure, au niveau physique, sous forme de base de données relationnelle ou réseau, ou autres (ex: simples fichiers).

La transcription d'un MCD en modèle relationnel s'effectue selon quelques règles simples qui consistent d'abord à transformer toute entité en table, avec l'identifiant comme clé primaire, puis à observer les valeurs prises par les cardinalités maximum de chaque association pour représenter celle-ci soit (**ex:** card. max 1-n ou 0-n) par l'ajout d'une clé étrangère dans une table existante, soit (**ex:** card. max n-n) par la création d'une nouvelle table dont la clé primaire est obtenue par concaténation de clés étrangères correspondant aux entités liées, exemple :

- MCD
- MLD / Modèle relationnel

PAYS(code_pays)

USINE(id_usine,@code_pays,date_implantation)

EXPORT(@id_usine,@code_pays)

Les opérateurs de l'algèbre relationnelle (projection, sélection, jointure, opérateurs ensemblistes) Langage d'interrogation de données peuvent ensuite directement s'appliquer sur le modèle relationnel ainsi obtenu et normalisé Formes normales.

Cette démarche algorithmique ne fournit pas à ce niveau d'élément sur l'optimisation de la durée ou des ressources nécessaires pour exécuter les traitements dans l'environnement de production cible.

La transcription du MCD en MLD doit également être précédée d'une étape de synchronisation et de validation des modèles de données (MCD) et de traitement (MCT et MLT), au moyen de vues . Cela afin d'y introduire les informations d'organisation définies au MLT, d'éliminer les propriétés conceptuelles non utilisées dans les traitements ou redondantes et enfin de vérifier que les données utilisées pour un traitement sont bien atteignables par 'navigation' entre les entités/rerelations du MCD.

- MCD

La modèle Physique des données (MPD ou MPhD) permet de préciser les systèmes de stockage employés. Les données qui sont stockées et gérées dans un ordinateur le sont souvent par un système de gestion de base de données (SGBD). Le MPD est l'implémentation du MLD dans le SGBD retenu.

Une fois le système d'information analysé et modélisé en Modèle Conceptuel de Donnée (MCD), et après être passé par le Modèle Logique de Donnée Relationnel (MLDR), nous arrivons au Modèle Physique de Donnée (MPD). Il s'agit maintenant de créer la base correspondante à l'étude entamée. C'est à ce stade seulement que le système de gestion de base de données choisie intervient.

Le SQL (Structured Query Language), ou Langage d'Interrogation Structuré, a été reconnu en tant que norme officielle de langage de requête relationnelle.

Toutefois, les syntaxes d'extractions des données et de créations des tables varient quelques peu d'un système de gestion de base de données à l'autre.

Le modèle physique consiste donc à ressortir le script SQL de création des tables en précisant la longueur des champs et les différentes clés.

Ensuite il faudra évaluer le poids global de la base de données et faire une projection sur un certains nombre d'années. Ce qui permet de tabler sur la capacité du disque requise pour l'exploitation de la base de données pendant cette période.

Des exemples de logiciels supportant Merise sont :

- • AMC Designer
- • TRAMIS

- • SELECT

Les modèles des traitements

Si les modèles de données se penchent sur l'aspect statique, les modèles des traitements s'occupent de l'aspect dynamique. Ils portent sur les "manipulations" que subissent les données. On en distingue également trois :

- le modèle conceptuel des traitements
- le modèle organisationnel des traitements
- le modèle opérationnel des traitements

Le modèle conceptuel des traitements (MCT)

le Modèle Conceptuel de Traitement est un schéma représentant les traitements, en réponse aux événements à traiter (par exemple : la prise en compte de la commande d'un client).

Le MCT repose sur les notions d'événement et d'opération, celle de processus en découle.

- **L'événement**

Un événement est assimilable à un message porteur d'informations donc potentiellement de données mémorisables (par exemple : l'événement 'commande client à prendre en compte' contient au minimum l'identification du client, les références et les quantités de chacun des produits commandés).

Un événement peut

- déclencher une opération (**ex**: 'commande client à prendre en compte' déclenche l'opération 'prise en compte commande'),
- être le résultat d'une opération (**ex**: 'colis à expédier' suite à l'opération de 'préparation colis'), et à ce titre être, éventuellement, un évènement déclencheur d'une autre opération.

- **L'opération**

Une opération se déclenche uniquement par le stimulus d'un ou de plusieurs évènements synchronisés

Elle est constituée d'un ensemble d'actions correspondant à des règles de gestion de niveau conceptuel, stables pour la durée de vie de la future application (ex: pour la prise en compte d'une commande : vérifier le code client (présence, validité), vérifier la disponibilité des articles commandés, ...).

Le déroulement d'une opération est ininterrompible : les actions à réaliser en cas d'exceptions, les évènements résultats correspondants doivent être formellement décrits (**ex** : en reprenant l'exemple précédent, si le code client indiqué sur la commande est incorrect prévoir sa recherche à partir du nom ou de l'adresse indiqués sur la commande, s'il s'agit d'un nouveau client prévoir sa création et les informations à mémoriser, ...).

- **Le processus**

Un processus est une vue du MCT correspondant à un enchaînement pertinent d'opérations du point de vue de l'analyse (**ex** : l'ensemble des événements et opérations qui se déroulent entre la prise en compte d'une nouvelle commande et la livraison des articles au client).

Le modèle organisationnel des traitements (MOT) encore appelé MLT (Modèle Logique des Traitements), il décrit avec précision l'organisation à mettre en place pour réaliser une, ou le cas échéant plusieurs, opérations figurant dans le MCT : c'est à dire qui fait quoi, où, quand, comment. A un MCT correspond donc généralement plusieurs MLT.

Les notions introduites à ce niveau sont le poste de travail, la phase, la tâche et la procédure.

- **Le poste de travail**

Le poste de travail décrit la localisation, les responsabilités, et les ressources nécessaires pour chaque profil d'utilisateurs du système (ex : client-web, responsable commercial, responsable des stocks, etc.).

- **La phase**

La phase est un ensemble d'actions (CF MCT/Opération) réalisées sur un même poste de travail.

La phase peut être soit manuelle (ex : confectionner des colis), soit automatisée et interactive (ex : saisie d'un formulaire client) ou automatisée batch (ex : production et envoi de tableaux de bord quotidiens dans les boîtes aux lettres électroniques).

- **La tâche**

La tâche est une description détaillée d'une phase automatisée interactive : spécification de l'interface et du dialogue homme-machine, localisation et nature des contrôles à effectuer, etc.

- **La procédure**

La procédure est un regroupement de phases, équivalent organisationnel des notions d'opération et d'actions conceptuelles, mais se déroulant sur une période de temps homogène.

Des procédures d'origines non conceptuelles peuvent être rajoutées du fait des choix d'organisation retenus (ex : procédures d'échanges d'informations liées à l'externalisation de certaines activités, prise en compte des questions de sécurité en cas de choix de solution Web, ...)

Le modèle opérationnel des traitements (MOT ou MOpT)

Le Modèle Opérationnel des Traitements permet de spécifier les fonctions telles qu'elles seront ensuite réalisées par le programmeur.

Les formalismes Les modèles merise utilisent des formalismes, une manière formelle et standard de décrire le modèle. Nous donnons dans les quelques sections qui suivent les principaux formalismes utilisés par merise.

Formalisme

Pour le MCC

Diagramme de contexte

Le diagramme de contexte a pour but de représenter les flux d'informations entre l'organisation et les acteurs externes selon une représentation standard dans laquelle chaque objet porte un nom :

- l'organisation est représentée par un rectangle
- les acteurs externes sont représentés par des ellipses en pointillés
- les flux d'information sont représentés par des flèches dont l'orientation désigne le sens du flux d'information

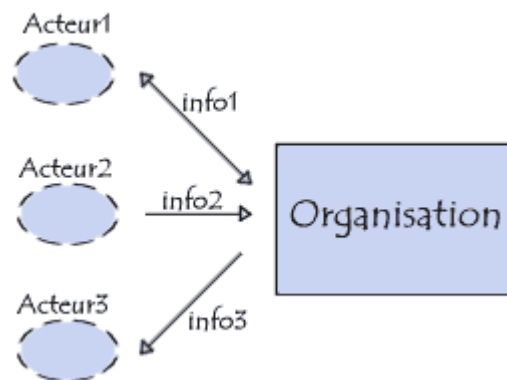
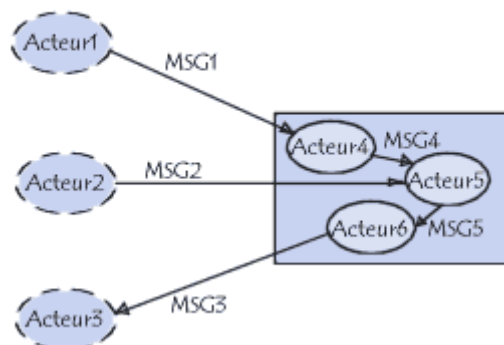


Diagramme conceptuel de flux

Ce diagramme (appelé aussi *modèle conceptuel de la communication*) permet de compléter le diagramme de contexte en décomposant l'organisation en une série d'acteurs internes. Dans ce diagramme la représentation standard est la suivante :

- Les acteurs internes sont représentés par des ellipses
- les messages internes sont représentés par des flèches



Pour le MCD

Entités et classe d'entité

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

On appelle **classe d'entité** un ensemble composé d'entités de même type, c'est-à-dire dont la définition est la même. Le classement des entités au sein d'une classe s'appelle *classification* (ou *abstraction*). Une entité est une *instanciation* de la classe. Chaque entité est composée de propriétés, données élémentaires permettant de la décrire.

Prenons par exemple une *Ford Fiesta*, une *Renault Laguna* et une *Peugeot 306*. Il s'agit de 3 entités faisant partie d'une classe d'entité que l'on pourrait appeler *voiture*. La *Ford Fiesta* est donc une instanciation de la classe *voiture*. Chaque entité peut posséder les propriétés *couleur*, *année* et *modèle*.

Les classes d'entités sont représentées par un rectangle. Ce rectangle est séparé en deux champs :

- le champ du haut contient le libellé. Ce libellé est généralement une abréviation pour une raison de simplification de l'écriture. Il s'agit par contre de vérifier qu'à chaque classe d'entité correspond un et un seul libellé, et réciproquement
- le champ du bas contient la liste des propriétés de la classe d'entité

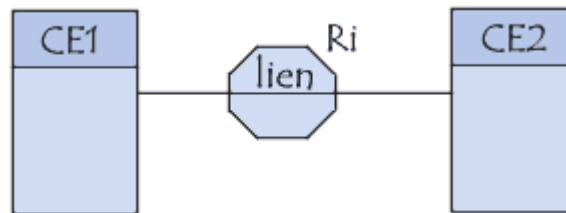


Relations et classes de relation

Une relation (appelée aussi parfois *association*) représente les liens sémantiques qui peuvent exister entre plusieurs entités. Une classe de relation contient donc toutes les relations de même type (qui relient donc des entités appartenant à des mêmes classes d'entité). Une classe de relation peut lier plus de deux classes d'entité. Voici les dénominations des classes de relation selon le nombre d'intervenants :

- une classe de relation **récursive** (ou *réflexive*) relie la même classe d'entité
- une classe de relation **binaire** relie deux classes d'entité
- une classe de relation **ternaire** relie trois classes d'entité
- une classe de relation **n-aire** relie n classes d'entité

Les classes de relations sont représentées par des hexagones (parfois des ellipses) dont l'intitulé décrit le type de relation qui relie les classes d'entité (généralement un verbe). On définit pour chaque classe de relation un identificateur de la forme R_i permettant de désigner de façon unique la classe de relation à laquelle il est associé.

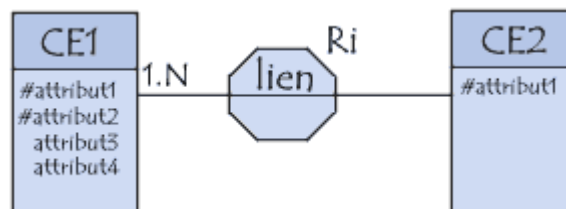


On peut éventuellement ajouter des propriétés aux classes de relation.

La cardinalité

Les cardinalités permettent de caractériser le lien qui existe entre une entité et la relation à laquelle elle est reliée. La cardinalité d'une relation est composée d'un couple comportant une borne maximale et une borne minimale, intervalle dans lequel la cardinalité d'une entité peut prendre sa valeur :

- la borne minimale (généralement 0 ou 1) décrit le nombre minimum de fois qu'une entité peut participer à une relation
- la borne maximale (généralement 1 ou n) décrit le nombre maximum de fois qu'une entité peut participer à une relation



Une cardinalité 1.N signifie que chaque entité appartenant à une classe d'entité participe au moins une fois à la relation.

Une cardinalité 0.N signifie que chaque entité appartenant à une classe d'entité ne participe pas forcément à la relation.

Les identifiants

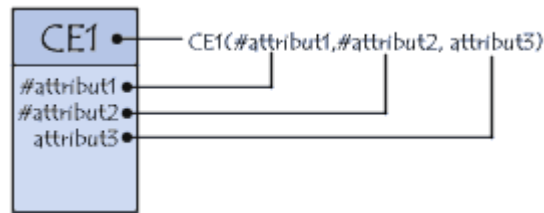
Un identifiant est un ensemble de propriétés (une ou plusieurs) permettant de désigner une et une seule entité. La définition originale est la suivante :

L'identifiant est une propriété particulière d'un objet telle qu'il n'existe pas deux occurrences de cet objet pour lesquelles cette propriété

pourrait prendre une même valeur.

Les attributs d'une classe d'entité permettant de désigner de façon unique chaque instance de cette entité sont appelés identifiants absolus.

Le modèle conceptuel des données propose de faire précéder d'un # les identifiants (parfois de les souligner).



Ainsi, chaque classe d'entité doit posséder au moins un attribut identifiant, et l'ensemble de ses attributs identifiants doivent être renseignés à la création de l'entité.

Agrégation (ou identification relative)

Lorsqu'un identifiant est constitué uniquement d'attributs intrinsèques à une entité, c'est-à-dire ne faisant référence à aucune autre entité, on le nomme **identifiant absolu**. Les entités comportant des identifiants absolus peuvent être définies indépendamment des autres occurrences d'entités, on dit que ces entités sont indépendantes.

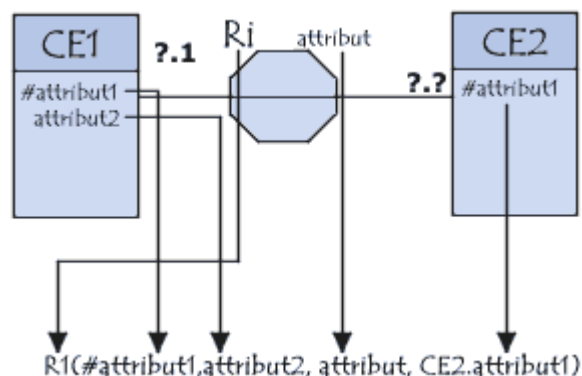
Certaines entités ne peuvent toutefois être identifiées que par l'intermédiaire d'autres entités, c'est la raison pour laquelle on parle d'**identification relative**.

On parlera par exemple de la 4^{ème} porte au 2^{ème} étage du bâtiment B au lieu de dire la porte n°3451...

Ainsi, l'**agrégation** (appelée aussi *identification relative*) permet de spécifier qu'une entité est nécessaire pour en identifier une autre.

- la classe d'entité permettant d'identifier est appelée *classe d'entité agrégeante*
- la classe d'entité identifiée est appelée *classe d'entité agrégée*

La représentation de ce type de relation est la suivante :



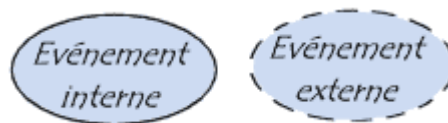
Pour le MCT

Le concept d'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

On représente un événement par une ellipse en trait plein pour les événements internes à l'organisation, en trait pointillé pour les événements externes.



Définition d'un processus

Un **processus** est un sous-ensemble de l'activité de l'entreprise, cela signifie que l'activité de l'entreprise est constituée d'un ensemble de processus. Un processus est lui-même composé de traitements regroupés en ensembles appelés *opérations*.

Opération

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

Cet ensemble d'actions est ininterrompible, c'est-à-dire que les événements ne sont pas pris en compte (ils ne sont pas forcément ignorés pour autant) tant que l'opération n'a pas été accomplie.

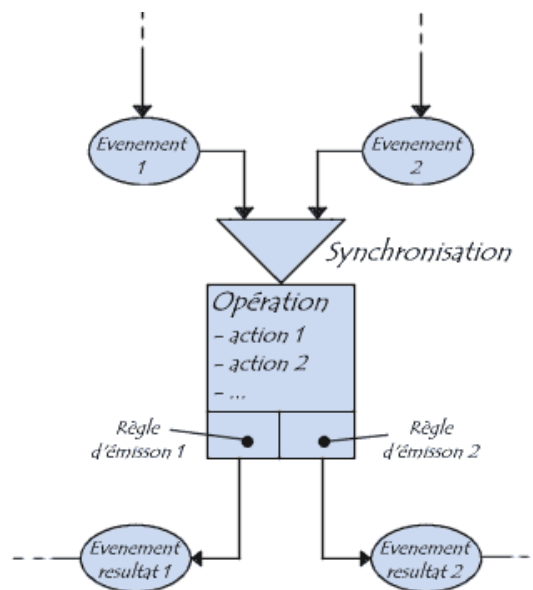
La synchronisation

La **synchronisation d'une opération** définit une condition booléenne sur les événements contributifs devant déclencher une opération. Il s'agit donc de conditions au niveau des événements régies par une condition logique réalisée grâce aux opérateurs :

- OU
- ET
- NON

Construction du MCT

Le modèle conceptuel des traitements permet de représenter schématiquement la gestion des événements :



Pour le MLD

Le modèle logique des données consiste à décrire la structure de données utilisée sans faire référence à un langage de programmation. Il s'agit donc de préciser le type de données utilisées lors des traitements.

Ainsi, le modèle logique est dépendant du type de base de données utilisé.

Le modèle relationnel

Traduction d'une classe d'entité

Chaque classe d'entité du modèle conceptuel devient une table dans le modèle logique. Les identifiants de la classe d'entité sont appelés *clés de la table*, tandis que les attributs standards deviennent des attributs de la table, c'est-à-dire des colonnes.

Traduction d'une classe de relation

Le passage du modèle conceptuel au modèle logique au niveau des classes de relation se fait selon les cardinalités des classes d'entité participant à la relation :

- Si une des classes d'entités possède une cardinalité faible : la table aura comme attributs, les attributs de la classe ayant une cardinalité faible, puis le (ou les) attribut(s) de relation et enfin les attributs de la seconde classe précédés du nom de la classe
- si les deux classes d'entités possèdent une cardinalité forte : la table aura comme attributs, les attributs des deux classes de relation précédés des noms des classes respectives, puis le (ou les) attribut(s) de relation

Traduction d'une classe d'agrégation

Dans le cas de la présence d'une classe d'agrégation, la classe d'entité agrégée a comme attributs supplémentaires les attributs de la classe d'entité agrégeante

Pour le MOT

Le modèle organisationnel des traitements

Le modèle organisationnel des traitements s'attache à décrire les propriétés des traitements non traitées par le modèle conceptuel des données, c'est-à-dire :

- le temps
- les ressources
- le lieu

Le modèle organisationnel des traitements consiste donc à représenter le modèle conceptuel des traitements dans un tableau dont les colonnes sont la durée, le lieu, les responsables et ressources nécessaires à une action.

Le tableau des procédures fonctionnelles

La première étape du modèle organisationnel des traitements consiste à découper les opérations en **procédures fonctionnelles**, une succession de traitements déclenchée par un événement.

Il s'agit d'associer dans un tableau:

- les procédures fonctionnelles
- l'heure de début et de fin (Période)
- la nature
- le lieu du poste de travail
- le responsable du poste de travail
- les ressources du poste de travail

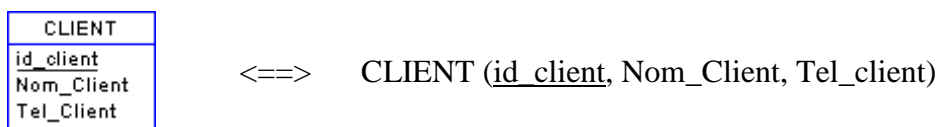
Période ou Temps		Enchaînement des Procédures fonctionnelles	Nature	Poste de Travail		
Début	Fin			Lieu	Responsable	Ressources

Règles de passage du MCD au MLDR

Ces règles sont à appliquer scrupuleusement selon le cas.

1 : Une entité se transforme en une relation (table)

Toute entité du MCD devient une relation du MLDR, et donc une table de la Base de Donnée. Chaque propriété de l'entité devient un attribut de cette relation, et dont une colonne de la table correspondante. L'identifiant de l'entité devient la **Clé Primaire** de la relation (elle est donc soulignée), et donc la **Clé Primaire** de la table correspondante.



2 : Relation binaire aux cardinalités (X,1) - (X,n), X=0 ou X=1

La **Clé Primaire** de la table à la cardinalité (X,n) devient une **Clé Etrangère** dans la table à la cardinalité (X,1) :

Exemple de Système d'Information (SI) :

Un employé a une et une seule société. Une société a 1 ou n employés.

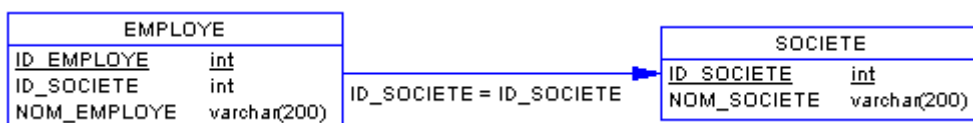
Modèle Conceptuel de Donnée (MCD) :



Modèle Logique de Donnée Relationnelle (MLDR) :

EMPLOYE (id_Employe, Nom_Employe, #id_Societe)
 SOCIETE (id_Societe, Nom_Societe)

Modèle Physique de Donnée (MPD), ou schéma de base :



3 : Relation binaire aux cardinalités (X,n) - (X,n), X=0 ou X=1

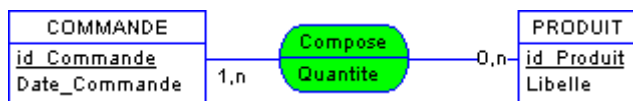
Il y a création d'une table supplémentaire ayant comme **Clé Primaire** une clé composée des **identifiants** des 2 entités. On dit que la **Clé Primaire** de la nouvelle table est la **concaténation** des **Clés Primaires** des deux autres tables.

Si la relation est porteuse de donnée, celles ci deviennent des attributs pour la nouvelle table.

S.I. :

Une commande est composée de 1 ou n produits distincts en certaine quantité. Un produit est présent dans 0 ou n commandes en certaine quantité.

MCD :



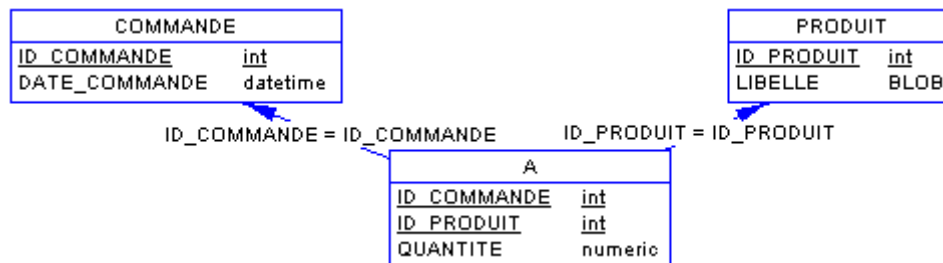
MLDR :

COMMANDE (id_Commande, Date_commande)

PRODUIT (id_Produit, libelle)

COMPOSE (id_Commande, id_Produit, quantité)

MPD :



4 : Relation n-aire (quelles que soient les cardinalités).

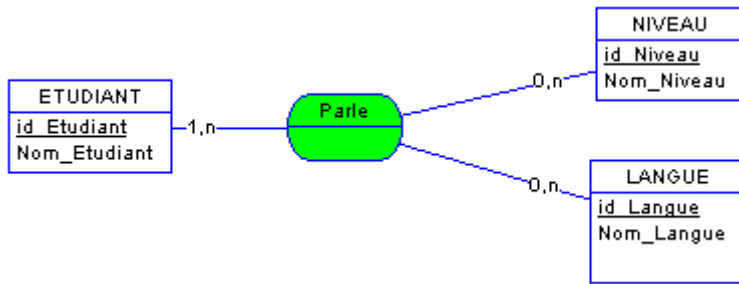
Il y a création d'une table supplémentaire ayant comme **Clé Primaire** la **concaténation** des **identifiants** des entités participant à la relation.

Si la relation est porteuse de donnée, celles ci deviennent des attributs pour la nouvelle table.

S.I. :

Un étudiant parle une ou plusieurs langues avec un niveau. Chaque langue est donc parlée par 0 ou n étudiants avec un niveau. Pour chaque niveau, il y a 0 ou plusieurs étudiants qui parlent une langue.

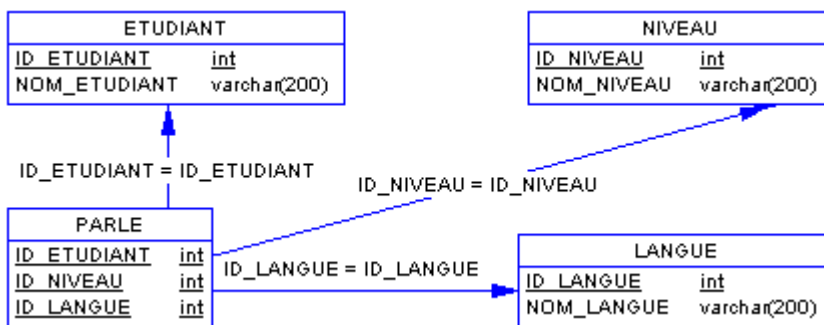
MCD :



MLDR :

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

MPD :



5 : Association Réflexive.

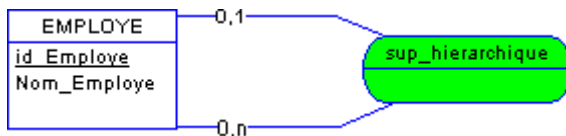
- Premier cas : cardinalité (X,1) - (X,n), avec X=0 ou X=1.

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

S.I. :

Prenons l'exemple d'une société organisée de manière pyramidale : chaque employé a 0 ou 1 supérieur hiérarchique direct. Simultanément, chaque employé est le supérieur hiérarchique direct de 0 ou plusieurs employés.

MCD :



MLDR :

EMPLOYE (id_Employe, Nom_Employe, #id_Sup_Hierarchique)

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

MPD :



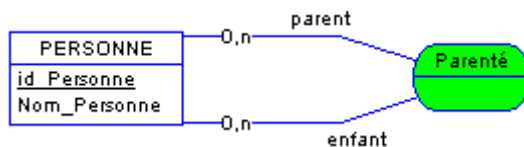
-
- Deuxième cas : cardinalité (X,n) - (X,n), avec X=0 ou X=1.

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

S.I. :

Prenons cette fois l'exemple d'une organisation de type familiale : chaque personne a 0 ou n descendants directs (enfants), et a aussi 0 ou n descendants directs (enfants).

MCD :



MLDR :

PERSONNE (id_Personne, Nom_Personne)

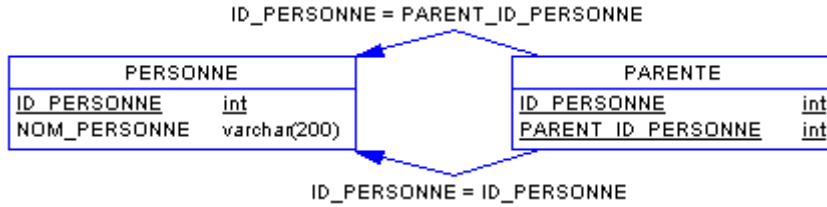
PARENTE (#id_Parent, #id_Enfant)

#id_Parent est l'identifiant (id_Personne) d'un ascendant direct de la personne. #id_Enfant est l'identifiant (id_Personne) d'un descendant direct de la personne.

La table PARENTE sera en fait l'ensemble des couples (parents-enfants) présent dans cette

famille.

MPD :



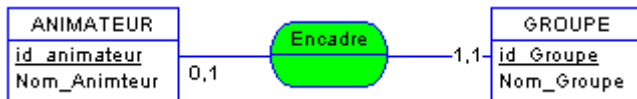
6 : Relation binaire aux cardinalités (0,1) - (1,1).

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

S.I. :

Dans ce centre de vacances, Chaque animateur encadre en solo 0 ou 1 groupe, chaque groupe étant encadré par un et un seul animateur.

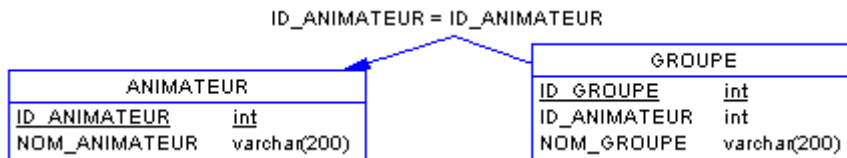
MCD :



MLDR :

ANIMATEUR (id_Animateur, Nom_Animateur)
 GROUPE (id_Groupe, Nom_Groupe, #id_animateur)

MPD :



Ces 6 règles représentent TOUS les cas que vous pourrez rencontrer.

La démarche Merise Description

Merise est une méthode de conception des système d'information qui utilisent, comme nous l'avons vu, un certain nombre de modèles. Mais mérise est également une démarche. La démarche est la méthode qui organise en étape la conception d'un système d'information à l'aide des modèles sur une période donnée. La démarche Merise passe par les étapes successives suivantes :

1. le schéma directeur
2. l'étude préalable
3. l'étude détaillée
4. l'étude technique
5. La production
6. la qualification
7. la maintenance

Le schéma directeur

« Un schéma directeur est une opération de courte durée visant l'anticipation à moyen terme, la prospective et la planification stratégique afin d'assurer la cohérence entre les finalités, les stratégies, les objectifs et sa politique de mobilisation des ressources organisationnelles, technologiques, humaines et financières. »

Appliqué à l'informatique, le schéma directeur d'une organisation a pour but de proposer un plan d'automatisation progressive des tâches de gestion d'une entreprise, d'une administration... c'est-à-dire des tâches bien définies dans un ensemble organisé.

Le concept de schéma directeur de l'informatique (SDI) est né dans la banque et la grande administration face au besoin de planifier des investissements de plus en plus lourds, de maîtriser les aspects humains liés à la modification des méthodes de travail.

Le schéma directeur de l'informatique permet d'envisager des scenarii qui portent sur les domaines.

Un domaine est un découpage du système d'information de l'entreprise, défini dans un schéma directeur en fonction de sa stratégie. Par exemple, une banque peut avoir défini les domaines Gestion de clientèle, Gestion des participations financières, Gestion de trésorerie, Gestion des Ressources Humaines, etc. Le découpage en domaines correspond grosso modo aux grandes fonctions de l'entreprise, hiérarchisées par ses objectifs stratégiques (centres d'intérêt).

Le schéma directeur peut se faire en plusieurs étapes.

Etape 1 : définir les objectifs et les structures de travail.

Etape2 : déterminer les composants de l'entreprise.

Etape 3 : formaliser et chiffrer les solutions.

Etape 4 : planifier les actions.

Etape 5 : définir la structure de pilotage.

Le schéma directeur donne lieu à un document appelé *plan de développement*

L'étude préalable

Une étude préalable s'applique à un domaine ou à une grande fonction de l'entreprise.

Également appelée: étude de faisabilité

L'objectif de cette étape est d'obtenir le descriptif complet de la nouvelle solution pour le domaine envisagé, mais en plusieurs phases, de sorte à envisager les différentes hypothèses possibles et à s'orienter progressivement vers la solution optimum.

Le point final de l'étude préalable est de:

* décider d'une solution type en parfaite connaissance de cause quant à sa faisabilité: coût, rentabilité, délai, budget, moyens à mettre en œuvre, impact organisationnel, ...

* établir le cahier des charges pour la réalisation; sélectionner un sous-traitant ou un progiciel.

Pour mener à bien une étude préalable, on part de la situation existante. L'étude préalable est faite par des utilisateurs et des organisateurs en liaison avec la direction générale et les informaticiens. On en distingue plusieurs étapes dont chacune est divisée en phases:

Étape 1 : Étude de l'existant

Phase 1 : Rédiger une **Fiche de présentation générale du problème**. Cette fiche situe le domaine dans le système d'information, à partir du schéma directeur et de la demande de la direction générale. Par interview de la direction générale, se faire préciser les grands objectifs relatifs au domaine et les consigner dans une **Fiche d'objectifs D.G.**

Phase 2 : Procéder à des interviews des postes de travail concernés et rédiger des **comptes rendus d'interviews**, en y joignant tous les documents concernés actuellement en vigueur. S'il y a des traitements qui se font à l'aide ou par ordinateur, joindre les **fiches de description de fichiers actuels**.

Phase 3 : A partir des comptes rendus, établir les documents suivants:

- le **graphe de circulation actuel**;
- la **carte de circulation des informations du domaine** ;
- le **graphe des flux** ;
- le **MOT actuel** ;

Phase 4 : à partir des documents recueillis lors des interviews, établir un **dictionnaire des données actuel** pour toutes les données manipulées et épurer ce dictionnaire.

Phase 5 : a partir du dictionnaire des données, construire la **structure d'accès théoriques (SAT)** en ne prenant que les données élémentaires, sans les données calculées, en listant toutes les dépendances fonctionnelles et en éliminant les transitivités. En déduire une **représentation graphique du MCD actuel**.

Phase 6 : Déduire les règles de gestion à partir du MCD et du MCT actuels ;

Phase 7 : à partir des comptes rendus d'interviews établir une fiche de synthèse faisant état des points les plus épineux, des souhaits des utilisateurs, en s'efforçant de critiquer l'organisation actuelle. Dégager de cette fiche un certain nombre d'objectifs de la "fiche d'objectifs D.G.", établir une liste hiérarchisée de tous les objectifs sur une fiche d'objectifs synthèse. Faire approuver celle-ci par la direction générale et par les divers responsables.

Phase 8 : étudier les interfaces avec les autres domaines (d'après la carte générale de circulation de l'information du schéma directeur, par exemple) et les consigner dans une fiche des interfaces.

Étape 2 : Scénarios futurs

Phase 1 : à partir de la fiche d'objectif synthèse et des règles de gestion actuelles, inventorier les règles de gestion futures et les inscrire sur une fiche de règles de gestion futures.

A partir du MCD actuel, de la fiche des interfaces et des règles de gestion futures, établir le MCD futur.

Phase 2 : à partir du MCT actuel et des règles de gestion futures, concevoir le MCT futur.

Phase 3 : pour chaque scénario faire l'étude organisationnelle qui donnera :

- une **fiche des règles d'organisation** ;
- une esquisse de **MOT futur**.

Faire ensuite l'étude opérationnelle (toujours par scénario) en définissant le matériel et les logiciels nécessaires. S'il existe plusieurs sites, cette étude tient compte des aspects spécifiques de chaque site. L'étude opérationnelle donnera lieu à :

- une fiche de configuration de matériel pour l'architecture générale et pour chaque site s'il y en a plusieurs ;
- une liste des logiciels.

Étape 3 : Une évaluation des Scénarios

Pour chaque scénario rédiger un rapport d'évaluation du scénario comportant :

- une évaluation des coûts de :
 - * matériel (achat, location)
 - * stockage
 - * logiciels (programmation, achat de progiciels...)
 - * traitement (unités centrales, consommation d'électricité, ...)
 - * communication (location de lignes de transmission)
 - * exploitation (fournitures de consommable, entretien des locaux ...)
 - * maintenance (contrats de maintenance chez le constructeur)
 - * personnel (utilisateurs, service informatique, sous-traitant, ...)
 - * formation (utilisateurs, informaticiens)
- une évaluation des avantages
 - * quantifiables et chiffrables au plan financier (recette attendues, économies de personnel ...)
 - * quantifiables et non chiffrables au plan financier (diminution des temps de gestion, simplification des procédures, ...)

- * non quantifiables (rapidité des traitements, aide à la décision, image de marque, fidélisation d'une clientèle, ...)
- une évaluation de l'impact sur l'organisation
 - * impact sur les postes de travail
 - * acceptabilité du personnel
- une évaluation de la faisabilité
 - * en matériels (rédiger un compte rendu de l'étude technique)
 - * en logiciels
 - * en personnel (personnel disponible, embauche, formation, sous-traitance)
- une évaluation approximative des délais
 - * de livraison des matériels
 - * de programmation des logiciels (délais rarement respectés)
 - * de recrutement et de formation
- une évaluation de la mise en œuvre
 - * cadencement du lancement des applications
 - * périodes transitoires
 - * etc.

L'étude préalable donne lieu à un document appelé **dossier de choix**.

L'étude détaillée

L'étude préalable ne porte que sur les processus majeurs. La description des données et des traitements y est succincte. L'étude détaillée va décrire tous les processus composants le fonctionnement du futur système; définir précisément les informations utilisées et mémorisées; spécifier complètement les tâches à effectuer. Elle se déroule en plusieurs étapes:

Étape 1 : Étude générale du MOT futur

cette étude se fait en suivant les phases suivantes :

- réalisation du **tableau des procédures fonctionnelles** ;
- construction du **diagramme d'enchaînement des procédures fonctionnelles** ;
- construction du **graphe de circulation**.

Étape 2 : Étude poussée de chaque procédure fonctionnelle

dans cette étape, chaque procédure fonctionnelle est étudiée de manière poussée. Cela donne lieu à :

- une fiche de description de la procédure fonctionnelle
- une description des documents éventuelle
- des tables de décision éventuelles
- une description éventuelle des états de sortie
- pour les transactions :
 - * des grilles d'écrans (à faire approuver par les utilisateurs concernés)
 - * des grilles de contrôles
 - * une fiche de répartition des tâches entre l'homme et la machine (à faire approuver par les utilisateurs)
- le modèle externe non validé.

Étape 3 : Validation du MCD

Les données et les traitements ayant été étudiés de manière indépendante, la validation du MCD permet de confronter les données aux traitements et de rendre l'ensemble cohérent.

Étape 4 : Passage au MLD

L'étude détaillée donne lieu, pour chaque application à un cahier des charges utilisateurs.

L'étude technique

L'étude technique est la traduction informatique des spécifications issues de l'étude détaillée. Elle s'occupe des spécifications techniques pour la réalisation.

Les objectifs de l'étude technique sont de présenter les spécifications informatiques nécessaires à la production du logiciel. A cet effet, elle définit complètement :

- la structure de mémorisation informatique des données (fichiers ou base de données) ;
- l'architecture interne du système
- les procédures de sécurités
- le plan de réalisation.

L'étude technique peut être menée en suivant les étapes suivantes :

Étape 1 : structuration des données

- établir la description physique des données à partir du MLD (Modèle Logique des données)
- définir les clés d'accès
- établir une quantification de l'activité des éléments de stockage des données
- procéder à l'optimisation physique
- déduire l'allocation des espaces physiques, l'organisation et le mode d'accès
- décrire les dispositifs de protection d'accès et de confidentialité

Étape 2 : spécification de l'architecture interne du système

découpage des Unité Fonctionnelle (UF) en Unités de Traitement (UT)

- découper chaque UF en UT

découpage des UT en modules

- découper chaque UT en modules
- établir la liste des Entrées/Sorties pour chaque module
- établir la liste des données utilisées ou échangées
- établir une description pour chaque module sous forme d'algorithmes ou d'organigramme
- déterminer les jeux d'essai à élaborer répondant aux exigences de la qualité

terminer de concevoir les Entrées/Sorties

- définir les spécifications des écrans ainsi que les dialogues associés
- définir les formats détaillés et les spécifications des sorties
- établir une liste des entrées/sorties pour chaque UT

Étape 3 : les procédures techniques de sécurité

contrôle des données :

- description des caractéristiques liées aux contrôles des données

sécurité et confidentialité :

- description des mécanismes relatifs à la sécurité et à la confidentialité des données
- description des procédures de reconstitution des supports de données, de redémarrage et de sauvegarde

Étape 4 : planification de la réalisation

- réaliser le planning pour les phases suivantes

L'étude technique donne lieu à un **cahier des charges technique**.

La production

La production consiste en l'écriture effective des programmes dans le langage approprié, à la génération des fichiers ou des bases de données et aux tests. La production du logiciel doit permettre de faciliter la maintenance future.

Ses objectifs sont :

- la production du logiciel conforme aux spécifications de l'étude technique
- de tester techniquement le logiciel de manière qu'il soit prêt à être livré aux utilisateurs

La production du logiciel peut se faire en suivant les étapes suivantes :

Étape 1 : programmation de l'application

durant laquelle il faudra :

- intégrer les normes et standards de programmation
- organiser l'environnement de programmation
- définir une organisation pour permettre la gestion de l'environnement de programmation (matériel, temps machine, ..)

coder les modules

- transformer les spécifications techniques des modules en code source
- vérifier et mettre au point le code source
- corriger les défauts et erreurs constatées

préparer les jeux d'essai

- définir les jeux d'essai
- préparer les données en vue des tests unitaires et d'enchaînement de modules

Étape 2 : tests, mise au point

tests unitaires

- chaque module écrit, correspondant à une Unité de Traitement, doit faire l'objet d'un premier test à partir du jeu d'essai

tests d'intégration

- ces tests doivent permettre de vérifier le bon fonctionnement de l'ensemble du logiciel. Ils sont généralement effectués par le responsable technique du projet.

La production donne lieu à un **dossier de réalisation**.

La mise en service

L'objectif principal de la mise en service est de rendre opérationnel le système d'information. Elle consiste à installer l'ensemble du logiciel développé et à mettre l'ensemble du SI au service des utilisateurs. Elle permet de conclure l'ensemble du cycle conception-développement.

Son objectif principal est de mettre en place tous les moyens nécessaires à la réception et au lancement du nouveau SI

La mise en service se déroule en deux étapes:

Étape 1 : l'installation

elle a pour but de livrer le logiciel aux utilisateurs et de préparer le démarrage de l'exploitation. Elle comprend :

- la commande, l'installation, le contrôle et la réception du matériel ;
- la mise en place des procédures nouvelles liées aux changements d'organisation ;
- l'organisation et le déroulement de la formation des utilisateurs et des exploitants ;
- l'exécution des jeux d'essai utilisateurs ;
- l'appréciation de la conformité des résultats ;
- la réception provisoire.

Étape 2 : la mise en exploitation

elle consiste en :

- l'exploitation en grandeur réelle du nouveau système
- le lancement progressif du nouveau système en parallèle au système existant si possible
- le fonctionnement en vraie grandeur pendant une période déterminée
- le bilan de l'exploitation
- la réception définitive du produit

Elle donne lieu au **dossier d'exploitation** et au **manuel utilisateur**.

La maintenance

La maintenance est définie selon BOEHM comme l'ensemble des opérations de modification d'un logiciel opérationnel laissant intactes ses fonctionnalités de base. Elle consiste en la rectification des anomalies, les améliorations et les évolutions.

Ses objectifs sont :

- maintenir le système en fonctionnement ;
- rectifier les anomalies de fonctionnement et prendre en compte les demandes d'évolution ;
- établir un scénario de développement des versions ;
- planifier les corrections apportées au système pour minimiser les coûts d'intervention ;
- s'assurer du bon fonctionnement des nouvelles révisions avant toute nouvelle mise en exploitation ;
- procéder à cette mise en exploitation après l'accord des parties prenantes ;
- assurer une bonne gestion des différentes configurations du logiciel par la tenue à jour de la documentation du système.

Il existe plusieurs niveaux, catégories et formes de maintenance.

Parmi les **niveaux et catégories**, on a : la maintenance corrective ; la maintenance adaptative ; la maintenance perfective et la maintenance préventive.

La maintenance corrective ne porte que sur le logiciel développé. Elle ne remet pas en cause les modèles de données ou de traitements. Elle consiste en la correction des erreurs du logiciel. Elle représente souvent 17 à 20 % du temps de maintenance.

La maintenance adaptative est liée à l'environnement du logiciel (contexte d'utilisation du logiciel, génération des ordinateurs, exploitation logiciel sur des matériels distincts ...). Elle peut faire évoluer le MCD et le MCT. Elle représente 18 à 25 % du temps de maintenance.

La maintenance perfective consiste à améliorer le fonctionnement du logiciel. Elle peut entraîner une remise en question des modèles physiques et du MOT. C'est la catégorie de maintenance la plus fréquente (environ 60 %).

La maintenance préventive a pour objectif de diminuer le nombre d'opérations de maintenance pour en diminuer le coût. Elle n'est vraiment possible que si certaines conditions sont réunies :

- le système est développé dans un esprit de maintenance ultérieure ;
- le système est continuellement amélioré pour faire face et intégrer les nouvelles technologies ;
- le système est maintenu en pensant à la maintenance ultérieure.

Webographie

fdigallo.online.fr/cours/merise.pdf

www.commentcamarche.net/.../affich-1081964-cours-et-exercices-merise

merise.developpez.com/

www.scribd.com/doc/7472422/Cours-de-Merise