La méthode MERISE

1 : Introduction

Bertrand LIAUDET

BIBLIOGRAPHIE

La méthode MERISE. Tome 1 : Principes et outils, Les éditions d'organisation, 1986. Tardieu, Rochfeld, Colletti.

La méthode MERISE. Tome 2 : Démarche et pratiques, Les éditions d'organisation, 1985. Tardieu, Rochfeld, Colletti, Panet, Vahée.

Ingéniérie des systèmes d'information : Merise - Deuxième génération, , Eyrolles, 2001, 4ème édition (ISIM). Nanci, Espinasse.

Ingéniérie des systèmes d'information, sous la direction de Corine Cauvet et Camille Rosenthal-Sabroux, Hermes, 2001.

SOMMAIRE

INTRODUCTION 4

1. Généralités sur la méthode 5

Définition 5
Objectifs centraux de toutes méthode 5
Méthode et méthodologie 6
Intérêt de la méthode 6
Comment pallier la non maîtrise des opérations complexes 7
Application à la pédagogie : bottum-up et top-down 8

2. Développement d'un logiciel : Les 5 distinctions capitales 9

Première distinction : Développement = Conception + Réalisation 9
Deuxième distinction : Conception = Analyse fonctionnelle + Analyse organique 10
Troisième distinction : Architecture des sous-systèmes + Analyse par sous-système 11
Quatrième distinction : Analyse générale + Analyse détaillée 12
Cinquième distinction : données versus traitements : l'analyse des données. 13

3. Le cycle en V 14

Logique du V 14
Relations cycle en V, MOA, MOE, fonctionnel, technique 15
La production des documents 16
Réalisation et langage de programmation 18
Cycle en V et analyse des données 18

4. Du cycle en V aux méthodes « agiles » : cycle de vie itératif 19

Principal défaut du cycle en V : le résultat apparaît à la fin 19
Avant le V : cycle de vie en tunnel et cycle de vie en cascade 19
Après le V : cycle de vie itératif 19
Méthodes agiles 20
Les deux écueils à éviter : l'effet tunnel et l'usine à gaz 20

5. Génie Logiciel *vs* Ingénierie des Systèmes d'Information 21

Génie logiciel - Software 21
Ingénierie des systèmes d'information - Brainware 22
Relations entre software engineering et brainware engineering 23

6. Méthode analytique vs méthode systémique 24

Méthode analytique 24
Brève présentation de la méthode systémique 26

7. Notion de Système d'Information – SI – Vers la méthode MERISE 27

Définition Wikipédia 27
Présentation 27
Schéma théorique d'un système d'information 28
Distinction entre SIO et SII 29
Distinction entre système entreprise et système logiciel 30
Notion d'acteur 30
Notion de poste de travail 30
Exemple : une bibliothèque parisienne 31

8. La méthode MERISE 32

Définition 32
Historique 32
Les 3 cycles de la démarche MERISE 33
La distinction entre données et traitements 33
Le cycle d'abstraction 33
Le cycle de vie 36
Le cycle de décision 37
Les plans types 38
MCD, MEA… clarification sur la modélisation et les différents types de modèles 40
Etapes détaillées de la construction d'un modèle des données selon la méthode MERISE 42

9. MERISE aujourd'hui 43

Type d'offre d'emploi 43

10. Questions de cours 44

Edition octobre 2018

INTRODUCTION

Il est facile de décrire la méthode MERISE, encore que son application exige à coup sûr savoir et pratique.

MERISE : c'est une méthode systémique de conception des systèmes d'information. Elle est en relation avec le développement des bases de données relationnelles (SQL).

Conception : c'est une partie du développement du logiciel.

Système d'information : c'est un ou plusieurs logiciels manipulant un ensemble d'informations structurées cohérentes. Par exemple : l'intra de l'INSIA : des cours, des élèves, des profs, des horaires, des projets, des groupes, des notes, etc. On peut les consulter, les créer, les modifier, les détruire.

Avant de présenter la méthode MERISE, on va présenter quelques notions générales sur la méthode, la conception et le système d'information.

1. Généralités sur la méthode

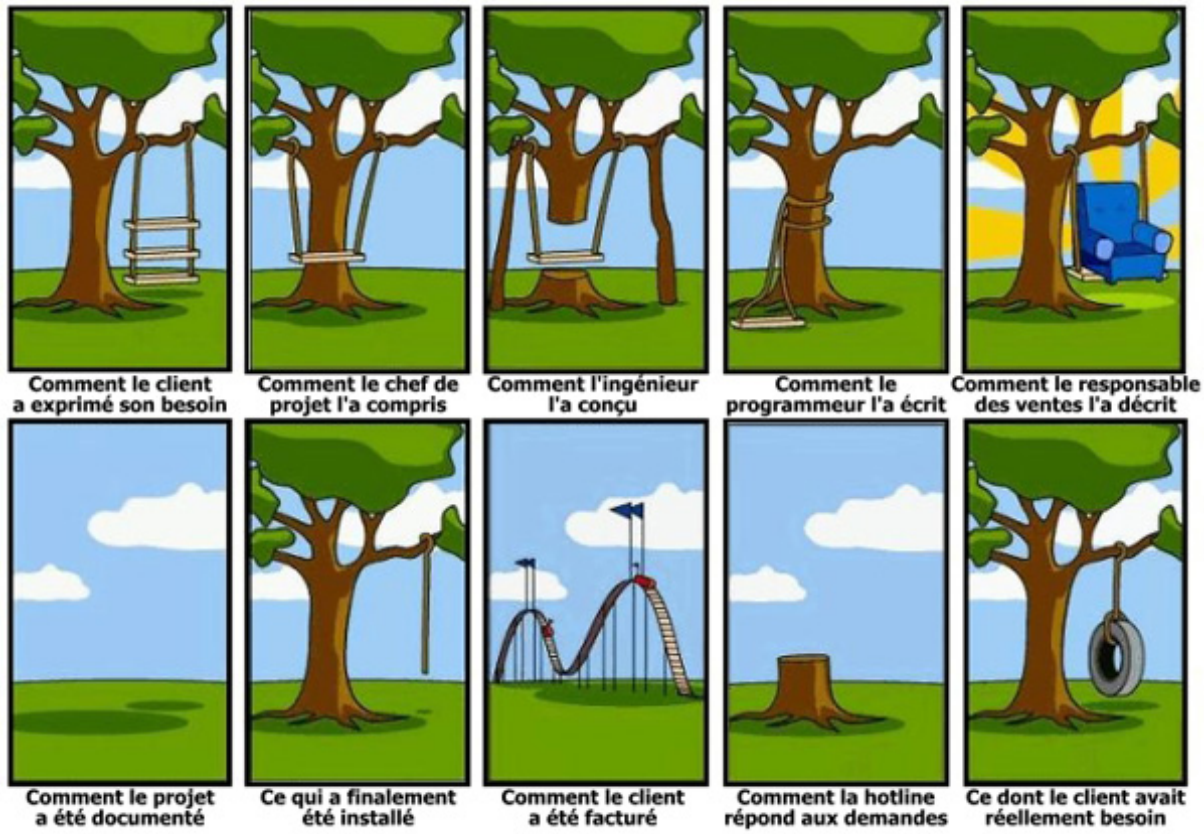
Définition

Une méthode c'est une procédure (c'est-à-dire la suite des opérations) mise en œuvre pour arriver à un but.

Objectifs centraux de toutes méthode

- **Répondre aux besoins** et même aider à l'exprimer.
- Répondre **au meilleur coût.**
- Répondre avec une architecture qui permet de **faciliter les mises à jour.**

Les risques entre le besoin et sa réalisation



Méthode et méthodologie

La méthodologie est la **science des méthodes**.

Par abus de langage, on parle souvent de méthodologie quand on devrait simplement parler de méthode, comme on parle de technologie quand on pourrait simplement parler de technique.

La méthodologie c'est un programme formel qui règle à l'avance une suite d'opérations à réaliser pour arriver à un résultat en signalant les difficultés à contourner.

C'est en général le résultat de principes théoriques et de retour d'expérience.

La méthodologie est une **connaissance très concrète et qui semble abstraite** car elle décrit la mise en œuvre concrète d'opérations complexes. Si la maîtrise concrète de ces opérations complexes n'est pas acquise, la méthodologie paraîtra d'autant plus abstraite.

> *C'est [la méthode] que l'on place le plus souvent en tête dans les écoles, comme propédeutique des sciences, alors que, selon le parcours de la raison humaine, elle est l'ultime étape, à laquelle la raison parvient uniquement quand la science est déjà terminée depuis longtemps et n'a plus besoin que de la dernière main pour être mise en ordre et atteindre la perfection. Car il faut que l'on connaisse les objets déjà à un assez haut degré, si l'on veut indiquer les règles selon lesquelles une science s'en peut mettre en œuvre.*
>
> **Critique de la raison pure, 1781, Emmanuel Kant (1724-1804)**
> **Introduction de la logique transcendantale**

Intérêt de la méthode

L'intérêt d'une méthode est que si on la suit correctement, on a plus de chance d'arriver au but, même si on ne maîtrise pas bien les opérations individuelles à mettre en œuvre.

La méthode permet aussi une séparation des tâches et des compétences.

Comment pallier la non maîtrise des opérations complexes

Pour pallier la non maîtrise des opérations décrites dans la méthode, il faut acquérir un savoir suffisant sur ces opérations.

Nom, image, science et pratique chez Platon

Le savoir chez Platon se décline en 4 parties : le nom, l'image, la science et la pratique

- ***Le nom***

C'est la simple connaissance du nom des opérations et des définitions. C'est une connaissance très limitée mais c'est toujours mieux que rien !

- ***L'image***

C'est une connaissance par représentation imagée. Il s'agit d'une connaissance par l'exemple. La compréhension d'un exemple est souvent une étape nécessaire pour arriver un comprendre quelque chose !

- ***La science***

C'est la connaissance théorique systématique. Elle passe par une maîtrise des concepts et des opérations en jeu.

- ***La pratique***

C'est la connaissance qu'on acquiert en pratiquant les opérations, même si on ne les comprend pas ! La pratique conduit forcément à une connaissance par nom minimum et à une connaissance par image.

Application en informatique : les interfaces

La connaissance des interfaces, par leur nom et leur mode d'emploi (leur définition) et par une représentation imagée de leur utilité permet de maîtriser des opérations complexes sans rentrer dans le détail.

La notion d'interface rejoint celle de fonction en tant que boîte noire. Une interface est comme une fonction dont on connaît les entrées et les sorties, et les résultats attendus en fonction des entrées fournies. On ne sait pas ce qui se passe à l'intérieur (principe de la boite noire). On sait juste à quoi ça sert et comment s'en servir.

Application à la pédagogie : bottum-up et top-down

Il existe fondamentalement deux types de pédagogie : bottum-up et top-down.

Pédagogie top-down

C'est la pédagogie universitaire classique : on part du nom, on monte à la science en présentant des images des différentes notions abordées. Pour finir, on met en pratique la science acquise.

C'est une pédagogie théorique

Pédagogie bottum-up

C'est une pédagogie par la pratique. On part d'abord de la pratique pour remonter ensuite à la science.

C'est une pédagogie pratique.

2. Développement d'un logiciel : Les 5 distinctions capitales

Il y a 5 distinctions capitales dans le développement d'un logiciel.

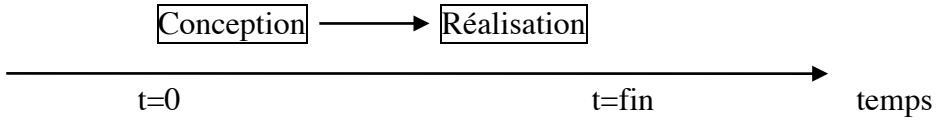
Première distinction : Développement = Conception + Réalisation

Le développement se compose de deux activités qu'on peut distinguer : la conception et la réalisation.

- **La conception** consiste à comprendre et prévoir ce qu'il a à faire.
- **La réalisation** consiste à faire concrètement ce qu'il y a à faire.

La distinction entre la conception et la réalisation est une façon d'organiser la division du travail.

Le premier principe de la méthode consiste à considérer ces deux activités comme deux étapes successives :



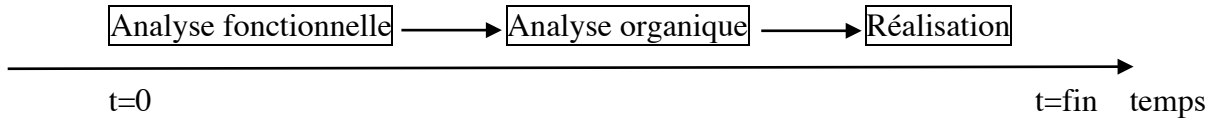
Le projet se déroule dans le temps : il commence avec la conception, il se termine avec la réalisation.

La division du travail consiste à mettre en évidence les étapes de la réalisation d'un logiciel.

Deuxième distinction : Conception = Analyse fonctionnelle + Analyse organique

La CONCEPTION se divise en deux parties : **le POURQUOI et le COMMENT** (le fonctionnel et le technique).

- **Le POURQUOI = l'analyse fonctionnelle** (ou analyse générale, ou spécifications fonctionnelles) d'abord. L'analyse fonctionnelle s'occupe des fonctionnalités (ou des services) que le système offre à ses utilisateurs. Autrement dit elle analyse **POURQUOI** faire le système et aussi **POUR QUI ?** Elle s'occupe aussi des relations que les fonctionnalités ont avec d'autres systèmes (analyse des interfaces). Elle répond alors à la question **AVEC QUI ?**
- **Le COMMENT = l'analyse organique** (ou technique ou architectonique[1]) ensuite. L'analyse organique s'occupe de la façon dont sera construit le système pour répondre aux attentes de l'analyse fonctionnelle.



ANALYSE FONCTIONNELLE	**ANALYSE ORGANIQUE**
EXTERNE	INTERNE
Le POURQUOI – POUR QUI – AVEC QUI	Le COMMENT
Point de vue de l'utilisateur et du client, le maître d'ouvrage (**MOA**) : celui qui commande le logiciel	**Point de vue** de l'informaticien et du maître d'œuvre (**MOE**) : celui qui réalise le logiciel
Build the right system	*Build the system right*

Avec cette distinction, on fait apparaître :

- le point de vue de l'utilisateur : le maître d'ouvrage (l'utilisateur, le client)
- le point de vue de l'informaticien : le maître d'œuvre.

Pour l'utilisateur, ce qui compte, c'est l'usage du système : les cas d'utilisation (vocabulaire UML). L'analyse fonctionnelle permettra de modéliser l'ensemble des cas d'utilisation.

Pour l'informaticien, ce qui compte c'est l'architecture interne du système.

L'analyse fonctionnelle garantit qu'on va bien faire ce qui est demandé : répondre aux exigences du client.

L'analyse organique garantit que ce qu'on va faire, on va bien le faire.

---

1 L'architectonique est un terme d'architecture. De façon générale, c'est la technique de la construction, mais aussi la structure ou l'organisation de la construction.

Troisième distinction : Architecture des sous-systèmes + Analyse par sous-système

Principe

Tout système peut se décomposer, se découper en sous-systèmes autonomes. On parle d'architecture des sous-systèmes ou plus simplement **d'architecture système**. Chaque sous-système autonome peut ensuite s'étudier de façon autonome.

Ce principe s'applique au niveau fonctionnel et au niveau organique.

Analyse fonctionnelle = Architecture des sous-systèmes fonctionnels + Analyse par sous-système fonctionnel

L'analyse fonctionnelle se divise en deux parties :

- **L'architecture des sous-systèmes fonctionnels :** elle s'occupe de l'organisation des sous-systèmes fonctionnels, autrement dit des **POSTES de TRAVAIL**. Par exemple, dans une bibliothèque, la borne automatique d'emprunt, la borne de consultation et de recherche, le poste du bibliothécaire, le site internet sont autant de sous-systèmes fonctionnels ou postes de travail permettant l'accès à des fonctionnalités de la bibliothèque.
- **L'analyse par sous-système fonctionnel :** on peut analyser les fonctionnalités au niveau de chaque poste de travail.

Analyse organique = Architecture des sous-systèmes logiciels et matériels + Analyse par sous-système logiciel et/ou matériel

L'analyse organique se divise en deux parties :

- **L'architecture des sous-systèmes logiciels et/ou matériels** : Elle analyse la division du logiciel en matériels et en programme distincts. Par exemple, la distinction client-serveur qui met au jour deux sous-systèmes logiciels et matériel. On peut aussi un microcontrôleur (Arduino par exemple) et un serveur.
- **L'analyse organique par sous-système :** L'analyse organique par sous-système s'occupe de la façon dont sera construit chaque sous-système pour répondre aux attentes de l'analyse fonctionnelle.

Quatrième distinction : Analyse générale + Analyse détaillée

Principe

Tout système peut s'analyser de façon générale par une **analyse des différentes structures (architectures)** qui vont le constituer et qui peuvent être partagées ou pas par les différents sous-systèmes. On parle d'analyse générale qui est donc transverse aux sous-systèmes.

L'analyse détaillée permet de préciser les contenus de façon détaillée.

Les analyses générale et détaillé peuvent s'appliquer au niveau du système complet au niveau des sous-systèmes ou au niveau des architectures.

Ces principes s'appliquent aux niveaux fonctionnel et organique.

Analyse générale et détaillé au niveau fonctionnel

- **L'analyse générale** consiste à analyser les cas d'utilisation et les utilisateurs en organisant des regroupements adaptés (les généralisations). C'est l'architecture des cas d'utilisation. Elle consiste aussi à analyser les processus métier (succession de UC correspondant à un usage métier pour le sous-système) en précisant les interfaces si nécessaires (les « avec qui/quoi »).
- **L'analyse détaillée** permet de préciser le contenu des cas d'utilisation. L'UML offre des outils pour cela : les inclusions, les diagrammes de séquence système, diagramme d'activités, diagrammes d'état-transition.

Analyse générale et détaillé au niveau organique

- **Analyse générale = les architectureS transversales :** elle s'occupe des différentes **architectureS** qu'on trouve dans le système et qui sont plus ou moins partagées par les sous-systèmes :
    - **Architectures logicielles :** MVC, Dossiers, Fichiers, Bibliothèques, Classe, Package, etc. C'est fonction des langages et des usages.
    - **Architecture des données :** organisation des données. Structures des données, base de donnée, classes. Elle reprend la 5ème et dernière distinction.

- **L'analyse détaillée** (ou spécifications détaillées) : elle s'occupe du découpage en fonctions informatiques ou en classe avec des méthodes pour chacun des sous-systèmes logiciels. A ce niveau vont apparaître les en-têtes des fonctions, leurs modes d'emploi, leurs principes de résolution, voir leurs pseudo-codes.

Cinquième distinction : données versus traitements : l'analyse des données.

Principes

Les 4 distinctions précédentes sont centrées sur la questions des traitements (c'est-à-dire des fonctionnalités).

Seul le point d'architecture des données de l'analyse organique générale aborde la question des données.

La dernière distinction est celle qui est faites entre les données et les traitements.

Les données seront analysées pour elle-même, indépendamment des traitements qu'on leur appliquera.

Les données peuvent être analysées à deux niveaux : celui de l'analyse fonctionnelle et celui de l'analyse organique.

Approche « base de données »

Dans une approche « base de données », l'analyse concerne les données de la BD. Selon le niveau de l'analyse (conceptuel, organisationnel, logique ou physique), l'analyse des données se fera indépendamment de l'architecture (les sous-systèmes), en fonction des sous-systèmes, en fonction du modèle des données utilisé (modèle relationnel en général) et enfin en fonction de choix concret de réalisation en vue d'une optimisation des usages.

Ainsi le niveau conceptuel correspond au fonctionnel, le niveau organisationnel à l'architecture organique et les niveaux logique et physique à l'analyse organique détaillée.

Approche « orientée objet »

Dans une approche « orientée objet », il n'y a pas de découplage entre les données et les traitements. L'analyse des données (le diagramme de classes) se situe donc uniquement au niveau de l'analyse organique et pas du tout au niveau de l'analyse fonctionnelle (les cas d'utilisation).

Cependant, on pourra distinguer un niveau intermédiaire entre le fonctionnel et l'organique : celui des classes « métier » qui reprennent globalement l'organisation de la BD et dans lesquelles seront réparties les responsabilités mises au jour par l'analyse fonctionnelle.

Au niveau organique, on entrera dans le détail du diagramme des classes.

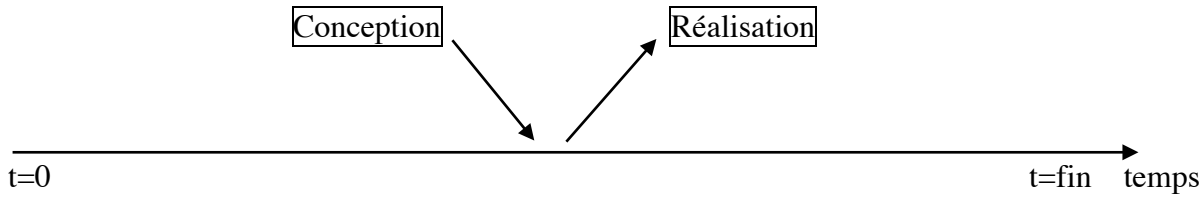
3. Le cycle en V

Logique du V

Le cycle en V c'est une méthode classique de développement du logiciel.

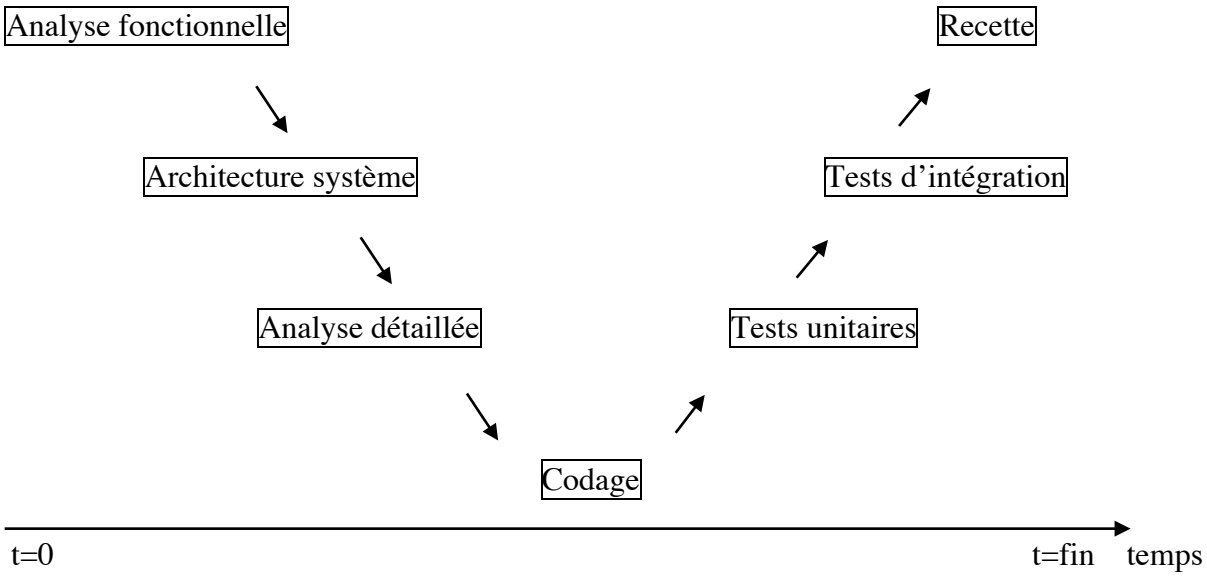
Construction du V

- ***la conception et la réalisation forment les deux branches du cycle en V :***



- ***Ces deux étapes sont détaillées :***

On reprend les 3 premières distinctions abordées précédemment et en ajoutant des distinctions dans la réalisation :



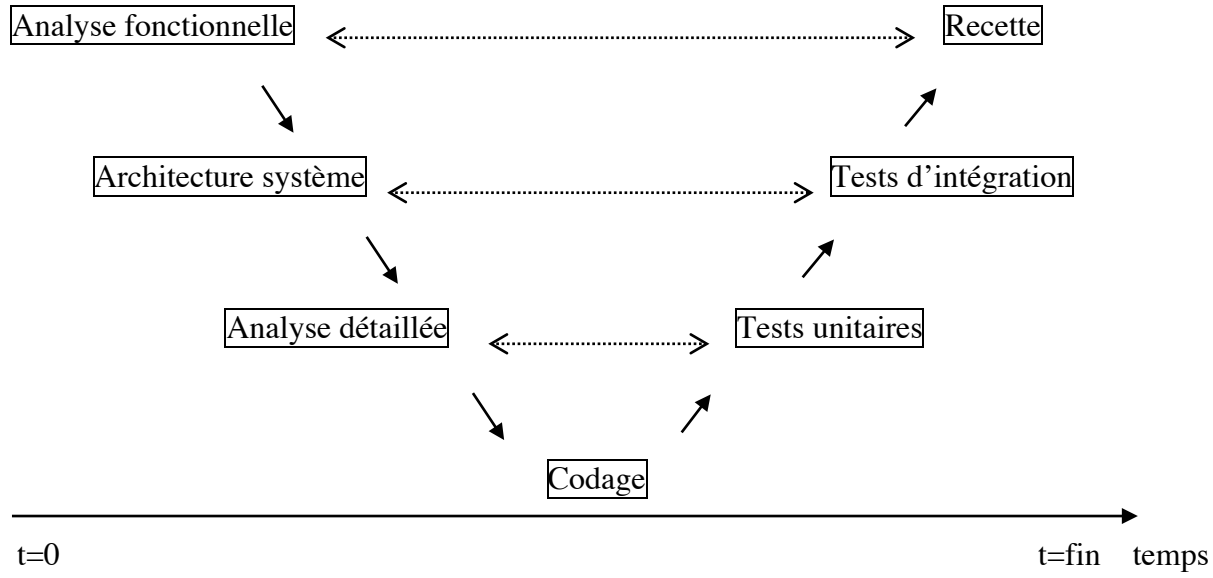
Conception = Analyse fonctionnelle + Architecture système + Analyse détaillée.

Réalisation = Codage + Tests unitaires + Tests d'intégration + Recette.

➢ ***C'est le lien entre les étapes de chaque branche qui justifie le cycle en V :***

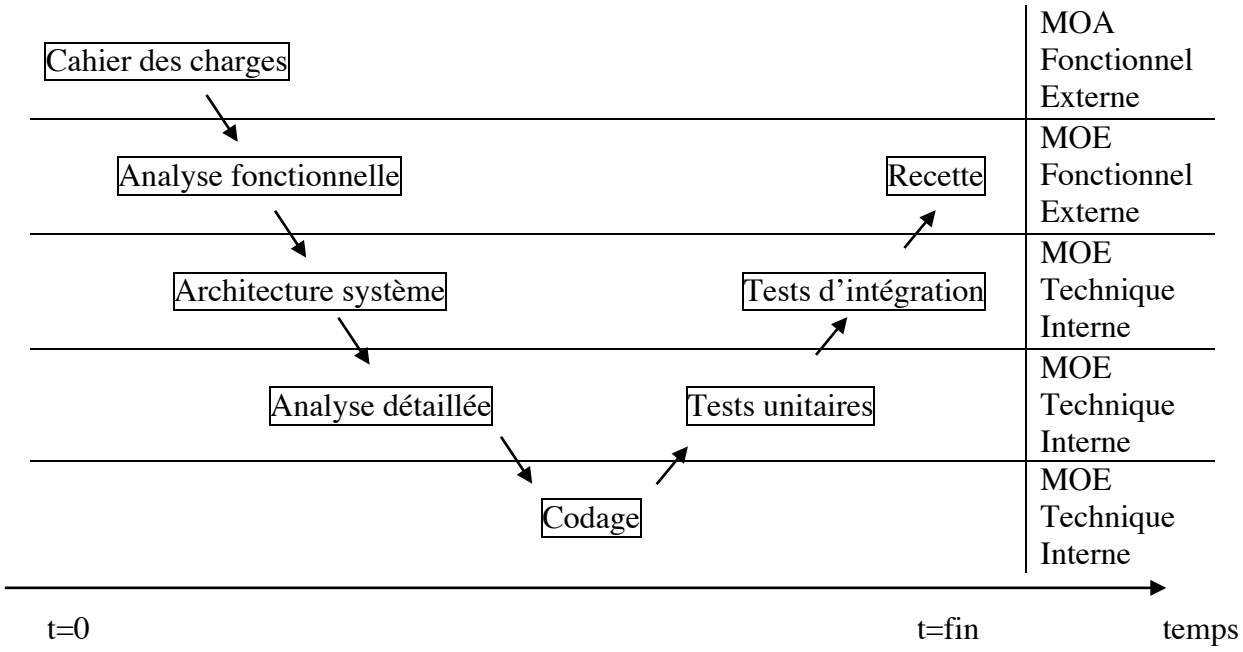
- Quand on fait l'analyse fonctionnelle, on peut préparer la procédure de recette.
- Quand on fait l'architecture système, on peut préparer les tests d'intégration des sous-systèmes.
- Quand on fait l'analyse détaillée, on peut préparer les tests unitaires

Ainsi, cela permettra, en cas de problème de test (unitaire, d'intégration ou de recette), de revenir facilement à la partie de la conception à laquelle le problème correspond.



Relations cycle en V, MOA, MOE, fonctionnel, technique

La MOA produit le cahier des charges qui est en entrée du cycle en V.



La production des documents

Le cycle en V correspond aussi à un cycle de consommation et de production des documents.

Les activités du cycle en V utilisent les documents des activités précédentes et produisent des documents qui seront utilisés aux étapes suivantes : l'étape immédiatement suivante dans le cycle en V et l'étape de même niveau dans le cycle en V.

Le client (maître d'ouvrage) produit le cahier des charges.

L'analyse fonctionnelle se base sur le cahier des charges.

Elle aboutit à un document d'analyse fonctionnelle. Ce document pourra être validé par le client de façon à vérifier la bonne compréhension du cahier des charges par l'informaticien. Ce document servira d'entrée pour l'architecture et l'analyse détaillée.

Elle aboutit aussi à un document de recette et au manuel utilisateur. Le document de recette servira d'entrée pour la recette. Le manuel utilisateur servira au client une fois le produit livré.

L'architecture se base sur le document d'analyse fonctionnelle et éventuellement sur le cahier des charges.

Elle aboutit à une document d'architecture et à un document d'intégration.

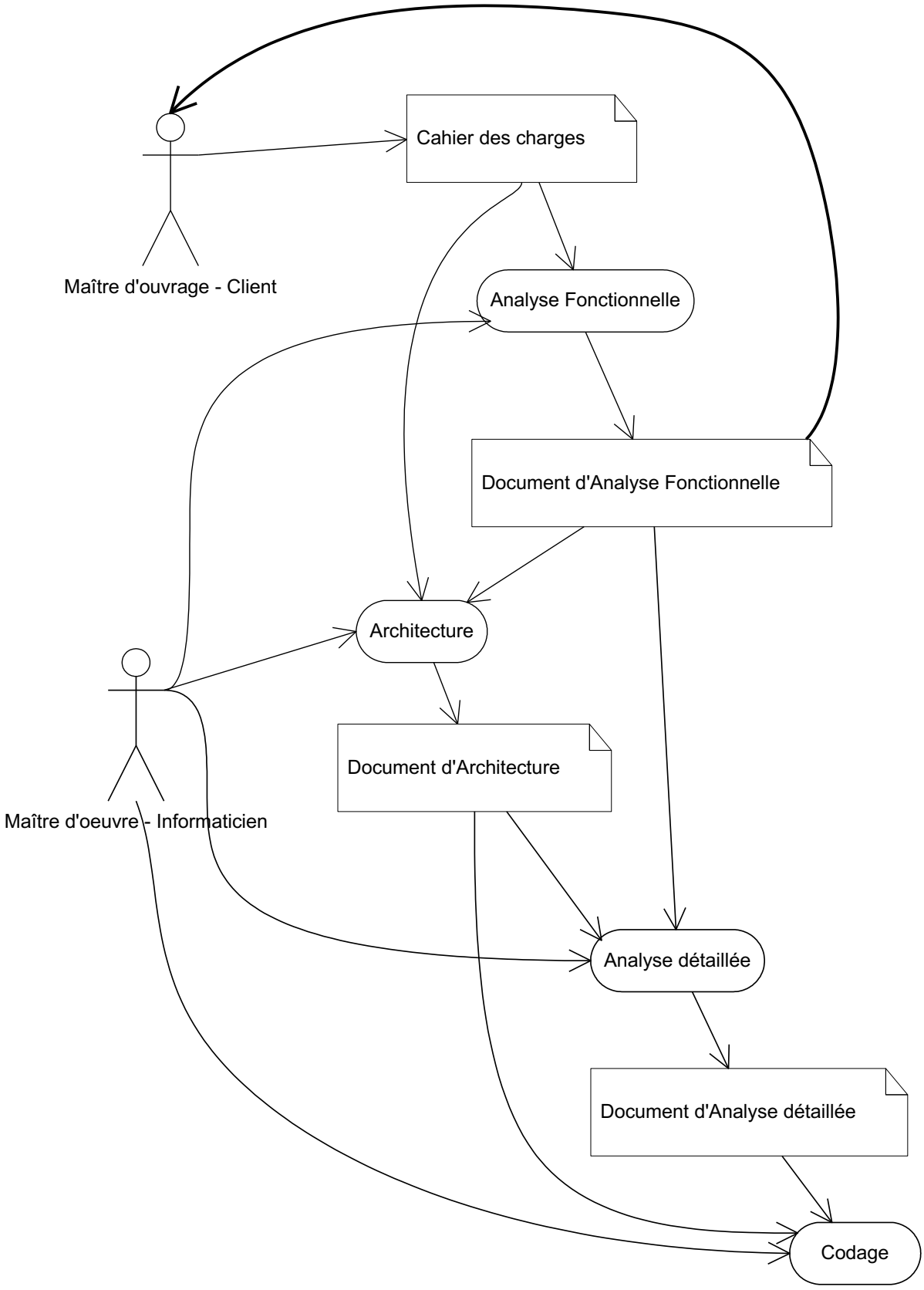
L'analyse détaillée se base sur le document d'analyse fonctionnelle et sur le document d'architecture. A ce niveau, on n'utilise plus le cahier des charges. Elle aboutit à un document d'analyse détaillée et à un document de test unitaire.

Remarques

L'analyse fonctionnelle produit aussi un document de recettes qui sera utilisé à la fin par l'activité de recette.

L'architecture produit aussi un document d'intégration qui sera utilisé par l'activité d'intégration.

L'analyse détaillée produit aussi un document de tests unitaires qui sera utilisé par l'activité de tests unitaires.



Cycle de la documentation

Cycle de la documentation dans la branche « conception » du logiciel. Les documents sont les rectangles cornés, les ovales aplatis sont les process. A noter que les documents à destination de la branche des tests ne sont pas présentés.

Réalisation et langage de programmation

Une fois la conception terminée, on passe à la réalisation.

La réalisation peut se faire avec n'importe quel langage.

Toutefois, dans le cas d'un SI centré sur une base de données, on utilisera probablement le SQL pour la partie directement liée à la base de données.

Pour l'interface utilisateur, on utilisera indifféremment des langages directement (C, Java, php, etc.), mais aussi des framework (Zend, Symfony, etc.) ou des environnements de développement rapide du type de 4D (quatrième dimension) ou de Oracle Database XE (freeware depuis mars 2006).

Cycle en V et analyse des données

Le cycle en V ne prend pas explicitement en compte l'analyse des données des données.

L'analyse des données peut être considérée comme une partie de chaque étape de la conception. On commence au niveau de l'analyse fonctionnelle : c'est le MCD MERISE et le diagramme des classes métier.

On continue au niveau de l'architecture : c'est le MOD MERISE.

On finit au niveau de l'analyse détaillée : c'est le MLD et MPD MERISE et le diagramme de classes détaillé.

4. Du cycle en V aux méthodes « agiles » : cycle de vie itératif

Principal défaut du cycle en V : le résultat apparaît à la fin

Si le cycle en V est appliqué sur tout le projet, on ne verra le résultat final qu'au cours de la recette, à la dernière étape du projet.

Cela risque de créer des déceptions ! C'est l'effet tunnel.

Avant le V : cycle de vie en tunnel et cycle de vie en cascade

Le cycle de vie en tunnel

Il part de l'analyse du cahier des charges pour arriver à la mise en production en passant par un « tunnel » : aucune étape intermédiaire n'est planifiée !

Le cycle de vie en cascade

Il part de l'analyse du cahier des charges pour arriver à la mise en production en planifiant chaque étape mais sans faire les corrélations du cycle en V.

Après le V : cycle de vie itératif

Le cycle en V est une méthode rigide et très formelle qui est efficace pour des gros développements industriels (ex : poste de contrôle et de commandement d'un métro automatique).

La méthode peut être assouplie en ajoutant la **notion d'itération** : dans la méthode itérative, on réalise une partie des fonctionnalités selon le cycle en V, puis une autre, etc. jusqu'à la réalisation de toutes les fonctionnalités.

Différents modèles sont apparus dès les années 80 comme évolutions du cycle en V pour corriger ce défaut : cycle en W, spirale de Boehm, etc.

Du fait du caractère systémique du paradigme objet, et non pas hiérarchique comme celui du paradigme procédural classique, **la programmation objet se prête bien à cette méthode** puisqu'elle consiste concrètement à faire croître le diagramme de classes par ajout de nouveaux composants logiciels (classes et paquetages).

L'avantage de cette méthode est que **le client peut « voir » des résultats concrets au fur et à mesure** du développement. La méthode permet aussi de faciliter les évolutions du cahier des charges.

Ces évolutions permettent la **production** de **prototype**.

Précisions sur itération et incrément

Une itération c'est une première **esquisse** grossière du projet complet qui sera affinée à l'itération suivante. **Un incrément** c'est un **morceau** du projet.

Les méthodes post cycle en V sont itératives et / ou incrémental : http://romy.tetue.net/mona-lisa-agile.

Méthodes agiles

La méthode agile est une méthode incrémentale.

Il existe plusieurs méthodes agiles.

La notion de **méthode « agile »** prend en compte le fait que **le client n'est pas forcément capable dès le début de fixer son cahier des charges**. Celui-ci peut donc évoluer au fur et à mesure du développement qui sera itératif permettant ainsi de pouvoir commencer à tester, voir utiliser en production le produit en cours de développement.

Ce type de situation correspond particulièrement aux projets web.

Principes des méthodes agiles

- Itératif
- Incrémental
- Adaptatif
- Livraisons régulières de versions opérationnelles (les « sprints »)
- Accepter le changement

Les deux écueils à éviter : l'effet tunnel et l'usine à gaz

L'effet tunnel

L'effet tunnel consiste à rentrer dans un tunnel en début de projet et à n'en ressortir qu'à la fin. Entre temps, personne ne sait ce qui se passe.

L'effet tunnel risque de conduire à ne pas répondre aux besoins.

Il faut produire des incréments pour éviter ça : régulièrement, on doit pouvoir mesurer et critiquer l'état d'avancement ainsi que la réalisation.

L'usine à gaz

« L'usine à gaz », c'est une architecture de « patch » : on ajoute des petits bouts sur d'autres petits bouts pour arriver au résultat finale.

L'agilité permanente risque de conduire à une architecture d' «usine à gaz » : les incréments architecturaux peuvent finalement rendre la structure très difficile à faire évoluer.

Il faut avoir une vision globale de l'architecture même si on avance par par incrément successifs. C'est très évident pour la base de données.

5. Génie Logiciel *vs* Ingénierie des Systèmes d'Information

Génie logiciel - Software

Le terme de génie logiciel (*software engineering*) est né en Europe à la fin des années 60.

Le G.L vise à transformer les besoins et attentes des utilisateurs en une application informatique.

Besoins et attentes ⟶ **Application informatique**

Quoi : software

Qui : les informaticiens.

Le G.L. regroupe :

- Des **METHODES** (organisation du travail)
- Des **TECHNIQUES** (langages de programmation, API, documentation des programmes)
- Des **OUTILS** (IDE, systèmes de gestion de la documentation, etc.) de développement du logiciel.

Ingénierie des systèmes d'information - Brainware

Ingénierie des systèmes d'information

Le terme d'ingénierie des systèmes d'information (*requirement engineering*) est né au début des années 90.

L' I.S.I vise à transformer les besoins et attentes des utilisateurs en spécifications formalisées d'une future application informatique.

Besoins et attentes ⟶ **Spécifications formalisées**

Quoi : brainware

Qui : les informaticiens, les gestionnaires et les autres utilisateurs du système d'information

L' I.S.I. regroupe :

- Des **METHODES** d'organisation du travail de spécification : MERISE, SCRUM, etc.
- Des **TECHNIQUES** de modélisations : MCD, MLD, UML, architectures, etc.
- Des **OUTILS** de modélisation et de spécifications : POWER designer, etc.

utilisés pour le développement des spécifications.

Le brainware

Le concept de brainware, très peu usité, a été introduit par Tosio Kitagawa en septembre 1974 dans le n°39 des Research Report of Research Institute of Fundamental Information Science.

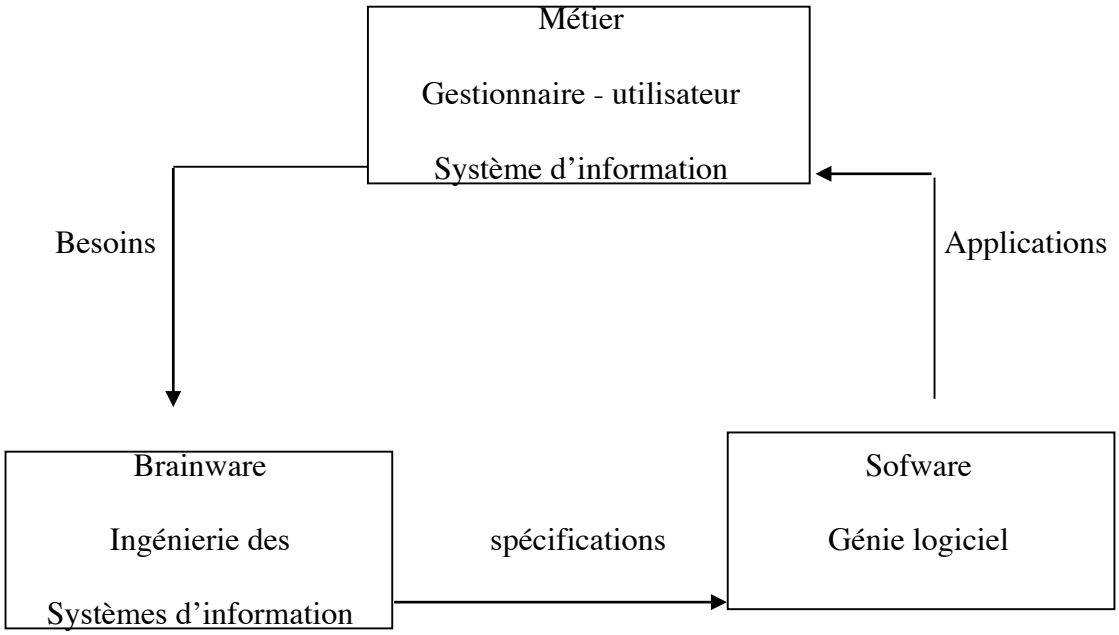
Le brainware est l'analyse intellectuelle qui fonde le software.

Le brainware est un matériau (ware) en ce sens que c'est un stock objectif de connaissances et d'informations.

On peut donc distinguer entre :

software ingenering - brainware ingenering

Relations entre software engineering et brainware engineering



ISIM, p. 2

6. Méthode analytique vs méthode systémique

Méthode analytique

Principe : du général au particulier

La méthode analytique est la méthode de décomposition classique : elle consiste à aller du général au particulier.

Le principe est de « diviser pour régner ». C'est une détournement de la signification d'origine de l'expression qui est politique, mais qui donne une bonne image, par une formule, du principe.

Discours de la méthode

On retrouve les fondements de la méthode analytique chez Descartes :

> *Certains chemins m'ont conduit à des considérations et des maximes dont j'ai formé une méthode par laquelle il me semble que j'ai moyen d'augmenter par degrés ma connaissance, et de l'élever peu à peu au plus haut point…*
>
> *Au lieu de ce grand nombre de préceptes dont la logique est composée, je crus que j'aurais assez des quatre suivants, pourvu que je prisse une ferme et constante résolution de ne manquer pas une seule fois a les observer.*
>
> ***Le premier*** *était de ne recevoir jamais aucune chose pour vraie que je ne la connusse évidemment être telle; c'est-à-dire, d'éviter soigneusement la précipitation et la prévention, et de ne comprendre rien de plus en mes jugements que ce qui se présenterait si clairement et si distinctement à mon esprit, que je n'eusse aucune occasion de le mettre en doute.*
>
> ***Le second,*** *de diviser chacune des difficultés que j'examinerais, en autant de parcelles qu'il se pourrait, et qu'il serait requis pour les mieux résoudre.*
>
> ***Le troisième,*** *de conduire par ordre mes pensées, en commençant par les objets les plus simples et les plus aisés à connaître, pour monter peu à peu comme par degrés jusques à la connaissance des plus composés, et supposant même de l'ordre entre ceux qui ne se précèdent point naturellement les uns les autres.*
>
> ***Et le dernier,*** *de faire partout des dénombrements si entiers et des revues si générales, que je fusse assuré de ne rien omettre.*
>
> Discours de la méthode, 1637, Descartes (1596-1650)

Ce discours met en avant quatre principes :

1. L'évidence contre les préjugés pour décrire la réalité.
2. L'analyse : La division du tout en partie.
3. La synthèse : la reconstruction du tout à partir des parties.
4. La totalité : ne rien omettre.

L'analyse descendante et ses étapes

La bonne méthode consiste à diviser le tout en parties, puis à réaliser les parties, avant de les réintégrer toutes ensembles. C'est l'analyse descendante.

Les étapes sont les suivantes :

1. Partir de la réalité : le problème à traiter.
2. S'en faire une idée claire et complète.
3. Diviser cette idée en parties.
4. Construire les parties une par une.
5. Intégrer les parties toutes ensemble.

Les erreurs de méthode

- ***Ne pas partir du tout***

Une erreur classique consiste à ne pas partir du tout, mais à partir directement des parties. C'est **l'analyse ascendante : on part de l'étape 4.**

Le défaut de cette méthode est que l'absence d'analyse initiale de l'idée de la totalité va conduire à des grandes difficultés au moment de l'intégration des parties.

- ***Se tromper sur le tout***

En appliquant l'analyse descendante, on peut aussi se tromper en appliquant mal l'étape 2.

L'idée qu'on se fait du problème à traiter est incomplète ou fausse. En conséquence de quoi l'intégration des parties ne pourra pas donner un bon résultat.

Brève présentation de la méthode systémique

Systémique et système d'information

La science des systèmes, ou systémique, est à l'origine du développement de la notion de système d'information.

Principe : la recherche d'invariants

La méthode systémique consiste à analyser le système non pas par fonction, mais par structure.

Pour analyser la structure, on recherche les **invariants au niveau fonctionnel**.

Ces invariants sont la structure du système :

- **les données brutes : elles aboutissent au modèle des données**
- **les règles d'organisation : elles aboutissent au modèle des traitements.**

Ces invariants permettent la modélisation : la représentation du réel par un modèle.

Caractéristiques des systèmes étudiés par la systémique

- Ils ont un projet identifiable : c'est **l'hypothèse téléologique** (télos = finalité)
- Ils sont ouverts sur leur environnement : c'est **l'hypothèse d'ouverture.**
- Ils sont décrits totalement, dans l'espace et le temps : **c'est l'hypothèse structuraliste.**

Caractéristiques des systèmes ouverts

- La **rétroaction** : la rétroaction consiste à ce que les informations en sortie du système reviennent en entrée dans le système.
- **Equifinalité** : les mêmes conséquences peuvent avoir des origines différentes.
- La **simulation** : du fait de la rétroaction et de l'équifinalité, les systèmes ouverts s'étudient avec des simulations.

7. Notion de Système d'Information – SI – Vers la méthode MERISE

Définition Wikipédia

Le système d'information (SI) est un ensemble organisé de ressources qui permet de collecter, stocker, traiter et distribuer de l'information, en général grâce à un ordinateur. Il s'agit d'un système socio-technique composé de deux sous-systèmes, l'un social et l'autre technique.

L'apport des nouvelles technologies de l'Information est à l'origine du regain de la notion de système d'information.

Présentation

La notion de système d'information est une notion issue de la science des systèmes (ou systémique).

Un système est un **ensemble d'éléments** :

- **reliés entre eux,**
- **capables d'actions**,
- **compris dans un tout** (le système)
- **pouvant communiquer (échanger) avec l'extérieur**.

Le système d'information est une représentation possible de n'importe quel système, notamment de tout système humain organisé, donc de toute entreprise en général (l'épicier du coin, un poids lourd de la grande distribution, une bibliothèque, une école, l'Etat, une association sportive, une association caritative, etc.).

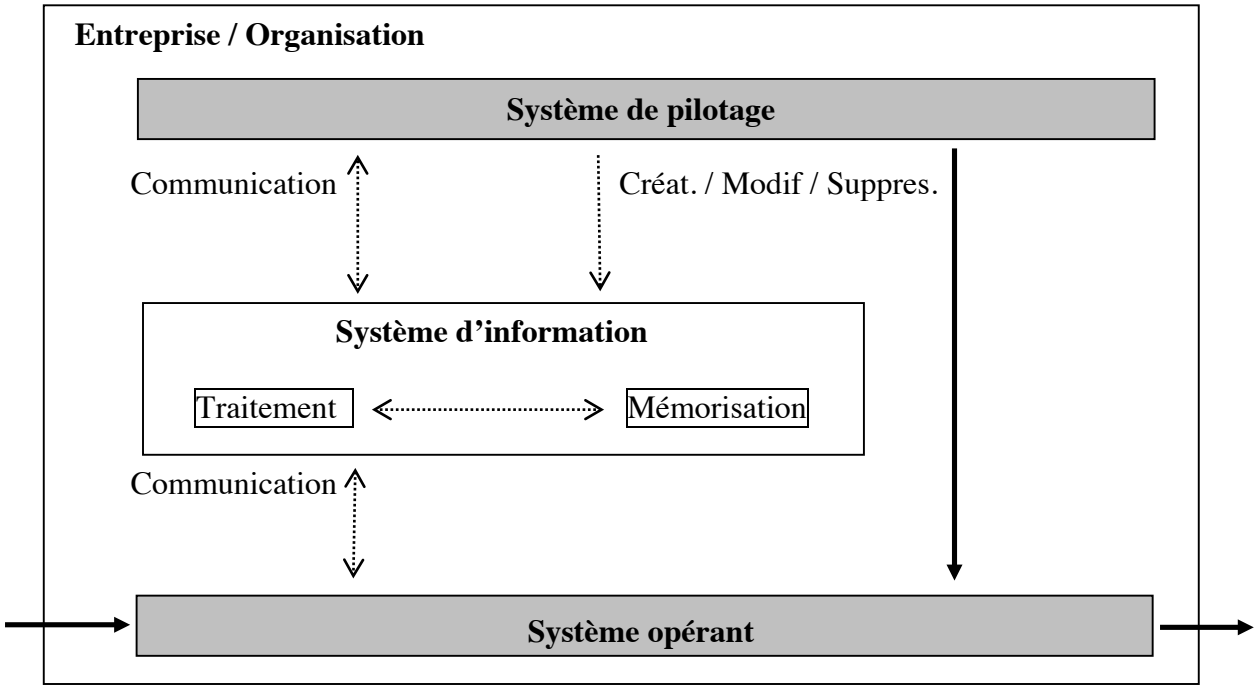
Les systèmes d'information préexistent donc à l'informatique.

Toutefois, les nouvelles technologies de l'Information sont à l'origine du regain de la notion de système d'information.

Schéma théorique d'un système d'information

L'analyse systémique permet d'arriver à la modélisation suivante de l'entreprise :

Environnement



Le système d'information est un système de mémorisation et de traitement de l'information au sens large, interfacé avec le système opérant et le système de pilotage. Ce système est en partie informatisé

Le système opérant – SOP - est le siège de l'**activité productive** de l'entreprise. Cette activité consiste en une transformation de données en entrée (les flux primaires). Ces flux primaires peuvent être des flux de matière, de finance, de personnel ou d'information. Par exemple : le système opérant reçoit une commande et la traite.

Le système de pilotage est le siège de l'**activité décisionnelle** de l'entreprise. Il permet le pilotage, la régulation et l'adaptation, par la communication avec le SI, la mise à jour du SI et l'envoi de décisions au SOP. Cette activité décisionnelle est très large et elle est assurée par de nombreux acteurs de l'entreprise à des niveaux divers. Par exemple : le système de pilotage décide d'une campagne publicitaire ou de l'installation d'une nouvelle application informatique dans le système d'information.

: flèche des flux d'entrée et de sortie du système opérant

La communication : consiste à consulter, insérer, supprimer, modifier des données dans des cadres déjà définis (consulter, insérer, supprimer, modifier les tuples des tables de la BD par exemple).

La création / Modification / Suppression : consiste à créer, modifier ou supprimer les cadres (les tables de la BD par exemple).

Distinction entre SIO et SII

On distingue dans le SI entre :

- **Le système d'information organisationnel**, SIO. C'est une représentation possible de **n'importe quel système**, notamment de tout système humain organisé (donc de toute entreprise). **Une entreprise** peut donc être considérée comme un SIO.

Et

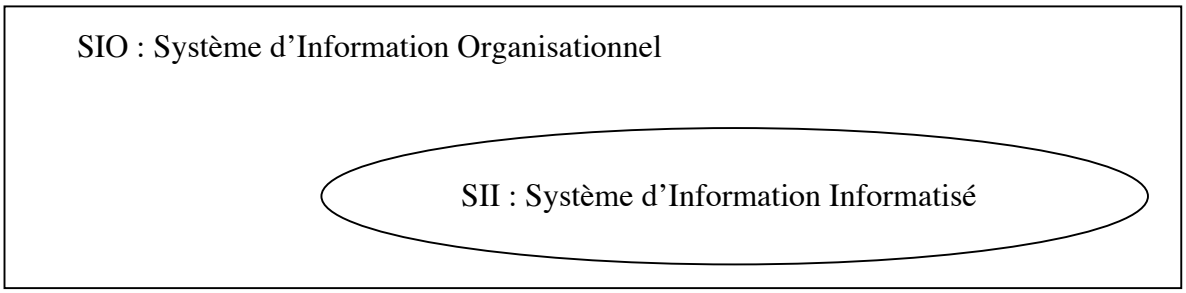
- **Le système d'information informatisé**, SII, c'est la **partie informatisée du SIO** d'information organisationnel à laquelle les utilisateurs ont accès. Il est constitué par les machines, les logiciels et les fichiers qu'on utilise dans l'entreprise.

Un SIO peut contenir un SII, mais ce n'est pas obligatoire. Tout SII est inclus dans un SIO.

Le SIO est tourné vers les utilisateurs et fera appel à certaines disciplines des sciences de la gestion.

Le SII est sous la responsabilité des informaticiens et fera appel aux disciplines de l'administration des systèmes et des réseaux et du génie logiciel.

Cependant, **un SII est au service du SIO mis en place par les dirigeants de l'entreprise, et non l'inverse** ! La conception du SII doit s'appuyer sur celle du SIO et non l'inverse !



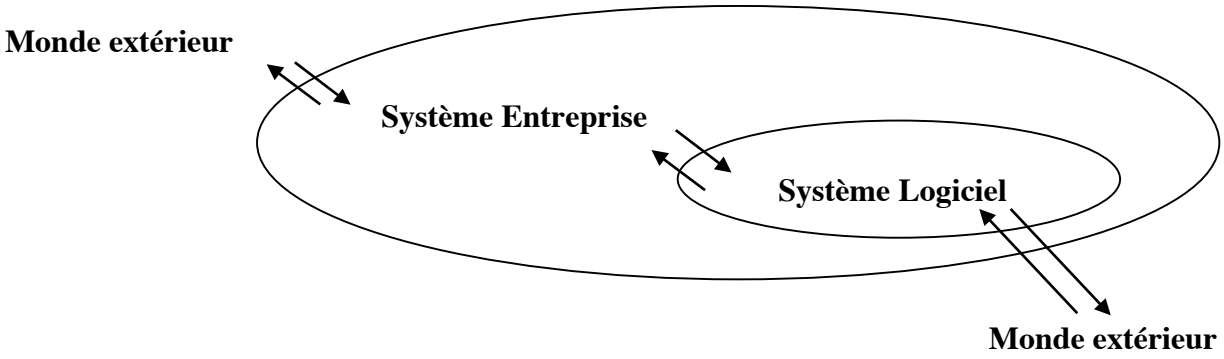
Distinction entre système entreprise et système logiciel

Dans une première analyse, on ne va pas s'intéresser directement à l'activité réalisée via un logiciel, mais plus généralement à l'activité réalisée par l'entreprise.

On peut distinguer **3 lieux** :

- **L'entreprise** (le système entreprise).
- **Le monde extérieur.**
- **Le logiciel** (système logiciel).

Ces trois lieux sont des abstractions concentriques : l'entreprise inclut le logiciel et le monde extérieur inclut l'entreprise.



On peut ensuite décrire les échanges entre ces **trois lieux :**

- Le monde extérieur communique avec l'entreprise.
- L'entreprise communique avec son système logiciel.
- Le monde extérieur peut aussi communiquer directement avec le système logiciel (borne automatique, site internet…).

Notion d'acteur

Les acteurs sont ceux qui interagissent avec le système, que ce soit le système entreprise ou le système logiciel.

Le ME contient donc une série d'acteurs interagissant avec le SE et le SL.

Le SE contient lui aussi une série d'acteurs interagissant avec le SL.

L'analyse de ces acteurs fait partie de l'analyse des traitements du système.

Notion de poste de travail

Un poste de travail c'est une machine à partir de laquelle un acteur peut interagir avec le SI.

L'analyse de ces acteurs fait partie de l'analyse des traitements du système et particulièrement elle met au jour l'architecture fonctionnelle.

Exemple : une bibliothèque parisienne

Description simplifiée

Dans les bibliothèques parisiennes, les adhérents peuvent emprunter et rendre les livres qu'ils empruntent directement sur un poste de travail dédié à cela. Ce poste scanne les livres et les cartes d'adhérent pour pouvoir fonctionner.

Les adhérents peuvent aussi consulter le catalogue à partir de machine dédiées à cette tâche.

Les bibliothécaires peuvent aussi rendre ces services à partir de leur machine.

Les adhérents peuvent aussi consulter leur compte sur internet et prolonger leurs emprunts.

Bilan en terme de postes de travail

On a donc 4 postes de travail :

- La borne d'emprunt et de retour
- La borne de consultation
- Le poste de travail du bibliothécaire
- Le site internet

Première conclusion technique

Ces 4 postes de travail vont partager les informations concernant les livres et les adhérents pour pouvoir fonctionner. Ces informations sont rangées dans une base de données.

8. La méthode MERISE

Définition

MERISE est une méthode systémique de conception des systèmes d'information.

Elle est en relation avec le développement des bases de données relationnelles.

Historique

1970	Modèle Relationnel de Codd.
Années 70	Premiers prototypes de SQL
1976	Modèle Entité Association de Chen
1974-78	Le noyau de MERISE est établi par une équipe d'ingénieurs et de chercheurs aixois.
1978	Développement de MERISE : méthode française de conception de systèmes d'information, sous l'égide du ministère de l'industrie.
1979	Conception du système d'information, construction de la base de données, H. Tardieu, D. Nanci, D. Pascot (préfacé par J.-L. Le Moigne), Editions d'Organisation.
1979	Première version de SQL, proposé par ORACLE.
1983	La méthode MERISE - Tome 1 : principes et outils. H. Tardieu, A. Rochfeld, R. Colletti. Éditions d'Organisation.
1985	La méthode MERISE - Tome 2 : démarche et pratique. H. Tardieu, A. Rochfeld, R. Colletti, G. Panet, G. Vahée. Éditions d'Organisation.
1986	SQL ANSI (American National Standard Institute)
1989	SQL-1, ISO et ANSI (International Standard Organisation)
1989	La méthode MERISE - Tome 3 : gamme opératoire. A. Rochfeld, J. Moréjon. Édition d'Organisation.
1992	Ingéniérie des systèmes d'information : MERISE. 1ère édition. D. Nanci, B. Espinasse. Sybex.
1992	SQL-2, ISO et ANSI
fin années 90	PHP-MySQL
1999	SQL-3, ISO et ANSI
2001	Ingéniérie des systèmes d'information : MERISE. 4ème édition. D. Nanci, B. Espinasse. Vuibert.
2006	Oracle Database XE

En 2001, la méthode MERISE était encore la méthode de conception de systèmes d'information la plus largement pratiquée en France.

MERISE a pris en compte les évolutions de l'informatique et continue de s'adapter aux nouvelles technologies : architectures clients/serveur, interfaces graphiques, démarche de développement rapide, approche objet, applications intra/internet.

Aujourd'hui, la méthode MERISE correspond encore globalement aux savoir-faire actuels en ingénierie des systèmes d'information de gestion.

MERISE constitue un standard de fait en conception des systèmes d'information.

Les 3 cycles de la démarche MERISE

La démarche de développement proposée par MERISE s'inscrit dans trois cycles :

- **Le cycle de vie :** c'est le découpage du projet en trois périodes : conception, réalisation et maintenance. Le cycle de vie rejoint le cycle en V.
- **Le cycle de décision :** c'est la liste de tous les moments où une décision est prise sur le projet (décision de faire le projet après une étude préalable, décision de valider l'analyse fonctionnelle et de passer à l'architecture, validation de la recette, etc.)
- **Le cycle d'abstraction :** c'est l'organisation structurelle des données et des traitements.

On va surtout s'intéresser au cycle d'abstraction.

La distinction entre données et traitements

Le cycle d'abstraction est basé sur une distinction entre les données et les traitements.

C'est la dichotomie fondamentale de MERISE.

Elle est directement issue de l'approche base de données.

Le cycle d'abstraction

Le cycle d'abstraction est découpé en quatre niveaux : conceptuel, organisationnel, logique et physique.

- **Le niveau conceptuel :** il exprime des choix fondamentaux de gestion (recherche d'éléments stables indépendamment des moyens à mettre en œuvre, de leurs contraintes et de leur organisation). Répond à la question : **QUOI**.
- **Le niveau organisationnel :** il exprime les choix d'organisation de ressources humaines et matérielles, au travers notamment de la définition d'acteurs et de postes de travail. Répond aux questions : **QUI, OU, QUAND**.
- **Le niveau logique :** il exprime les choix de moyens et de ressources informatiques, en faisant abstraction de leurs caractéristiques techniques précises. C'est le niveau du modèle relationnel (moyen informatique : base de données relationnelle), du diagramme des classes et des diagrammes de séquence objets (moyen informatique : langage orienté objet). Répond à la question : **COMMENT**.
- **Le niveau physique :** il traduit les choix techniques et la prise en compte de leurs spécificités. C'est le niveau du code dans un langage particulier.

LE CYCLE D'ABSTRACTION		
Niveaux	**DONNEES**	**TRAITEMENTS**
CONCEPTUEL **QUOI**	**M C D** ***Modèle conceptuel des données*** Signification des informations sans contraintes techniques, organisationnelle ou économique. **Modèle entité – association**	**M C T** ***Modèle conceptuel des traitements*** Activité du domaine sans préciser les ressources et leur organisation
ORGA-NISATIONNEL **QUI, OU, QUAND**	**M O D** ***Modèle organisationnel des données*** Signification des informations avec contraintes organisationnelles et économiques. (Répartition et quantification des données ; droit des utilisateurs)	**M O T** ***Modèle organisationnel des traitements*** Fonctionnement du domaine avec les ressources utilisées et leur organisation (répartition des traitements sur les postes de travail)
LOGIQUE **COMMENT**	**M L D** ***Modèle logique des données*** Description des données tenant compte de leurs conditions d'utilisation (contraintes d'intégrité, historique, techniques de mémorisation). **Modèle relationnel**	**M L T** ***Modèle logique des traitements*** Fonctionnement du domaine avec les ressources et leur organisation informatique.
PHYSIQUE **COMMENT**	**M P D** ***Modèle physique des données*** Description de la (ou des) base(s) de données dans la syntaxe du Système de Gestion des données (SG.Fichiers ou SG Base de Données) Optimisation des traitements (indexation, dénormalisation, triggers).	**M P T** ***Modèle physique des traitements*** Architecture technique des programmes

D'après ISIM, p. 37

SIO et SII dans le cycle d'abstraction

Niveau	DONNÉES	TRAITEMENTS	SI
Conceptuel	**M C D**	**M C T**	**SIO** Système d'information organisationnel
Organisationnel	**M O D**	**M O T**	
Niveau logique	**M L D**	**M L T**	**SII** Système d'information informatisé
Niveau physique	**M P D**	**M P T**	

ISIM, p. 218

Le cycle de vie

Le cycle de vie MERISE est une méthode de développement au même titre que le cycle en V.

Le cycle de vie est découpé en trois périodes: la conception, la réalisation et la maintenance.

LE CYCLE DE VIE		
Etapes de la démarche		**Explications**
Conception	Schéma directeur	Définition des orientations générales du développement à moyen terme des systèmes d'information
	Étude préalable	Proposition et évaluation de solutions d'organisation et de solutions techniques pour le SI d'un domaine. Cette étape porte sur un sous-ensemble représentatif du domaine étudié.
	Étude détaillée	Spécifications complètes du futur SIO du point de **vue de l'utilisateur** (point de vue externe). Elle comporte deux phases : • la conception générale (extension de l'étude préalable à tout le domaine) • la conception détaillée (description complète de chacune des tâches à automatiser).
Réalisation	Étude technique	Spécifications complètes du futur SII du **point de vue du réalisateur** (point de vue interne).
	Production logicielle	Écriture des programmes, générations des fichiers ou des bases de données, tests.
	Mise en service	Installation de l'application informatique, vérification du bon fonctionnement, mise en place de la nouvelle organisation, formation des utilisateurs.
Maintenance	Maintenance	Rectification des anomalies, améliorations, évolutions.

ISIM, p. 32

Le cycle de décision

Le cycle de décision représente l'ensemble des choix qui doivent être faits durant le déroulement du cycle de vie.

Étapes de la démarche		Résultats	Décisions
Schéma directeur	MOA	Plan de développement des SI	Approbation et mise en application
Étude préalable	MOA	Dossier des choix, n solutions	Choix d'une solution ou arrêt .
Étude détaillée	MOE	Spécifications fonctionnelles	Accord des utilisateurs sur les spécifications fonctionnelles
Étude technique	MOE	Spécifications techniques pour la réalisation	Accord des réalisateurs sur les spécifications techniques
Réalisation logicielle	MOE	Système réalisé en ordre de marche	Recette provisoire : conformité du système
Mise en service	MOE	Système installé dans l'organisation	Recette définitive : système en service
Maintenance	MOE	Système maintenu	Recette simplifiée : fin de maintenance

ISIM, p. 41

Les plans types

Les étapes de la démarche du cycle de vie donnent lieu à la production de documents.

Comme toute autre méthode, la méthode MERISE propose des plans types pour tous les documents prévus par la méthode.

L'étude préalable et de l'étude détaillée sont les deux études sont les plus spécifiques à MERISE car elles font intervenir l'essentiel du cycle d'abstraction.

On présente ci-dessous les plans type de ces deux étapes.

Plan type de l'étude préalable : production du cahier des charges

- ***1. Recueil***
  - Préparation et réalisation des interviews
  - Recherche de la documentation
  - Description et bilan de l'existant
- ***2. Conception***
  - Élaboration des divers scénarios
  - Élaboration des MCD et MCT
  - Maquette et prototype
  - Élaboration du cahier des charges fonctionnel
- ***3. Qualité***
  - Définition des exigences qualité globale
  - Définition des exigences qualité par fonction
- ***4. Chiffrage***
  - Estimation prévisionnelle des charges, coût, délais
  - Planning prévisionnel
- ***Résultats obtenus :***
  - Cahier des charges fonctionnel
  - Dossier de choix

Plan type de l'étude détaillée : production de spécifications

- ***1. Recueil complémentaire***

  - Préparation et réalisation des interviews des utilisateurs
  - Recherche de la documentation
  - Actualisation de l'étude préalable

- ***2. Conception***

  - Mise à jour des MCD et MCT
  - Élaboration du MOT
  - Description des états et des écrans
  - Validation croisée MCD / MOT
  - Élaboration du MLD

- ***3. Qualité***

  - Définition des facteurs qualité

- ***4. Chiffrage***

  - Estimations globale et détaillée
  - Plannings global et détaillé

- ***Résultats obtenus :***

  - Dossier des spécifications fonctionnelles
  - Plan de développement logiciel

MCD, MEA… clarification sur la modélisation et les différents types de modèles

Définition

La modélisation est l'activité qui consiste à produire un modèle.

Un modèle est ce qui sert ou doit servir d'objet d'imitation pour faire ou reproduire quelque chose.

Principes des modèles MERISE

Les modèles MERISE (MCD, MLD, MCT) sont définis par les contraintes d'analyse qu'ils intègrent et non pas par les langages de représentation qu'ils utilisent.

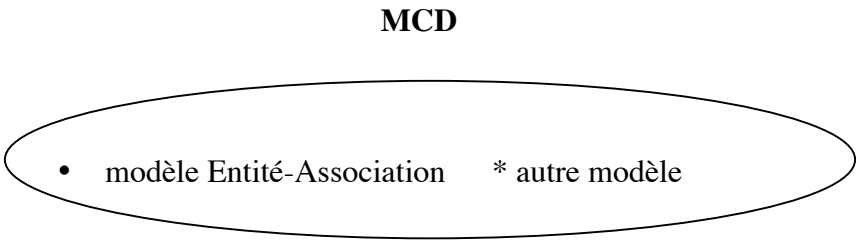
Les modèles MERISE sont des méthodes et non pas des langages (inversement, UML est un langage et non pas une méthode).

Un MCD peut être décrit avec n'importe quel langage de représentation. En général, on utilise le MEA comme langage, mais on pourrait utiliser le modèle relationnel, le graphe des dépendances fonctionnelles, le dictionnaire des attributs, etc.

MCD et modèle Entité-Association.

Le MCD, c'est l'ensemble des modèles qui intègrent les contraintes conceptuelles définies par MERISE (signification des informations sans contraintes techniques, organisationnelle ou économique). Parmi ces modèles, le plus couramment utilisé est le modèle Entité-Association.

Le MCD est donc une abstraction (un modèle abstrait), tandis que le modèle Entité-Association est un modèle concret. C'est une instance possible du MCD.

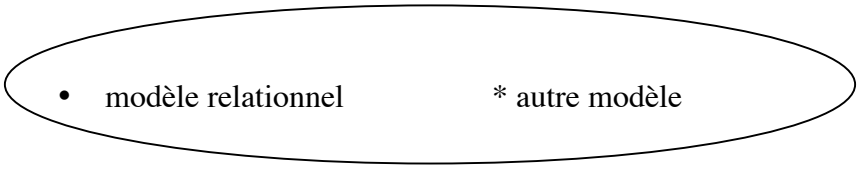


Toutefois, quand on parle du MCD, le plus souvent, on parle du modèle concret réalisé pour intégrer les contraintes conceptuelles définies par Merise (donc on parle d'un modèle Entité-Association).

MLD et modèle relationnel

La notion de MLD correspond à l'ensemble des modèles qui intègrent les contraintes organisationnelles et logiques définies par MERISE (description des données tenant compte de leurs conditions d'utilisation : techniques de mémorisation (un SGBD-R par exemple), contraintes d'intégrité, historique). Parmi ces modèles, le plus couramment utilisé est le modèle relationnel.

La notion de MLD est donc une abstraction (un modèle abstrait), tandis que le modèle relationnel est un modèle concret.



Toutefois, quand on parle du MLD, le plus souvent, on parle du modèle concret réalisé pour intégrer les contraintes organisationnelles et logiques définies par Merise (donc on parle d'un modèle relationnel).

MPD et MySQL

Le MPD c'est le modèle des données réalisé (d'où l'adjectif « physique »). Il est donc réalisé avec une technologie particulière : par exemple MySQL, ou ORACLE, ou un système de fichiers, ou un système de fiches cartonnées !

Il se traduit concrètement par du code MySQL par exemple.

A ce niveau, MERISE prend en compte les contraintes d'optimisation : choix des index, restructuration des tables et gestion par trigger, ajouts d'attributs calculés et gestion par trigger.

Etapes détaillées de la construction d'un modèle des données selon la méthode MERISE

La construction des modèles dans la méthode MERISE se fait selon l'ordre des 7 étapes présentées dans le schéma suivant :

N° étape	Nom du modèle MERISE	Nom du formalisme ou du langage utilisé	Explication du modèle MERISE
1	MCD	MEA	
2	MOD(s)	MEA	
3	MLD brut	MR	C'est la traduction du MEA en MR
4	MLD brut valorisé	MR	C'est l'ajout de toutes les contraintes d'intégrité : dictionnaire des attributs. Triggers.
5	MLD optimisé	MR	C'est la modification du MLD brut en prenant en compte des contraintes d'optimisation. C'est le niveau de la dénormalisation. Triggers.
6	MPD brut	SQL	C'est la traduction du MLD optimisé dans le langage du SGBD utilisé.
7	MPD optimisé	SQL	C'est la modification du MPD brut en prenant en compte des contraintes d'optimisation propres au SGBD utilisé. Indexation.

Source : Ingéniérie des systèmes d'information : MERISE – Nanci, Espinasse.

9. MERISE aujourd'hui

Type d'offre d'emploi

Consultant AMOA junior (AMOA : Assistance à la Maîtrise d'Ouvrage).

- ***Profil***

  Vous avez de solides aptitudes à l'écoute active et à la communication, ainsi que de bonnes capacités d'analyse et beaucoup de rigueur.

  Vous possédez un excellent niveau d'expression, tant écrite qu'orale, et vous savez traiter des problématiques fonctionnelles ou techniques.

  De formation informatique ou scientifique Bac+5, doublée d'une bonne culture technologique, vous pratiquez la modélisation UML ou Merise et pouvez justifier d'expériences en maîtrise d'œuvre.

10. Questions de cours

1. Quelles différences faites-vous entre le génie logiciel et l'ingénierie des systèmes d'information ?
2. Quelle différence faites-vous entre méthode et méthodologie.
3. Quel est le principe de la méthode analytique ? Quel est l'auteur de référence pour la méthode analytique ?
4. Quel est le principe de la méthode systémique ?
5. Faites le schéma du cycle en V avec ses étapes classiques. Montrez les relations horizontales.
6. Qu'est-ce que la recette ?
7. Quelle relation y a-t-il entre l'analyse fonctionnelle et la recette ?
8. Faites le schéma théorique d'un système d'information.
9. Faites le schéma des échanges entre le monde extérieur, le système entreprise et le système logiciel.
10. Qu'est-ce que le SII ? Quelles relations y a-t-il entre le SII et le SIO ?
11. Donnez le nom des trois cycles de la démarche MERISE. Définissez-les en 3 lignes.
12. Donnez le nom des 4 niveaux du cycle d'abstraction. Pour chaque niveau dîtes à quelle(s) question(s) il répond.
13. Faites le schéma complet du cycle d'abstraction en précisant ce que signifie chaque élément.
14. Quelle relation y a-t-il entre le MCD et le MEA ?
15. Quelle relation y a-t-il entre le MLD et le Modèle Relationnel.