Chapitre 1 : introduction

L'expérience, se dit le concierge, on en parle toujours comme si elle servait à quelque chose.

Jean Sareil, *Le pipelet n'a pas pipé*

Cet ouvrage a pour objectif la compréhension de Merise, méthode d'informatisation universelle. Toute méthode informatique doit répondre à quatre objectifs principaux :

- définir ce que l'utilisateur final veut informatiser (quitte à lui faire comprendre ce qu'il veut), et sa faisabilité ;
- vérifier la cohérence de sa demande ;
- structurer les données à informatiser. Cela est primordial en informatique de gestion ;
- rester simple. Ce point est largement battu en brèche quand une méthode est appliquée "pour la méthode". Elle doit rester un outil d'aide à la conception ou à la réalisation.

Merise sert de langage de référence entre les différents acteurs, informaticiens et utilisateurs. Elle représente, sous forme de représentations graphiques appelées modèles, les différents concepts manipulés. Merise possède des modèles spécifiques. Les exemples sont choisis dans cet ouvrage afin de mettre en évidence l'apport spécifique de chaque modèle. Les méthodes de conduite de projet, découpe du projet en tâches ou en sous-projets, estimation des charges, affectation des ressources aux tâches, ne sont pas abordées ici.

1 Rappel sur Merise

Merise répond aux critères précédents à sa manière :

1.1 Définir ce que l'utilisateur final désire

Des étapes de validation jalonnent le travail effectué. L'utilisateur est contraint de valider un "niveau" avant de passer au suivant. Par exemple, les enchaînements d'écrans de saisie de ristournes ou de promotion consommateur ne seront pas dessinés si des définitions ne sont pas données ou ne sont pas claires pour le concepteur et l'utilisateur. Des étapes sur lesquelles il est possible de revenir ont été créées.

Les étapes retenues dans Merise correspondent aux trois niveaux suivants :

- un niveau indépendant de l'organisation, fonctionnel, et appelé *conceptuel* ;
- un niveau indépendant de l'informatique, l'*organisationnel,* et ;
- l'*informatique*. Ce niveau est découpé en deux "sous-niveaux" *logique* et *physique*. Le "sous-niveau" logique, indépendant du matériel, peut encore être découpé en spécification *externe*, visible par l'utilisateur, et spécification *interne,* ou invisible à l'utilisateur. Il est inutile de faire valider ou approuver la spécification interne à l'utilisateur final.

Niveaux	
CONCEPTION	
ORGANISATION	
INFORMATIQUE Logique Physique	

D'autres étapes auraient pu être choisies. Certains niveaux, en particulier le niveau organisationnel, n'existent pas dans des méthodes anglo-saxonnes telles que Ssadm ou Sadt.

1.2 Vérifier la cohérence de sa demande

Le domaine à informatiser est abordé par trois côtés ou approches : communication, traitement et données. La validation permet de vérifier la cohérence de ces modèles entre eux.

Communication, traitement et données.

Dans tout projet impliquant un dialogue ou un découpage nécessaire des projets (construction d'usine avec un découpage génie civil, électricité, instrumentation, informatique, tuyauterie...), les quiproquos viennent d'une définition insuffisante des fonctions couvertes par chaque métier. C'est pourquoi, avant de démarrer un projet, il est fondamental de fixer les limites de ce projet et de définir ses liens avec les autres projets. A chaque projet est rattaché un domaine de l'entreprise. Les liens entre projets sont représentés par les échanges entre domaines fonctionnels. La découpe de l'entreprise et les échanges entre systèmes internes ou externes à l'entreprise sont représentés dans les modèles de *communication.*

La deuxième approche qui vient naturellement à l'esprit quand il s'agit d'informatique est la description des *traitements* : "Que provoquent ou comment sont générés ces messages ou ces échanges d'information ?"

Enfin, vient la structuration des *données,* sur laquelle nous reviendrons au point trois.

Vérification de la cohérence entre les modèles de communication, données et traitements.

Une première validation, décrite dans tous les manuels concernant Merise, doit être effectuée entre données et traitements. Toute donnée ou information est utilisée dans un traitement et tout traitement peut accéder aux données nécessaires.

Toute méthode accordant une importance privilégiée et justifiée aux données, telle que Niam ou Merise, doit garder son objectif de vérifier la faisabilité de la demande utilisateur en croisant ses besoins, exprimés sous forme de données, et ses besoins de traitement. *Les données sont au service des traitements*.

Une deuxième validation, intervenant avant la validation entre les données et les traitements, est la validation entre données et communication. Cette validation est plus facile et suppose que les modèles de communication ont été effectués : ne pas modéliser des données de lieu de livraison quand les messages contiennent des données de publicité consommateur ou de marketing.

Approche	Communication	Données	Traitement	Vérification cohérence

1.3 Les modèles de Merise

La combinaison des 4 niveaux et des 3 approches donne lieu à la "création" de 12 modèles de référence. Par exemple, le croisement du niveau conception et de l'approche données crée le MCD, ou modèle conceptuel de données.

	Communication	Données	Traitement
Conception	**MCC**	**MCD**	**MCT**
Organisation	MOC	**MOD**	**MOT**
Informatique Logique Physique	 **MLC** MPC	 MLD **MPD**	 **MLT** MPT

Certains modèles ne seront pas abordés dans cet ouvrage.

- Le modèle logique de données ou MLD, indépendant du système de gestion de base de données ou SGBD, n'est pas traité. La transformation entre les modèles entité relation (MCD ou MOD) et les modèles physiques relationnel et réseau est directe. Ceux-ci sont considérés comme logiques par les administrateurs de base de données. Certains appellent modèles logiques de données les modèles dépendant du SGBD, traités ici comme physiques.
- Le modèle organisationnel de communication ou MOC, traite les messages échangés entre sites différents : demande de présentation, demande de lancement de programme, mise à jour ou interrogation de données à distance. Ce

domaine en pleine évolution n'est pas stable actuellement (architecture client serveur). Aucun exercice ne traite cet aspect.

- Les modèles physiques de communication et de traitement ne sont pas décrits car l'ouvrage ne traite pas de programmation.

1.4 Structurer les données

La construction des représentations graphiques des structures de données, appelés modèles de données, est couverte par la plupart des méthodes actuelles : Merise, Niam, modèles de Chen, Normalisation de tables relationnelles. Cela entraîne un sens de l'abstraction (inné ou acquis ?) non négligeable. *Une bonne définition des modèles de données est indispensable.* Certaines méthodes, comme les méthodes anglo-saxonnes, sont plus orientées vers la gestion de projet. Une représentation des données plus compréhensible par l'utilisateur et non couverte par les méthodes de conception est la *construction d'un jeu d'essai*.

Merise formalise des ensembles de données, "client", "produit", "animal", dont les occurrences sont "sympathique", "orgueilleux", "nouveauté", "commode", "avide", "sécurité" ou "pomme", "tomate" ou "hérisson", "taureau" ou "chat", par exemple. L'application finale créera "M. Sécurité", "une pomme" et "un chat", les occurrences des concepts manipulés par Merise, "client", "produit" et "animal". Il est difficile de modéliser les ensembles d'occurrences et les occurrences elles-mêmes. Merise manipule les ensembles d'occurrences, le jeu d'essai manipule les ensembles et les occurrences.

Construire un jeu d'essai est primordial. Il permet à l'utilisateur de préciser sa demande et au concepteur de construire le modèle de données si l'utilisateur ne sait pas interpréter les modèles et les dessins de ses enfants. C'est pourquoi ce livre comprend un exercice de construction de jeu d'essai. Celui-ci se situe après la modélisation des données. Un jeu d'essai permet aussi la fourniture d'un jeu de test pour la réception des programmes ou la sélection d'un progiciel.

1.5 Rester simple.

Modifier une application existante revient 100 fois plus cher que de la concevoir correctement dès son origine. Malheureusement, il est difficile de rester simple quand tout s'agite autour de vous, et l'application "naturelle" de Merise peut laisser croire à une méthode complexe. Vous verrez par la pratique qu'en gardant à l'esprit ce souci de simplicité, vous aurez le plaisir d'avancer sans remettre en question les étapes précédentes. Cette simplicité va de pair avec la maîtrise du sujet de l'utilisateur final.

2 Etapes d'une étude informatique.

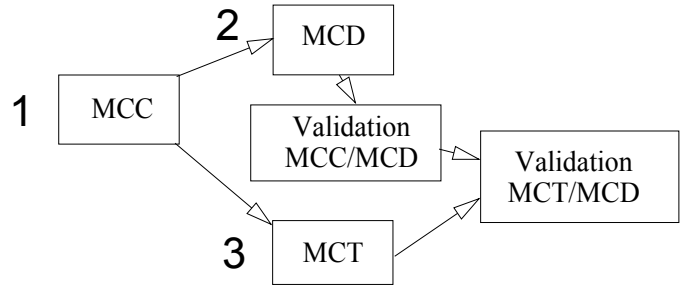
Les étapes principales d'une étude préalable sont, pour le niveau conceptuel :

- construction du modèle conceptuel de communication, domaines, partenaires et messages ;

- construction du modèle conceptuel de données ;
- validation des modèles de communication et de données ;
- construction du modèle conceptuel de traitement ;
- validation des modèles conceptuels de données et de traitement.

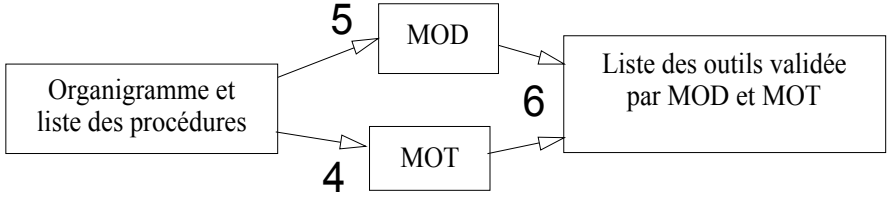
Après le MCC, le MCD ou le MCT peuvent être construits.

Les étapes de validation sont transparentes dans la correction des exercices ne traitant pas de cette validation.



Les étapes du niveau organisationnel et de définition des outils sont :

- construction de l'organigramme et de la liste des MOT ou procédures ;
- construction du ou des modèles organisationnels de données ;
- construction des modèles organisationnels de traitement ou procédures ;
- construction de la liste des outils validée par les modèles de données et de traitements.

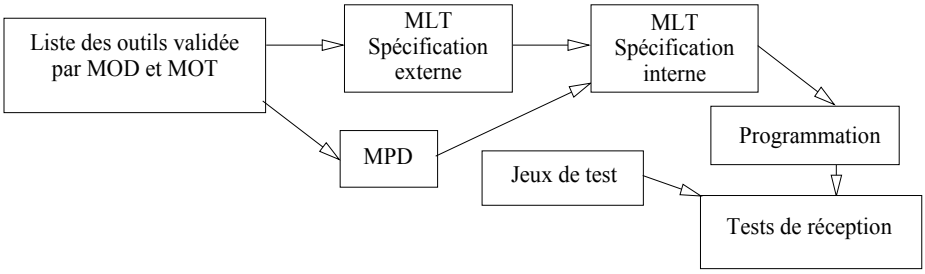


Ce plan sera repris dans le corrigé des exercices complets :

- construction du modèle conceptuel de communication : domaines, partenaires et messages ;
- construction du modèle conceptuel de données ;
- construction du modèle conceptuel de traitement ;
- construction des modèles organisationnels de traitement ou procédures ;
- construction du ou des modèles organisationnels de données ;
- construction de la liste des outils validée par les modèles de données et de traitement.

A la fin de l'étude préalable, les étapes sont les suivantes :

- spécification externe : construction des enchaînements d'écrans et description des champs des écrans ou MLT, et validation par l'utilisateur final ;
- construction des modèles de données dépendant du SGBD choisi ;
- description des actions des écrans sur la base de donnés : spécification interne ;
- construction du jeu de test ;
- programmation : modèles physiques des traitements ;
- tests de réception appelés *recettes* en informatique.



Quatre exercices sont consacrés à la construction du modèle physique de données relationnel et réseau. Un exemple de MLT est donné lors du corrigé du premier exercice complet. La construction du jeu de test est identique à la construction du jeu d'essai.

3 Plan de l'ouvrage.

Le plan de l'ouvrage reprend les fonctions essentielles à couvrir par Merise. Il est divisé en trois parties et trois annexes.

Première partie : rappels et objectifs

Chapitre 1 Introduction

Chapitre 2 Rappel sur les modèles

Deuxième partie : énoncés des exercices

Chapitre 3 Questionnaire à choix multiples permettant de vérifier ses connaissances.

Chapitre 4 Définir le sujet de l'étude (MCC, domaine). Trois cas sont pris : le cas d'une école, de services généraux et d'une bibliothèque.

Chapitre 5 Formaliser les données : exercices de plus en plus riches.

Chapitre 6 Construire un jeu d'essai qui se rapporte au cas des services généraux.

Chapitre 7 Valider les données et les traitements qui se rapportent au cas bibliothèque.

Chapitre 8 Définir l'organisation influe sur la solution informatique (utilisation de la table des coûts en annexe).

Chapitre 9 Comprendre les différentes manières de construire les outils informatiques (données ou traitements) qui se rapportent au cas bibliothèque.

Chapitre 10 Planifier les spécifications et la réalisation informatique à l'aide de la liste des outils se rapportant au cas bibliothèque.

Chapitre 11 Appliquer les règles de passages du modèle entité/relation aux modèles relationnel et navigationnel.

Chapitre 12 Appliquer le tout à quelques exemples complets, dont les trois cas du chapitre 4.

Troisième partie : corrigés des exercices

Les chapitres 13 à 22 comprennent les corrigés. Il n'existe pas de corrigé universel. Votre solution peut être bonne. Dans ce cas, envoyez-la-moi !

Annexes : Trois annexes décrivent les coûts de différentes architectures micro-ordinateurs à gros systèmes, un glossaire, la légende des symboles utilisés et les références des ouvrages cités.