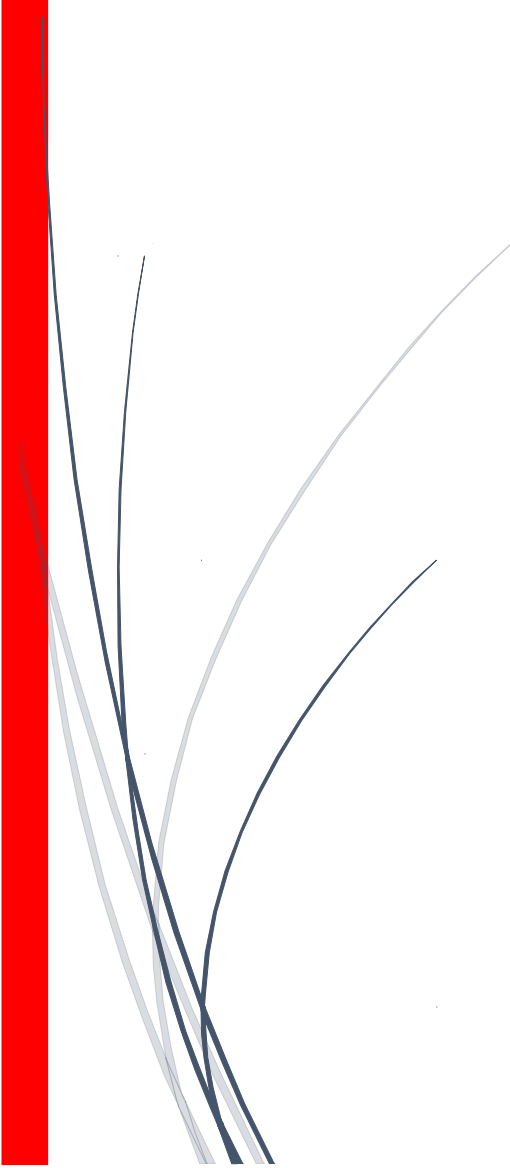


10/14/2020

CER UE 1.5

MODELISATION



SOMMAIRE

I. ANNALYSE DU CONTEXTE

1 – Mots clés

2 – Contexte

3 – Besoins

II. DEFINITION DE LA PROBLÉMATIQUE

III. DEFINITION DE LA CONTRAINTE

IV. PLANS D'ACTION

V. RÉALISATION DU PLAN D'ACTION

VI. VALIDATION DES HYPOTHESES

VII. CONCLUSION ET RETOUR SUR LES OBJECTIFS

VIII. BILAN CRITIQUE DU TRAVAILLE EFFECTUÉ

IX. RÉFÉRENCES DES MÉTHODES ET OUTILES UTILISÉ

X. RÉFÉRENCES BIBLIOGRAPHIQUE FOURNIES DANS LE PROSIT

I – ANALYSE DU CONTEXTE

1- Mots clés

- Association, Normalisation, Modélisation, Dependence.

1- Contexte :

Giselle est maintenant capable d'implémenter des requêtes SQL et de créer des bases de données. Cependant, elle constate que ces dernières sont mal structurées. Elle voudrait avoir une meilleure structuration.

2- Besoin :

- Besoin de trouver le moyen de mieux structurer ses données.
- Besoin d'établir les dépendances entre les données.

II- DÉFINITION DE LA PROBLEMATIQUE

Comment mieux organiser les données en s'appuyant sur les dépendances ?

III- DÉFINITION DES CONTRAINTES

➤ Données existantes.

IV- Généralisation

Structuration de données

V- PLAN D'ACTION

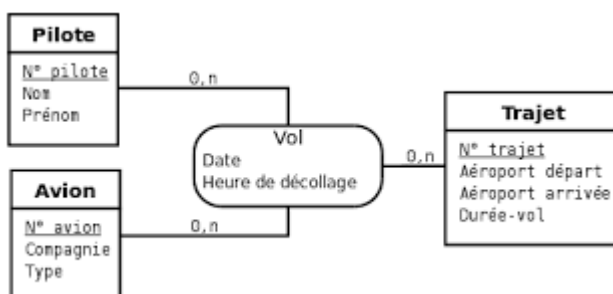
1. Définir des mots clés
2. Étudier les bases de merise
3. Établir le dictionnaire de données
4. Étudier et établir les dépendances des bases de données
5. Normalise les données

VI- RÉALISATION DU PLAN D'ACTION

1- DÉFINITION DES MOTS CLÉS

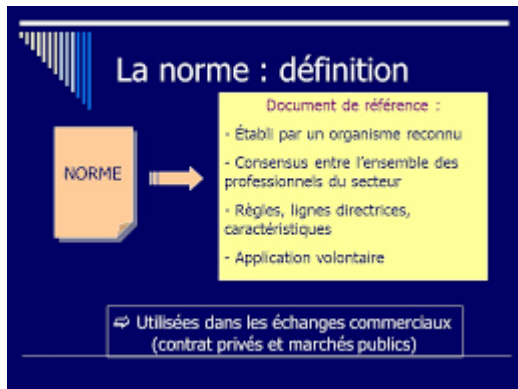
Association :

En conception de **base de données**, on dit qu'une **association** précise et explicite le lien entre plusieurs entités. Une **association** se fait entre 2 entités minimum mais il n'y a pas de maximum.



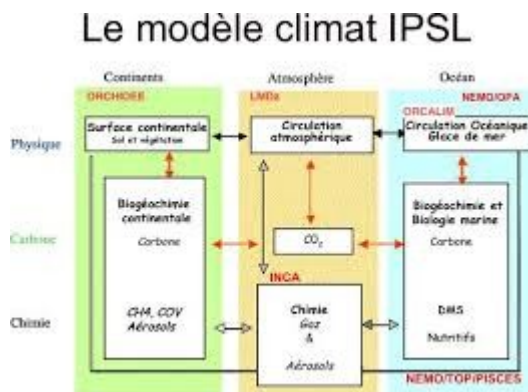
Normalisation:

Action qui consiste à édicter des normes ; fait d'être soumis à des normes. Ensemble des techniques qui ont pour objet de définir les produits et/ou les méthodes de fabrication aptes à satisfaire des besoins spécifiés ; standardisation.



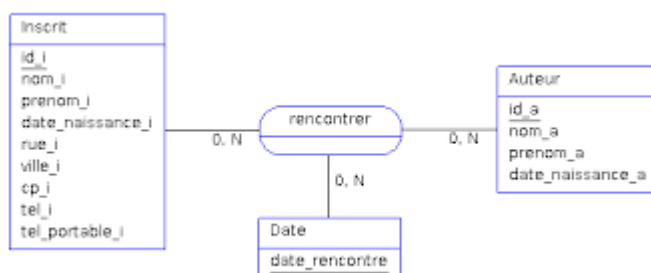
Modélisation:

Dans la conception d'un système d'information, la **modélisation** des **données** est l'analyse et la conception de l'information contenue dans le système afin de représenter la structure de ces informations et de structurer le stockage et les traitements informatiques.



Dépendance :

Dans l'étude des **bases de données**, les dépendances fonctionnelles sont les liens existants entre deux propriétés présentes dans la **base de données**. Exemple : Il existe une **dépendance** fonctionnelle entre le numéro d'un candidat et son nom et prénom.



2- Étudier les bases de merise

Merise est une méthode d'analyse, de conception et de gestion de projet informatique.

Merise a été très utilisée dans les années 1970 et 1980 pour l'informatisation massive des organisations. Cette méthode reste adaptée pour la gestion des projets internes aux organisations, se limitant à un domaine précis. Elle est en revanche moins adaptée aux projets transverses aux organisations, qui gèrent le plus souvent des informations à caractère sociétal (environnemental et social) avec des parties prenantes.

Historique :

Issue de l'[analyse systémique](#), la méthode **Merise** est le résultat des travaux menés par René Colletti, Arnold Rochfeld et Hubert Tardieu dans les [années 1970](#) et qui s'inséraient dans le cadre d'une réflexion internationale, autour notamment du [modèle relationnel](#) d'[Edgar Frank Codd](#). Elle est devenue un projet opérationnel au début des [années 1980](#) à la demande du ministère de l'industrie, et a surtout été utilisée en [France](#), par les [SSI](#) de ses membres fondateurs ([Sema-Metra](#), ainsi que par la [CGI Informatique](#)) et principalement pour les projets d'envergure, notamment des grandes administrations publiques ou privées.

Merise, méthode spécifiquement française, a d'emblée connu la [concurrence](#) internationale de méthodes anglo-saxonnes telles que [SSADM \(en\)](#), SDM/S ou Axial. Elle a ensuite cherché à s'adapter aux évolutions rapides des technologies de l'informatique avec Merise/[objet](#), puis Merise/2 destinée à s'adapter au [client-serveur](#). Merise était un courant majeur des réflexions sur une « Euro Méthode » qui n'a pas réussi à percer.

Dans le livre de référence présentant la méthode *Merise*, la préface rédigée par Jacques Lesourne introduisait une analogie avec le [merisier](#) « qui ne peut porter de beaux fruits que si on lui greffe une branche de cerisier : ainsi en va-t-il des méthodes informatiques bien conçues, qui ne produisent de bons résultats que si la greffe sur l'organisation réussit », même si beaucoup de gens ont voulu y voir un [acronyme](#) comme Méthode d'Étude et de Réalisation Informatique par les Sous-Ensembles ou pour les Systèmes d'Entreprises, l'acronyme le plus répandu parmi les mauvaises langues étant Méthode Éprouvée pour Retarder Indéfiniment la Sortie des Études.

Définition générale du système (Merise 2)



Synoptique général de la méthode MERISE 2 pour la conception du SI

Dans Merise deuxième génération, la définition générale du système comprend les étapes suivantes¹ :

1. Objectifs de la définition générale du système ;
2. Les phases de la définition générale du système ;
3. Phase d'appréciation du projet ;
4. Positionnement du projet ;
5. Délimitation des fonctions à informatiser ;
6. Évaluation des enjeux ;
7. Phase de spécifications générales ;
8. Modélisation conceptuelle des données ;
9. Macro-modélisation organisationnelle des traitements ;
10. Rédaction du dossier de spécifications générales.

Positionnement de la méthode

La méthode Merise est une méthode d'analyse, de conception et de réalisation de systèmes d'informations.

En amont, elle se situait dans le prolongement naturel d'un schéma directeur, souvent conduit suivant la méthode RACINES, très présente notamment dans le secteur public.

Les projets Merise étaient généralement des projets de grande ampleur de refonte d'un existant complexe, dans un environnement grand système. La méthode a aussi connu des tentatives d'adaptation avec les SGBD relationnels, les différentes interfaces homme-machine IHM, l'Orienté objet, le développement micro, les outils CASE, la rétro-ingénierie... mais qui n'ont pas connu le même succès.

La méthode est essentiellement française. Elle a des équivalents à l'étranger en ce qui concerne les modèles de données (avec des différences, par exemple les cardinalités ne sont pas aussi détaillées dans les modèles anglosaxons). En revanche la modélisation des traitements est beaucoup plus complexe que dans les méthodes anglo-saxonnes.

Sa mise en œuvre peut paraître lourde. On consacre beaucoup de temps à concevoir et à pré-documenter avant de commencer à coder, ce qui pouvait sembler nécessaire à une époque où les moyens informatiques n'étaient pas aussi diffusés qu'aujourd'hui. Cela dit, elle évite l'écueil inverse du développement micro, qui souffre du manque de documentation, et où les erreurs sont finalement très coûteuses à réparer *a posteriori*.

Même si les échanges et la consultation entre concepteurs et utilisateurs sont formellement organisés, on a aussi reproché à Merise d'utiliser un formalisme jugé complexe (surtout pour les modèles de données), qu'il faut d'abord apprendre à manier, mais qui constitue ensuite un véritable langage commun, puissant et rigoureux pour qui le maîtrise.

L'articulation très codifiée et bien balisée des différentes étapes, avec un descriptif très précis des résultats attendus est ce qui reste aujourd'hui de mieux connu et de plus utilisé.

Méthode d'analyse et de conception



MERISE est une méthode d'analyse et de conception des systèmes d'information basée sur le principe de la séparation des données et des traitements. Elle possède plusieurs modèles qui sont répartis sur 3 niveaux (Le niveau conceptuel, le niveau logique ou organisationnel, le niveau physique)².

La méthode Merise d'analyse et de conception propose une démarche articulée simultanément selon 3 axes pour *hiérarchiser* les préoccupations et les questions auxquelles répondre lors de la conduite d'un projet :

- **Cycle de vie** : phases de conception, de réalisation, de maintenance puis nouveau cycle de projet.
- **Cycle de décision** : des grands choix (GO-NO GO : Étude préalable), la définition du projet (étude détaillée) jusqu'aux petites décisions des détails de la réalisation et de la mise en œuvre du système d'information. Chaque étape est documentée et marquée par une prise de décision.
- **Cycle d'abstraction** : niveaux conceptuels, d'organisation, logique et physique/opérationnel (du plus abstrait au plus concret) L'objectif du *cycle d'abstraction* est de prendre d'abord les grandes décisions métier, pour les principales activités (Conceptuel) sans rentrer dans le détail de questions d'ordre de l'organisation ou technique.

La méthode Merise, très analytique (attention méthode systémique), distingue nettement les données et les traitements, même si les interactions entre les deux sont profondes et s'enrichissent mutuellement (validation des données par les traitements et réciproquement). Certains auteurs (Merise/méga, puis Merise/2) ont également apporté la notion complémentaire de communications, vues au sens des messages échangés. Aujourd'hui, avec les SGBD-R, l'objet, les notions de données et de traitements sont de plus en plus imbriquées.

« Courbe du soleil »

La littérature parle de « courbe du soleil », établissant une analogie entre la démarche Merise et le lever puis le coucher du soleil : de même, le projet doit élaborer une analyse critique de l'existant (en partant du niveau physique et en s'élevant jusqu'au conceptuel : démarche [Bottom-up](#), phase ascendante de la courbe), puis décliner la solution retenue (en partant du niveau conceptuel et revenant au niveau physique : démarche [top-down](#), phase descendante de la courbe).

Le recensement de l'existant est très décrié de nos jours, car il augmente la durée du projet et inciterait à reconduire les solutions existantes. Il semble néanmoins assez rationnel de commencer par un bilan du passé. Sur ce point, la démarche Merise est à l'opposé des méthodes itératives de type [RAD](#), ou de l'adoption systématique des bonnes pratiques observées dans d'autres entreprises du secteur, qui constituent une démarche typique dans l'implémentation de progiciels. Merise/Méga insiste aussi beaucoup plus sur l'analyse de l'état de l'art pour chercher des solutions innovantes.

Niveau conceptuel

L'étude *conceptuelle* Merise s'attache aux *invariants* de l'entreprise ou de l'organisme du point de vue du métier : quelles sont les activités, les métiers gérés par l'entreprise, quels sont les grands [processus traités](#), de quoi parle-t-on en matière de données, quelles notions manipule-t-on ?... et ce indépendamment des choix techniques (comment fait-on ?) ou d'organisation (qui fait quoi ?) qui ne seront abordés que dans les niveaux suivants.

Au niveau conceptuel on veut décrire, après *abstraction*, le modèle (le *système*) de l'entreprise ou de l'organisme :

- le **Modèle conceptuel des données** (ou MCD), schéma représentant la structure du [système d'information](#), du point de vue des données, c'est-à-dire les dépendances ou relations entre les différentes données du [système d'information](#) (par exemple : le client, la commande, les produits, etc.),
- et le **Modèle conceptuel des traitements** (ou MCT), schéma représentant les traitements, en réponse aux événements à traiter (par exemple : la prise en compte de la commande d'un client).

Dans l'idéal, le MCD et le MCT d'une entreprise sont stables, à périmètre fonctionnel constant, et tant que le métier de l'entreprise ne varie pas. La modélisation ne dépend pas du choix

d'un [progiciel](#) ou d'un autre, d'une automatisation ou non des tâches à effectuer, d'une organisation ou d'une autre, etc.

MCD : modèle conceptuel des données[\[modifier\]](#) | [modifier le code](#)

Article détaillé : [Modèle conceptuel des données](#).

Le MCD repose sur les notions d'entité et d'association et sur les notions de relations. Le modèle conceptuel des données s'intéresse à décrire la *sémantique* du domaine (*entity/relationship* en anglais)

Entité ou objet[\[modifier\]](#) | [modifier le code](#)

L'entité est définie comme un *objet de gestion* considéré d'intérêt pour représenter l'activité à modéliser (exemple : entité pays). À son tour, chaque entité (ou **objet**) est porteuse d'une ou plusieurs **propriétés** simples, dites atomiques (exemples : code, nom, capitale, population, superficie) dont l'une, unique et discriminante, est désignée comme **identifiant** (exemple : code).

L'entité représente le concept qui se décline, dans le concret, en occurrences d'individus.

Par exemple :

- (FR, France, Paris, 60,4 millions d'hab., 550 000 km²), et
- (DE, Allemagne, Berlin, 82 537 000 hab., 357 027 km²),

sont deux occurrences de l'entité « pays » et sont constituées de n-uplets de propriétés, que le code FR ou DE suffit à identifier sans risque de doublon.

Par construction, le MCD impose que toutes les propriétés d'une entité ont vocation à être renseignées (il n'y a pas de propriété « facultative »).

Le MCD doit, de préférence, ne contenir que le cœur des informations strictement nécessaires pour réaliser les traitements conceptuels (cf. MCT). Les informations calculées (ex : montant taxes comprises d'une facture), déductibles (ex : densité démographique = population / superficie) et *a fortiori* celles liées aux choix d'organisation conçus pour effectuer les traitements (cf. MOT) ne doivent pas y figurer.

Association ou relation

L'association est un lien sémantique entre entités :

- 1 entité reliée à elle-même : la relation est dite **réflexive**,
- 2 entités : la relation est dite **binaire** (ex : une usine 'est implantée' dans un pays),
- Plus rarement 3 ou plus : **ternaire**, voire de dimension supérieure. En fait, hormis le cas d'une date (la table date disparaît) si une relation à 3 points d'attache ou plus, on peut réécrire la relation en transformant la relation en table et en transformant les liens en relations.

Une association peut également être porteuse d'une ou plusieurs propriétés (ex : 'date d'implantation' d'une usine dans un pays)

Cette description sémantique est enrichie par la notion de **cardinalité**, celle-ci indique le *nombre minimum* (0 ou 1) et *maximum* (1 ou n) de fois où une occurrence quelconque d'une entité peut participer à une association (ex : une usine est implantée dans un (card. min=1) et un seul (card. max=1) pays; et réciproquement un pays peut faire l'objet soit d'aucune (card. min=0) implantation d'usine soit de plusieurs (card. max=n). On a donc les combinaisons suivantes

- 0,1 \Rightarrow NULL, les clés de l'entité migrent
- 1,1 \Rightarrow NOT NULL, les clés de l'entité migrent
- (1,1) \Rightarrow NOT NULL PRIMARY KEY, les clés de l'entité migrent
- 0,n \Rightarrow NULL, les clés de l'entité ne migrent pas
- 1,n \Rightarrow NOT NULL, les clés de l'entité ne migrent pas

Il existe deux types d'associations : les **CIF (contrainte d'intégrité fonctionnelle)** et les **CIM (contrainte d'intégrité multiple)**. Les CIF ont pour particularité d'être binaires et d'avoir une cardinalité min à 0 ou 1 et une cardinalité max à 1 ou n, de plus elles ne sont pas porteuses de propriétés. Les CIM sont n-aires et ont toutes leurs cardinalités max à n, de plus elles peuvent être porteuses de propriétés. Les associations ne sont plus utilisés aujourd'hui avec l'avènement de la programmation MVC. Le Modèle n'est pas censé contrôler les données, travail dévolu au contrôleur.

MCT : modèle conceptuel des traitements

Article détaillé : [Modèle conceptuel des traitements](#).

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

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' à la suite de l'opération de 'préparation colis'), et à ce titre être, éventuellement, un événement déclencheur d'une autre opération.

Opération

Une opération se déclenche uniquement par le stimulus d'un ou de plusieurs événements 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 événements 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...)

Processus[[modifier](#) | [modifier le code](#)]

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).

Niveau logique ou d'organisation

À ce niveau de préoccupation, les modèles conceptuels sont précisés et font l'objet de choix d'organisation. On construit :

- un **Modèle Logique des Données** (ou MLD), qui reprend le contenu du MCD précédent, mais précise la volumétrie, la structure et l'organisation des [données](#) telles qu'elles pourront être implémentées. 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](#)
- un **Modèle Logique des Traitements** (ou MLT), qui précise les acteurs et les moyens qui seront mis en œuvre. C'est ici que les traitements sont découpés en [procédures fonctionnelles](#) (ou PF).

Comme son nom l'indique, l'étude d'organisation s'attache à préciser comment on organise les données de l'entreprise (MLD) et les tâches ou procédures (MLT). Pour autant, les choix techniques d'implémentation, tant pour les données (choix d'un [SGBD](#)) que pour les traitements ([logiciel](#), [progiciel](#)), ne seront effectués qu'au niveau suivant.

La façon dont seront conservés les [historiques](#) des [données](#) fait également partie de ce niveau de préoccupation.

MLD : modèle logique des données

(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 [1-1 ou 0-1]) par l'ajout d'une **clé étrangère** dans une table existante, soit (ex : card. max n [0-N ou 1-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)
```

De plus en plus, nous voyons arriver une nouvelle façon d'écrire le modèle relationnel :

```
PAYS(code_pays)
```

code_pays : clé primaire de la relation PAYS

```
USINE(id_usine, code_pays, date_implantation)
```

id_usine : clé primaire de la relation USINE

code_pays : clé étrangère de la relation USINE en référence à code_pays de la relation PAYS

```
EXPORT(id_usine, code_pays)
```

id_usine + code_pays : clé primaire de la relation EXPORT [concaténation de id_usine et de code_pays]

id_usine : clé étrangère de la relation EXPORT en référence à id_usine de la relation USINE

code_pays : clé étrangère de la relation EXPORT en référence à code_pays de la relation PAYS

Les opérateurs de l'algèbre relationnelle (projection, sélection, [jointure](#), opérateurs ensemblistes) peuvent ensuite directement s'appliquer sur le modèle relationnel ainsi obtenu et normalisé. (Voir [Langage de requête](#) et [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/reliations du MCD.

MLT : modèle logique des traitements

Le **MLT**, appelé aussi **MOT** pour « modèle organisationnel des traitements », décrit avec précision l'organisation à mettre en place pour réaliser une ou, le cas échéant, plusieurs opérations *figurant dans le MCT*. Il répond aux questions suivantes : qui ? quoi ? où ? quand ? À un MCT correspondent 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'utilisateur* du système.

Par exemple, on peut identifier les profils suivants : client-web, responsable commercial, responsable des stocks, etc.

La phase

La phase est un *ensemble d'actions* (cf. la notion d'opération pour le MCT) réalisées sur un *même* poste de travail.

La phase peut être :

- Soit manuelle : par exemple, la confection d'un colis ;
- Soit automatisée et interactive : par exemple, la saisie d'un formulaire client ;
- Soit automatisée et planifiée (on parle aussi de *batch*) : par exemple, la production et l'envoi quotidiens de tableaux de bord dans les boîtes aux lettres électroniques.

La tâche

La tâche est une description détaillée d'une *phase automatisée interactive*.

Par exemple, elle correspond à la spécification de l'interface et du dialogue humain-machine, à la localisation et la nature des contrôles à effectuer, etc.

La procédure

La procédure est un *regroupement de phases*. Elle équivaut sur le plan de l'organisation aux notions d'opérations et d'actions conceptuelles. La différence est que l'on considère ici ces dernières comme se déroulant sur une période de temps homogène.

Des procédures d'origines non conceptuelles peuvent être ajoutées du fait des choix d'organisation effectués.

Par exemple, on peut citer les procédures d'échanges d'informations liées à l'externalisation de certaines activités, la prise en compte des questions de sécurité en cas de choix de solution Web, etc.

Niveau physique

Les réponses apportées à ce dernier niveau permettent d'établir la manière concrète dont le système sera mis en place.

. le **Modèle physique des données** (ou MPD ou MPhD) permet de préciser les systèmes de stockage employés (implémentation du MLD dans le [SGBD](#) retenu)

. le **Modèle Opérationnel des Traitements** (ou MOT ou MOPT) permet de [spécifier](#) les fonctions telles qu'elles seront ensuite réalisées par le programmeur.

Différentes phases d'un projet Merise

Un projet élaboré selon la méthode Merise est composé de différentes phases :

- Les acteurs d'un projet : il s'agit ici d'identifier les acteurs d'un projet, les personnes intervenants dans une quelconque phase de celui-ci. Ces acteurs apparaîtront logiquement dans la modélisation des flux de données.
- [Schéma directeur](#) : « le schéma directeur définit le cadre organisationnel et informatique des futurs projets » ³, et donc doit définir le projet relativement aux objectifs de l'entreprise, sa stratégie. Il ne s'agira pas ici de donner les détails du projet, mais plutôt de fournir le cadre, les objectifs, et moyens du projet.
- L'[étude préalable](#) : elle décrit les [besoins](#) et les [attentes](#) des [utilisateurs](#), les [traitements](#) ([processus métier](#)) pour la procédure représentative (modèle conceptuel des traitements, modèle logique des traitements, ébauche de modèle physique des données), et les principales [données](#) ([modèle conceptuel des données](#), [modèle logique des données](#), ébauche de modèle physique externe des traitements),
- L'[étude détaillée](#) : elle décrit les besoins, traitements, et données de façon plus détaillée pour chaque [procédure fonctionnelle](#). L'étude détaillée se décompose elle-même en :
 - [Spécifications fonctionnelles générales](#) ([Tableau des opérations par processus](#), TOP), écrites par la [maîtrise d'ouvrage](#),

- [Spécifications fonctionnelles détaillées](#), écrites par la [maîtrise d'œuvre](#),
- L'[étude technique](#) : elle décrit les moyens techniques nécessaires à la réalisation de l'application (environnement technique, [SGBD](#), [langages informatiques](#), consignes de développement, ...).
- Production : elle décrit la mise en production.
- Maintenance : elle décrit la maintenance du système, et fournira donc au moins les éléments suivants :
 - o Les acteurs
 - o Les documentations
 - o Les formations

Méta-modèle de Merise pour les données

Types de « modèles » (méta-modèles) dits « Entity-Relationship »

[Peter Chen](#) présente la classification suivante des différents modèles « entité-association »⁴ :

- N-aires :
 - 1-Avec attributs pour entités et associations
 - 2-Avec attributs pour entités seulement
 - 3-Sans attribut
- Binaires :
 - 4-Avec attributs pour entités et associations
 - 5-Avec attributs pour entités seulement
- 51-Avec associations « plusieurs-à-plusieurs »
- 52-"Sans" association « plusieurs-à-plusieurs »
- 521-Associations non orientées
 - 522-Associations orientées (avec seulement un « parent », avec un « parent physique » et un « parent logique », avec des parents multiples, par exemple CODASYL)
 - 6-Sans attribut : "modèle binaire [ABR74], Entity Set Model, Modèle fonctionnel

Merise utilise un « modèle » avec entités, attributs (ou propriétés) et relations (ou associations)

En termes formels, on dira qu'un MCD est un invariant.

On y spécifie des ensembles, des relations dont on donne les propriétés (fonction (totale ou partielle), fonction injective, surjective, relation quelconque). On utilise pour cela des « cardinalités » (appelées en [UML](#), des multiplicités). Il y a 16 cas de relations.

En termes de mathématiques ensemblistes, un attribut est une fonction. Par exemple *date_de_naissance* est une fonction de l'ensemble *Personnes* vers l'ensemble *Dates*, *date_de_décès* est une fonction partielle de l'ensemble *Personnes* vers l'ensemble *Dates*.

Les fonctions (au sens mathématique) sont exprimées par ce qui est appelé "clé" (même sens que celui du "modèle relationnel n-aire" de [Codd](#)) et aussi par les "cardinalités" (0-1 pour les fonctions partielles) et (1-1 pour les fonctions totales). Quand la fonction a comme partie gauche un produit cartésien entre entités de types différents (entre plusieurs rectangles), on parle de CIF (Contrainte d'Intégrité Fonctionnelle).

1- Établir le dictionnaire de données

2- Étudier et établir les dépendances des bases de données

3- Normalise les données