

SOMMAIRE

1.	Introduction.....	4
2.	Le système d'information	4
2.1.	Qu'est-ce qu'un système ?	4
2.2.	Qu'est-ce qu'un système d'information ?	5
2.3.	Qu'est-ce qu'un système automatisé d'information ?	5
2.4.	Intérêt de définir un système d'information	5
3.	La méthode MERISE	6
3.1.	Une approche systémique	6
3.2.	Une démarche à trois axes	6
3.2.1.	Démarche par étapes	6
3.2.2.	Démarche par niveaux	7
3.2.3.	Cycle de décision	7
3.3.	L'étude et le recueil de l'existant	8
3.3.1.	Etudier l'organisation et découper en domaine d'activité	8
3.3.2.	Le recueil de l'existant	8
3.3.3.	Identifier les acteurs et les documents.....	8
3.4.	Présentation des niveaux de description et des modèles associés	8
3.4.1.	Le niveau conceptuel	9
3.4.1.1.	Le Modèle Conceptuel de Communication (MCC)	9
	• Acteur :	9
	• Flux :	9
	• Exemple de MCC :	9
	• Résumé du MCC :	10
3.4.1.2.	Le dictionnaire des données	11
3.4.1.3.	Le Modèle Conceptuel des Données (MCD)	11
	• Propriété (ou attribut) :	11
	• Entité (ou objet) :	12
	• Identifiant (ou propriété identifiante) :	12
	• Relation (ou association) :	13
	• Cardinalité :	14
	• Occurrence :	15
	• Dépendance fonctionnelle :	16
	• Contrainte d'intégrité fonctionnelle (CIF) :	17
	• Généralisation/spécialisation :	18
	• Le concept d'héritage :	19
	• Les contraintes d'extension sur les relations ou sur les entités :	19
	• La contrainte d'exclusion :	20
	• La contrainte d'inclusion :	20
	• La contrainte de totalité :	21
	• La contrainte de partition (ou de « ou exclusif ») :	22
	• La contrainte d'égalité (ou de simultanéité) :	22
	• Règles de vérification du MCD :	23
	• Résumé du MCD :	25
3.4.1.4.	Le Modèle Conceptuel des Traitements (MCT).....	26
	• Processus :	27
	• Evénement :	27
	• Opération :	27
	• Action :	28
	• Règle de gestion :	28
	• Synchronisation :	28
	• Les règles d'émission :	29
	• Le résultat (ou l'émission)	30
	• Comment élaborer le MCT	30
	• Vérification du MCT.....	30
	• Résumé du MCT :	31

• Exemple d'un MCT complet :.....	32
3.4.2. Le niveau organisationnel	32
3.4.2.1. Le Modèle Organisationnel des Données (MOD).....	33
• Choix d'informatisation des données.....	33
• Quantification et historisation des données.....	33
• Droits d'accès aux données.....	33
3.4.2.2. Le modèle Organisationnel des Traitements (MOT).....	33
• Procédure :	33
• Acteur :	34
• Evénement :	34
• Phase :	34
• Tâche :	34
• Règle d'organisation :	34
• Règle d'émission :	35
• Synchronisation :	35
• Résultat :	35
• Module :	35
• Correspondance entre les concepts du MCC, du MCT et du MOT	35
• Comment élaborer le MOT	36
• Exemple de MOT complet	36
• Résumé du MOT	36
3.4.3. Les niveaux logique et physique	38
3.4.3.1. Le Modèle Relationnel des Données (MRD)	39
• Table (ou relation).....	39
• Attribut.....	39
• Tuple	39
• Clé primaire	39
• Clé étrangère	40
• Exemple d'une table.....	40
• Correspondance entre les concepts du MCD et ceux du MLD	41
• Règles de passage d'un MCD à un MLD.....	41
• Normalisation du modèle relationnel	46
4. Positionnement de la méthode Merise	49
4.1. Positionnement dans le temps.....	49
4.2. Interaction des modèles de la méthode Merise	50
5. Conclusion	51
6. Bibliographie	52

Table des illustrations

Figure 1 : Le système d'information	5
Figure 2 : Les étapes d'un projet MERISE.....	6
Figure 3 : Exemple de MCC	10
Figure 4 : Une entité sans ses propriétés.....	12
Figure 5 : Une entité avec ses propriétés	12
Figure 6 : Exemple d'une entité	12
Figure 7 : Une entité avec un identifiant.....	13
Figure 8 : Des exemples d'identifiants	13
Figure 9 : Formalisme de la relation	13
Figure 10 : Relation binaire	13
Figure 11 : Relation porteuse de données	13
Figure 12 : Relation ternaire	14
Figure 13 : Relation réflexive	14
Figure 14 : Le lien identifiant de la commande et des lignes de commande.....	14
Figure 15 : Le lien identifiant de l'hôtel et de ses chambres.....	14
Figure 16 : Cardinalité 0,1	15
Figure 17 : Cardinalité 1,1	15
Figure 18 : Cardinalité 0,n	15
Figure 19 : Cardinalité 1,n	15

Figure 20 : Explication des cardinalités	15
Figure 21 : Les dépendances fonctionnelles	17
Figure 22 : MCD du contrat d'assurance.....	17
Figure 23 : MCD avec une CIF (Relation portant un nom)	18
Figure 24 : MCD avec une CIF (relation nommée CIF)	18
Figure 25 : Exemple du concept généralisation/spécialisation.	18
Figure 26 : Généralisation simple	19
Figure 27 : Généralisation multiple	19
Figure 28 : La contrainte d'exclusion.....	20
Figure 29 : La contrainte d'inclusion.....	21
Figure 30 : La contrainte de totalité.....	21
Figure 31 : La contrainte de partition.....	22
Figure 32 : La contrainte d'égalité	23
Figure 33 : Règle de validation n°1	23
Figure 34 : Règle n°5, résolution de la dépendance fonctionnelle transitive	24
Figure 35 : Modèle sans l'application de la règle n°6	25
Figure 36 : Application de la règle n°6	25
Figure 37 : Formalisme de l'événement	27
Figure 38 : Exemple d'événement.....	27
Figure 39 : Formalisme de l'opération	28
Figure 40 : Exemple d'opération.....	28
Figure 41 : Formalisme de l'action.....	28
Figure 42 : Exemple d'actions.....	28
Figure 43 : Formalisme de la synchronisation	29
Figure 44 : Exemple de synchronisation.....	29
Figure 45 : Formalisme des règles d'émission	29
Figure 46 : Exemple de règles d'émission.....	30
Figure 47 : Formalisme du résultat	30
Figure 48 : Exemple de résultats.....	30
Figure 49 : Un MCT complet.....	32
Figure 50 : Exemple de MOT complet	36
Figure 51 : Représentation graphique du MLD relationnel	40
Figure 52 : Un MCD avant de le Transformer en MLD	40
Figure 53 : Exemple de MLD	41
Figure 54 : Une relation binaire à cardinalités (0,n) ou (1,n) – (0,1) ou (1,1).....	41
Figure 55 : MLD issue d'un MCD à relation de type (0,n) ou (1,n) - (0,1) ou (1,1)	42
Figure 56 : Relation reflexive de type (0,n) ou (1,n) - (0,1) ou (1,1)	42
Figure 57 : MLD issu d'un MCD à relation reflexive de type (0,n) ou (1,n) - (0,1) ou (1,1)	42
Figure 58 : MCD à relation de type (0,n) ou (1,n) – (0,n) ou (1,n).....	42
Figure 59 : MCD issu d'un MCD à relation de type -(0,n) ou (1,n) - (0,n) ou (1,n)	43
Figure 60 : Relation reflexive de type (0,n) ou (1,n) - (0,n) ou (1,n)	43
Figure 61 : MLD issu d'un MCD à relation reflexive de type (0,n) ou (1,n) - (0,n) ou (1,n)	43
Figure 62 : MCD à relation n-aire.....	43
Figure 63 : MLD issu d'un MCD à relation n-aire	44
Figure 64 : Une relation binaire à cardinalités (1,1) - (0,1).....	44
Figure 65 : MLD issu d'un MCD à relation de type (1,1) - (0,1)	44
Figure 66 : Un héritage.	45
Figure 67 : Lien d'héritage avec généralisation.	45
Figure 68 : Lien d'héritage avec spécialisation.....	45
Figure 69 : Lien d'héritage mixte - Seul l'identifiant est hérité.	46
Figure 70 : Lien d'héritage mixte - Toutes les propriétés sont héritées.....	46
Figure 71 : La table "Commande" avant normalisation en 1FN	47
Figure 72 : La première forme normale	47
Figure 73 : La table "Ligne" avant normalisation en 2FN	47
Figure 74 : La deuxième forme normale.....	47
Figure 75 : La table "Commande" avant normalisation en 3FN	48
Figure 76 : La troisième forme normale	48
Figure 77 : La courbe du soleil	49
Figure 78 : Les modèles de la méthode Merise.....	50

1. Introduction

Merise est une méthode de conception et de développement des systèmes d'information conçue par un ensemble de sociétés de service, sous la direction du centre technique informatique du ministère de l'industrie. Cette méthode a été développée entre 1977 et 1978.

Son véritable démarrage s'est fait entre 1983 et 1985, et, depuis, la méthode Merise s'est imposée comme un standard de fait. Le nombre d'utilisateurs a aussi bien augmenté dans le secteur privé que public. Au début des années 90, Merise devenu Merise/2, a bénéficié d'un certain nombre d'évolutions.

Aujourd'hui, la majorité des systèmes d'information en services ont été développés avec la méthode Merise. En France, c'est actuellement la méthode la plus utilisée, même si, l'approche objet ; avec le langage de modélisation UML ; commence à être de plus en plus utilisée.

Cependant, tant que les systèmes de gestion de base de données objet n'auront pas remplacé les systèmes de gestion de base de données relationnelle (SGBDR), Merise restera la référence des méthodes de conception.

L'objectif de ce cours est de présenter la méthode de conception des systèmes d'information Merise, afin d'être capable de concevoir ou de participer à la conception d'un système d'information.

2. Le système d'information

2.1. Qu'est-ce qu'un système ?

Un système est un ensemble d'éléments matériels ou immatériels en interaction (hommes, machines, règles...) transformant par un processus des éléments en d'autres éléments.

Une entreprise qui fonctionne en vue de réaliser ses objectifs est une forme de système.

Il existe différents types de système :

Le système opérant : comparable à une boîte noire, il s'agit d'un système physique qui transforme un flux physique entrant (matière première, flux financier...) en un flux physique de sortie (produit, service...).

Le système de pilotage : son rôle est de réguler et contrôler le système opérant, et adapte son fonctionnement en fonction des objectifs prédéfinis.

Afin de faire l'interface entre le système opérant et le système de pilotage, il est nécessaire de mettre en place le système d'information.

2.2. Qu'est-ce qu'un système d'information ?

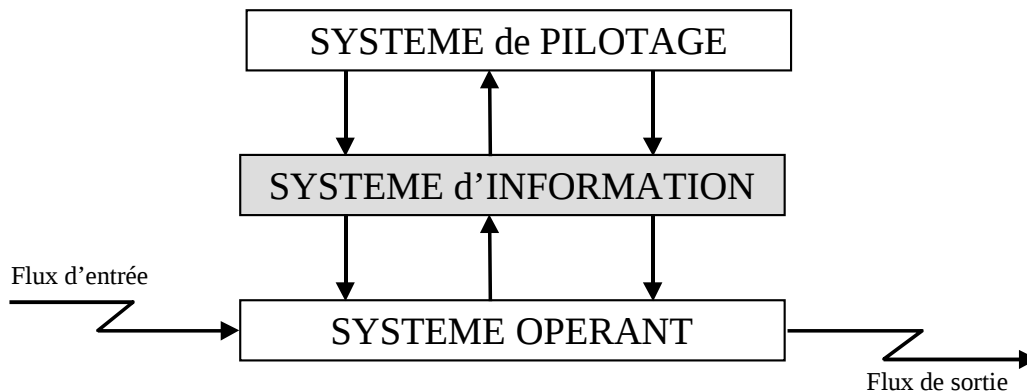


Figure 1 : Le système d'information

Le système d'information (SI) est composé d'éléments divers (personnels, systèmes informatiques, les règles de travail, le matériel...) chargés de stocker et de traiter les informations relatives au système opérationnel, afin de les mettre à la disposition du système de pilotage. Il peut recevoir des informations décisives du système de pilotage destinées à son propre pilotage. Il réagit sur le système opérationnel.

On peut comparer le système d'information à une « boîte noire » par laquelle transitent les principaux flux d'information entre le système de pilotage et le système opérant d'une part (règles de fonctionnement, ressources allouées, priorités d'exécution) et entre le système opérant et le système de pilotage d'autre part (variables de mesure de l'activité, de l'efficacité technique et commerciale, explicitant le nombre et la variété des résultats obtenus ainsi que la quantité de ressources consommées).

Le système d'information assure donc l'interface entre les systèmes opérant et de pilotage, mais il peut aussi assurer l'interface avec le pôle client et/ou le pôle fournisseur. Le système d'information n'est pas fermé sur une organisation interne.

2.3. Qu'est-ce qu'un système automatisé d'information ?

Le système automatisé d'information est un sous-ensemble du système d'information dans lequel toutes les opérations automatisables sont effectuées par des machines.

Un système automatisé d'information permet aussi bien de traiter et sauvegarder l'information de l'entreprise, de simplifier et améliorer le travail lorsque les tâches sont répétitives, de mettre à disposition des outils d'aide à la décision, etc.

Le système d'information est automatisable s'il est formalisable.

2.4. Intérêt de définir un système d'information

Auparavant, les systèmes d'information se constituaient au fur et à mesure des besoins. On agrégeait des fonctions les unes après les autres. On avait alors aucune vision globale du système. Puis, on a cherché à concevoir des systèmes qui intègrent l'ensemble des informations de l'entreprise. On s'est alors rendu compte de la nécessité de définir le système d'information, car l'étude menée permet d'améliorer le traitement de l'information nécessaire au fonctionnement de l'organisation sous tous ses aspects (collecte, traitement, archivage, diffusion, transmission). L'étude peut mener à la conclusion qu'une information n'est pas utile. Elle doit également permettre de mettre en évidence les flux d'information inutiles, les traitements redondants, les procédures d'archivage à améliorer etc.

3. La méthode MERISE

La méthode MERISE pour la conception des systèmes d'information, se caractérise selon une approche systémique et selon une démarche à trois axes : démarche par étapes, par niveaux, et par cycle de décisions. Elle est à la fois une méthode de conception des systèmes d'information et une méthode de développement des SI. Le découpage du développement a été repris et normalisé par l'AFNOR (Z67-101 : recommandation pour la conduite des projets informatiques).

3.1. Une approche systémique

L'approche systémique a pour objet l'étude des systèmes dans un sens large. Cette approche a été reprise par les auteurs de la méthode MERISE.

Cette approche consiste, à partir de l'ensemble des informations et des règles de gestion de l'entreprise, à définir pour le domaine d'étude concerné le futur système d'information.

La méthode MERISE s'appuie sur la représentation des structures de l'organisation, de son activité, de son environnement et de ses finalités.

Cette méthode propose la construction du futur système d'information par approches successives.

3.2. Une démarche à trois axes

La démarche MERISE est simultanément organisée en étapes, en niveaux d'abstraction, et en cycle de décision.

3.2.1. Démarche par étapes

La démarche par étapes ou par cycles de vie, décrit la vie du système d'information au travers de différentes étapes. Certaines étapes sont regroupées en grandes périodes : conception, réalisation, et maintenance. La conception du SI aboutit à une description détaillée des spécifications fonctionnelles et techniques. La réalisation consiste à produire les programmes et les consignes d'utilisation correspondants aux spécifications détaillées. La maintenance du système a pour objectif de l'adapter aux évolutions de son environnement.

Ces grandes étapes s'enchaînent et se décomposent de la façon suivante :

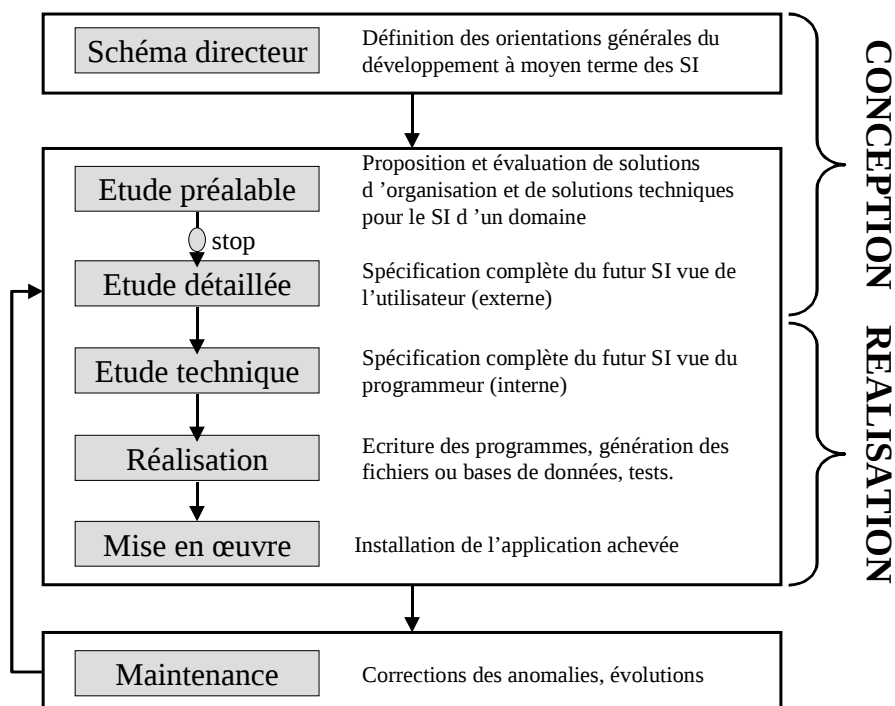


Figure 2 : Les étapes d'un projet MERISE

3.2.2. Démarche par niveaux

La démarche par niveaux ou cycles d'abstraction propose un ensemble de concepts pour la formalisation du système d'information. Dans cette démarche, nous trouvons trois niveaux d'abstraction (conceptuel, organisationnel et opérationnel) et une séparation des données et des traitements.

NIVEAUX	Point de vue	Modèles	
		Données	Traitements
Conceptuel Quoi ?	Gestionnaire	MCD	MCT
Logique / organisationnel Qui ? Quand ? Où ?	Organisateur	MLD/MOD	MOT
Physique Comment ?	Informaticien	MPD	MPT

Ce tableau fait clairement apparaître les modèles que propose la méthode MERISE pour la formalisation des données et des traitements à chacun des niveaux supportés. Les différents modèles Merise sont donc :

- MCD : Modèle Conceptuel de Données ;
- MCT : Modèle Conceptuel de Traitements ;
- MLD : Modèle Logique de Données ;
- MOT : Modèle Organisationnel de Traitements ;
- MPD : Modèle Physique de Données ;
- MPT : Modèle Physique de Traitements.

Les trois niveaux conceptuel, logique, et physique facilitent l'analyse d'un problème en se focalisant respectivement sur les aspects de gestion, d'organisation, et d'implémentation. Aborder un problème selon ces trois axes en facilite l'analyse. Ainsi, le gestionnaire définira les traitements relatifs à une commande reçue dans l'entreprise, les aspects organisationnels seront étudiés par l'organisateur, tandis que l'informaticien précisera les moyens techniques à mettre en œuvre pour satisfaire les besoins relatifs aux niveaux précédents.

La séparation des données et des traitements est conforme aux principes des bases de données relationnelles. Elle met en évidence la relative invariance des données par rapport aux traitements. Par exemple, un processus de facturation peut consister en l'envoi d'une facture et l'attente du paiement ou en prélèvement automatique. Pour autant, les données nécessaires à chacun de ces traitements restent sensiblement les mêmes.

Un modèle supplémentaire, appelé Modèle Conceptuel de Communication (MCC) ou graphe des flux, a été rajouté dans MERISE pour l'étude des échanges d'information intra-entreprise ou avec des tiers (fournisseurs, clients...). Il se situe au niveau conceptuel avant l'étude séparée MCD/MCT.

3.2.3. Cycle de décision

Le cycle de décision regroupe les choix à faire lors du cycle de vie pour passer d'une étape à une autre. Par exemple, à l'issue de l'étude préalable, un certain nombre de solutions possibles seront proposées pour atteindre les objectifs assignés au nouveau système. Certaines de ces offres seront

retenues et donneront lieu à une étude détaillée. Le choix effectué constitue la décision qui conditionne le passage de l'étude préalable à l'étude détaillée.

L'ensemble des résultats produits à chaque étape constitue le cycle de décision.

3.3. L'étude et le recueil de l'existant

Avant de se lancer dans la conception du système d'information, il est essentiel d'étudier et de recueillir l'existant.

3.3.1. Etudier l'organisation et découper en domaine d'activité

Le travail préliminaire est d'étudier l'organisation de l'entreprise qui en général considérée de manière globale, représente un ensemble trop vaste pour que l'on puisse aborder directement l'étude de l'ensemble de son système d'information. C'est pourquoi on va chercher à découper l'entreprise en différents domaines d'activités.

Ces domaines sont en général déterminés à partir des grandes fonctions de l'entreprise : vendre, stocker, commander, payer le personnel... Chaque domaine peut être considéré comme autonome, avec son propre système d'information. Ces différents domaines échangent des informations entre eux.

3.3.2. Le recueil de l'existant

Ensuite, on peut procéder au recueil de l'existant. Cette démarche consiste à mener des interviews auprès des personnes concernées et à collecter l'ensemble des documents (papiers, numériques...) manipulés au sein de l'entreprise et qui concerne un domaine particulier.

Le but de ces entretiens est de mieux cerner la définition de chaque poste de travail, du point de vue de la circulation de l'information. Il convient donc de poser des questions telles que :

- Quels sont les types de documents ou messages reçus ou émis ?
- Quels sont les opérations effectuées, il ne faut négliger aucune opération (par exemple calculer le total de la facture, archiver un dossier, envoyer un fax de confirmation...)
- Quels sont les problèmes rencontrés (informations manquantes, délais d'obtention, répétitivité d'une opération...)

Il convient ensuite de faire une synthèse de ces entretiens afin de formaliser l'existant sous forme de modèles représentant le système d'information.

3.3.3. Identifier les acteurs et les documents

Un document est un support d'informations. Il peut être unique (planning mural) ou exister en plusieurs exemplaires (doubles des factures). Un document peut posséder plusieurs occurrences.

Bien que l'idée du document soit attachée à son support matériel, il faut également prendre en compte les informations véhiculées sur des supports immatériels (messages oraux, téléphoniques, ...).

Un acteur est une personne, ou un groupe de personne qui accomplit des actions sur les informations du domaine. Les acteurs internes font partie du sous-ensemble de l'organisation étudiée. Les acteurs externes échangent de l'information avec les acteurs internes au domaine, mais ils n'en font pas partie.

3.4. Présentation des niveaux de description et des modèles associés

Comme vu ci-dessus, la méthode MERISE propose de décrire un SI suivant différents niveaux d'abstraction allant de l'abstrait vers le concret. A chaque niveau, correspond une préoccupation du concepteur du SI sur la description des données et des traitements. Ces niveaux de description peuvent se regrouper en deux ensembles. Tout d'abord le niveau *conceptuel* et le niveau *organisationnel* correspondent à la préoccupation de description du SI indépendamment des aspects

techniques liés à l'informatisation. Ensuite les niveaux *logique* et *physique* de description d'un SI prennent en compte la technologie informatique retenue pour l'informatisation. Il n'existe pas aujourd'hui de formalisme universel propre à la méthode MERISE pour la description du modèle logique et physique des traitements. Cependant, le modèle relationnel constitue un standard de fait pour la description du niveau logique des données.

Globalement, les quatre niveaux de description du SI constituent le cycle d'abstraction du SI.

3.4.1. Le niveau conceptuel

Le niveau conceptuel qui reflète les aspects de gestion, peut-être défini grâce à trois modèles :

- MCC : Modèle Conceptuel de Communication ;
- MCD : Modèle Conceptuel de Données ;
- MCT : Modèle Conceptuel de Traitements.

Dans ce niveaux, il s'agit de répondre à la question « QUOI ? », tout en faisant abstraction des contraintes d'organisation et techniques.

3.4.1.1. Le Modèle Conceptuel de Communication (MCC)

Le Modèle Conceptuel de Communication représente la première vue formelle du système d'information. Il permet de recenser l'ensemble des échanges informationnels entre acteurs pour le domaine étudié. Particulièrement simple à élaborer, il constitue un puissant outil de communication et de validation. Les concepts utilisés (acteurs et flux) étant très simples et compréhensibles de tous.

- **Acteur :**

Définition :

C'est une personne morale ou physique capable d'émettre ou de recevoir des informations. Ainsi, un client ou le service commercial d'une société sont des acteurs du domaine gestion commerciale. Les acteurs peuvent être internes ou externes. Ils sont internes s'ils appartiennent au domaine fonctionnel étudié, à l'inverse ils sont externes s'ils n'appartiennent pas au domaine fonctionnel étudié.

Formalisme :

On représente les acteurs internes par un cercle et les acteurs externes par un cercle en pointillés.

- **Flux :**

Définition

C'est un échange d'informations entre un acteur émetteur et un acteur récepteur.

Formalisme :

On représente le flux par un lien fléché et nommé, entre acteurs.

- **Exemple de MCC :**

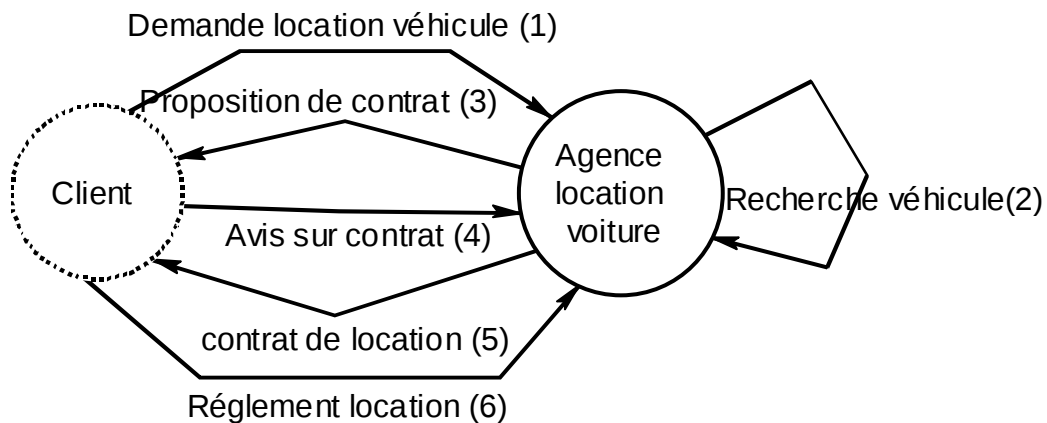


Figure 3 : Exemple de MCC

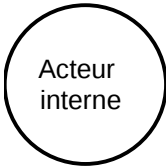
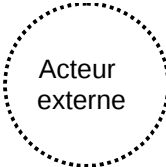
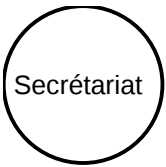
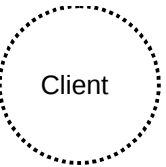
Dès lors qu'un flux représente un échange d'informations, on peut très bien dessiner un flux réflexif. Cela dépend notamment du niveau de précision lors de la définition des acteurs. Dans l'exemple ci-dessus, on pourra décomposer « *agence location voiture* » en deux acteurs : « *secrétariat* » et « *responsable du parc automobile* ».

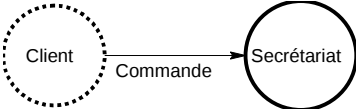
Cet exemple de MCC représente les flux d'informations qui circulent entre un client souhaitant louer un véhicule et une agence de location de voiture. Le client est en pointillé, puisqu'il est acteur externe du système d'information.

Il est à noter qu'aucune référence organisationnelle ou technique apparaît dans le MCC. Nous sommes bien au niveau du concept où l'on ne précise pas les moyens de communication (téléphone, fax...) ni si le système est informatisé. De même, le MCC ne fait pas apparaître d'informations chronologiques, même si dans l'exemple ci-dessus on numérote les flux afin d'en faciliter la lecture.

Enfin, on se rend compte que ce modèle ne requiert aucune connaissance approfondie de Merise tant le formalisme utilisé est simple.

- **Résumé du MCC :**

Acteur		
Définition	Symbole Associé	Exemple
Personne morale ou physique qui émet ou reçoit des informations. L'acteur est typé interne ou externe selon qu'il appartient ou non au champ d'étude	 Acteur interne  Acteur externe	 Secrétariat  Client

Flux		
Définition	Symbole Associé	Exemple
Matérialisation de l'échange	 Flux	

3.4.1.2. Le dictionnaire des données

Lors de la phase du recueil de l'existant, les documents ont été étudiés de manière globale. A cette étape de la méthode Merise, on étudie attentivement chaque document. Un même document peut se présenter sous plusieurs formes (papier, écran, numérique...).

Un document comprend en général des informations de forme (en-tête, formule de politesse...) qui n'ont pas d'importance pour notre système d'information, et des informations « vitales » (nom, adresse client, téléphone client...). Ces informations sont rassemblées dans des rubriques éventuellement répétitives.

Par définition, une donnée élémentaire est la représentation d'informations élémentaires manipulées dans le domaine étudié. Rubriques de document et données sont deux notions différentes. Une donnée, comme une rubrique, peut-être composée (ex : numéro de sécurité sociale). Cependant, il est exclu de décomposer une telle donnée, dans le cadre des activités du domaine. Ce n'est pas nécessairement le cas pour les rubriques de documents. Par exemple, l'adresse du client peut constituer qu'une seule rubrique sur le document, et correspondre à trois ou quatre données : rue, code postal, ville, pays.

D'autre part, une donnée a un type bien déterminé (alphanumérique, numérique, date, booléen...) alors qu'une rubrique n'est pas nécessairement normalisée (code numérique ou nom pour un département).

Au fur et à mesure de l'étude, et pour chaque donnée identifiée :

- On lui attribue un nom, qui sera utilisé dans toute la suite de l'étude ;
- On lui indique le type, la plage de valeurs ;
- On en étudie des propriétés telle que :
 - ♦ S'agit-il d'une donnée élémentaire de base ou d'une donnée calculée ?
 - ♦ S'agit-il d'une donnée stable ou non stable ?

On constitue ainsi une liste. A chaque ajout de donnée dans la liste des données, il est très important de vérifier si :

- La donnée n'est pas déjà répertoriée ?
- La donnée n'est pas déjà répertoriée mais avec un nom différent ? (cas des synonymes : plusieurs termes ayant le même sens, ex : briser, casser, et rompre)
- Le nom attribué à la donnée n'est pas déjà utilisé pour désigner une autre donnée ? (cas des polysèmes : mot qui présente plusieurs sens, ex : le son)

A la fin de ce travail, on dispose d'une liste de données sans redondance, sans synonyme, et sans polysème : Le dictionnaire des données.

3.4.1.3. Le Modèle Conceptuel des Données (MCD)

Conformément à l'approche séparée des données et des traitements, le Modèle Conceptuel des Données (MCD) reflète la vue du système d'information côté données uniquement. Il doit être le plus complet possible. Sa représentation doit se faire en toute indépendance de considérations techniques et/ou organisationnelles. Le MCD est une représentation statique des données et, par conséquent, ne doit comporter aucune référence aux traitements effectués.

Contrairement au MCC, le MCD n'a pas vocation d'être lu par des non-initiés. En revanche, c'est un excellent outil de validation pour le concepteur du SI.

Le Modèle Conceptuel de Données (MCD) permet de faire la description des données et des relations entre les données, grâce aux concepts du formalisme Entité-Association.

- **Propriété (ou attribut) :**

Définition :

Une propriété est une donnée élémentaire susceptible de prendre une valeur. C'est le plus petit élément manipulé du système d'information et qui a un sens en lui-même. Une propriété peut-être élémentaire ou calculée, simple ou composée.

La note d'un élève est une propriété élémentaire, en revanche sa moyenne est une propriété calculée.

L'âge de l'élève est une propriété simple alors que l'adresse complète de l'élève est une propriété composée.

Formalisme :

Le nom de la propriété est inscrit à l'intérieur de l'entité.

- **Entité (ou objet) :**

Définition :

Une entité est un objet ou un individu manipulé par l'entreprise, pourvu d'une existence propre. L'entité est caractérisée par un certain nombre de propriétés dont l'ensemble des valeurs spécifiques d'un objet donné correspond à une occurrence de l'entité.

L'élève est une entité, l'élève « Pierre Dupond » est une occurrence de l'entité élève.

Formalisme :

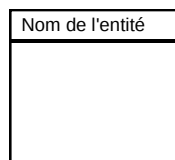


Figure 4 : Une entité sans ses propriétés

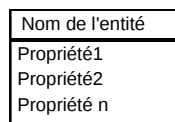


Figure 5 : Une entité avec ses propriétés

Exemple :

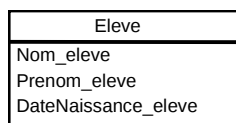


Figure 6 : Exemple d'une entité

- **Identifiant (ou propriété identifiante) :**

Définition :

L'identifiant est une propriété particulière de l'entité telle qu'à chaque valeur de la propriété corresponde une et une seule occurrence de l'entité. La valeur de l'identifiant rend unique chaque occurrence de l'entité, ainsi pour éviter les synonymes, des numéros ou des codes font les meilleurs identifiants.

Formalisme :

Dans le MCD, l'identifiant figure en première position dans la liste des propriétés et est souligné.

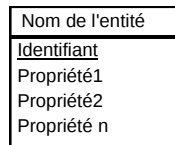


Figure 7 : Une entité avec un identifiant

Exemple :

Le numéro d'un élève sera l'identifiant pour l'entité élève, son nom ne suffirait pas pour l'identifier de manière unique dans le SI.

Le numéro INSEE pourra être l'identifiant d'une personne.

Le numéro d'immatriculation pourra être l'identifiant de l'entité voiture.

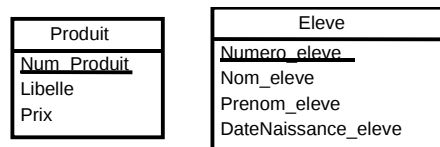


Figure 8 : Des exemples d'identifiants

- Relation (ou association) :

Définition :

Une association est la représentation d'une relation entre entités. Elle est dépourvue d'existence propre et est subordonnée à l'existence des entités qui la composent.

L'entité « élève » est en association avec l'entité « classe »

La relation peut être porteuse d'une ou plusieurs propriétés : L'entité « élève » est en association avec l'entité « matière » et porte la propriété « note ». Cela se traduit par l'élève a une note dans une matière.

Formalisme :

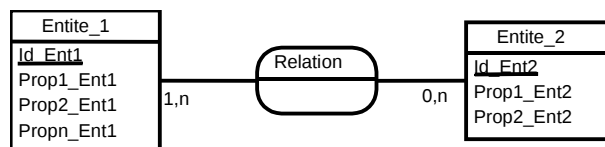


Figure 9 : Formalisme de la relation

Exemples :

La relation peut être binaire :

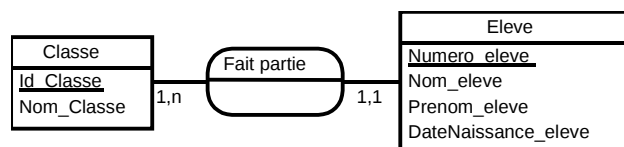


Figure 10 : Relation binaire

La relation peut être porteuse de données :

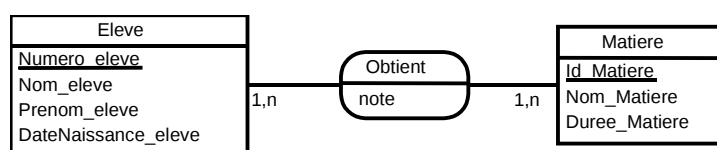


Figure 11 : Relation porteuse de données

La relation peut être ternaire :

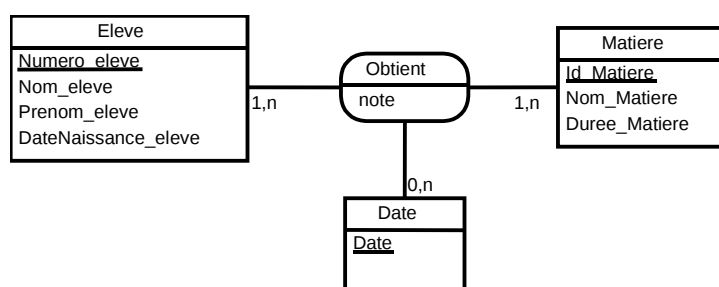


Figure 12 : Relation ternaire

NB : Dans une relation ternaire, il n'y a jamais de cardinalités à 1,1 ou 0,1

Une relation à n entités, est une relation n-aire

La relation peut être réflexive :

C'est une relation d'une entité sur elle-même.

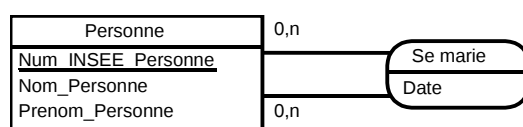


Figure 13 : Relation réflexive

La relation peut-être identifiante :

L'association identifiante traduit la relation de composition entre une entité composante et une entité composée. L'exemple classique est la relation entre une commande et ses lignes de commande. Une ligne de commande n'a pas d'existence propre. Elle est toujours relative à une commande. Un autre exemple classique est celui de l'hôtel et des ses chambres. Les chambres seules n'ont pas d'existence propre. L'association identifiante (appelée lien identifiant) est notée (1,1) dans le MCD.

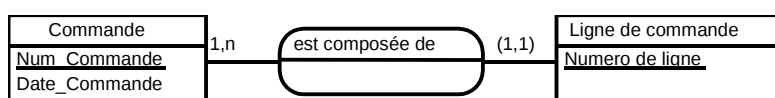


Figure 14 : Le lien identifiant de la commande et des lignes de commande

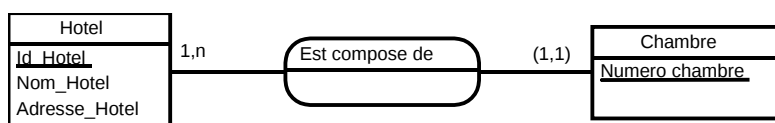


Figure 15 : Le lien identifiant de l'hôtel et de ses chambres

- **Cardinalité :**

Définitions :

La cardinalité d'une entité par rapport à une relation s'exprime par deux nombres appelés cardinalité minimale et cardinalité maximale.

La cardinalité minimale, égale à 0 ou 1, est le nombre de fois minimum qu'une occurrence d'une entité participe aux occurrences de la relation. Si la cardinalité minimale est égale à 0, c'est qu'il existe parmi toutes les occurrences de l'entité, au moins une occurrence qui ne participe pas à la relation. Par exemple, dans une équipe de sport, tous les membres de l'équipe ne participent pas forcément à un match. En revanche, si la cardinalité minimale est égale à 1, cela implique que toutes les occurrences d'une entité participent à toutes les

occurrences de la relation. Dans notre exemple, cela se traduit par le fait qu'un joueur joue tous les matchs.

La cardinalité maximale, égale à 1 ou n, indique le nombre de fois maximum qu'une occurrence de l'entité participe aux occurrences de la relation (n est équivalent à infini).

Formalisme :

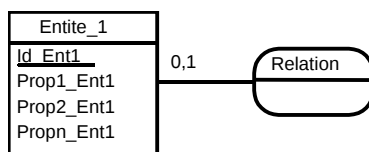


Figure 16 : Cardinalité 0,1

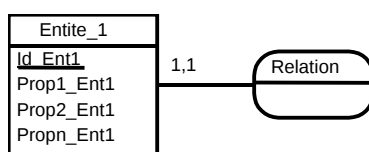


Figure 17 : Cardinalité 1,1

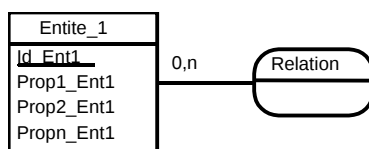


Figure 18 : Cardinalité 0,n

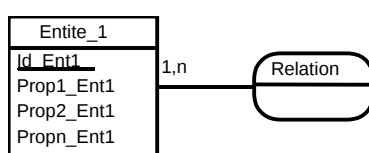


Figure 19 : Cardinalité 1,n

Exemple :

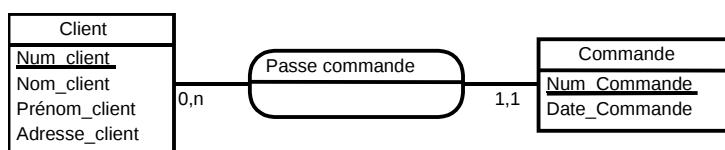


Figure 20 : Explication des cardinalités

Dans l'exemple ci-dessus, la cardinalité minimale entre l'entité « client » et la relation « passe commande » qui est à « 0 » exprime le fait qu'un client peut ne pas passer de commande. C'est un client potentiel.

La cardinalité maximale entre l'entité « client » et la relation « passe commande » qui est à « n » exprime le fait qu'un client peut passer au plus « n » commandes.

La cardinalité minimale entre la relation « passe commande » et l'entité « commande » qui est à « 1 » exprime le fait qu'à une commande correspond toujours un client.

La cardinalité maximale entre la relation « passe commande » et l'entité « commande » qui est à « 1 » exprime le fait qu'à une commande correspond un seul client au maximum.

- **Occurrence :**

Définition :

L'occurrence d'une entité ou d'une association est le nombre de fois qu'apparaît l'entité (ou l'association) ayant des propriétés à valeurs distinctes.

Dans le cas d'une équipe de foot sans remplaçants, l'entité « joueurs » aurait 11 occurrences. Cette notion d'occurrence sert à estimer la taille de la base de données.

- **Dépendance fonctionnelle :**

Dépendance fonctionnelle entre propriétés :

On dit que 2 propriétés A et B sont reliées par une dépendance fonctionnelle (notée : $A \xrightarrow{\text{df}} B$) si la connaissance de la valeur de A détermine une et une seule valeur de B.

Exemple : Code_Client $\xrightarrow{\text{df}}$ Nom_Client

La connaissance de « Code_Client » détermine une et une seule valeur de « Nom_Client ». Autrement dit, si l'on connaît « Code_Client » on doit pouvoir connaître « Nom_Client » et celui-ci sera unique.

La réciproque est fausse. « Nom_Client » ne permet pas de déterminer son code, car plusieurs clients peuvent avoir le même nom.

La dépendance fonctionnelle peut porter sur la concaténation de plusieurs propriétés.

Exemple : N°_Bon_Commande + Ref_Art $\xrightarrow{\text{df}}$ Qté_Demandée

« Ref_Art » seul ne suffit pas à déterminer la quantité commandée, une même référence pouvant figurer sur plusieurs bons de commande.

« N°_Bon_Commande » ne suffit pas non plus puisqu'un même bon de commande peut concerner plusieurs références.

En revanche, la connaissance de « N°_Bon_Commande » **ET** « Ref_Art » détermine celle de « Qté_Demandée ».

Dépendance fonctionnelle entre entités :

On dit qu'il existe une dépendance fonctionnelle entre 2 entités A et B (notée : $A \longrightarrow B$) si toute occurrence de A détermine une et une seule occurrence de B.

Exemple : Commande $\longrightarrow$ Client

Toute commande a un et un seul client (cas des commandes individuelles uniquement).

Une dépendance fonctionnelle entre entité se traduit toujours par une cardinalité maximale à 1.

Les dépendances fonctionnelles entre entités ne portent jamais de propriétés.

Il existe deux types de dépendance fonctionnelle :

- ♦ La dépendance fonctionnelle faible;
- ♦ La dépendance fonctionnelle forte.

La dépendance fonctionnelle faible : lorsque la cardinalité minimale est à 0 (relation partielle ou potentielle entre 2 entités) la dépendance est dite faible.

La dépendance fonctionnelle forte : lorsque la cardinalité minimale est à 1 (relation fonctionnelle totale) cette dépendance est alors dite forte.

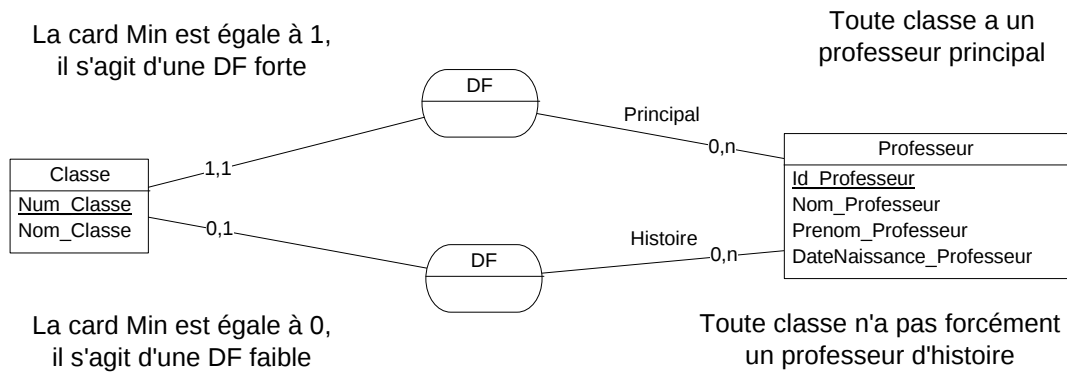


Figure 21 : Les dépendances fonctionnelles

- **Contrainte d'intégrité fonctionnelle (CIF) :**

Définition :

La notion de contrainte d'intégrité fonctionnelle (CIF) correspond à celle de la dépendance fonctionnelle (DF) forte **ET** stable. (stable signifiant que les occurrences des entités mises en jeu ne pourront jamais changer).

Exemple : Un enfant ne pourra jamais changer de père. De même la dépendance fonctionnelle entre commande et client est une contrainte d'intégrité fonctionnelle car le client qui a passé la commande ne peut pas changer.

La CIF traduit un lien fort et permanent (non modifiable sauf son annulation) de dépendance d'une entité par rapport à une ou plusieurs autres entités. Dans le cas où ce lien n'est pas permanent dans le temps, il s'agira donc d'une dépendance fonctionnelle entre objets.

L'intérêt de mettre en évidence une CIF dans une relation de dimension supérieure à 2, réside dans le fait que l'on peut diminuer de 1 la dimension de la relation.

Exemple :

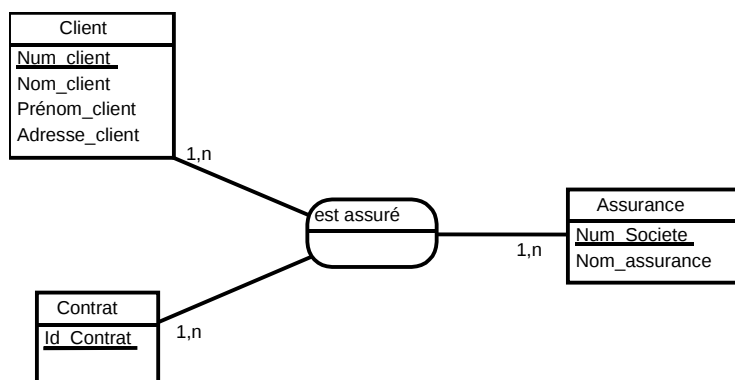


Figure 22 : MCD du contrat d'assurance

Dans l'exemple ci-dessus, l'existence d'un contrat d'assurance implique la connaissance de l'assuré. Il y a donc un lien fort et permanent de dépendance entre l'entité « client » et l'entité « contrat ». La mise en évidence de cette CIF permet de décomposer le modèle en deux associations binaires (au lieu d'une association ternaire) :

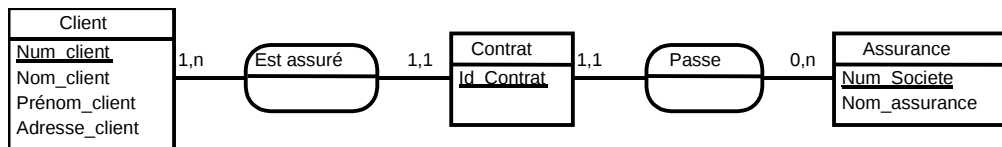


Figure 23 : MCD avec une CIF (Relation portant un nom)

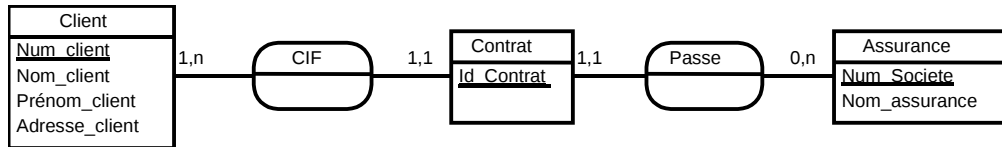


Figure 24 : MCD avec une CIF (relation nommée CIF)

Dans l'exemple ci-dessus, il n'est pas nécessaire de nommer la relation de la CIF. Afin de faciliter le nommage de la relation, il est possible de la nommer « CIF », notamment pour bien la mettre en évidence.

- **Généralisation/spécialisation :**

La modélisation d'un SI à l'aide du formalisme entité-relation connaît dans certains cas des difficultés pour représenter dans une même entité des propriétés qui ne concernent pas la totalité des occurrences de cette même entité.

Par exemple, dans un SI comportant deux catégories d'employés, les employés mensuels, et les employés vacataires. Très rapidement on va se rendre compte que ces entités ne seront pas facile à formaliser. Vaut-il mieux rassembler les propriétés dans une seule entité, mais avec le risque d'avoir des propriétés sans valeur. Ou bien, est-il préférable de décomposer une entité en deux entités, mais cette fois-ci avec le risque d'avoir une redondance des propriétés.

Pour solutionner ce problème, le concept de généralisation/spécialisation a été porté avec l'évolution de MERISE 2.

Ce concept est fondé sur le fait de regrouper toutes les propriétés communes aux deux populations d'individus dans une entité dite générique et de créer deux entités spécialisées ne contenant que les propriétés spécifiques à chaque population d'individus. Ces deux types d'entités étant liés par un lien de généralisation/spécialisation.

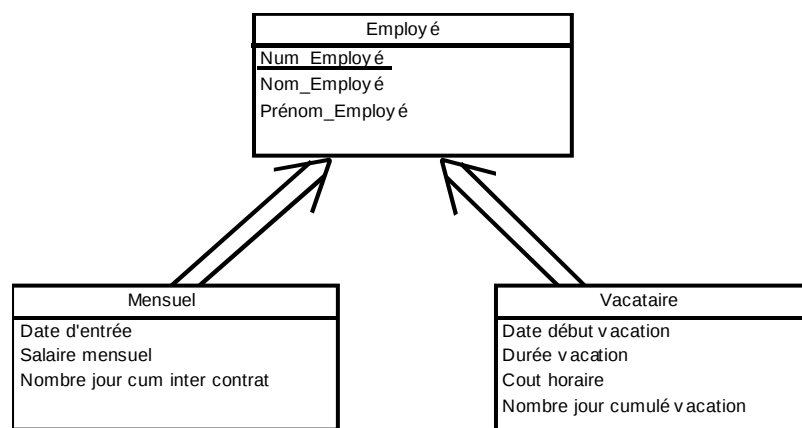


Figure 25 : Exemple du concept généralisation/spécialisation.

Cette solution permet une représentation plus fidèle de la réalité.

Définition :

La généralisation est un processus de modélisation permettant de rassembler dans une même entité toutes les propriétés communes relatives à cette entité, vis-à-vis d'autres

entités spécialisées regroupant des propriétés propres à un sous-ensemble d'occurrences de l'entité générique.

Formalisme :

La généralisation peut être de deux types : simple ou multiple.

La généralisation simple est caractérisée par l'unicité du lien de généralisation pour une même entité, comme l'exemple ci-dessus.

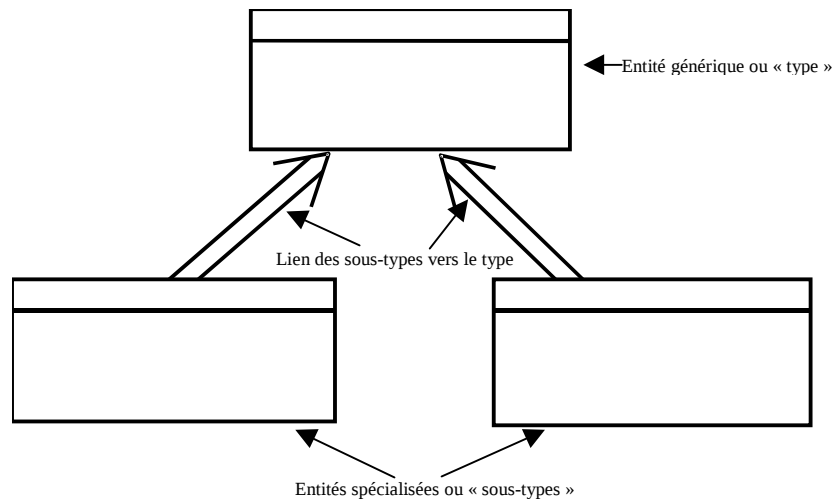


Figure 26 : Généralisation simple

La généralisation multiple est caractérisée par les liens multiples pour une même entité en tant que sous-type vis à vis d'autres entités types.

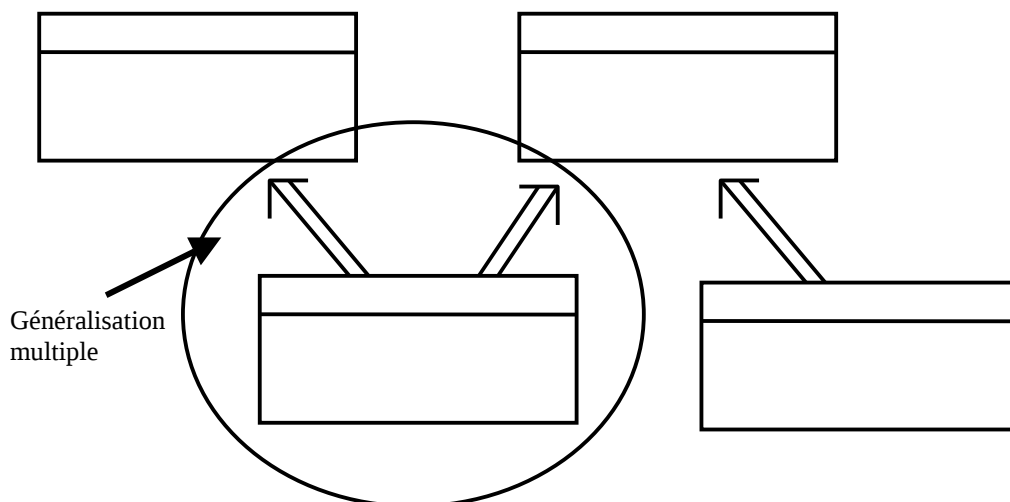


Figure 27 : Généralisation multiple

- **Le concept d'héritage :**

Le concept d'héritage consiste à transmettre les propriétés de l'entité type vers les sous-types. Ces propriétés sont toujours disponibles pour les objets spécialisés, mais par convention de représentation, elles ne sont mentionnées que par l'entité type. (Figure 24: Exemple du concept généralisation/spécialisation)

- **Les contraintes d'extension sur les relations ou sur les entités :**

Le formalisme étudié jusqu'à présent ne permet pas de représenter certaines contraintes et particularités liées aux occurrences de relations ou d'entités. Pour ce faire, il existe des contraintes d'extension suivantes :

- ♦ La contrainte d'exclusion ;
- ♦ La contrainte d'inclusion ;
- ♦ La contrainte de totalité ;
- ♦ La contrainte de partition ou de « ou exclusif » ;
- ♦ La contrainte d'égalité (ou de simultanéité) ;

La présentation des contraintes est réalisée sur les relations. Ces contraintes peuvent aussi exister sur des entités ou sous-types d'entités.

Préalable :

Le **pivot** désigne l'ensemble des entités communes aux associations concernées par la contrainte.

- **La contrainte d'exclusion :**

Définition :

La contrainte d'exclusion traduit le fait que **toute occurrence** d'une entité pivot participe à **l'une ou l'autre** des associations de la contrainte **ou à aucune des deux, mais pas aux deux**.

Formalisme :

La contrainte d'exclusion est représentée par un cercle contenant un X, et est reliée au pivot par un trait pointillé.

Exemple :

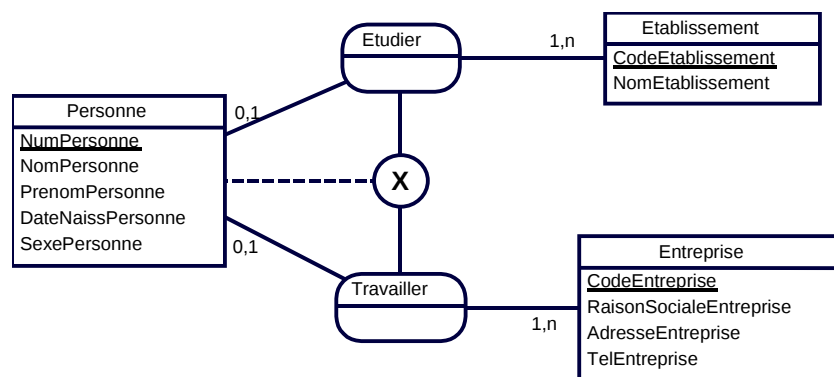


Figure 28 : La contrainte d'exclusion

Dans l'exemple ci-dessus, l'entité PERSONNE est le pivot de la contrainte.

La contrainte exprimée est : « Une personne étudie dans un établissement **ou bien** elle est salariée d'une entreprise, **ou bien** elle est ni étudiante ni salariée, **mais elle ne peut être à la fois étudiante et salariée** ».

- **La contrainte d'inclusion :**

Définition :

La contrainte d'inclusion traduit le fait que **les occurrences** des entités participant à une relation R1 participent **implicitement** à une autre relation R2. En revanche, **la réciproque est fausse**.

Formalisme :

La contrainte d'inclusion est représentée par un cercle contenant un **I**, et est reliée au pivot par un trait pointillé.

Exemple :

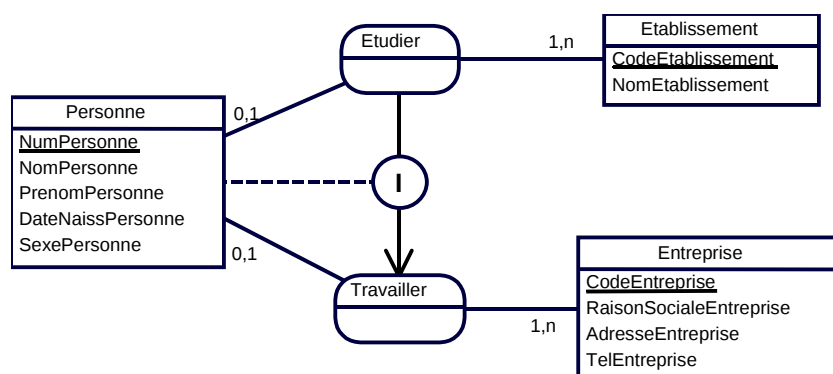


Figure 29 : La contrainte d'inclusion

Dans l'exemple ci-dessus, les occurrences de l'entité PERSONNE qui participent à la relation ETUDIER participent **également et obligatoirement** à la relation TRAVAILLER. En revanche, le fait qu'une occurrence de l'entité personne participe à la relation TRAVAILLER **n'implique pas** qu'elle participe à la relation ETUDIER.

Cela se traduit par « Une personne qui étudie dans un établissement implique qu'elle travaille dans une entreprise ». (Cas d'une personne étudiant dans un lycée professionnel et ayant l'obligation de faire un stage en entreprise). En revanche, la réciproque est fausse.

- **La contrainte de totalité :**

Définition :

La contrainte de totalité traduit le fait que **toute occurrence** du pivot participe à **l'une ou l'autre** des associations de la contrainte **ou aux deux**.

Formalisme :

La contrainte de totalité est représentée par un cercle contenant un **T**, et est reliée au pivot par un trait pointillé.

Exemple :

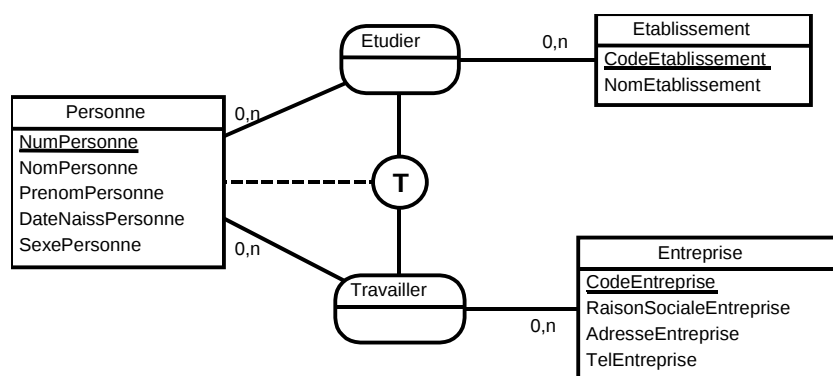


Figure 30 : La contrainte de totalité

Dans l'exemple, **toute occurrence** de l'entité PERSONNE participe **soit** à la relation ETUDIER, **soit** à la relation TRAVAILLER **soit au deux relations à la fois**.

Cela se traduit par « Une personne peut être étudiant dans un établissement de formation ou salarié dans une entreprise ou les deux à la fois » (Cas d'un étudiant en formation continue par exemple).

- **La contrainte de partition (ou de « ou exclusif ») :**

Définition :

La contrainte de partition traduit le fait que **toute occurrence** du pivot participe **obligatoirement** à **l'une ou l'autre** des associations de la contrainte, **mais pas aux deux**.

Formalisme :

La contrainte de partition est représentée par un cercle contenant un **+**, et est reliée au pivot par un trait pointillé.

Exemple :

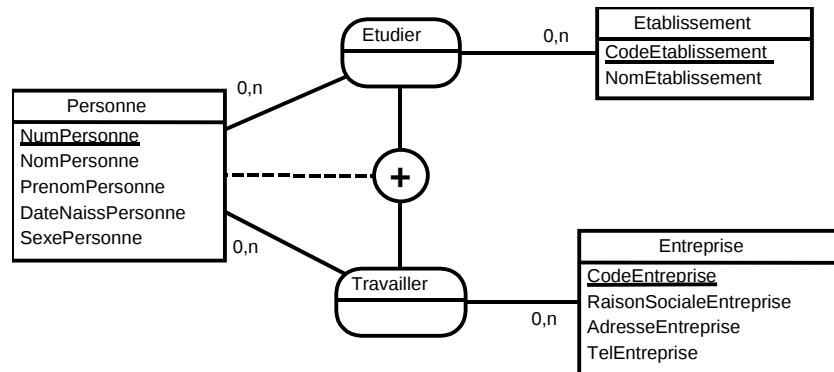


Figure 31 : La contrainte de partition

Dans l'exemple, **toute occurrence** de la table PERSONNE participe **soit** à la relation ETUDIER, **soit** à la relation TRAVAILLER **mais pas au deux à la fois**.

Cela se traduit par « Une personne est obligatoirement soit étudiant dans un établissement de formation soit salarié dans une entreprise »

Cette contrainte est équivalente à une contrainte de totalité et une contrainte d'exclusion.

- **La contrainte d'égalité (ou de simultanéité) :**

Définition :

La contrainte d'égalité traduit deux inclusions symétriques. C'est à dire que **toute** occurrence d'une entité participant à une relation R1, participe **simultanément** à une relation R2.

Formalisme :

La contrainte d'égalité est représentée par un cercle contenant un **=**, et est reliée au pivot par un trait pointillé.

Exemple :

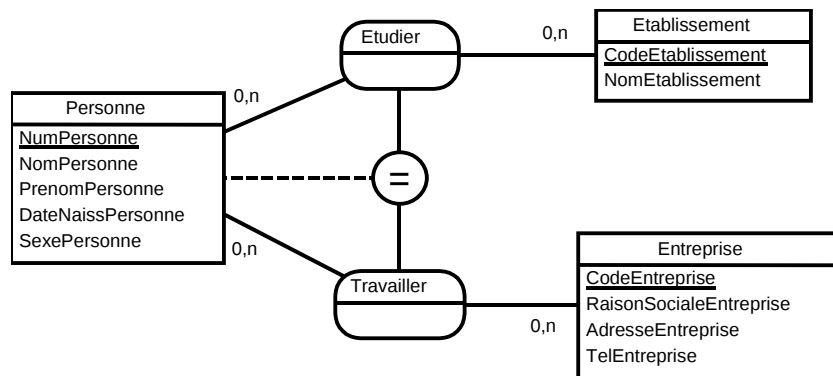


Figure 32 : La contrainte d'égalité

Dans l'exemple ci-dessus, **toutes les occurrences** de l'entité PERSONNE qui participent à la relation ETUDIER participent **également et obligatoirement** à la relation TRAVAILLER. Inversement **toutes les occurrences** de l'entité PERSONNE qui participent à la relation TRAVAILLER participent **également et obligatoirement** à la relation ETUDIER.

Cela se traduit par « Une personne qui étudie dans un établissement implique qu'elle travaille dans une entreprise » ou « une personne qui est salariée dans une entreprise implique qu'elle étudie dans un établissement ».

- **Règles de vérification du MCD :**

Le MCD a progressivement été élaboré à partir du dictionnaire de données et des règles de gestion issues du recueil et de l'étude de l'existant.

Cependant, il reste à vérifier et valider ce MCD. Pour ce faire, des règles de vérification ont été élaborées.

Règle n°1 : toute entité ou association doit avoir un identifiant et un seul

Dans l'exemple ci-dessous, les identifiants des entités CLASSE et ELEVE sont respectivement « Id_Classe » et « Numero_eleve ». Les identifiants d'entités apparaissent donc de manière explicite sur le schéma.

L'identifiant d'une entité peut-être composé de plusieurs propriétés mais il n'y a toujours qu'un seul identifiant par entité.

L'identifiant de l'association n'apparaît pas clairement sur le modèle. Il se compose des identifiants des entités de la relation. L'identifiant de l'association FAIT PARTIE est donc le couple de valeurs (« Id_Classe », « Numero_eleve »).

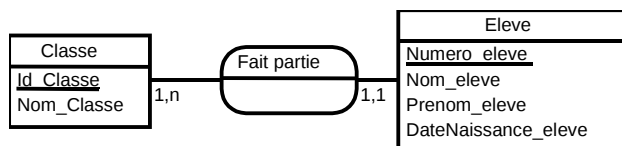


Figure 33 : Règle de validation n°1

Règle n°2 : toutes les propriétés doivent être élémentaires, c'est-à-dire non décomposables

Cette règle a aussi une autre définition : toute propriété doit être dans sa forme atomique. Cela signifie que toute propriété ne doit pas être décomposable. Par exemple, la propriété « Adresse » pourrait être décomposée en « rue », « ville », « code postal », et « pays ».

Règle n°3 : Les propriétés d'une entité autres que l'identifiant doivent être en dépendance fonctionnelle monovaluée de cet identifiant

Pour une occurrence de l'identifiant d'une entité, chacune des propriétés ne peut prendre qu'une valeur. Par conséquent, il est impossible d'avoir une valeur répétitive pour une même propriété, comme il ne doit y avoir d'absence de valeur pour une même propriété.

Par exemple, l'entité PERSONNE définie par les propriétés « Num_SS », « Nom_Pers », « Prénom_Pers », « Diplômes_Pers ». Cette entité peut recevoir des valeurs pour le cas où la personne a au plus un diplôme. Une solution consisterait à prévoir plusieurs propriétés concernant les diplômes : « Diplôme 1 », « Diplôme 2 », « Diplôme 3 ». Rapidement, on se rend compte que dans certains cas ces propriétés ne serviraient pas, alors que dans d'autres, elles seraient insuffisantes. Une autre solution, la bonne, consiste à créer une entité DIPLOME et une relation l'associant à l'entité PERSONNE.

Règle n°4 : une propriété ne peut qualifier qu'un seul objet ou qu'une seule relation

Une propriété ne peut être présente à la fois dans plusieurs entités. Par exemple, la propriété « Nom du client » ne peut être présente à la fois dans l'entité CLIENT et dans l'entité FACTURE.

De même, une propriété ne peut porter le même nom pour des sens différents. Par exemple, la propriété ne peut s'appeler « Adresse » dans l'entité FOURNISSEUR et « Adresse » dans l'entité CLIENT (cas de polysèmes). Dans ce cas, il faut appeler l'une « Adresse fournisseur » et l'autre « Adresse client ».

Règle n°5 : la dépendance fonctionnelle transitive doit être écartée.

Si une propriété est en dépendance fonctionnelle de l'identifiant, et d'une autre propriété de l'objet, elle-même en dépendance fonctionnelle simple de cet identifiant, il y a une entité imbriquée dans celle que l'on étudie. Chaque entité doit décrire un concept sémantique unique, et en conséquence, il faut éclater en deux entités celle qui contient une dépendance fonctionnelle transitive.

Dans l'exemple ci-dessous, considérons la règle de gestion suivante : le prix de vente au client est calculé sur le prix de vente public, diminué d'une remise dont le montant est fonction de la catégorie à laquelle appartient le client. Le client est forcément rattaché à une catégorie, et, au plus une.

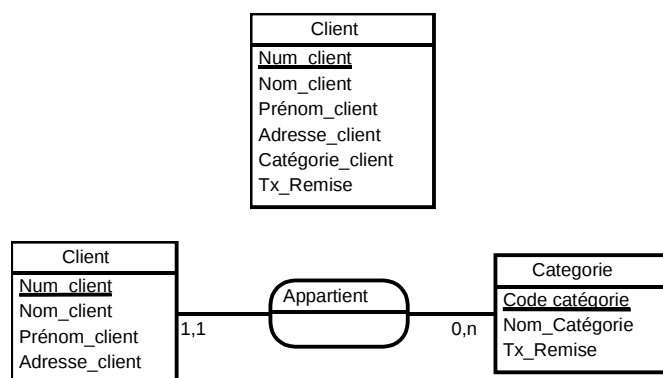


Figure 34 : Règle n°5, résolution de la dépendance fonctionnelle transitive

Règle n°6 : toute propriété d'entité ou d'association ne peut prendre qu'une seule valeur

Dans l'exemple ci-dessous, la propriété « note » caractérise le couple (« Num_etudiant », « Num_matiere »). Un étudiant pouvant avoir plusieurs notes pour la même matière, ce schéma ne respecte donc pas cette règle

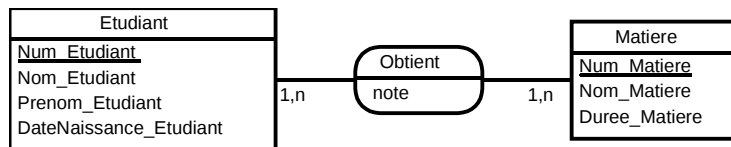


Figure 35 : Modèle sans l'application de la règle n°6

Pour obtenir un modèle correct, il suffit de préciser qu'un étudiant ne peut avoir plusieurs notes le même jour dans une même matière. Dans ces conditions, la « date » est suffisamment discriminante. Il suffit de modifier le modèle de telle sorte que la propriété « note » soit une caractéristique du triplet (« Num_etudiant », « Num_matiere », « date »).

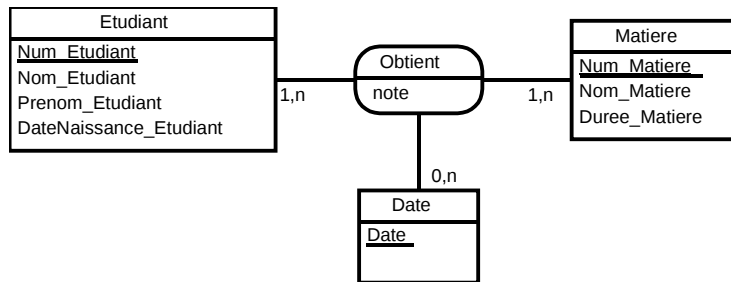


Figure 36 : Application de la règle n°6

- **Résumé du MCD :**

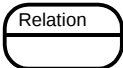
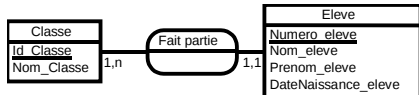
Règle de gestion		
Définition	Symbole Associé	Exemple
Description littérale d'une loi qui régit l'activité de l'entreprise. Elle peut être de type définition, fait, validation, ou formule	Aucun	Tout client possède une adresse (def) Tout client passe au moins une commande (fait) Un client ne peut commander à nouveau s'il a des factures impayées (val) La chiffre d'affaire réalisé avec un client est égal à la somme des montants des commandes passées. (for)

Information		
Définition	Symbole Associé	Exemple
Plus petit élément du SI présentant un intérêt pour l'activité étudiée.	Aucun	Nom client, Prénom client, date de commande...

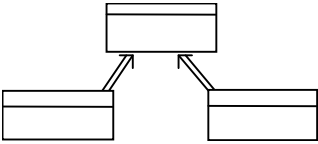
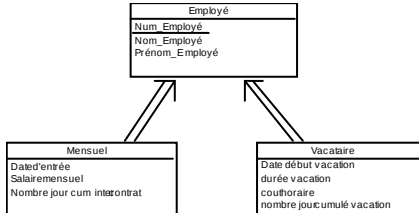
Entité		
Définition	Symbole Associé	Exemple
Objet de gestion pour lequel on souhaite conserver des informations.	Ent_1	Hotel <u>Id_Hotel</u> Nom_Hotel Adresse_Hotel Eleve <u>Numero_eleve</u> Nom_eleve Prenom_eleve DateNaissance_eleve

Propriété		
Définition	Symbole Associé	Exemple
Information rattachée à une entité	Aucun	Nom client, Prénom client, date de commande...

Propriété identifiante		
Définition	Symbole Associé	Exemple
Propriété identifiant de manière unique tout élément d'entité	Propriété soulignée	Numéro INSEE, Num Immatriculation

Association		
Définition	Symbole Associé	Exemple
Traduit l'existence d'une relation entre entités		

Cardinalités		
Définition	Symbole Associé	Exemple
Nombre minimal et maximal de fois qu'un élément de l'entité est associé aux éléments de l'association	Couples de valeurs : 1,1 0,1 1,n 0,n	

Héritage		
Définition	Symbole Associé	Exemple
Relation type/sous-types ou père/fils. Relation de décomposition suivant un critère discriminant.		

3.4.1.4. Le Modèle Conceptuel des Traitements (MCT)

Après avoir étudié les données du système, et conformément à l'approche séparée des données et des traitements, il convient d'étudier la dynamique du système au travers du Modèle Conceptuel de Traitements (MCT). Comme pour le MCD, le MCT relève du niveau conceptuel, et ne décrit que la dimension fonctionnelle du domaine étudié à l'exclusion de toute considération technique et organisationnelle. En revanche, il peut être construit, avant, pendant ou après la construction du MCD.

Le but du MCT est de décrire les traitements que le système d'information doit effectuer pour passer d'un événement à un autre dans le cadre d'un processus. Cette description dynamique est faite à l'aide des concepts suivants :

- **Processus :**

Définition :

Le processus est un ensemble homogène de traitements constituant la réaction du système d'information à la réalisation d'événements.

Exemple :

Dans une entreprise ayant pour activité la commercialisation de produits auprès de sa clientèle. Cette activité de commercialisation peut se découper en sous domaines ; gestion des commandes, facturation, et expédition. Chacun de ces sous-domaines est couvert par un processus distinct.

- **Événement :**

Définition :

L'événement est un fait qui, en se produisant, déclenche une réaction du système. Il peut être associé à un alias pour être référencé simplement dans une condition de synchronisation.

Le système d'information est un système « à états ». A tout moment, il est possible d'établir un constat de l'état dans lequel se trouve le système. L'arrivée d'un événement nouveau vient modifier cet état, et oblige le système à réagir pour retrouver un état stationnaire.

L'événement est porteur d'informations. Les seuls événements pris en compte dans le modèle sont les événements dits externes qui sont indépendants de l'organisation mise en place dans le domaine étudié. Les différentes valeurs prises par chaque événement sont appelées occurrences de l'événement.

Formalisme

L'événement est représenté par un cercle portant son nom.



Figure 37 : Formalisme de l'événement

Exemple

Dans le processus « gestion des remboursements » d'une compagnie d'assurance, la déclaration de sinistre constitue un événement.

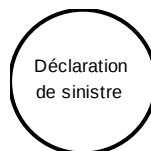


Figure 38 : Exemple d'événement

- **Opération :**

Définition :

Une opération est un ensemble de traitements ininterrompu déclenché par un événement ou un ensemble d'événements. Une fois l'opération déclenchée elle prend en compte les informations portées par les événements, et consulte ou met à jour les données du système.

Formalisme :

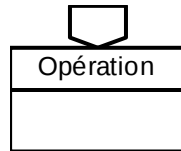


Figure 39 : Formalisme de l'opération

Exemple :

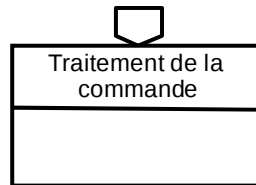


Figure 40 : Exemple d'opération

- **Action :**

Définition :

L'action compose l'opération et représente un traitement élémentaire (créer, modifier, supprimer, lire...)

Formalisme :

Les actions sont nommées dans l'opération, dans l'ordre où elles se déroulent. Elles sont représentées par un verbe à l'infinitif.

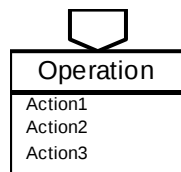


Figure 41 : Formalisme de l'action

Exemple :

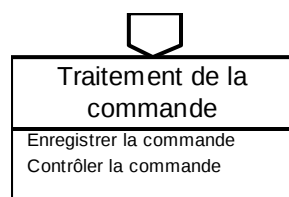


Figure 42 : Exemple d'actions

- **Règle de gestion :**

Définition :

La règle de gestion est l'expression littérale décrivant les contraintes de gestion applicables à l'action.

Exemple :

La facturation des produits est soumise à la TVA.

- **Synchronisation :**

Définition :

Il est fréquent de rencontrer des opérations dont le déclenchement est conditionné par plusieurs événements. Il est donc nécessaire de représenter ces conditions de déclenchement, c'est à dire de préciser quelles sont les associations d'événements dont la présence est indispensable au déclenchement de l'opération. Cette représentation se fait par la synchronisation.

La synchronisation est à la fois une association d'événements « candidats » et une expression booléenne formée à partir des opérateurs « ET » et « OU ». Toutes les combinaisons entre ces opérateurs sont admises.

Formalisme :

Le ou les événements sont mentionnés par un label dans la partie supérieure de l'opération.

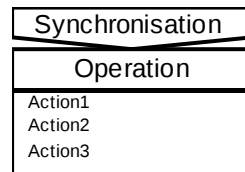


Figure 43 : Formalisme de la synchronisation

Exemple :

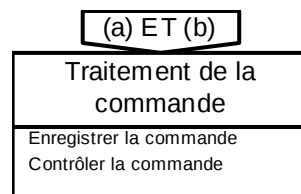


Figure 44 : Exemple de synchronisation

Dans l'exemple ci-dessus, l'opération « Traitement de la commande » sera déclenchée par la présence simultanée des événements (a) **ET** (b).

- **Les règles d'émission :**

Définition :

Les règles d'émission sont les précisions concernant le déclenchement d'un résultat ou d'un autre en fonction d'un traitement conditionnel (vrai, faux, solde positif, solde négatif...)

Formalisme :

La ou les règles d'émission sont mentionnées dans la partie inférieure de l'opération par un terme simple et compréhensible (vrai, faux, arrhes versées, arrhes non versées...)

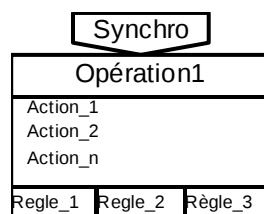


Figure 45 : Formalisme des règles d'émission

Exemple :

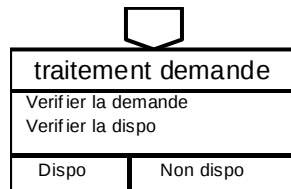


Figure 46 : Exemple de règles d'émission

Dans l'exemple ci-dessus, l'opération traitement de la demande se décompose en deux actions : « Vérifier la demande », puis « Vérifier la disponibilité ». A cet instant, le traitement est interrompu en fonction du résultat. Il y aura ensuite deux traitements différents selon qu'il y ait disponibilité ou non.

- **Le résultat (ou l'émission)**

Définition :

Le résultat est produit par l'opération. Il est la réponse du système à la contrainte de traitement de l'information portée par le ou les événements déclencheurs de l'opération.

Une même opération peut générer plusieurs résultats.

Le résultat, peut devenir ensuite un événement déclencheur pour une opération suivante.

Formalisme :

La représentation du résultat est la même que celle de l'émission.



Figure 47 : Formalisme du résultat

Exemple :

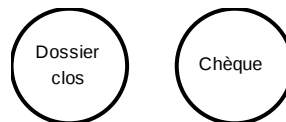


Figure 48 : Exemple de résultats

- **Comment élaborer le MCT**

Il faut commencer par répertorier les événements. Cette étape est relativement simple, car, si les flux ont déjà été identifiés, ces derniers se traduisent en événements. Les événements sont à l'origine des traitements en entrée ou en sont le résultat en sortie. Il faut identifier les processus et distinguer les événements externes et internes. Bien sûr, on écarte les acteurs (puisque nous sommes au niveau conceptuel). On place ensuite les événements et les résultats par ordre d'arrivée logique. D'où l'intérêt de numéroter les flux afin de faciliter ce travail. Ensuite, on définit les opérations en regroupant les traitements ininterrompibles. Il reste ensuite à modéliser le MCT. Pour ce faire, il est nécessaire d'identifier les événements d'entrée qui déclenchent les opérations, et les résultats de ces opérations. Puis, il faut déterminer les conditions de synchronisation avec les règles d'émissions.

- **Vérification du MCT**

L'élaboration d'un MCT répond à un certain nombre de règles qu'il convient de respecter.

Non interruption des opérations :

Il faut considérer l'opération comme une boîte noire, qui, une fois mise en route doit aboutir à une fin. Pour cela, il lui faut les données dont elle a besoin pour son exécution, dès qu'elle se déclenche.

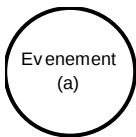
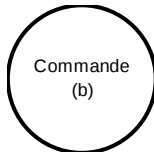
Notion de cycle :

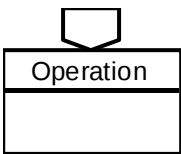
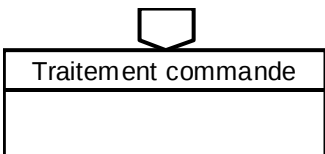
La présence d'un cycle dans un MCT n'est pas interdite dès lors qu'une sortie du modèle est possible.

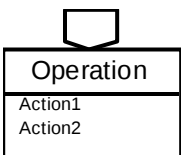
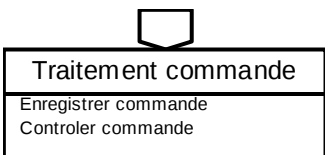
Emissions non accessibles :

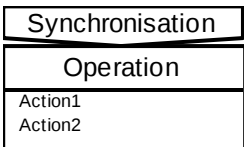
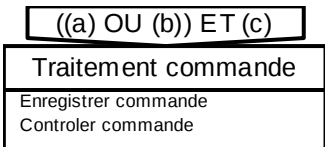
Chaque fois qu'une opération comporte des émissions, il faut vérifier que toutes les émissions sont accessibles.

• Résumé du MCT :

Événement		
Définition	Symbole Associé	Exemple
Représentation d'un fait qui intéresse le SI. Il peut être associé à un alias		

Opération		
Définition	Symbole Associé	Exemple
Traitement effectué sur le SI en réaction à l'apparition d'un événement.		

Action		
Définition	Symbole Associé	Exemple
Traitement élémentaire sur le SI. Composant d'une opération		

Synchronisation		
Définition	Symbole Associé	Exemple
Conditions logiques liant les événements déclencheurs à l'opération. Utilise les opérateurs booléens		

Règle d'émission		
Définition	Symbole Associé	Exemple
Condition de sortie d'une opération	Synchronisation Operation Action1 Action2 Regle 1 Regle 2	((a) OU (b)) ET (c) Traitement commande Enregistrer commande Controler commande Accord Refus

Résultat		
Définition	Symbole Associé	Exemple
Information nécessaire à la gestion, produite par une opération et pouvant constituer un événement pour une opération postérieure	Résultat (a)	Commande (b)

• Exemple d'un MCT complet :

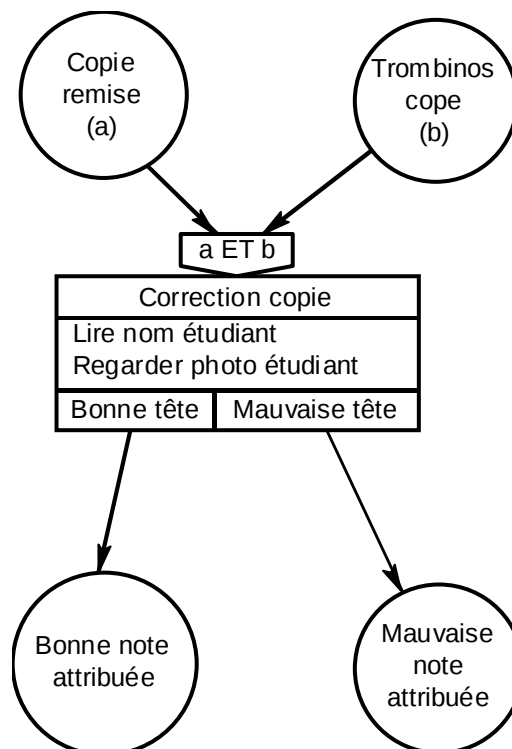


Figure 49 : Un MCT complet

3.4.2. Le niveau organisationnel

Après avoir étudié le niveau conceptuel, il est bon de s'attacher à l'organisation du futur SI. A ce stade, il va falloir étudier :

- La répartition des traitements entre l'homme et la machine ;
- Le mode de fonctionnement : temps réel ou différé ;

- Et l'affectation des données et des traitements par type de site organisationnel et par type de poste de travail.

Les modèles associés au niveau organisationnel sont :

- Le Modèle Organisationnel de Données (MOD) qui représente l'ensemble des données par type de site organisationnel ; le formalisme identique à celui du MCD ;
- Le Modèle Organisationnel des Traitements (MOT) qui permet de représenter par procédure les phases et les tâches exécutées par chaque poste de travail.

Le niveau organisationnel permet donc de décrire le « Qui fait quoi, quand et où ? »

3.4.2.1. Le Modèle Organisationnel des Données (MOD)

Bien que peu utilisé, le MOD permet de faire des choix d'informatisation des données et de faire la quantification et l'historisation des données et de définir la politique d'accès aux données.

- **Choix d'informatisation des données**

Il s'agit ici de choisir quelles seront les données mémorisées à l'aide des moyens informatiques. Les autres données resteront traitées manuellement.

- **Quantification et historisation des données**

L'informatisation est liée à la définition de l'espace de mémorisation. Il est donc nécessaire d'évaluer correctement le volume des données à mémoriser.

Cependant, ce travail peut-être fait au niveau conceptuel lors de l'établissement du dictionnaire de données. Il suffit de « typer » la donnée (par exemple : entier, caractère), de fixer sa longueur et de préciser le nombre d'occurrences.

- **Droits d'accès aux données**

Les données d'un système d'information ne sont pas nécessairement accessibles par l'ensemble du personnel ayant accès au SI. Par conséquent, il est nécessaire d'étudier les droits en consultation, d'ajout, de modification, et de suppression des données pour chacun des profils des utilisateurs.

Les résultats sont ensuite synthétisés dans une matrice.

3.4.2.2. Le modèle Organisationnel des Traitements (MOT)

Les objectifs du Modèle Organisationnel des Traitements (MOT) sont d'enrichir le Modèle Conceptuel des Traitements (MCT) en ajoutant les aspects organisationnels. Le MOT décrit l'organisation à mettre en place pour exécuter les traitements définis par le gestionnaire.

Les notions de temps, de déroulement (durée), de ressources, de lieu et de responsabilité (poste de travail) et de nature des traitements (manuels ou automatiques) vont s'intégrer au MCT et constituer le MOT.

Pour une même description conceptuelle, il y a diverses représentations organisationnelles.

Pour chaque ensemble de traitement, le MOT précise le poste de travail associé, la nature des tâches décrites (M pour manuel, I pour informatisé et immédiat, D pour informatisé et différé) et situe le traitement dans le temps. L'unité de traitement du MOT est une procédure fonctionnelle ou « pf ». A chaque opération du MCT correspond une ou plusieurs pf. Les pf peuvent elles-mêmes se décomposer en tâches.

La tâche informatisée interactive met en jeu l'homme et la machine, la tâche informatisée automatique ne fait appel qu'à la machine.

Les concepts utilisés par le MOT sont les suivants :

- **Procédure :**

Définition :

La procédure est un sous-ensemble organisationnel d'un processus. Le processus « Gestion des transactions » peut donner lieu aux procédures « Gestion des transactions à l'agence » et « Gestion des transactions sur Minitel ».

- **Acteur :**

Définition :

L'acteur est une entité organisationnelle présentant un intérêt pour le SI étudié. Ce concept d'acteur est le même que celui employé dans le MCC.

Formalisme :

L'acteur est représenté par son nom en tête de colonne. L'acteur peut être humain ou bien une machine.

- **Événement :**

Définition :

L'événement matérialise un fait qui, en se produisant, déclenche une réaction du système. Il peut être associé à un alias pour être référencé simplement dans une condition de synchronisation.

Formalisme :

Le formalisme de l'événement du MOT est le même que celui du MCT, c'est à dire qu'il est représenté par un cercle portant son nom.

- **Phase :**

Définition :

La phase est au MOT ce que l'opération est au MCT.

La phase représente la réaction du système d'information à l'apparition d'événements. Elle doit être ininterrompible. Elle est caractérisée par des attributs tels que la période (tous les jours, tous les mois, fin d'exercice, à la demande...), la durée, le type (manuel, interactif, automatique, temps différé).

Formalisme :

Le formalisme est le même que celui employé pour le MCT avec l'opération.

- **Tâche :**

Définition :

La tâche est au MOT ce que l'action est au MCT.

La tâche compose la phase et représente un traitement élémentaire (créer, modifier, supprimer, lire...). Comme la phase, elle est typée (manuelle, automatisée...).

Formalisme :

Le formalisme employé est le même que celui du MCT pour l'action.

- **Règle d'organisation :**

Définition :

La règle d'organisation est au MOT ce que la règle de gestion est au MCT.

La règle d'organisation est l'expression littérale décrivant les contraintes d'organisation applicables à la tâche. La règle d'organisation correspond souvent à une règle de gestion enrichie d'une dimension organisationnelle.

Exemple :

Une transaction à l'agence n'est possible que du lundi au vendredi.

- **Règle d'émission :**

Définition :

La règle d'émission est la précision concernant la condition de déclenchement d'un résultat ou d'un autre en fonction d'un traitement conditionnel (vrai, faux, solde positif, solde négatif...).

Formalisme :

Le formalisme employé est identique au formalisme employé dans le MCT.

- **Synchronisation :**

Définition :

La synchronisation est la condition logique reliant les événements déclencheurs à la phase.

Formalisme

Le formalisme employé est identique au formalisme employé dans le MCT

- **Résultat :**

Définition :

Le résultat est produit par la phase. Il est la réponse du système à la contrainte de traitement de l'information portée par le ou les événements déclencheurs de la phase.

Une même phase peut générer plusieurs résultats.

Le résultat, peut devenir ensuite un événement déclencheur pour une phase suivante.

Formalisme :

Le formalisme employé est identique au formalisme employé dans le MCT

- **Module :**

Définition :

Le module est un élément de logiciel (écran, programme, édition...) associé à un fichier de programme source et rattaché à une tâche. Le module permet de réaliser un lien entre la description graphique des traitements et leur implémentation sous forme logicielle.

- **Correspondance entre les concepts du MCC, du MCT et du MOT**

MCC	MCT	MOT
	Processus	Procédure
Acteur		Acteur
	Evénement	Evénement
	Opération	Phase
	Action	Tâche
	Règle de gestion	Règle d'organisation
	Condition de synchronisation	Condition de synchronisation
	Règle d'émission	Règle d'émission
	Résultat	Résultat
		Module

- **Comment élaborer le MOT**

Le MOT est élaboré à l'aide du MCC et du MCT.

Il suffit ensuite d'intégrer dans le MCT les aspects organisationnels en ajoutant autant de colonnes qu'il y a d'acteurs, une colonne pour préciser la notion de temps et de durée, et une colonne pour déterminer le type de traitement.

Il faudra toutefois vérifier que toutes les tâches de la phase sont effectuées par le même acteur. Sans quoi il faudra éclater la phase autant de fois qu'il y a d'acteurs différents qui participent à son exécution.

- **Exemple de MOT complet**

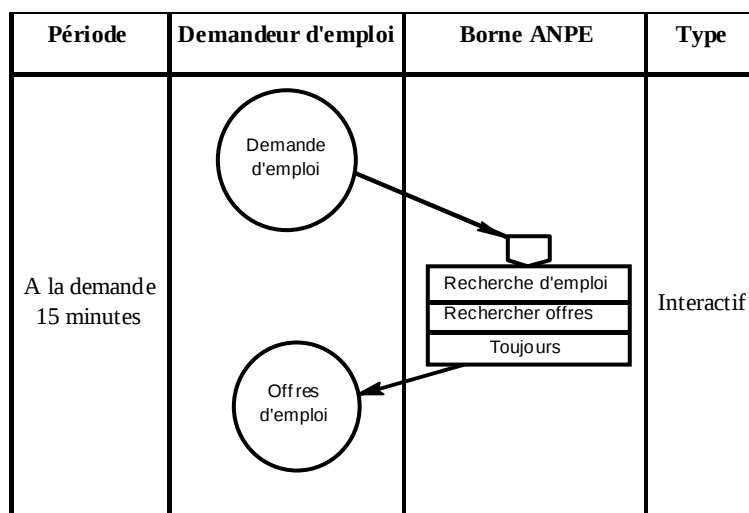
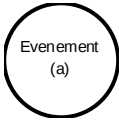


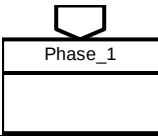
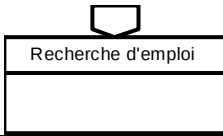
Figure 50 : Exemple de MOT complet

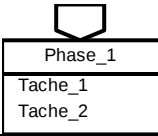
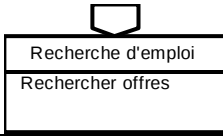
- **Résumé du MOT**

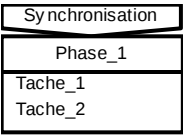
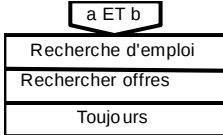
Procédure		
Définition	Symbole Associé	Exemple
Sous-ensemble organisationnel d'un processus	Aucun	Procédure « Gestion des demandes d'emploi à la borne ANPE »

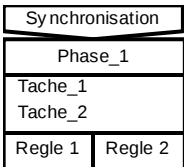
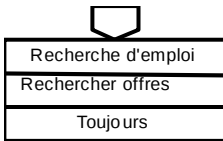
Acteur		
Définition	Symbole Associé	Exemple
Entité organisationnelle présentant un intérêt pour le SI	Colonne du MOT	Demandeur d'emploi, Borne ANPE

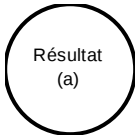
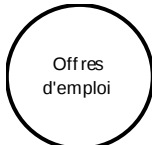
Evénement		
Définition	Symbole Associé	Exemple
Représentation d'un fait qui intéresse le SI. Il peut être associé à un alias		

Phase		
Définition	Symbole Associé	Exemple
Réaction du système d'information à l'apparition d'un ou plusieurs événements..		

Tâche		
Définition	Symbole Associé	Exemple
Traitement élémentaire sur le SI. Composant d'une phase.		

Synchronisation		
Définition	Symbole Associé	Exemple
Conditions logiques liant les événements déclencheurs à la phase.		

Règle d'émission		
Définition	Symbole Associé	Exemple
Condition de sortie d'une phase		

Résultat		
Définition	Symbole Associé	Exemple
Information nécessaire à la gestion, produite par une phase et pouvant constituer un événement pour une phase postérieure		

3.4.3. Les niveaux logique et physique

Alors que le SI a été décrit au niveau conceptuel et organisationnel, il est maintenant nécessaire de le décrire sur le plan informatique. Cette description est décomposée en deux niveaux : le niveau logique et le niveau physique.

▪ Le niveau logique :

Concernant les traitements, il est difficile aujourd'hui de dire qu'il existe une normalisation de la description logique des traitements qui correspondrait à un véritable Modèle Logique des Traitements. C'est pourquoi ce modèle ne sera pas étudié dans ce cours.

Concernant les données, le Modèle Logique des Données (MLD) permet de prendre en compte la structuration technique propre au stockage informatisé. L'utilisation de l'informatique permet d'améliorer l'organisation et la gestion d'un SI. Depuis le début du cours, il a été vu comment constituer théoriquement une base de données. Cette base de données peut être réalisée et gérée par un Système de Gestion de Bases de Données Relationnelles (SGBDR).

Un SGBDR va permettre d'organiser un SI, de saisir ses données, de les modifier, de les supprimer et de les consulter.

La plupart des SGBDR courants sont fondés sur le modèle relationnel défini par E.F Codd en 1970. Le modèle relationnel présente deux aspects fondamentaux : des concepts structuraux de base comme la table, un algèbre permettant de manipuler les tables, ou une collection de tables.

Le Modèle Relationnel des Données (MRD) (Parfois appelé Modèle Logique des Données (MLD)) s'est largement imposé depuis la fin des années 80, et est donc devenu un standard de fait pour la description du niveau logique.

Cependant, le MLD ne tient pas compte des contraintes techniques comme la capacité mémoire ou des particularités dues à un logiciel. C'est au niveau du Modèle Physique des Données qu'elles seront prises en compte.

▪ Le niveau physique :

Au niveau physique, les choix des outils techniques sont définis. Ainsi, les organisations physiques de données sont spécifiées au travers de la Description Physique des Données.

De la même manière, la spécification des traitements est réalisée pour chaque transaction (temps réel) ou chaque unité de traitement (temps différé) au travers de la description opérationnelle des traitements.

En résumé, au niveau physique, aucun formalisme universel n'est disponible aujourd'hui. Cette partie ne sera donc pas traitée.

3.4.3.1. Le Modèle Relationnel des Données (MRD)

Les MCD et MOD représentent donc l'ensemble des données d'un point de vue abstrait, sans tenir compte des contraintes d'implémentation dans une base de données ou des contraintes techniques associées.

Ces différents points sont justement décrits dans le Modèle Relationnel des Données (MRD) à l'aide des concepts suivants :

- **Table (ou relation)**

Définition :

La table constitue la principale structure de stockage dans une base relationnelle. Elle a une structure de tableau, composée de lignes et de colonnes dans lesquelles on stocke les informations. La table est désignée par un nom.

La table du modèle relationnel correspond à l'entité du MCD.

Exemple :

La table « VOITURE », la table « PERSONNE ».

- **Attribut**

Définition :

L'attribut représente une colonne d'une table caractérisée par un nom. Une table possède autant d'attributs que d'informations à gérer.

L'attribut du modèle relationnel correspond à la notion de propriété dans le MCD.

Exemple :

La table « VOITURE » contient les attributs « marque », « couleur ».

La table « PERSONNE » contient les attributs « Nom_Personne », « Prénom_Personne », « DateNaissance_Personne »...

- **Tuple**

Définition :

Un tuple, ou occurrence de la table (ou de la relation) correspond à une ligne du table. Il y aura autant de tuples qu'il y aura de valeurs. Le tuple correspond à l'enregistrement d'une ligne de la table.

Le tuple du MLD est l'équivalent de l'occurrence du MCD.

- **Clé primaire**

Définition :

La clé primaire est un attribut particulier dont les valeurs définissent de manière unique les tuples de la table (ou de la relation).

La clé primaire du MLD correspond à l'identifiant du MCD.

Exemple :

Soit la table « PERSONNE » contenant les attributs « Num_Personne », « Nom_Personne », « Prénom_Personne ». L'attribut « Num_Personne » permet de définir de manière unique toutes les personnes.

- ♦ Tuple1 : 0001 Rossi Tino
- ♦ Tuple2 : 0002 Haddock Capitaine
- ♦ Tuple3 : 0010 Lagaffe Gaston

Formalisme :

La clé primaire est soulignée d'un trait continu.

- **Clé étrangère**

Définition :

La clé étrangère est un attribut particulier correspondant à une clé primaire de la table de référence.

Une table peut contenir 0, 1 ou plusieurs clé étrangères.

Formalisme :

La clé étrangère est soulignée d'un trait pointillé, ou bien suivie du signe #.

- **Exemple d'une table**

Formalisme :

Table_1 (Cle_Primaire, Attribut_1, Attribut_2, Attribut_n, CleEtrangère_1, CleEtrangère_n)

Ou

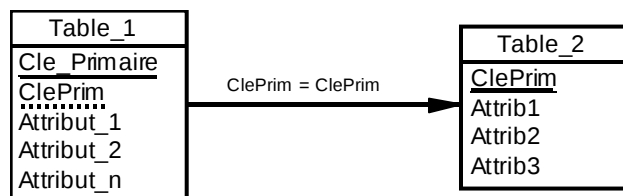


Figure 51 : Représentation graphique du MLD relationnel

Exemple :

A partir du MCD suivant :

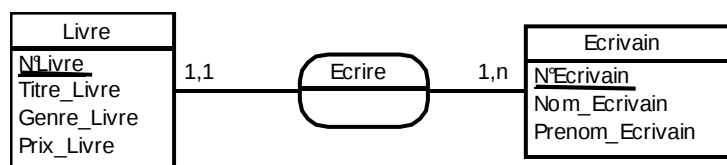


Figure 52 : Un MCD avant de le Transformer en MLD

On obtient le modèle relationnel suivant :

LIVRE (N°Livre, Titre_Livre, Genre_Livre, Prix_Livre, N°Ecrivain)

ECRIVAIN (N°Ecrivain, Nom_Ecrivain, Prenom_Ecrivain)

La représentation graphique du MLD donne :

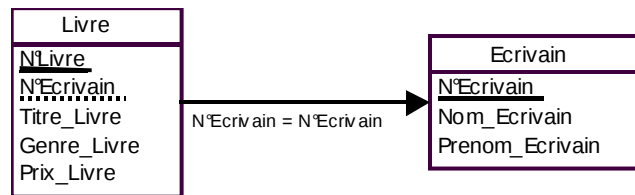


Figure 53 : Exemple de MLD

- **Correspondance entre les concepts du MCD et ceux du MLD**

MCD	MLD
Domaine	Domaine
Entité	Table (ou relation)
Événement	Événement
Occurrence	Tuple
Propriété	Attribut
Identifiant	Clé primaire
Association	Table ou référence
	Clé étrangère

- **Règles de passage d'un MCD à un MLD**

Règles pour les entités du MCD

- ♦ L'entité se transforme en une table ;
- ♦ L'identifiant de l'entité devient la clé primaire de la table ;
- ♦ Les propriétés de l'entité deviennent des attributs de la table.

Règle pour les relations binaires (ou réflexives) de type (0,n) ou (1,n) – (0,1) ou (1,1)

Une relation binaire (ou réflexive) ayant des type (0,n) ou (1,n) – (0,1) ou (1,1) se traduit par une redondance de l'identifiant de l'objet à cardinalité (1,n) ou (0,n) dans la table issue de l'entité à cardinalité (1,1) ou (0,1). L'identifiant de l'entité à cardinalité (1,1) devient la clé primaire de la table. La propriété dupliquée devient clé étrangère dans la table. Si la relation est réflexive, c'est l'identifiant de l'entité qui est dupliqué dans la table issue de ce même objet après avoir été renommé.

Si l'association est porteuse de données, celles-ci se retrouvent comme attributs dans la relation issue de l'entité à cardinalité (1,1) ou (0,1).

- ♦ Les MCD à relation binaire ci dessous :

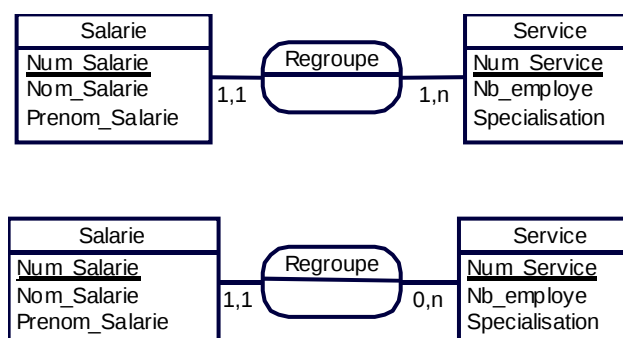


Figure 54 : Une relation binaire à cardinalités (0,n) ou (1,n) – (0,1) ou (1,1)

Deviennent dans les deux cas :

Dans leur forme graphique :



Figure 55 : MLD issue d'un MCD à relation de type (0,n) ou (1,n) - (0,1) ou (1,1)

Dans leur forme littérale :

SALARIE (Num_Salarie, Nom_Salarie, Prenom_Salarie, Num_Service)

SERVICE (Num_Service, Nb_employe, Specialisation)

♦ Le MCD à relation binaire et reflexive ci dessous :

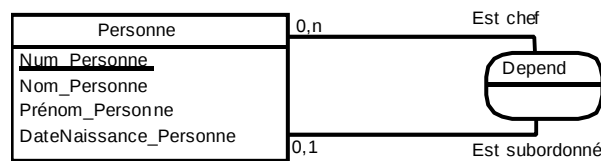


Figure 56 : Relation reflexive de type (0,n) ou (1,n) - (0,1) ou (1,1)

Devient :

Dans sa forme graphique :

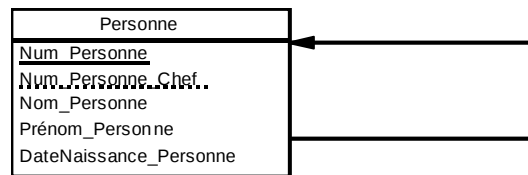


Figure 57 : MLD issu d'un MCD à relation reflexive de type (0,n) ou (1,n) - (0,1) ou (1,1)

Dans sa forme littérale :

PERSONNE (Num_Personne, Num_Personne_Chef, Nom_Personne, Prenom_Personne, DateNaissance_Personne)

Règle pour les relations binaire de type (0,n) ou (1,n) - (0,n) ou (1,n).

Dans le cas d'une relation de type (0,n) ou (1,n) – (0,n) ou (1,n), chaque entité se transforme en table, chaque identifiant se transforme en clé primaire de la table. L'association devient une table qui récupère les clé primaires des deux tables. Si l'association était porteuse de données, alors ces données deviennent des attributs.

Le MCD ci-dessous :

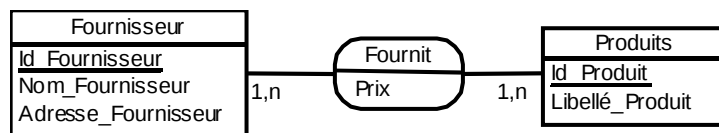


Figure 58 : MCD à relation de type (0,n) ou (1,n) – (0,n) ou (1,n)

Devient :

Dans sa forme graphique :

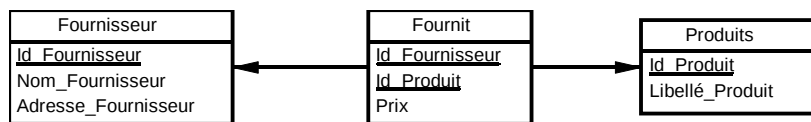


Figure 59 : MCD issu d'un MCD à relation de type $-(0,n)$ ou $(1,n)$ - $(0,n)$ ou $(1,n)$

Dans sa forme littérale :

FOURNISSEUR (Id_Fournisseur, Nom_Fournisseur, Adresse_Fournisseur)

FOURNIT (Id_Fournisseur, Id_Produit, Prix)

PRODUITS (Id_Produit, Libellé_Produit)

♦ Le MCD à relation binaire et reflexive ci dessous :

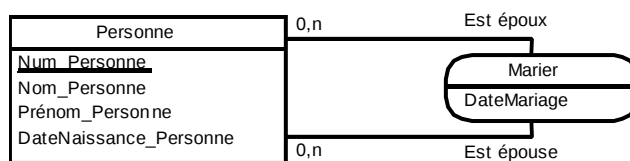


Figure 60 : Relation reflexive de type $(0,n)$ ou $(1,n)$ - $(0,n)$ ou $(1,n)$

Devient :

Dans sa forme graphique :

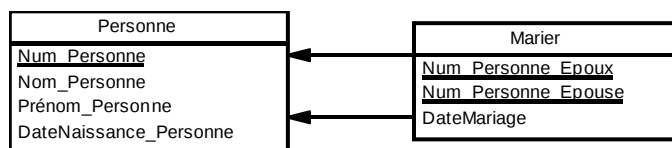


Figure 61 : MLD issu d'un MCD à relation reflexive de type $(0,n)$ ou $(1,n)$ - $(0,n)$ ou $(1,n)$

Dans sa forme littérale :

PERSONNE(Num_Personne, Nom_Personne, Prénom_Personne, DateNaissance_Personne)

MARIER (Num_Personne_Epoux, Num_Personne_Epouse, DateMariage)

Règle pour les relations n-aire

Une relation n-aire du MCD, porteuse ou non de données, se transforme en une table du modèle relationnel ayant comme clé primaire composite les identifiants des entités participant à cette association du MCD.

Le MCD ci dessous :

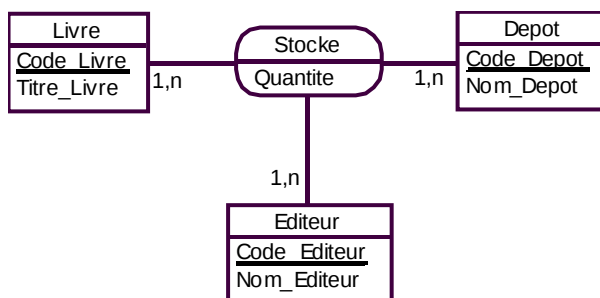


Figure 62 : MCD à relation n-aire

Devient :

Dans sa forme graphique :

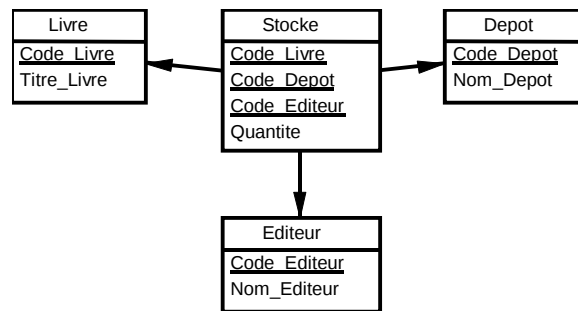


Figure 63 : MLD issu d'un MCD à relation n-aire

Dans sa forme littérale :

LIVRE (Code_Livre, Titre_Livre)

DEPOT (Code_Depot, Nom_Depot)

EDITEUR (Code_Editeur, Nom_Editeur)

STOCKE (Code_Livre, Code_Depot, Code_Editeur, Quantite)

Cas des relations de type (1,1) – (0,1)

La transformation d'une relation de type (1,1) – (0,1) se traduit par une double redondance des identifiants des deux entités.

Le MCD ci dessous :

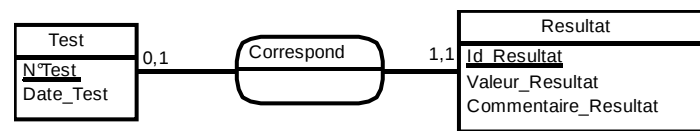


Figure 64 : Une relation binaire à cardinalités (1,1) - (0,1)

Devient :

Dans sa forme graphique :

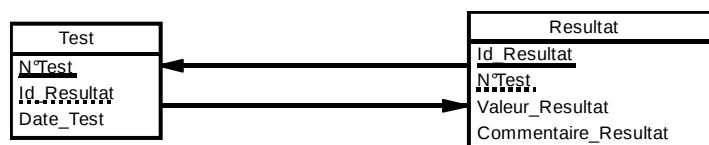


Figure 65 : MLD issu d'un MCD à relation de type (1,1) - (0,1)

Dans sa forme littérale :

TEST (N°Test, Date_Test, Id_Resultat)

RESULTAT (Id_Resultat, Valeur_Resultat, Commentaire_Resultat, N°Test)

Cas des héritages :

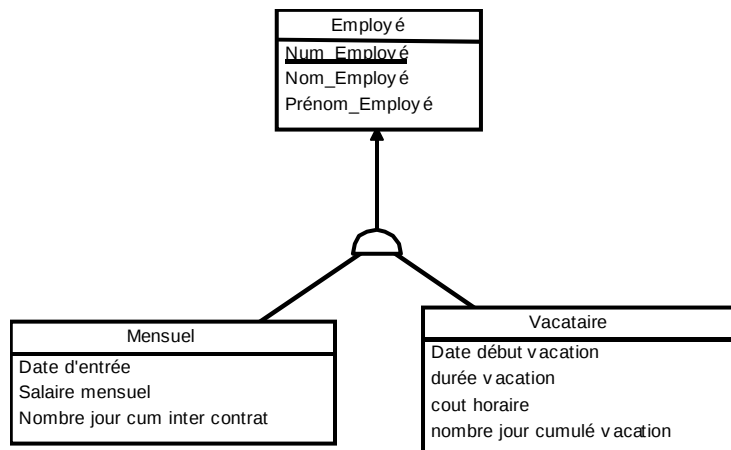


Figure 66 : Un héritage.

Pour la transformation d'un MCD contenant un héritage en un MLD, trois cas sont possibles.

♦ La généralisation :

L'entité générique se transforme en une table, il est possible d'insérer une propriété discriminante afin de différencier les occurrences de l'entité « père ». Toutes les propriétés des entités « filles » sont migrées dans l'entité générique.

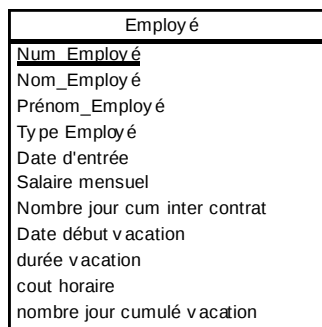


Figure 67 : Lien d'héritage avec généralisation.

♦ La spécialisation :

Les entités spécialisées deviennent des tables et héritent des propriétés de l'entité générique.

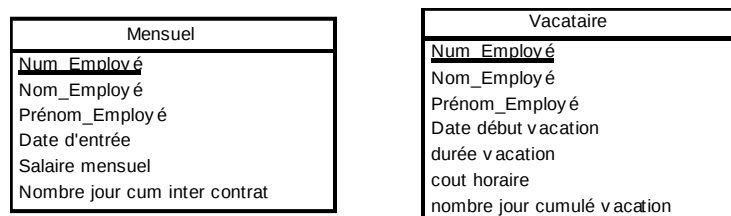


Figure 68 : Lien d'héritage avec spécialisation.

♦ La solution mixte :

L'entité type devient une table. Les entités spécialisées deviennent également des tables. Et, l'identifiant de l'entité type est dupliqué dans les tables issues des entités spécialisées.

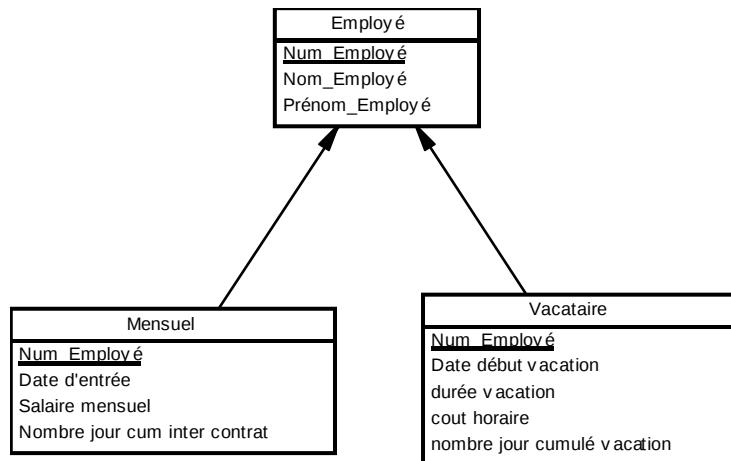


Figure 69 : Lien d'héritage mixte - Seul l'identifiant est hérité.

Il est également possible de faire hériter toutes les propriétés dans les tables issues des entités spécialisées.

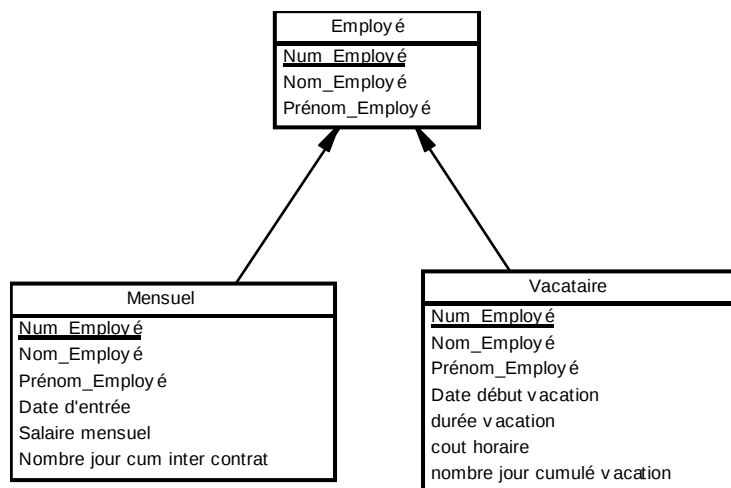


Figure 70 : Lien d'héritage mixte - Toutes les propriétés sont héritées.

- **Normalisation du modèle relationnel**

Afin de valider le modèle relationnel, le concepteur du modèle relationnel, Codd, a défini des règles de normalisation. Si, lors de l'application des formes normales, le modèle relationnel n'est pas validé, il est alors possible de la modifier ainsi que le MCD.

Première forme normale (1FN)

Une relation (ou une table) est en première forme normale si tout attribut est élémentaire et s'il ne peut pas prendre plusieurs valeurs. Plus clairement, une relation (ou une table) est en première forme normale si ses attributs ne sont pas décomposables et si le nombre d'attributs est constant, quelle que soit l'occurrence de la relation.

C'est le principe de la non répétitivité.

Exemple : Cas d'une commande

COMMANDE (Code_Commande, Date_Commande, N°Article, Quantité)

Commande
<u>Code_Commande</u>
Date_Commande
N°Article
Quantité

Figure 71 : La table "Commande" avant normalisation en 1FN

Cette table n'est pas en 1FN car « N°Article » et « Quantité » est un groupe répétitif.

Par conséquent, la table COMMANDE se décompose comme suit :

COMMANDE (Code_Commande, Date_Commande)

LIGNE (N°Article, Quantité, Code_Commande)

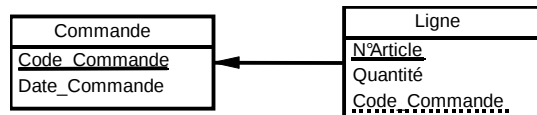


Figure 72 : La première forme normale

Deuxième forme normale (2FN)

Une table est en 2FN si :

- ♦ Elle est en 1FN.
- ♦ Aucun de ses attributs ne dépend partiellement de l'identifiant ou de la clé primaire.

Exemple :

LIGNE (N°Article, Code commande, Quantité, Description_Article)

Ligne
<u>N°Article</u>
<u>Code commande</u>
Quantité
Description_Article

Figure 73 : La table "Ligne" avant normalisation en 2FN

Cette table n'est pas en 2FN car « Description_Article » ne dépend que d'une partie de la clé « N°Article ».

Par conséquent, la table LIGNE se décompose comme suit :

LIGNE (N°Article, Code commande, Quantité)

ARTICLE (N°Article, Description_Article)

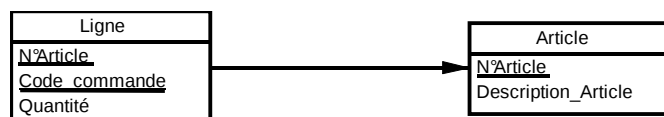


Figure 74 : La deuxième forme normale

Troisième forme normale (3FN)

Une table est en 3FN si :

- ♦ Elle est en 2FN.
- ♦ Aucun de ses attributs ne dépend d'un attribut autre que l'identifiant (clé primaire).

Exemple :

COMMANDE (Code_Commande, Date_Commande, N°_Client, Nom_Client)

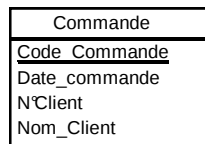


Figure 75 : La table "Commande" avant normalisation en 3FN

Cette table n'est pas en 3FN car « Nom_Client » ne dépend que de « N°_Client » qui n'est pas une clé primaire.

Par conséquent, la table COMMANDE se décompose comme suit :

COMMANDE (Code_Commande, Date_Commande, N°_Client)

CLIENT (N°_Client, Nom_Client)

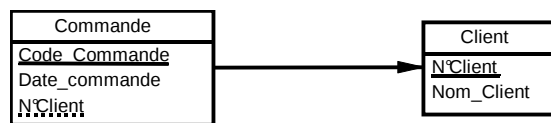


Figure 76 : La troisième forme normale

4. Positionnement de la méthode Merise

Après avoir vu l'ensemble des modèles qui composent la méthode Merise, il est essentiel de positionner la méthode dans un projet d'informatisation.

4.1. Positionnement dans le temps

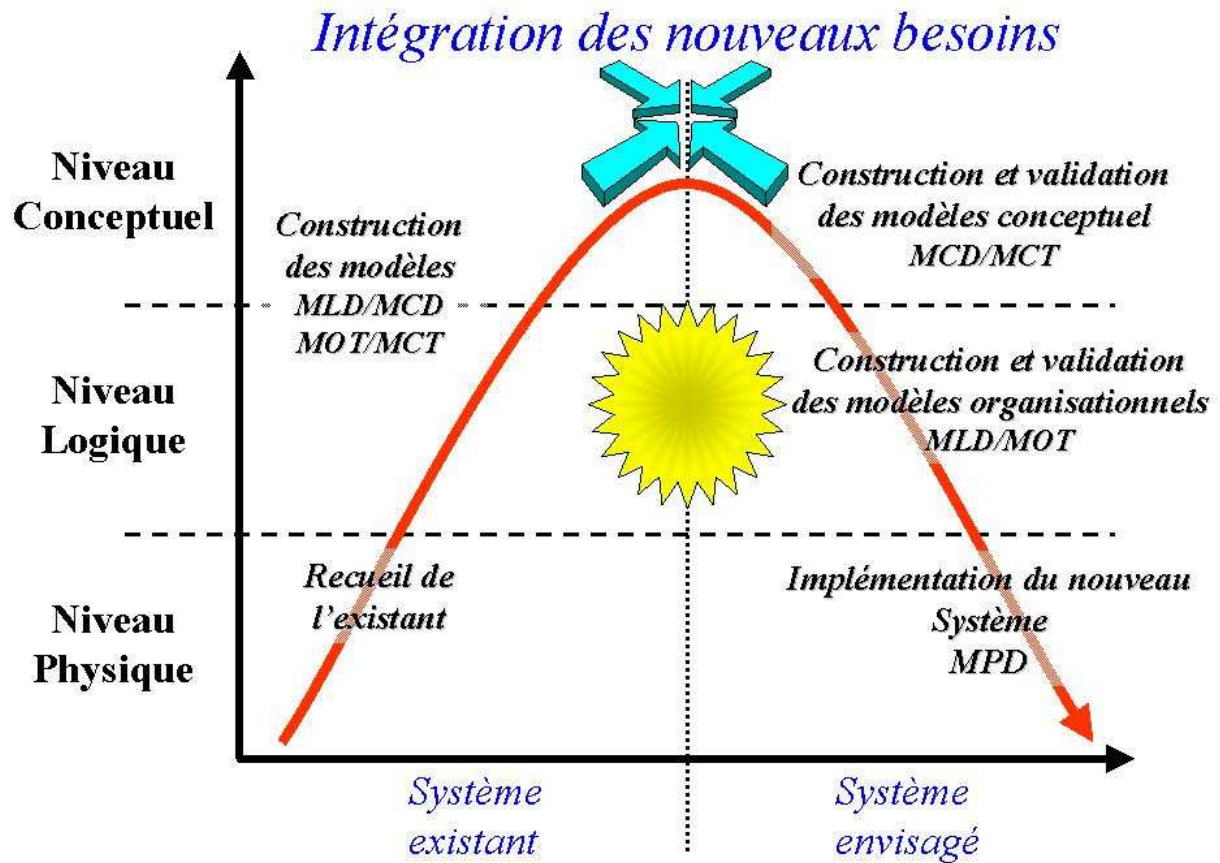


Figure 77 : La courbe du soleil

4.2. Interaction des modèles de la méthode Merise

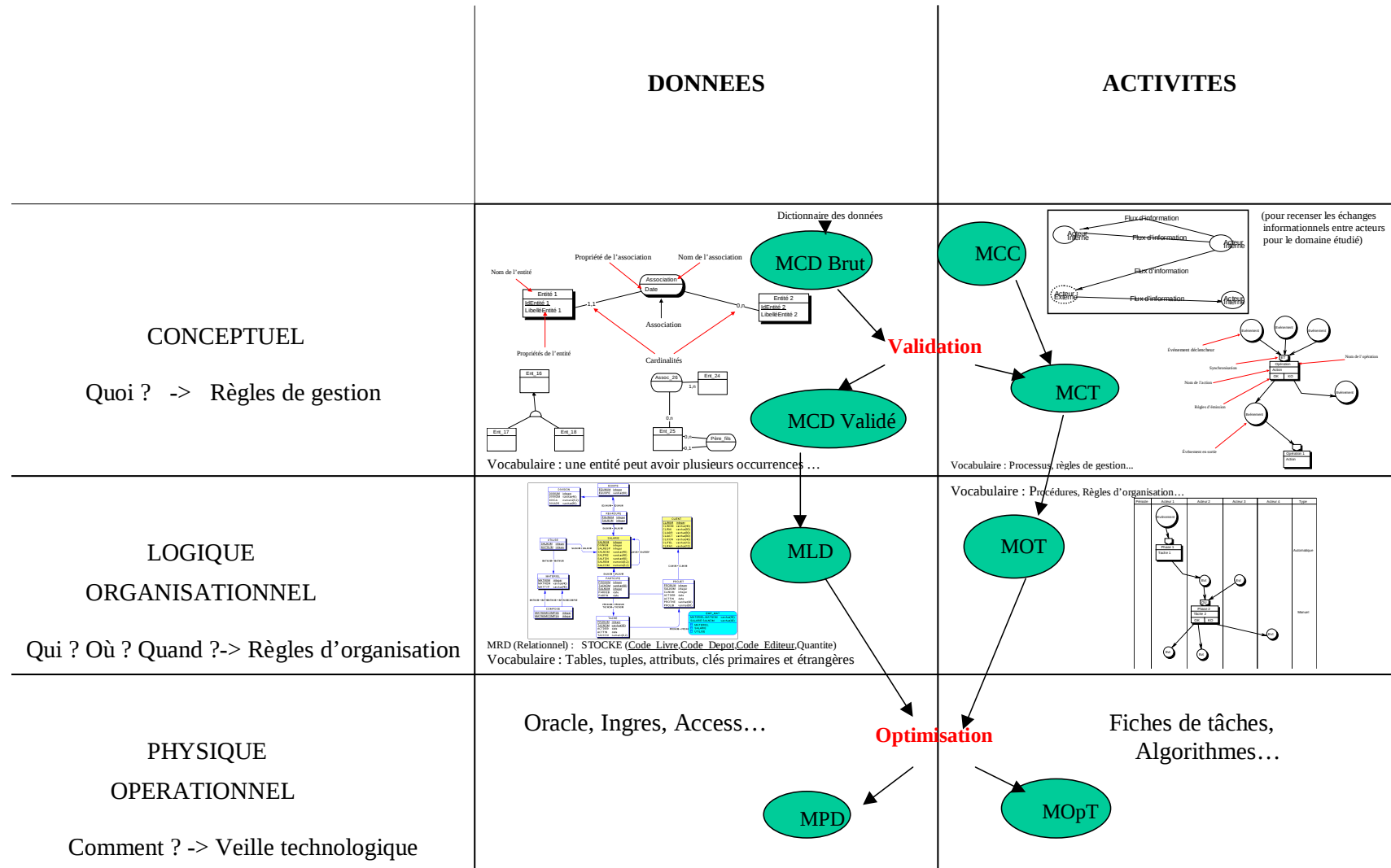


Figure 78 : Les modèles de la méthode Merise

Merise	Louis Pereira	R : INF-LP-MER4-2-I	V : 04/10-1.4
--------	---------------	---------------------	---------------

5. **Conclusion**

Merise est donc une méthode complète à la fois pour la démarche de conduite de projet, mais aussi et surtout pour la conception des systèmes d'information.

Cette méthode, au travers de la séparation des données et des traitements, à différents niveaux d'abstraction, permet d'aboutir à un nouveau système d'information conforme aux besoins.

Toutefois, depuis quelques années, une nouvelle approche émerge, et devrait prendre le pas d'ici quelques années dans la conception de nos futurs systèmes d'informations : la modélisation objet, avec le langage de modélisation UML (Unified Modeling Language).

Contrairement à la méthode Merise, cette méthode ne fait plus la séparation des données et des traitements, et est par ailleurs beaucoup plus complète dans l'analyse des traitements du SI. Cependant, la méthode objet ne propose pas de démarche de conduite de projet.

En bref, Merise n'est pas encore mort, mais UML arrive avec force. Mais, tant que nos bases de données resteront des SGBDR et non des SGBDO, Merise restera largement d'actualité.

6. **Bibliographie**

La bibliographie concernant la méthode MERISE est vaste.

Les ouvrages qui ont retenu mon attention, et m'ont aidé dans l'élaboration de ce cours sont les suivants :

Auteur	Titre	Editeur	Edition	N° ISBN	Nombre de pages prélevées
Joseph Gabay	MERISE et UML pour la modélisation des SI	DUNOD	4è édition	2-100-07821-6	0
Jean-Patrick Matheron	Comprendre Merise, outils conceptuels et organisationnels	EYROLLES	1è édition	2-212-07502-2	0
Dominique Dionisi	L'essentiel sur Merise	EYROLLES	2è édition	2-212-090046-3	0
Gilles Guedj	Mise en œuvre de merise et conception d'applications client-serveur	EYROLLES	1è édition	2-212-08931-7	0

Le dernier ouvrage est particulièrement intéressant pour la pratique de la modélisation des données et des traitements avec AMC Désignor.