

les modèles conceptuels

MERISE 2

concepts de base

démarche et modèles

Equipe d'analyse
département Informatique
IUT 2 Grenoble
Université Pierre Mendès-France

Sommaire

Les notes de ce cours sont entièrement issues de différents livres sur Merise cités en Bibliographie.

1. MERISE/MERISE2	5
1.1. QUELQUES CHIFFRES.....	5
1.2. OBJECTIFS MERISE 2	5
2. NOTIONS DE BASE	7
2.1. LES SYSTÈMES	7
2.2. LE SYSTÈME D'INFORMATION	8
2.3. CONCEPTION D'UN SI	9
<i>Cycle de vie</i>	9
<i>Différences principales entre Merise et Merise 2</i>	10
<i>Les niveaux d'abstraction</i>	12
<i>Parcours au sein des niveaux d'abstraction</i>	13
<i>Un parcours "standard"</i>	14
3. LES MODÈLES.....	17
3.1. MODÈLE.....	17
3.2. LES TROIS AXES DE MODÉLISATION DE MERISE 2.....	17
3.3. AXES DE MODÉLISATION ET NIVEAUX D'ABSTRACTION	18
4. MODÈLES CONCEPTUELS	20
5. LES MODÈLES DE FLUX	22
5.1. MODÈLE DE CONTEXTE (MC)	22
<i>Définition</i>	22
<i>Représentation graphique</i>	22
<i>Concepts associés</i>	23
<i>Utilisation du modèle</i>	23
5.2. MODÈLE DE FLUX CONCEPTUELS (MFC).....	23
<i>Définition</i>	23
<i>Exemple</i>	24
<i>Concepts associés</i>	25
5.3. LES CONCEPTS ASSOCIÉS	25
<i>Domaine d'étude</i>	25
<i>Activité</i>	26
<i>Flux de données</i>	26
<i>Acteur externe</i>	26
<i>Domaine connexe</i>	26
5.4. GAMMES OPÉRATOIRES.....	26
<i>Mécanisme de décomposition</i>	26

<i>Mécanisme de composition</i>	27
<i>Au niveau organisationnel ?</i>	27
6. MODÈLE CONCEPTUEL DE DONNÉES "ÉTENDU"	29
6.1. VOCABULAIRE	29
<i>Propriété</i>	29
<i>Objet type (ou individu)</i>	29
<i>Relation type</i>	29
<i>Cardinalités</i>	29
6.2. LA PROPRIÉTÉ	30
6.3. OBJET, OCCURRENCE D'OBJET	30
6.4. RELATION	32
6.5. CARDINALITÉS	33
6.6. CONTRAINTE D'INTÉGRITÉ FONCTIONNELLE	35
6.7. LE PROBLÈME DE LA REPRÉSENTATION DU TEMPS	36
6.8. VÉRIFICATION DU MODÈLE	39
6.9. LA DÉMARCHE	39
<i>Ecole pragmatique</i>	39
<i>Ecole "formelle" (ou plutôt exhaustive)</i>	39
6.10. SOUS-TYPE D'OBJET ET SOUS-TYPE DE RELATION	40
<i>Sous-type d'objet</i>	40
<i>Sous-type de relation</i>	42
6.11. CONTRAINTES D'INTÉGRITÉ STATIQUE	42
6.11.1. <i>Contraintes de base : couverture et disjonction</i>	42
6.11.2. <i>Contraintes d'intégrité entre les sous-types d'objets</i>	43
6.11.3. <i>Contraintes d'intégrité sur les relations</i>	45
7. LES MODÈLES DYNAMIQUES	49
8. MODÈLE CONCEPTUEL DES TRAITEMENTS (MCT)	49
8.1. VOCABULAIRE DE BASE	49
<i>Événement</i>	49
<i>Opération</i>	50
<i>Résultat</i>	50
<i>Synchronisation</i>	50
8.2. ÉVÉNEMENT	50
8.3. SYNCHRONISATION	51
8.4. OPÉRATION	51
8.5. LE PROCESSUS	53
8.6. LA CONSOMMATION	56
9. MODÈLE CONCEPTUEL DES TRAITEMENTS ANALYTIQUE (MCTA)	58
9.1. OPÉRATION CONCEPTUELLE	58
9.2. ÉTAT D'OBJET	59
9.3. ACTION	61
9.4. CONDITION SUR LES ACTIONS	64
<i>Condition de Déclenchement</i>	64
<i>Condition Itérative</i>	65
9.5. ÉVÉNEMENT INTERNE	65
9.7. PASSAGE DES MODÈLES DE FLUX AU MCTA	67
10. LES CYCLES DE VIE DES OBJETS	68
10.1. INTRODUCTION	68

<i>Evénement et Transition</i>	68
<i>Représentation graphique et exemple partiel</i>	68
<i>Objectifs</i>	68
<i>Commentaires MCTA / CVO</i>	69
10.2. LES PRINCIPALES TRANSITIONS	69
<i>Séquence</i>	69
<i>Alternative</i>	69
<i>Itération</i>	69
10.3. ETAT D'OBJET	70
<i>Le CVO d'un objet comporte :</i>	70
<i>Les états d'un objets sont identifiés à partir :</i>	71
<i>Etats d'objets et sous-types</i>	73
10.4. SPÉCIALISATION DE CVO	73
10.5. VARIABLES D'ÉTAT ET STRUCTURES PARALLÈLES	73
<i>Variables d'état et structures parallèles</i>	73
<i>Les différents cas de structures parallèle</i>	74
10.6. VÉRIFICATION ET ELABORATION DES CVO.....	77
<i>Règles de vérification</i>	77
<i>Enchaînement des modèles (cf. page suivante)</i>	77
<i>Démarche pratique</i>	77

11. BIBLIOGRAPHIE81

Merise 2

1. Merise/Merise2

1.1. Quelques chiffres

- premier contrat de recherche en 1974
- premier ouvrage en 1983
- aujourd'hui 50 000 praticiens et plus de 40 ouvrages
- démarrage du projet Merise 2 en 1989 (Sema Group)

1.2. Objectifs Merise 2

- **fournir** un ensemble complet de démarches, modèles et méthodes pour la conception et le développement de SI
- **conserver** une compatibilité totale avec Merise
- **amélioration** de certains points clés :
 - technique de raffinement
 - intégration des diagrammes de flux de données
 - enrichissement de la sémantique du modèle de données
 - représentation du cycle de vie du système et des objets
 - meilleure distinction entre les niveaux organisationnel et logique
- **extension** vers des aspects techniques
 - prise en compte des nouvelles architectures d'applications
 - prise en compte des nouvelles interfaces
- **adaptation** aux standards en cours de définition
 - démarche et modèles choisis en fonction des types de projets
- **outillage** par des logiciels d'aide à la conception et au développement

2. Notions de base

2.1. Les systèmes

Système : "un tout constitué d'éléments unis par des relations, ces éléments et ces relations étant munis de propriétés"

Exemple : le système "d'entreprise" est composé d'éléments tels que des employés, des services, des articles, des emplacements de stockage... Les propriétés décrivant ces éléments sont le matricule de l'employé, son nom, la référence de l'article.. Entre ces éléments on trouve des relations telles que "est rattaché" entre un employé et un service, "est stocké" entre un article et un emplacement de stockage... Les propriétés de ces relations sont la date d'entrée dans le service, la quantité stockée...

Etat d'un système : l'ensemble des valeurs, à un moment donné, des propriétés des éléments et des relations du système.

Un système vit dans un **environnement**. Il subit de cet environnement des stimuli qui viennent le perturber et l'obligent à réagir c.a.d à **déclencher des activités** qui vont le faire évoluer vers un nouvel état.

Système : "un tout constitué d'éléments unis par des relations, doté d'une activité et répondant à des stimuli".

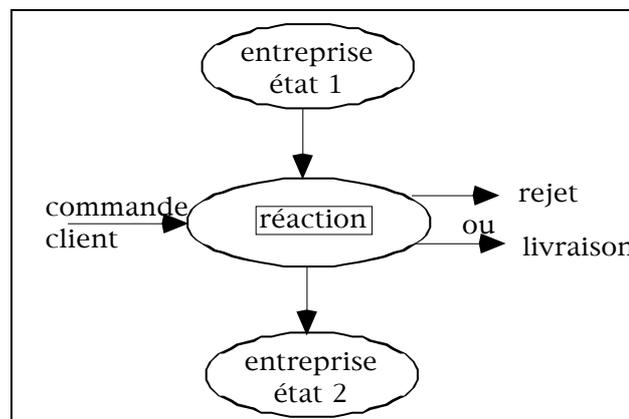


Figure 1 : réactivité du système

L'environnement n'est pas toujours "l'extérieur". Par exemple le processus "expédition" subit une perturbation sous la forme d'un bordereau d'expédition en provenance du processus "traitement de commandes".

Le Moigne propose en 1977 de définir un système comme :

- quelque chose (n'importe quoi, identifiable)
- qui fait quelque chose (activité, fonction),
- qui est doté d'une structure,
- qui évolue dans le temps,
- dans quelque chose (environnement),
- pour quelque chose (finalité).

2.2. Le système d'information

Le concept de SI d'une organisation recouvre deux notions:

- l'organisation réelle se transformant, agissant, communiquant et mémorisant des informations, notion qui apparente le SI à un objet naturel
- le système construit par l'homme pour représenter les actions, la communication et la mémorisation de l'information, notion qui apparente le SI à objet artificiel.

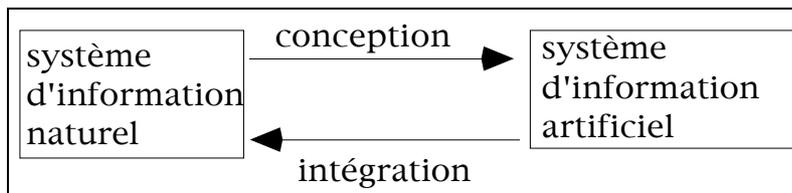


Figure 2 : le SI réel le SI artificiel

système d'information / système informatisé / système informatique

Une partie du S. d'information peut être (doit être) informatisé (le S. informatisé). Le système informatisé s'appuie sur un S. informatique (matériel, logiciel, réseau..)

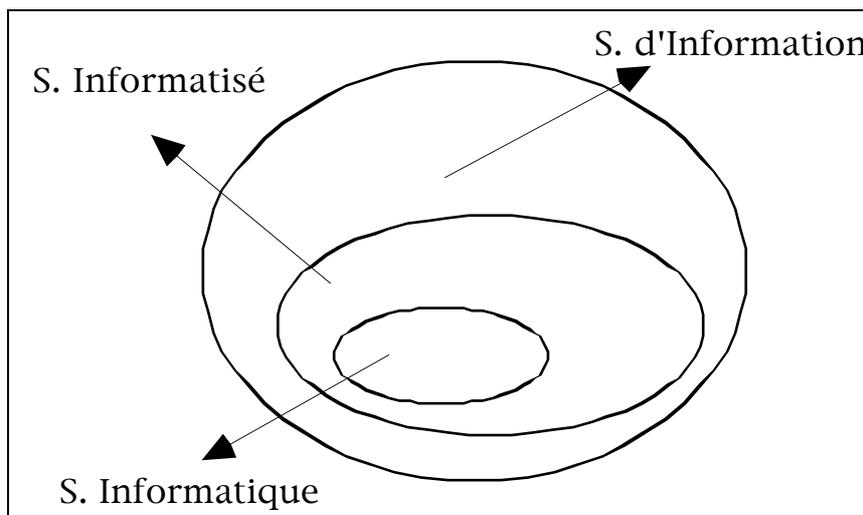


Figure 3 : S. Informatique/S. Informatisé/S. d'Information

Le schéma systémique de l'entreprise :

Le SI est avant tout un véhicule de communication dans l'entreprise et entre l'entreprise et son environnement. Au sein de l'entreprise on distingue:

- le système opérant chargé de la production (chaîne de fabrication, atelier d'assemblage, etc.),
- le système de pilotage dirigeant l'entreprise et maintenant les objectifs (directeur, chef de service, etc.),

- le système d'information assurant le lien entre les deux précédents. Il informe le système de pilotage des performances du système opérant. Inversement il transmet au système opérant les instructions du système de pilotage.

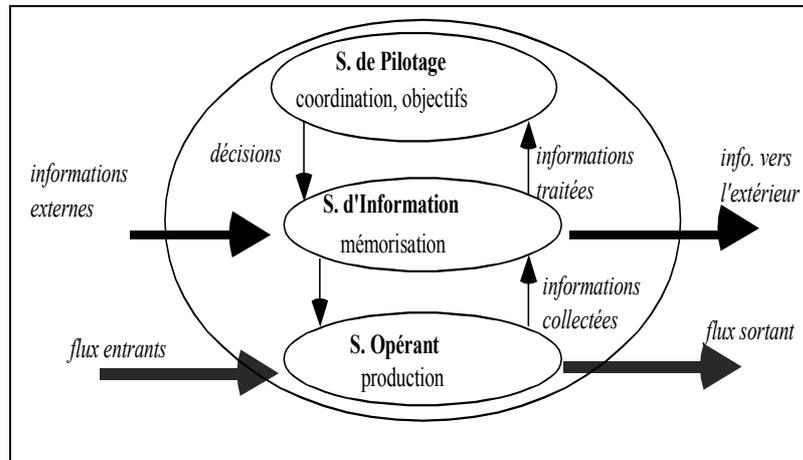


Figure 4 : S. Opérant / S. d'Information / S. de Pilotage

Les principes de la méthode Merise sont essentiellement applicables à la partie opérante : ils permettent une bonne appréhension de la gestion courante (gestion de la production, des stocks, facturation...), plutôt que la gestion de “ pilotage ” (états statistique, historiques, plans long terme...).

2.3. Conception d'un SI

Cycle de vie

Tout projet est mené dans le cadre d'une démarche par étapes, appelée cycle de développement ou cycle de vie. Ce cycle se situe sur une échelle de temps qui part de l'étude de l'objet naturel à l'intégration du système artificiel dans l'objet naturel.

La partie cycle de vie prise en compte par Merise est découpée en quatre périodes : la **conception** (descriptions détaillées des spécifications fonctionnelles), la **réalisation** (description logique et physique des données, production des programmes et des consignes d'utilisation...), la **mise en oeuvre** (mise en place effective du système dans son environnement réel) et la **maintenance** du système (adaptation du système aux évolutions de l'entreprise).

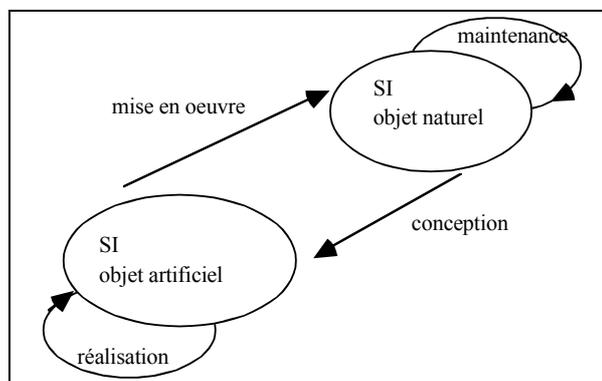


Figure 5 : cycle de vie du SI

Plus finement (Figure 6) on distingue :

- le **Schéma Directeur** : définition des domaines d'étude, planification du développement de chaque domaine. Il fixe les moyens en personne, machine... S'il n'y a pas de schéma directeur récent une **Etude d'Opportunité** est nécessaire.
- l'**Etude Préalable** : elle permet de déterminer le domaine sur lequel porte le projet, les structures concernées et l'organisation des postes de travail. Elle décrit les circuits de l'information et les procédures, en précisant pour chacune d'elle sa nature et son degré d'automatisation. Elle définit également les moyens informatiques à mettre en oeuvre, les coûts, les délais des différentes étapes de l'étude détaillée. Elle porte sur un sous-ensemble représentatif du système de manière à permettre aux dirigeants de prendre des décisions sur la globalité du projet. L'objectif est ici de définir la mission, d'établir un diagnostic de l'existant et de proposer de nouvelles orientations de gestion, d'organisation et technique. L'étude préalable peut être décomposée en trois étapes : un recueil se soldant par un bilan de l'existant, une conception permettant de préciser les nouvelles orientations de gestion, d'organisation et technique et une appréciation permettant en particulier de planifier la suite du projet.
- l'étude détaillée : elle détermine les spécifications fonctionnelles en respectant les solutions retenues à l'issue de l'étude préalable. Elle se scinde en deux étapes : la conception globale ou **Conception d'Ensemble** (Merise 2) et la conception détaillée ou **Conception Fonctionnelle Détaillée** (Merise 2).
- la réalisation. Elle se scinde en deux étapes : **Conception Technique Détaillée** (Merise 2) qui détermine les descriptions logique et physique des données, l'architecture des programmes, etc., la **Réalisation** proprement dite qui aboutit à production des programmes et comporte les tests unitaires et les tests d'intégration.
- **Préparation de la Mise en Oeuvre et Mise en Oeuvre** : concerne la préparation de la nouvelle organisation en particulier la conduite du changement et les supports de formation. La mise en oeuvre proprement dite concerne davantage le basculement des données et les changements de logiciels.
- **Maintenance** : adaptation du système aux évolutions de l'entreprise

Les trois premières étapes correspondent à la conception du système, les suivantes concernent sa réalisation et son lancement.

Différences principales entre Merise et Merise 2

Le cycle de référence de Merise 2 propose un découpage légèrement plus fin que le découpage initial de Merise.

- Il sépare en deux étapes distinctes le Conception d'Ensemble et la Conception Fonctionnelle détaillée. Dans Merise ces deux étapes étaient appelées Etude Détaillée.
- Il propose une nouvelle phase la Conception Technique Générale, définie lors de la Conception d'Ensemble. En effet, la plupart des choix techniques sont généralement faits à l'issue de cette étape.
- Il sépare la Préparation de la Mise en Oeuvre de la Mise en Oeuvre proprement dite.

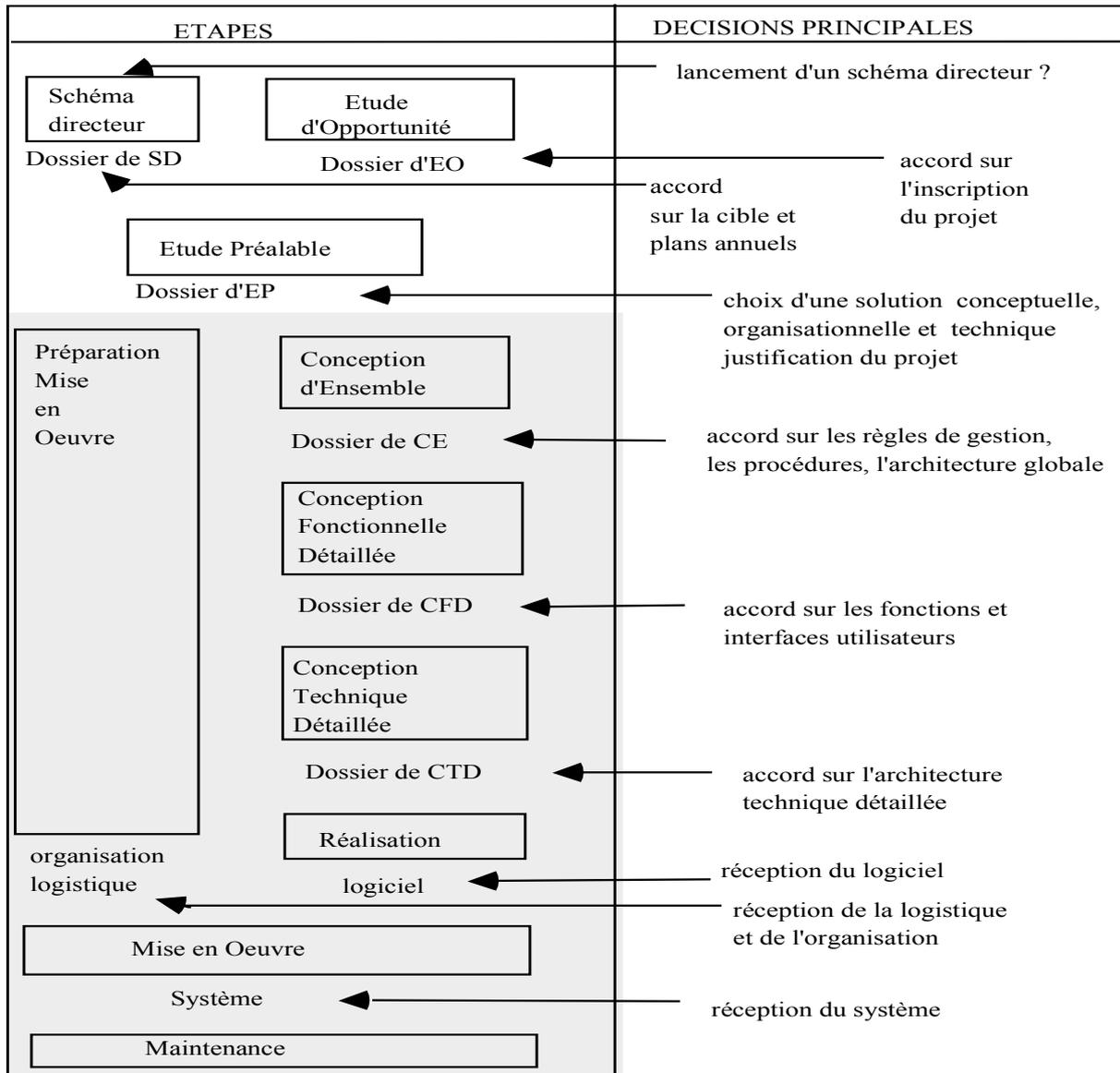


Figure 6 : cycle de vie détaillé

Le cycle de référence (Figure 6) est un cycle générique, à adapter aux types de projets et aux standards de l'entreprise. Il est par exemple possible de regrouper des étapes pour retrouver le cycle Merise. Certaines étapes sont parfois inutiles. Par exemple dans le cadre d'un "petit projet" la distinction entre Conception d'Ensemble et Conception Fonctionnelle Détaillée est inutile.

Les niveaux d'abstraction

La conception d'un SI se fait en utilisant des modèles permettant de représenter les données (aspects statiques) et les traitements (aspects dynamiques) du futur système.

Les niveaux d'abstraction ont pour but de permettre une modélisation progressive, par niveaux de préoccupation. Ces niveaux sont au nombre de 4.

- **Niveau conceptuel**

La description conceptuelle du système permet de représenter sa raison d'être et sa finalité en s'appuyant sur ses objectifs et les réalités qui le contraignent. Il s'agit dans un premier temps de décrire les **règles de gestion** qui permettront l'élaboration des modèles conceptuels de données et de traitements. **Une règle de gestion traduit un objectif prioritaire sans se soucier de la manière de le mettre en œuvre.**

Exemple intuitif : considérons les carrefours d'un système routier

Règles de gestion : - le système assure une circulation fluide des véhicules
- il s'agit d'éviter au mieux les collisions

Ces règles traduisent les deux objectifs primordiaux (fluidité de la circulation et limitation des collisions).

- **Niveau organisationnel**

La description organisationnelle du système représente l'organisation permettant d'atteindre les objectifs définis au niveau conceptuel. Il s'agit donc de décrire le fonctionnement du SI dans le cadre d'une organisation cible. Les descriptions du niveau organisationnel pour les traitements et les données ne préfigurent pas des moyens à mettre en œuvre pour y parvenir.

La description organisationnelle permet de décrire les vues partielles du système pour chaque type d'acteur par site de l'organisation. Il s'agit de décrire D'OU sont visibles les données et les traitements, QUI fait quoi en matière de données et de traitement, QUAND réalise-t-on les traitements et manipule-t-on les données.

Exemple intuitif :

Règles d'organisation : le système alterne les flots de circulation, en autorisant un passage en séquence dans une durée limitée des véhicules des différents axes.

- **Niveau logique**

Le niveau logique concerne la conception du logiciel correspondant aux parties à automatiser du système. Il prend en compte l'état de l'art technique général plutôt que les aspects physiques dans un contexte particulier. Il inclue une description logique des données c'est à dire une description dans un formalisme compatible avec l'état de l'art (modèle relationnel, modèle objet, fichiers, etc.) mais encore portable par rapport à des choix techniques précis. Il inclue également des modèles logiques de traitements décrivant le guidage fonctionnel, les boîtes de dialogue, l'arborescence des fenêtres...

Exemple intuitif : L'alternance sera assurée à l'aide de signaux lumineux, placés sur chaque axe de circulation...

• Niveau physique

le niveau physique tient compte des préoccupations et des choix techniques nécessaires à l'implantation physique des données et à la mise en place des traitements : langage de programmation, choix du SGBD, taille mémoire, etc.

Le passage de l'état ancien du système à l'état futur lors d'étapes de conception (en particulier lors de l'étape de conception de l'étude préalable) doit obligatoirement se faire par le niveau conceptuel, qui décrit l'invariant de l'entreprise. Toute étude qui proposerait des évolutions techniques non justifiées aux niveaux conceptuel et organisationnel est à proscrire.

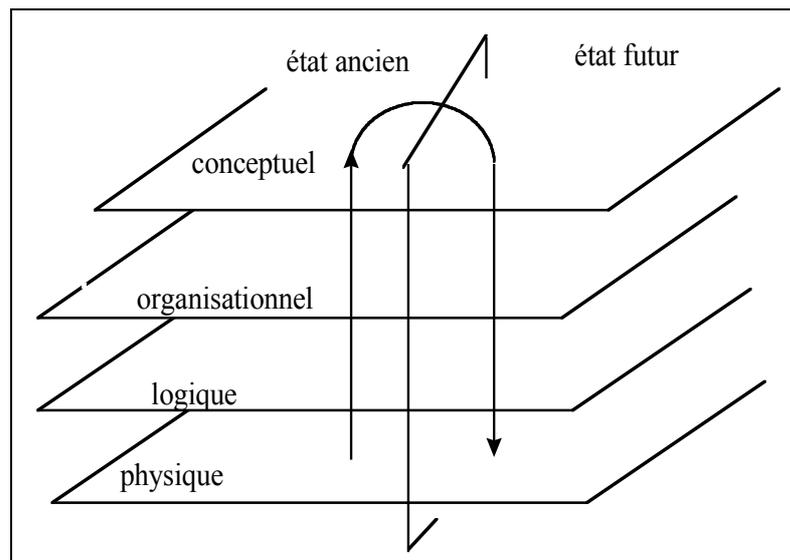


Figure 7 : évolution du SI / niveaux d'abstraction

Chaque niveau d'abstraction offrent une panoplie de modèles.

Chaque étape du cycle de vie d'un projet nécessite de s'intéresser :

- à un ou plusieurs niveaux d'abstraction,
- en utilisant un ou plusieurs modèles offerts par chaque niveau d'abstraction.

Parcours au sein des niveaux d'abstraction

Pour un projet donné il s'agit de fixer :

- **La démarche.** Par exemple, certaines étapes du cycle de vie peuvent être "fusionnées" dans le cas de "petits projets".
- **Le choix des modèles de différents niveaux d'abstraction** qui seront utilisés lors de chaque étape.

Le parcours au sein de ces niveaux d'abstraction lors des différentes étapes d'un projet d'informatisation est appelé **cycle d'abstraction** et est schématisé par une courbe dite "**courbe du soleil**".

Le processus est itératif, de plus suivant l'étape de la démarche dans laquelle on se trouve, la description d'un niveau d'abstraction peut être :

- plus ou moins **détaillée** (technique d'affinement)
- sur une couverture **plus ou moins large** du domaine (Sous Ensemble Représentatif : SER)

Le niveau de détail augmente au fur et à mesure que l'on progresse dans les étapes.

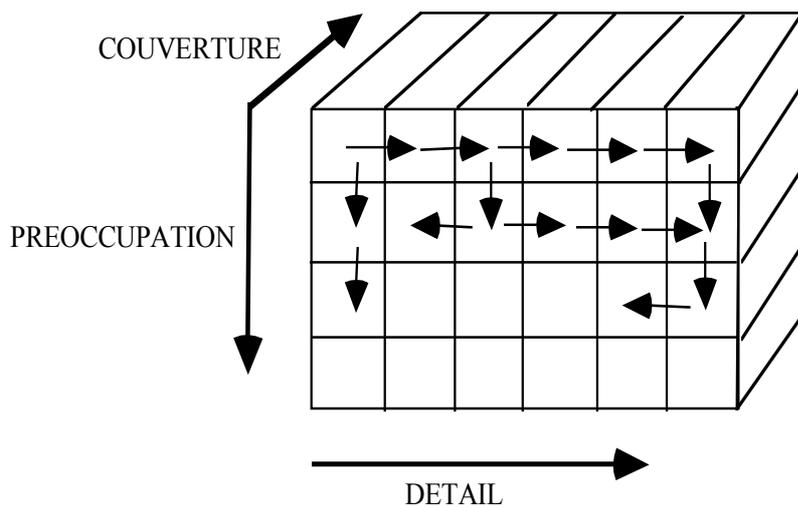


Figure 8 : courbe du soleil

Un parcours "standard"

A titre purement indicatif les indications suivantes peuvent permettre de suivre une démarche cohérente:

- **Etude Préalable.**

Si l'on part d'un existant, il s'agira de décrire le fonctionnement actuel (niveau organisationnel) de manière à en déduire une représentation conceptuelle (en tirer l'essentiel et l'incontournable). La conception du futur système partira du niveau conceptuel approuvé et validé par le client. Une étude préalable devant être courte mais complète, il s'agira alors de concevoir les niveaux organisationnel, logique et même éventuellement physique (prototype) d'un Sous Ensemble Représentatif (couverture limitée) du domaine à automatiser à un niveau de détail qui peut rester sommaire.

- **Etude Détaillée.**

Il s'agit de décrire les niveaux conceptuel et organisationnel (spécifications fonctionnelles) sur l'ensemble du projet. Au niveau de la conception d'ensemble le niveau de détail n'est pas encore très fin. Par exemple les données nécessaires à chaque procédure sont encore globalisées dans un modèle de données général. Les caractéristiques de moyens humains et matériels à mettre en oeuvre ne sont pas forcément abordées. Au niveau de la conception fonctionnelle détaillée les niveaux conceptuel et organisationnel doivent être entièrement décrits (niveau de détail le plus fin sur toute la couverture du projet).

• **Réalisation**

Il s'agit de se focaliser sur les parties à automatiser. On commencera (conception technique détaillée) par établir une description logique des données et des traitements. Puis une description physique des données. A l'issue de la conception technique détaillée la réalisation proprement dit aboutit à la production de programmes.

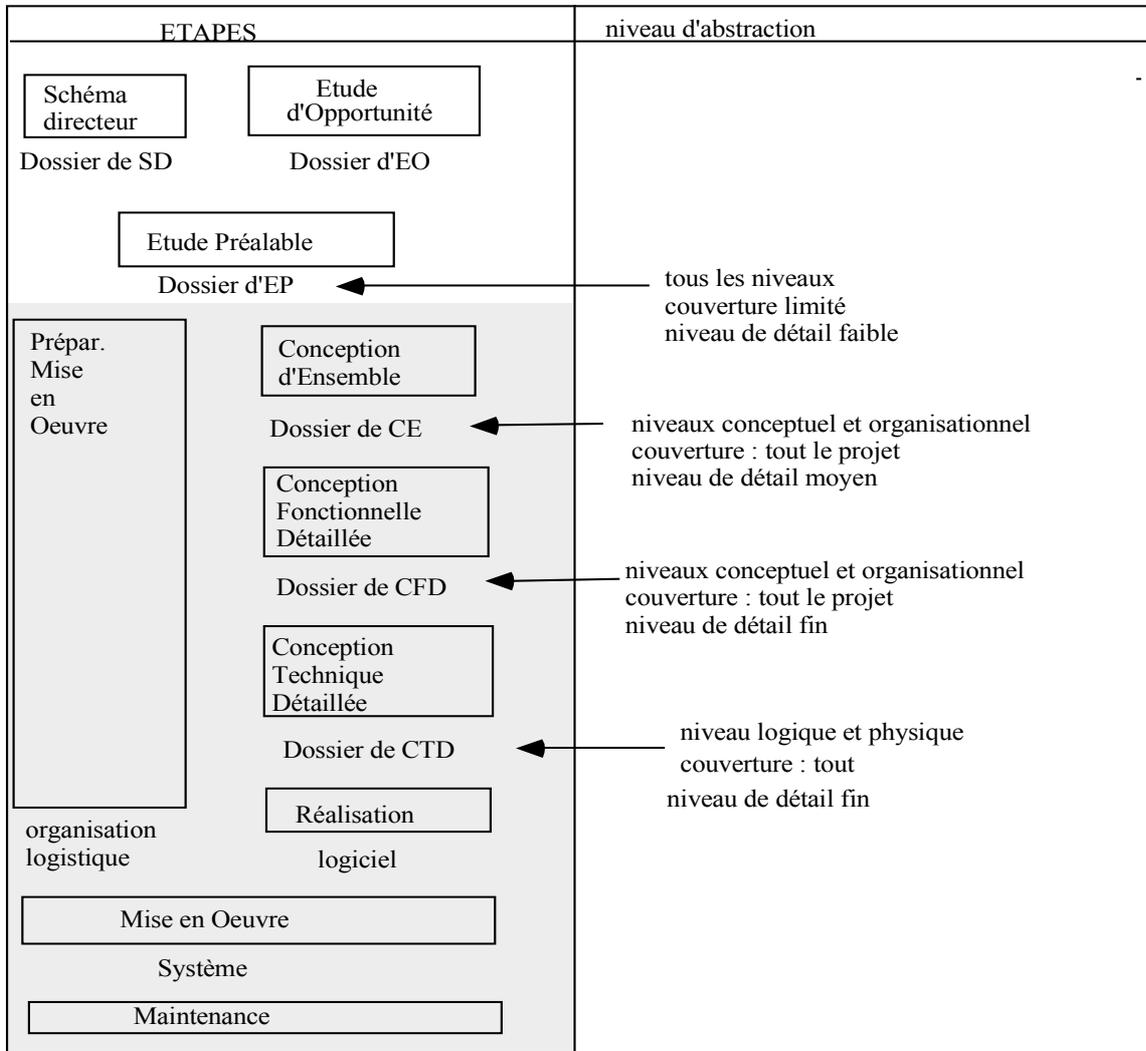


Figure 9 : parcours standard

3. Les modèles

3.1. Modèle

Un modèle est une représentation abstraite de la réalité. Par exemple une carte routière est une représentation abstraite de routes, un plan d'architecture est une représentation abstraite d'un bâtiment, etc. Les définitions du concept de modèle sont nombreuses :

- Weinberg

"Le modèle, c'est l'expression de quelque chose que nous cherchons à appréhender, représentée en des termes que nous pensons comprendre"

- Douglas Ross

" M modélise A si M répond à des questions concernant A "

- Minsky

" Pour un opérateur O, un objet M est un modèle d'un objet A dans la mesure où O peut utiliser M pour répondre aux questions qui l'intéressent au sujet de A "

Donc un modèle doit :

- correctement représenter la pensée du modélisateur,
- permettre de communiquer sans ambiguïté.

Pour ce faire il s'agit d'offrir des formalismes précis et normalisés pour modéliser les différents éléments (données et traitements) des systèmes d'information et ceci aux différents niveaux d'abstraction.

Par abus de langage nous utiliserons le mot "modèle" pour parler aussi bien des formalismes que de leur utilisation. C'est le contexte qui précise le sens. Lorsque nous parlerons par exemple, de modèles de données il s'agit du formalisme utilisé pour modéliser des données. Quand nous parlerons du modèle de donnée du domaine vente. Il s'agit de son utilisation pour modéliser les données concernées par le domaine "vente".

3.2. Les trois axes de modélisation de Merise 2

Merise 2 propose une démarche globale de modélisation basée sur 3 axes de modélisation (Figure 10 : Trois axes de modélisation) :

- l'axe d'architecture ou fonctionnel qui permet de décrire ce que **fait** le système (les activités) ;
- l'axe statique qui permet de décrire ce qu'**est** le système (les données) ;
- l'axe dynamique ou comportemental qui permet de décrire comment **se comporte** le système (les processus et la succession de transformations effectuées sur les données).

Exemple : On s'intéresse au traitement des commandes.

Dans ce cadre on distinguera plusieurs activités telles que le traitement de la commande, la facturation et la livraison. Ces activités décrivent les aspects fonctionnels du système (sous-système). Les aspects statiques correspondent à la représentation des données mises en jeu : informations sur les clients, les articles, les commandes, etc. Enfin les aspects comportementaux correspondent à la représentation du comportement du système lors de l'arrivée d'événement et l'impact sur les données. Par exemple l'arrivée d'une commande va déclencher la création d'une nouvelle commande.

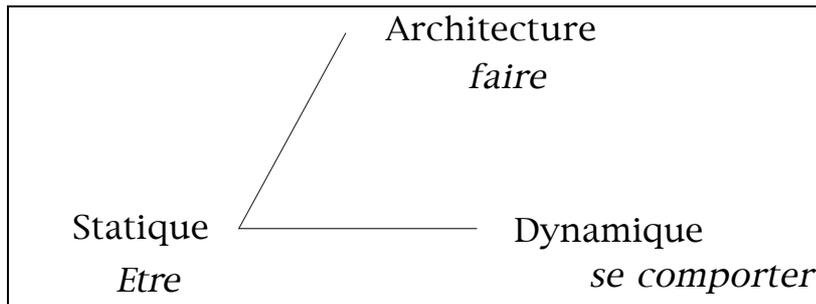


Figure 10 : Trois axes de modélisation

3.3. Axes de modélisation et niveaux d'abstraction

Merise 2 offrent différents modèles permettant de représenter les aspects **statique**, **fonctionnel** (architecture) et **dynamique** d'un système et ceci à différents niveaux d'abstraction : **conceptuel**, **organisationnel** et **logique** (Figure 11 : les modèles de Merise 2). Les modèles physique ne font pas partie de la méthode (et pour cause). Par rapport à Merise, Merise 2 propose de nouveaux modèles (modèle de flux par exemple) et établit une distinction beaucoup plus claire entre les modèles organisationnels et les modèles logiques.

	INTERFACE	APPLICATION		
		STATIQUE	DYNAMIQUE	ARCHITECTURE
quoi	MC	MCD	MCTA CVO	MFC
d'où qui quand	MOT	MOD	MOTA CVO	MFO
où comment	Maquettes IHM	MLD MLDr	MLT MLTr	SALMI, SAL SALR
		est	se comporte	fait

Figure 11 : les modèles de Merise 2

- Modèle de Contexte (MC)
- Modèle Conceptuel des Données (MCD)
- Modèle Conceptuel des Traitements Analytique (MCTA)
- Cycle de Vie des Objets (CVO)
- Modèle de Flux Conceptuel (MFC)
- + Règles de Gestion

- Modèle Organisationnel des Traitements (MOT)
- Modèle Organisationnel des Données (MOD)
- Modèle Organisationnel des Traitements Analytique (MOTA)
- Modèle de Flux Organisationnel (MFO)
- + Règles d'Organisation

- Modèle Logique des Données (MLD)
- Modèle Logique des Données Réparties (MLDr)
- Modèle Logique des Traitements (MLT)
- Modèle Logique des Traitements Répartis (MLTR)
- + Primitives

Ces modèles sont complétés pour les études d'architectures techniques complexes par les modèles suivants :

- Schéma d'Architecture Logique des Moyens Informatiques (SALMI)
- Schéma d'Architecture Logique (SAL)
- Schéma d'Architecture Logique Répartie (SALr)

Suivant le type de projet et l'étape du cycle de vie, il convient de sélectionner une partie de ces modèles.

Exemple : Dans le cas d'un "petit projet" (quelques hommes/mois, pas de répartition, logiciel classique) on ne développera que les MC, MFC, MCD, MCTA, CVO, MLD, MLT, maquettes.

Rappel : Ces modèles peuvent être élaborés à différents niveaux de détail, sur une couverture partielle ou totale du domaine d'étude et s'inscrivent dans une démarche itérative ("courbe du soleil").

4. Modèles Conceptuels

La description conceptuelle permet de représenter la finalité du système et sa raison d'être, en s'appuyant sur ses objectifs et les réalités externes qui le contraignent. Elle s'appuie sur un ensemble de **Règles de Gestion** qui décrivent le "quoi" de l'entreprise. Une Règle de Gestion est une traduction conceptuelle des objectifs choisis et des contraintes acceptées par l'entreprise. La plupart du temps il s'agit de règles d'actions liées aux traitements ou de règles de calcul liées aux données.

Exemples : "un inventaire doit être dressé périodiquement"
" tout produit livré sera entré en stock"
"la centrale d'achat sera libre d'imposer des jours de commandes"
"le salaire de base est égal à l'indice multiplié par la valeur du point"

Attention : l'analyste n'a aucune initiative sur les règles de gestion, son unique rôle est de les trouver (interview), les faire valider puis les utiliser pour élaborer les différents modèles conceptuels.

Le niveau conceptuel traite des événements et fournit des résultats sans se soucier de la manière dont sont acquises et restituées les informations portées par ces événements et résultats.

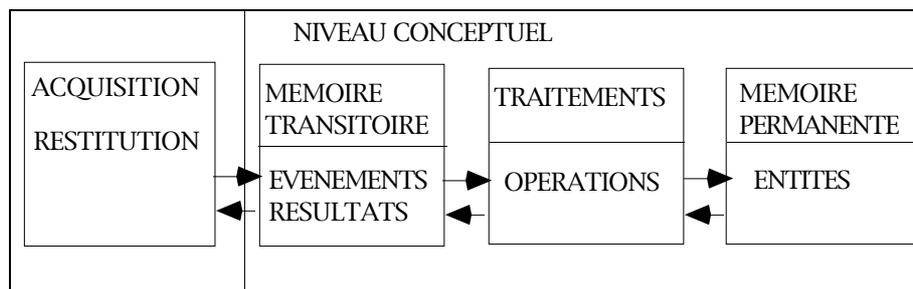


Figure 12 : niveau conceptuel

Les différents modèles proposés sont :

- le Modèle de Contexte (MC) et les Modèles de Flux Conceptuels (MFC)
- le Modèle Conceptuel des Données (MCD)
- le Modèle Conceptuel de Traitements (MCT), le Modèle Conceptuel des Traitements Analytique (MCTA) et les Cycles de Vie des Objets (CVO)

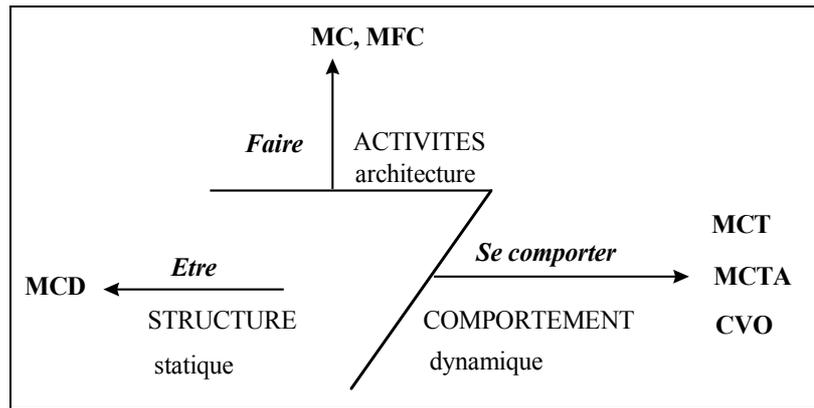


Figure 13 : Les 3 axes de modélisation / les modèles

Ces modèles peuvent être élaborés à différents niveaux de détail, sur une couverture plus ou moins grande et s'inscrivent dans une démarche itérative (tranche de la courbe du soleil).

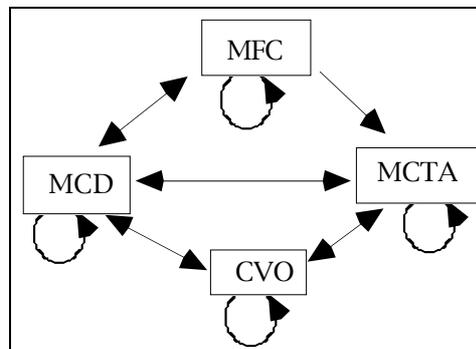


Figure 14 : démarche itérative

Les principales évolutions par rapport à Merise sont :

- l'utilisation des MFC avec des techniques d'affinement (*refinement*),
- la modification des modèles de traitement (MCT-> MCTA),
- l'extension du MCD,
- l'introduction des CVO.

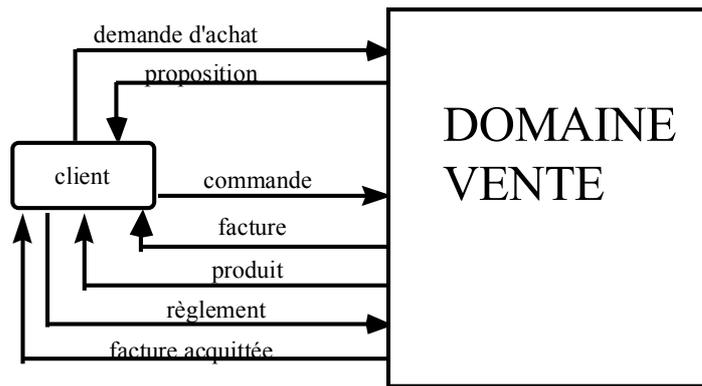


Figure 16 : Modèle de Contexte du domaine Vente

Concepts associés

domaine d'étude, acteur externe, domaine connexe, flux (agrégat de flux) de données

Utilisation du modèle

- au démarrage de l'étude : pour déterminer le domaine d'étude (correspondant la plupart du temps à un domaine fonctionnel) ;
- en fin d'analyse du système actuel : pour déterminer les activités existantes ;
- au début de l'analyse du système futur : pour préciser les contours du domaine d'étude.

5.2. Modèle de flux Conceptuels (MFC)

Définition

Un MFC détermine, par affinages successifs les activités du domaine d'étude sans décrire leur comportement.

Il correspond à une structure hiérarchique de diagrammes de flux, à partir du niveau 1.

On appelle diagramme de flux 1 (DFD1), le diagramme de 1er niveau décrivant les macros activités du domaine d'étude et les flux échangés.

On appelle diagramme de flux (DFDi), tout diagramme qui affine une activité.

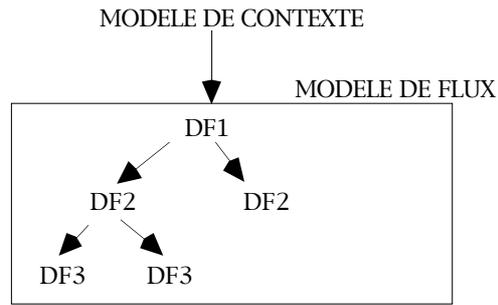


Figure 17 : Le Modèle de Flux Conceptuel

Exemple

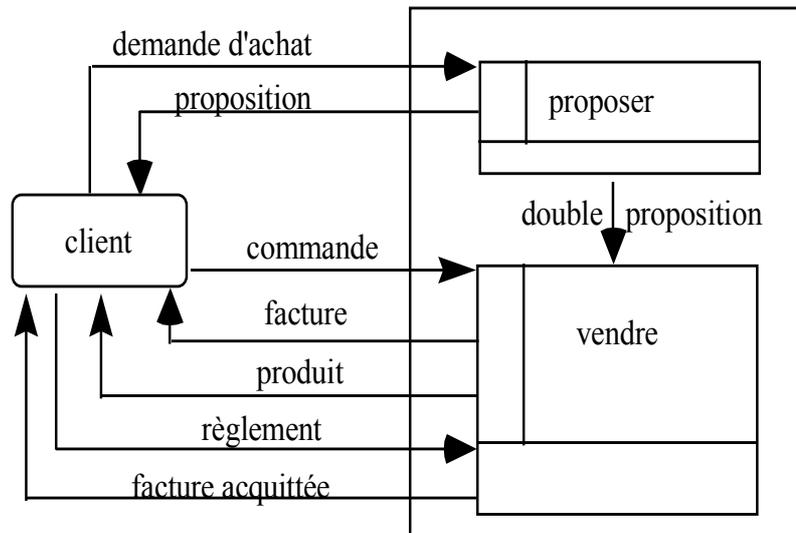


Figure 18 : diagramme de flux de niveau 1

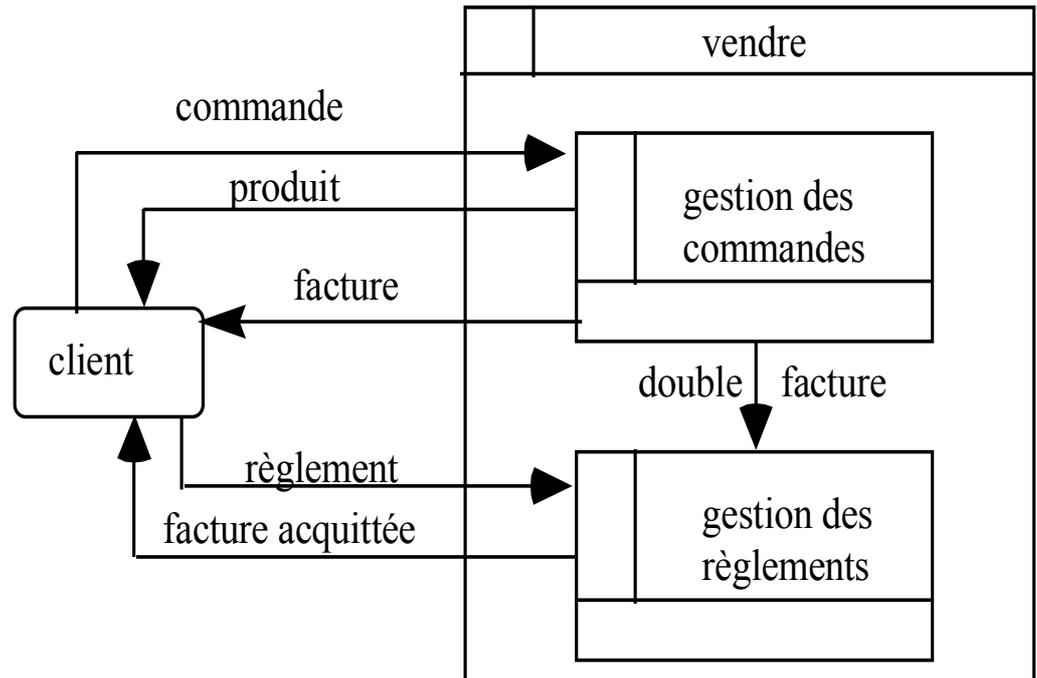


Figure 19 : diagramme de flux de niveau 2

Le MFC et les DFD complètent donc le MC par la description des activités du domaine recevant et générant les flux externes. Généralement établis sur deux niveaux, ils peuvent en comporter plus : la décomposition s'arrêtant lorsque les activités décrites correspondent aux processus ou aux opérations conceptuelles (cf. MCT et MCTA). En effet lors de la construction du Modèle Conceptuel de Traitement (Analytique) nous verrons que les événements remplacent les flux et que les opérations ou les processus remplacent les activités.

Concepts associés

les activités, les flux de données, les acteurs externes, les domaines connexes

5.3. Les concepts associés

Domaine d'étude

Un Domaine d'étude est un sous-ensemble cohérent de l'entreprise ou de l'organisme, bien délimité et formant le contenu à étudier. Il regroupe toutes les activités concernées par l'étude.

Il est:

- déterminé plus ou moins arbitrairement au démarrage de l'étude,
- précisé en fin d'analyse du système actuel,
- modifié éventuellement au début de l'analyse du système futur par la suppression ou l'ajout d'activités.

Il doit:

- correspondre à un projet de taille réaliste,

- être informatiquement opérationnel indépendamment du développement sur les autres domaines,
- minimiser les perturbations dans l'organisation lors de sa mise en place,
- minimiser les conséquences des choix organisationnels et techniques sur les autres domaines.

Activité

L'activité est un ensemble de traitements homogènes qui transforment ou manipulent des données.

L'activité est le concept sur lequel se concentre la décomposition du Modèle de Flux. Une activité du diagramme que l'on veut affiner devient le limite du diagramme de Flux de niveau inférieur. La décomposition d'une activité doit être indépendante des autres, mais peut par contre entraîner la décomposition des flux entrants et sortants.

Flux de données

Un flux de données est la représentation d'un échange d'informations entre deux composants du système ou entre un composant du système et un système extérieur. Un flux de données peut mettre en jeu des échanges de données informatisées, vocales, écrites, etc. mais aussi des flux physique de matières.

flux	acteur externe	activité	domaine connexe
acteur externe	non	oui	non
activité	oui	oui	oui
domaine connexe	non	oui	non

Figure 20 : flux de données

Acteur externe

L'acteur externe est une source ou une destination des données située dans l'environnement du système étudié. Il peut s'agir d'un service, d'une personne, d'un profil, etc.

Domaine connexe

Un domaine connexe est un composant du SI opérationnel ou de pilotage, interagissant avec le domaine étudié.

5.4. Gammes opératoires

Mécanisme de décomposition

Le niveau d'arrêt de la décomposition correspond au moment où l'activité représente une opération (au sens Merise) et donc vérifie la règle d'ininterruption (cf. MCT et MCTA). Une

activité est ininterrompue si son exécution ne peut être suspendue par l'attente d'un flux d'information.

Dans la Figure 19 par exemple l'activité Gestion des commandes est exécutée sans interruption dès l'arrivée d'une commande et se termine par l'émission des flux résultats (produit et facture). A partir de ce moment, il devient nécessaire de passer à la modélisation dynamique du système (cf. MCT) où nous verrons qu'une activité pourra être décomposée en plusieurs opérations conceptuelles si des interruptions temporelles le nécessitent.

On peut également s'arrêter un "cran" avant c'est à dire au moment où les activités représentent des processus.

Mécanisme de composition

Ce mécanisme peut être utilisé suite à une analyse de l'existant, dans le but d'architecturer le domaine d'étude de manière homogène.

Deux approches :

- par les objectifs en regroupant des activités qui participent à la même finalité opératoire,
- par les données en utilisant des matrices activités/objets par exemple.

Au niveau organisationnel ?

Au niveau organisationnel les MFO préciseront l'organisation des échanges d'informations (les sites, les postes, les supports d'échanges par exemple un courrier, un transfert informatique, un coup de téléphone...).

6. Modèle Conceptuel de Données "étendu"

Le Modèle Conceptuel de Données est une représentation statique du système d'information de l'entreprise. Au niveau conceptuel il s'agit de modéliser les données fondamentales de l'entreprise (les invariants décrits par des règles de gestion). Il ne doit être fait aucune hypothèse sur l'utilisation ultérieure de ces données.

Le Modèle Conceptuel de Données "étendu" (Merise 2) apporte des extensions au formalisme individuel adopté par Merise.

Ces extensions ont pour objectifs :

- de préciser et d'enrichir la description des objets en mettant en évidence des propriétés, des contraintes d'intégrité supplémentaires et des objets historiques,
- d'aider la validation du modèle de traitement (MCTA),
- de permettre à différents utilisateurs de partager le même modèle (accès à différents niveaux de décomposition ou de spécialisation),
- de permettre une relative stabilité par rapport aux évolutions de l'entreprise ou de l'application (une modification de détail laissant les niveaux supérieurs inchangés).

Nous commencerons par décrire le MCD originel, puis les extensions proposées dans Merise 2. Le formalisme Merise est conforme au modèle individuel (Entity-Relationship Approach).

6.1. Vocabulaire

Propriété

information élémentaire, conforme au choix de gestion de l'entreprise (adresse d'un client, référence d'un article, prix d'achat..).

Objet type (ou individu)

regroupement de propriétés, reflet d'une entité présentant un intérêt pour le système étudié, doté d'une existence propre et identifiable (par exemple Article doté des propriétés libellé, prix unitaire, nature... et identifié par sa référence).

Relation type

représente une association entre plusieurs objets; son existence est conditionnée par celles de objets mis en relation (par exemple la relation "commander" entre client et commande).

Cardinalités

indiquent pour chaque couple objet-relation les nombres minimum et maximum de valeurs de la relation pouvant exister pour chaque valeur d'objet.

Par exemple, dans la relation "commander" entre client et commande les cardinalités du couple client-commander sont (0,n) si un client peut exister comme prospect, il peut ne pas avoir

passer de commandes, d'autre part un client peut avoir n commandes, les cardinalités du couple commande-commander sont (1,1) une commande intervient dans une et une seule relation car elle concerne un et un seul client.

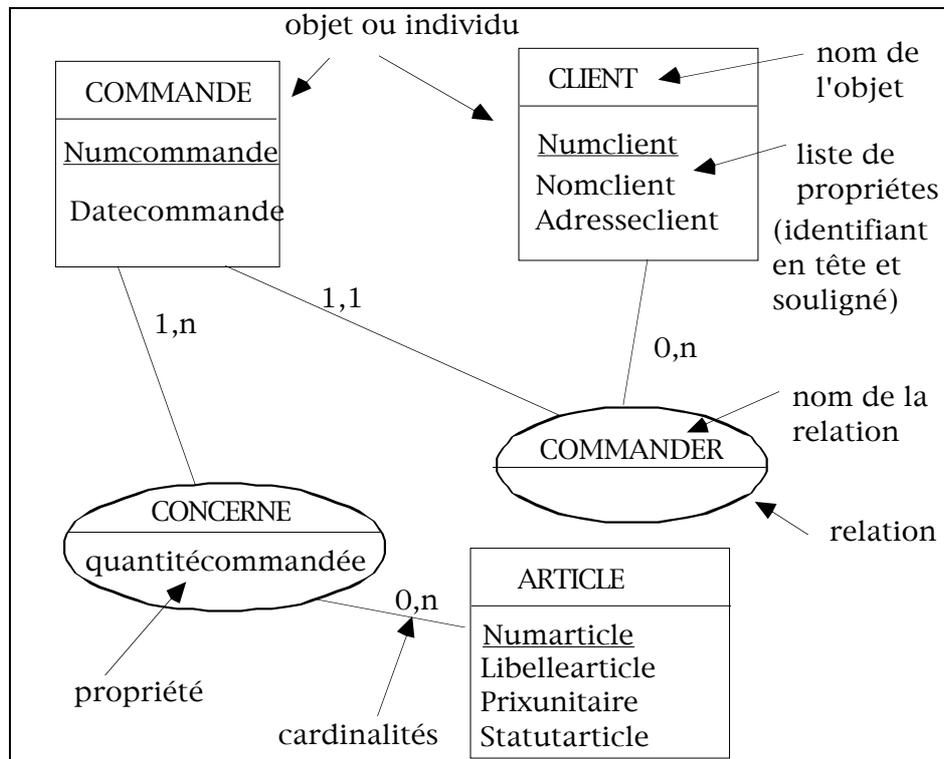


Figure 21 : notions de base du MCD

6.2. La Propriété

La propriété correspond à la plus petite partie logique d'information manipulée dans l'entreprise, il peut s'agir d'une propriété simple comme une référence ou un nom, mais aussi d'une propriété composée comme une date ou une adresse. Le degré d'atomisation est fonction de l'application. Si l'on s'intéresse à la gestion courante d'un fichier d'adresse il faudra éclater l'adresse en plusieurs propriétés (num, rue, code postal, ville).

Il faut autant que possible :

- éviter les propriétés calculables et redondantes,
- faire la chasse aux synonymies (référence article et num de produit) et aux polysèmes (préciser s'il s'agit de l'adresse du client ou celle du fournisseur).

6.3. Objet, Occurrence d'objet

L'objet est un regroupement de propriétés (ou attributs). Il est le reflet d'une entité de l'organisme dotée d'une existence propre. Il est identifiable et ne doit représenter qu'un seul concept sémantique.

Chaque propriété d'un objet peut être assimilée à une variable. Les valeurs qui lui seront affectées représentent les occurrences de cette propriété. Si l'on affecte une valeur à chacune des propriétés composant un objet, on obtient une occurrence de celui-ci.

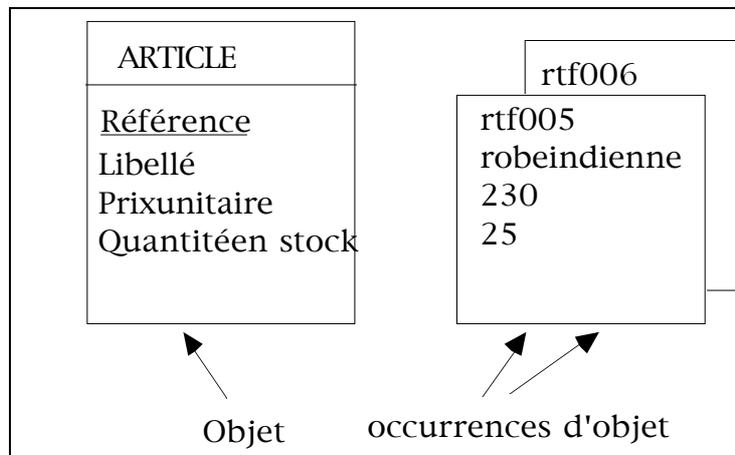


Figure 22 : objet / occurrences d'Objet

Identifiant (ou clé) : L'identifiant d'un objet est une propriété ou un ensemble de propriété qui permet de connaître sans ambiguïté chacune de ses occurrences. Dans l'exemple ci-dessus il s'agit de la propriété Référence.

Dépendance fonctionnelle monovaluée : Un ensemble B dépend fonctionnellement d'un ensemble A si la connaissance d'un élément a de A détermine au plus un élément b de B (si b existe). On dit que A détermine B. Les dépendances fonctionnelles monovaluées sont applicables aux objets : A et B sont alors des propriétés ; a et b sont des valeurs affectées à ces propriétés.

Dans un objet toute propriété dépend fonctionnellement de l'identifiant. Une propriété non identifiant ne dépend pas d'une autre propriété non identifiant. Une propriété non identifiant ne dépend pas que d'une partie de l'identifiant.

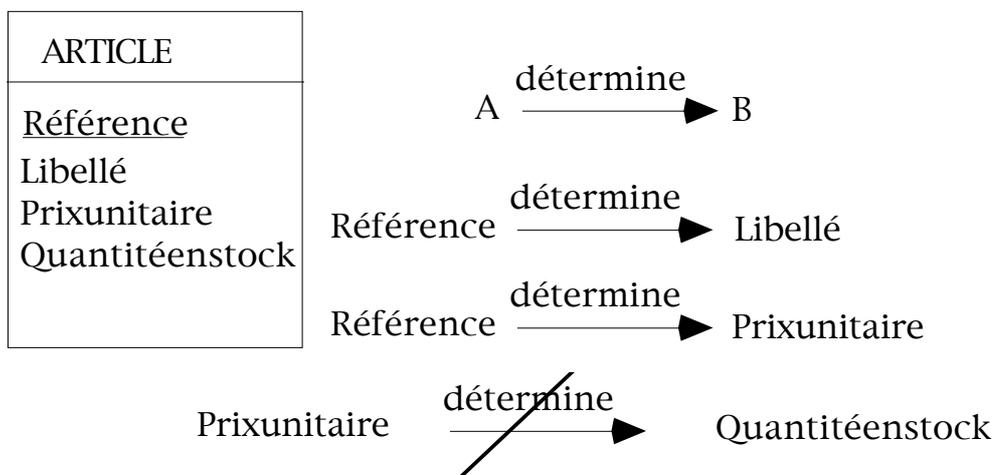


Figure 23 : dépendance fonctionnelle monovaluée

Construction des objets : c'est l'existence de cette propriété identifiant qui rapprochée à la règle de la dépendance fonctionnelle monovaluée, nous permet de construire les objets du modèle conceptuel. L'inventaire de la liste des données manipulées par l'entreprise nous conduit à dresser une liste de propriétés parmi lesquelles certaines sont reconnues comme identifiants nous révélant l'existence des objets. Les autres propriétés décrivant l'objet doivent être en dépendance fonctionnelle de son identifiant. Les propriétés de la liste qui ne dépendent pas d'un identifiant ou qui en apparence dépendent de plusieurs identifiants seront pour l'instant laissées de côté.

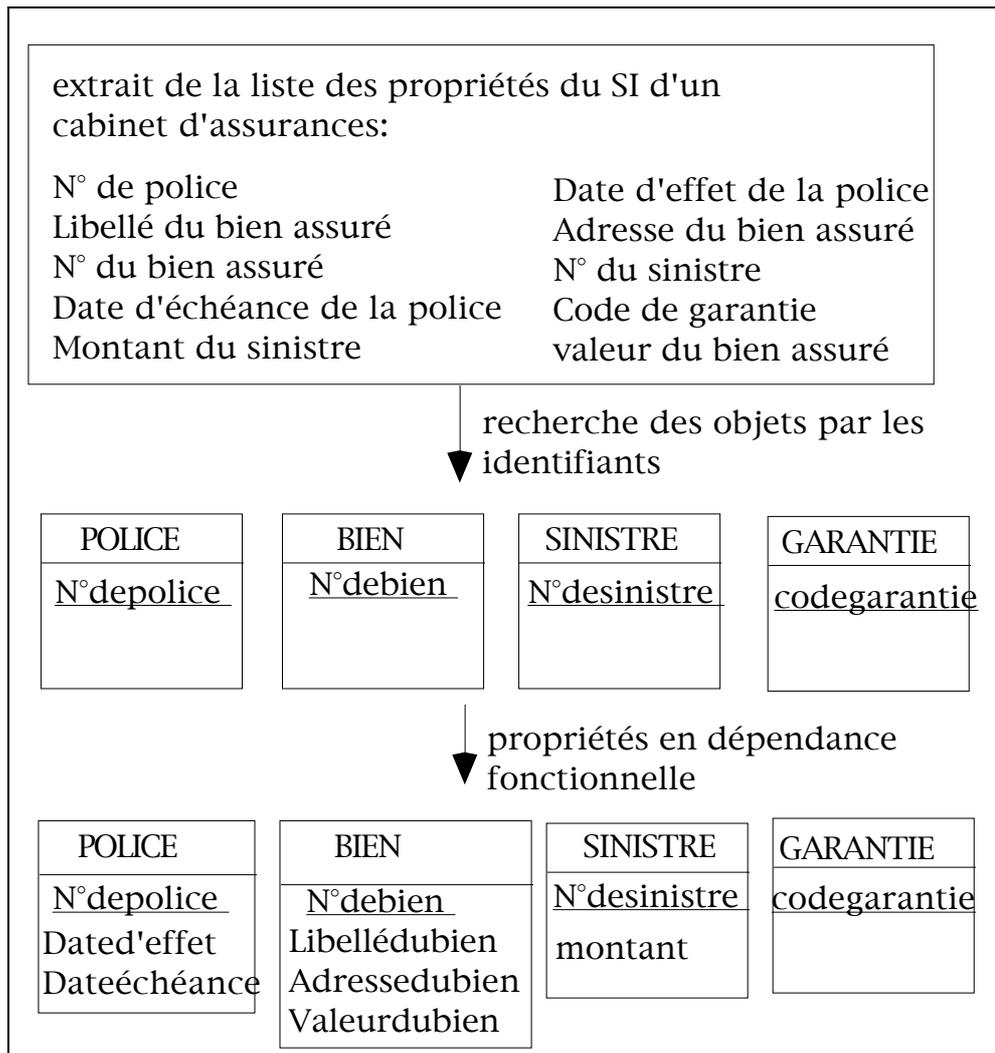


Figure 24 : construction des objets

6.4. Relation

La relation peut être vue comme une association entre divers objets du modèle. Elle est la traduction des verbes du langage de l'entreprise. La relation COMMANDER entre COMMANDE et CLIENT est la traduction de la phrase : "un client peut passer une ou des commandes". Cette association permet en particulier de savoir à qui livrer ou facturer une commande.

Contrairement aux objets, une association n'a pas d'existence propre. Son existence est conditionnée à celle des objets qui la composent.

L'ensemble des objets entrant dans la composition d'une relation est appelé **collection** de la relation, celle-ci comportant au moins deux objets. La dimension de la relation est égale au nombre d'objets de sa collection. Elle est dit n-aire (binaire, ternaire..) ou de dimension n (2, 3..).

Une relation peut ou non être porteuse de propriétés. Nous avons vu précédemment que lors de la construction des objets, certaines propriétés de la liste peuvent être restées en attente (car en dépendance fonctionnelle de plusieurs identifiants). La plupart du temps il s'agit de propriétés de relations. C'est par exemple le cas de "quantité commandée" qui est propriété de la relation CONCERNE entre COMMANDE et ARTICLE.

Identifiant : La relation possède un identifiant qui est la combinaison (concaténation, enchaînement) des identifiants des objets mis en relation. L'identifiant d'une relation n'est pas mentionné sur les schémas.

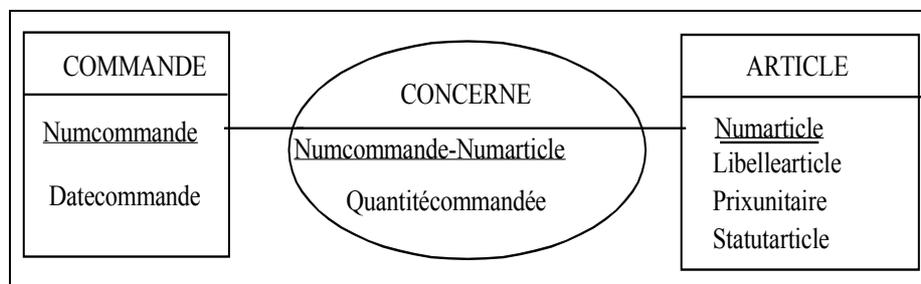


Figure 25 : Identifiant des Relations

Dépendance fonctionnelle monovaluée : Comme dans le cas des objets, toute propriété dépend fonctionnellement de l'identifiant. Une propriété non identifiant ne doit pas dépendre d'une autre propriété non identifiant. Une propriété non identifiant ne doit pas dépendre que d'une partie de l'identifiant.

Diagramme des occurrences: Construire un diagramme d'occurrences c'est attribuer à chacun des objets et relations du modèle étudié, un ensemble de valeurs plausibles pour en vérifier la validité (Figure 28).

Relation réflexive : Elle relie un objet à lui même. Une occurrence de la relation associe une occurrence de l'objet à une autre occurrence de ce même objet (exemple PARENTE).

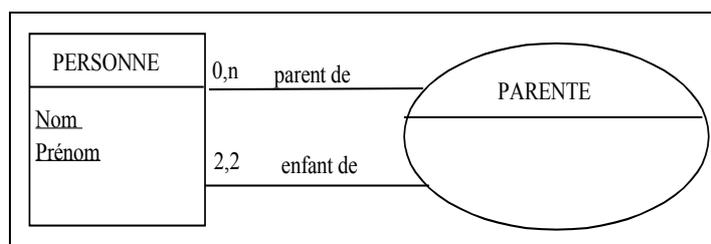


Figure 26 : relation réflexive

6.5. Cardinalités

Les cardinalités représentent pour chaque couple objet-relation les nombres minimum et maximum d'occurrences de la relation pouvant exister pour une occurrence d'objet. Elles mesurent donc la participation des occurrences d'objets aux relations.

- (0, 1) : une occurrence d'objet peut exister sans pour autant participer à la relation (0) et ne participe jamais plus d'une fois.
- (0,n) : c'est la cardinalité la plus ouverte; une occurrence d'objet peut exister sans pour autant participer à la relation (0) et peut participer sans limitation.
- (1,1) : une occurrence d'objet participe une et une seule fois à la relation.
- (1,n) : une occurrence d'objet participe au moins une fois à la relation et peut participer sans limitation.

Pour être complet, un modèle conceptuel doit comporter les cardinalités correspondantes à chaque couple objet-relation. Les diagrammes d'occurrences mais surtout les règles de gestion aident à déterminer les cardinalités. Elles sont des traductions de règles de gestion.

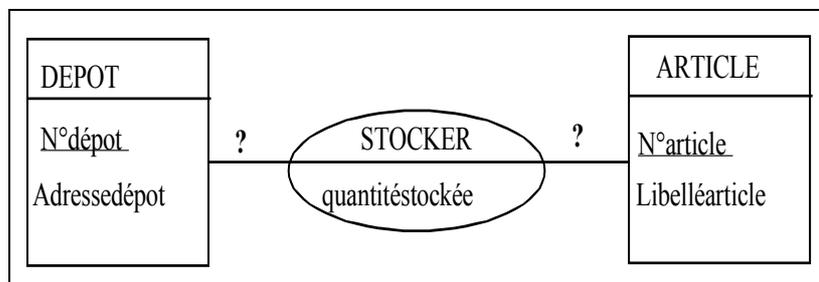


Figure 27 : cardinalités ?

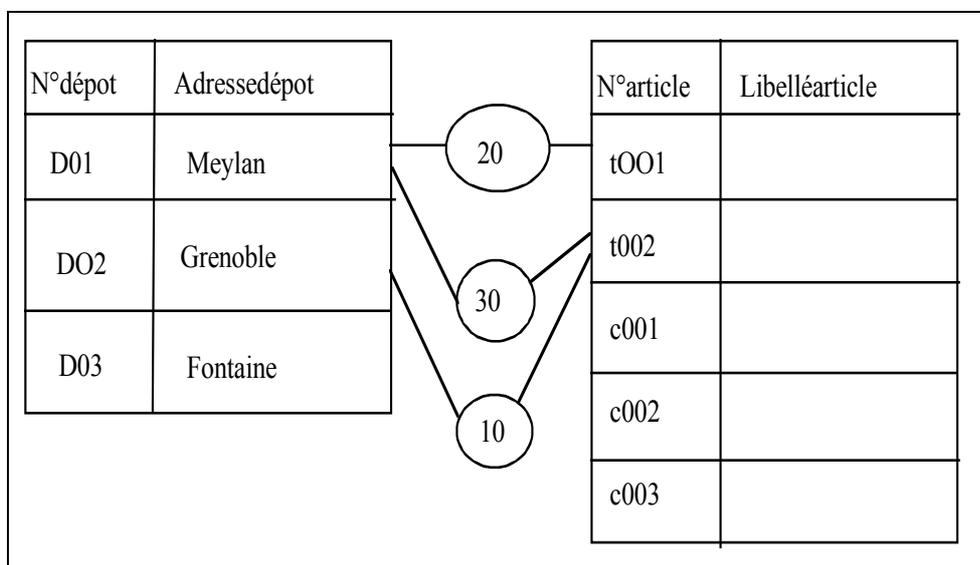


Figure 28 : diagramme d'occurrences

Du diagramme d'occurrences ci-dessus il ressort qu'un article peut être en stock dans plusieurs dépôts, qu'un article peut ne pas être en stock et qu'à un moment donné un dépôt peut être vide.

Règles de gestion :

- Un article peut exister sans être stocké.
- Il peut être stocké dans plusieurs dépôts.
- Un dépôt peut exister même s'il ne stocke rien.

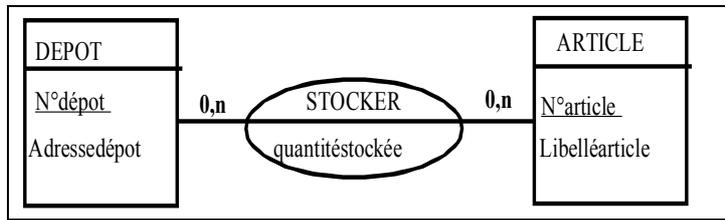


Figure 29 : cardinalités

6.6. Contrainte d'Intégrité Fonctionnelle

Une Contrainte d'Intégrité Fonctionnelle (CIF) définie sur une relation représente le fait que l'un des objets de sa collection est identifié sans aucun doute par la connaissance d'un ou plusieurs autres.

Une relation binaire ayant des cardinalités (0,1) ou (1,1) exprime une CIF.

Une CIF exprime une règle de gestion et est représentée par une flèche pointant sur l'objet déterminé.

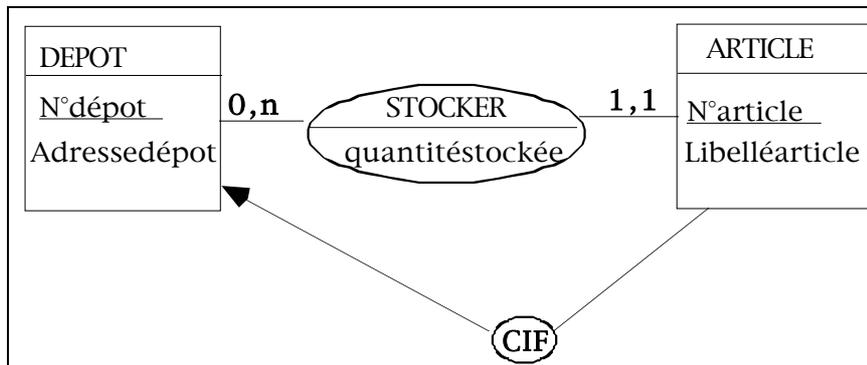


Figure 30 : Contrainte d'Intégrité Fonctionnelle

Le schéma ci-dessus (Figure 30) exprime les règles de gestion :

- Un article ne peut exister sans être stocké.
- Il ne peut être stocké qu'à un seul endroit.

Les CIF sont utilisées pour décomposer les relations. Elles permettent donc de simplifier des relations de dimensions supérieure à 2.

Exemple : dans une entreprise industrielle, les ordres de fabrication sont établis par le siège en direction de divers sites de production. On suppose la règle de gestion suivante:

- un ordre de fabrication ne concerne qu'un seul site.

La connaissance de l'ordre de fabrication permet donc de déterminer le site correspondant (existence d'une CIF de "ordre de fabrication" vers "site"). Cette contrainte d'intégrité permet de décomposer la relation ternaire (Figure 31) en deux relations binaires (Figure 32).

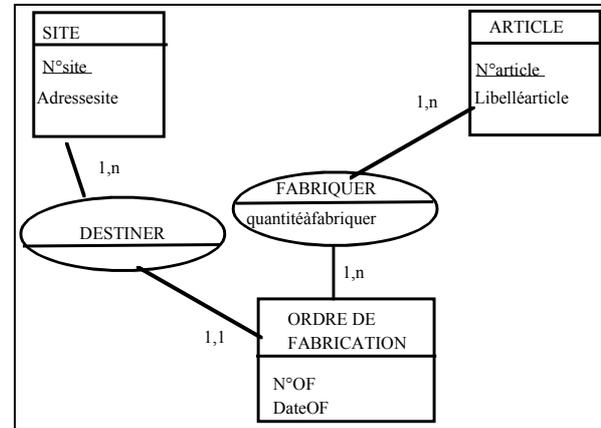
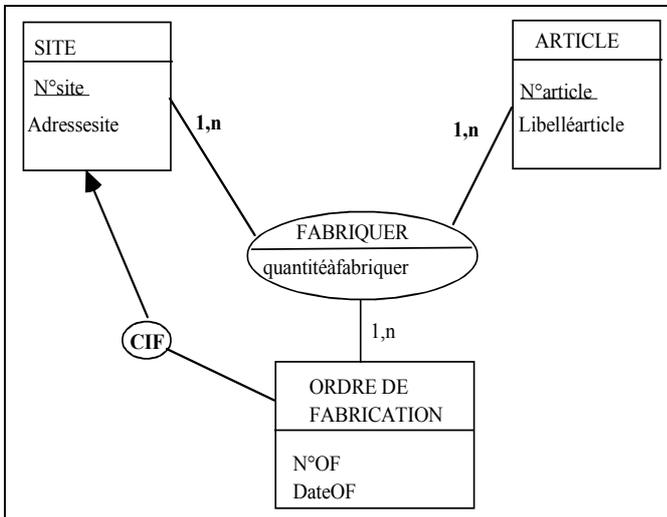


Figure 32 : décomposition de relation

Figure 31 : relation ternaire et CIF

6.7. Le problème de la représentation du temps

Il existe deux points de vue concernant la représentation du temps.

Représentation synchronique : Le temps n'intervient pas comme élément discriminant. Elle correspond à une vision instantanée de la réalité modélisée.

Représentation diachronique (ou historique) : Le temps intervient comme élément discriminant. Elle correspond à une vision historique de la réalité modélisée.

cas 1 (Figure 33) :

- Un étudiant assure une voiture pour un certain montant.
- Toute voiture est assurée par un et un seul étudiant.

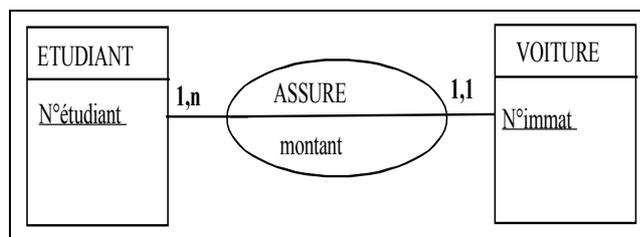


Figure 33 : le temps n'intervient pas

cas 2 (Figure 34): représentation synchronique

- Un étudiant assure à une certaine date, une voiture pour un certain montant.
- Toute voiture est assurée par un et un seul étudiant.

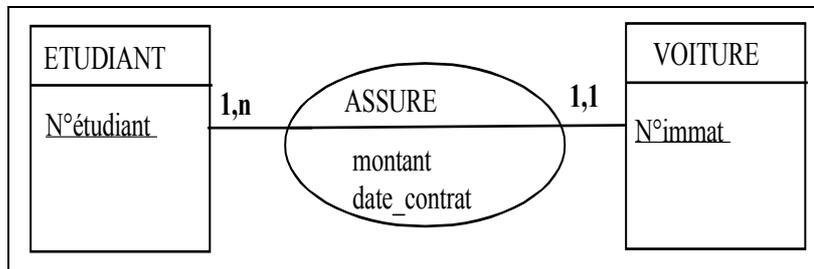


Figure 34 : représentation synchronique

L'identifiant de l'association est toujours N°étudiant, N°immat

cas 3 (Figure 35): représentation diachronique

- Un étudiant assure à une certaine date, une voiture pour un certain montant.
- Il peut avoir assuré plusieurs fois sa voiture.

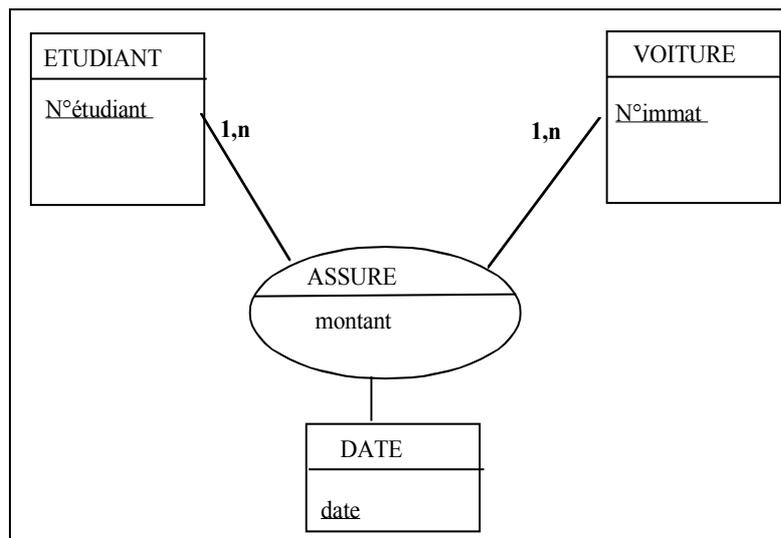


Figure 35 : représentation diachronique

La prise en compte du temps peut se faire:

- soit en introduisant une entité temporelle (date, jour...) (Figure 35),
- soit en introduisant des entités représentant des événements datés (Figure 36).

cas 4 (Figure 36): introduction d'événements datés

- Un étudiant peut pour une même voiture passer successivement plusieurs contrats

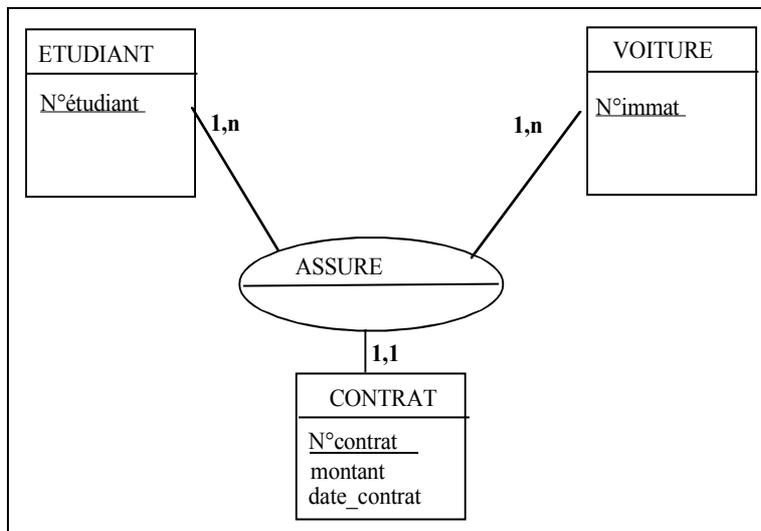


Figure 36 : entités correspondant à des événements datés

6.8. Vérification du modèle

- Toutes les propriétés doivent être élémentaires (non décomposables).
- Chaque objet doit posséder un identifiant et un seul.
- Les propriétés d'un objet, autres que l'identifiant, doivent être en dépendance fonctionnelle monovaluée de cet identifiant.
- Une propriété ne peut être qualifiée qu'un seul objet ou qu'une seule relation (éviter les redondances et les polysèmes).
- les dépendances fonctionnelles transitives doivent être éliminées. Autrement dit une propriété non identifiant ne doit pas être en dépendance fonctionnelle d'une autre propriété non identifiant.
- Pour chaque occurrence d'une relation, il doit exister une et une seule occurrence de chacun des objets de la collection (la participation d'un objet à une relation ne peut pas être optionnelle).
- les propriétés d'une relation doivent dépendre de la totalité de l'identifiant de cette relation. Autrement dit une propriété non identifiant ne doit pas dépendre que d'une partie de l'identifiant.

6.9. La démarche

Ecole pragmatique

- recherche des objets par "intuition",
- objet = reflet d'une entité d'intérêt pour l'organisation étudiée,
- inconvénients :
 - se laisser envahir par l'existant pour construire un modèle qui doit représenter l'invariant (c'est à dire se placer hors du temps) ;
 - trop proche de la "vue" de l'utilisateur et non de la globalité des objets de l'organisation.

Ecole "formelle" (ou plutôt exhaustive)

- établir la liste des propriétés à partir des documents de l'entreprise (ou de tout autre support),
- classer ces données par ordre alphabétique,
- épuration des polysèmes, des synonymes et des redondances,
- repérer les identifiants existants pour dégager les objets naturels,
- rattacher à ces objets les propriétés en dépendance fonctionnelle de leur identifiant,
- placer les relations et leur rattacher, si besoin est, les propriétés en dépendance fonctionnelle de plusieurs identifiants,

- considérer les propriétés restantes afin de les grouper en objets pour lesquels on définira des identifiants,
- étudier les cardinalités de chaque couple objet-relation,
- simplifier le modèle à l'aide des contraintes d'intégrité fonctionnelle,
- procéder à la vérification des règles.

Cohérence par rapport aux Modèles de Flux

Quel que soit le mode d'obtention du MCD, les propriétés des objets et des relations doivent être cohérentes avec celles drainées par les flux d'information des Modèles de flux.

C'est ainsi que l'arrivée d'une commande (un flux d'information du modèle de contexte) est porteuse de nombreuses informations (code client, nom client,... code produit, ... quantité commandée....) qui correspondent à des propriétés des objets et des relations du MCD. Inversement, une facture émise par le domaine étudié est également porteuse d'informations (prix total à payer, etc.) qui correspondent soit à des propriétés des objets ou des relations soient à des informations calculables à partir de celles ci.

Les paragraphes suivants présentent les extensions proposées dans Merise 2. Il s'agit d'une part de la définition des sous-types d'objets et de relations, d'autre part de l'introduction des contraintes d'intégrité statiques.

6.10. Sous-type d'objet et sous-type de relation

Sous-type d'objet

Un sous-type d'objet correspond à un sous-ensemble d'occurrences d'objet dotées de caractéristiques propres (propriétés spécifiques et/ou relations spécifiques). On pourra aussi employé les termes d'objet générique (sur-type) et **objet spécialisé** (sous-type).

- L'**objet générique** ou **sur-type** porte les caractéristiques communes aux objets spécialisés.

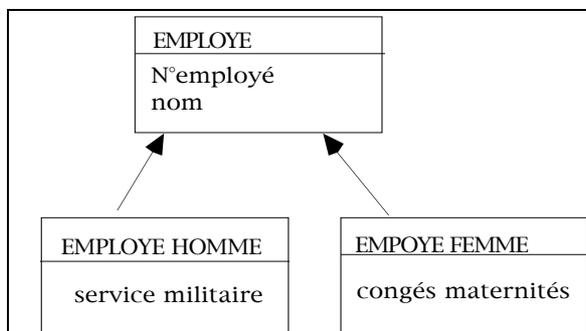


Figure 37 : sur-type/sous-type

- Chaque sous-type **hérite** des propriétés et des relations du sur-type.

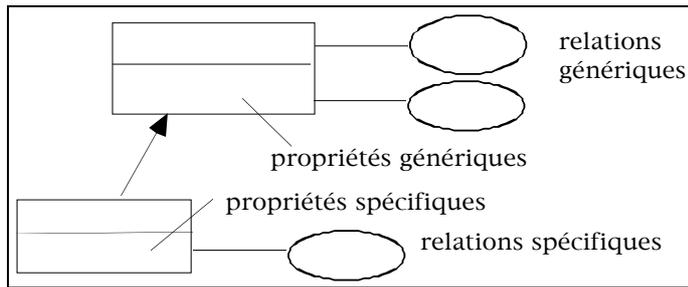


Figure 38 : héritage et spécialisation

- Le mécanisme de spécialisation permet d'avoir plusieurs niveaux de description d'un même objet : de la description la plus générale à la description la plus spécifique.

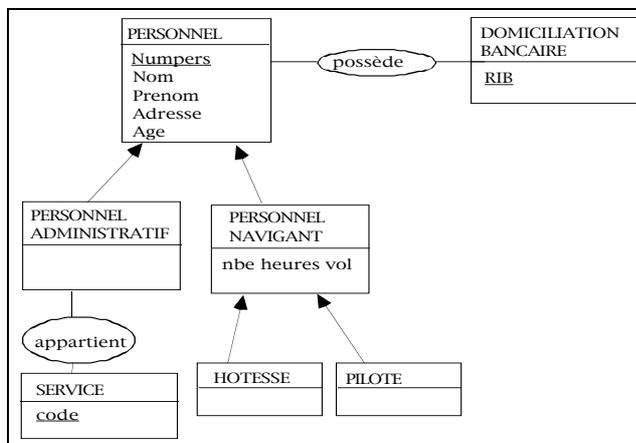


Figure 39 : hiérarchie d'héritage

- Cas particulier de spécialisation : la spécialisation par états

Un sous-type correspond dans ce cas à un état de l'objet. Il peut y avoir transfert d'une occurrence de l'objet d'un sous-type à un autre sous-type, contrairement au cas précédent (Figure 39) qui correspond à une spécialisation par "catégorie". Nous verrons que les CVO constitue une alternative à la modélisation des états.

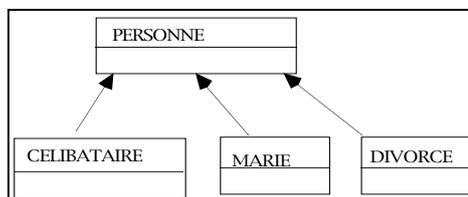


Figure 40 : spécialisation par états

Sous-type de relation

Un sous-type de relation correspond à un sous-ensemble d'occurrences d'une relation dotées de propriétés et/ou de cardinalités spécifiques.

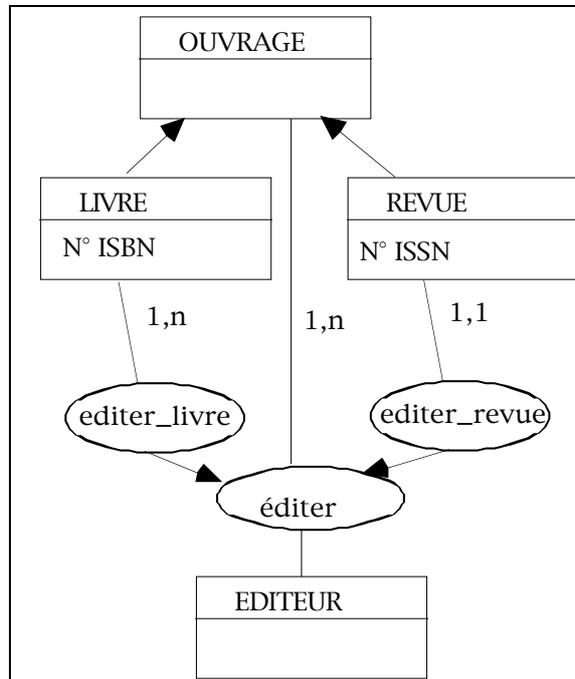


Figure 41 : sous-type de relation

6.11. Contraintes d'intégrité statique

Les contraintes d'intégrité statiques peuvent porter sur :

- Une ou plusieurs propriétés :
 - * plage de valeurs
 - * liste de valeurs
 - * contrainte de format ...
- Des sous-type d'objets :
 - * contrainte de partition (notée +)
 - * contrainte d'exclusion (notée x)
 - * contrainte de totalité (notée T)
- Des relations ou des pattes de relations :
 - * contrainte de partition (+)
 - * contrainte d'exclusion (x)
 - * contrainte de totalité (T)
 - * contrainte d'inclusion (I)
 - * contrainte d'égalité (=)
 - * contrainte d'unicité (1)
- Des propriétés et/ou des relations :
 - * contraintes d'identification

6.11.1. Contraintes de base : couverture et disjonction

contrainte de couverture : toute occurrence de l'objet générique doit appartenir à au moins un des sous-types.

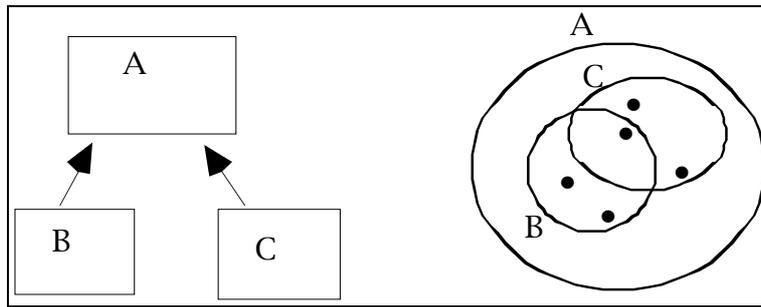


Figure 42 : couverture

contrainte de disjonction : toute occurrence de l'objet générique doit appartenir à un seul sous-type (les sous-types sont mutuellement exclusifs).

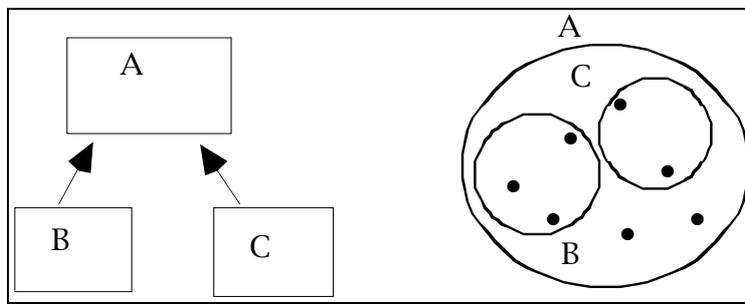


Figure 43 : disjonction

6.11.2. Contraintes d'intégrité entre les sous-types d'objets

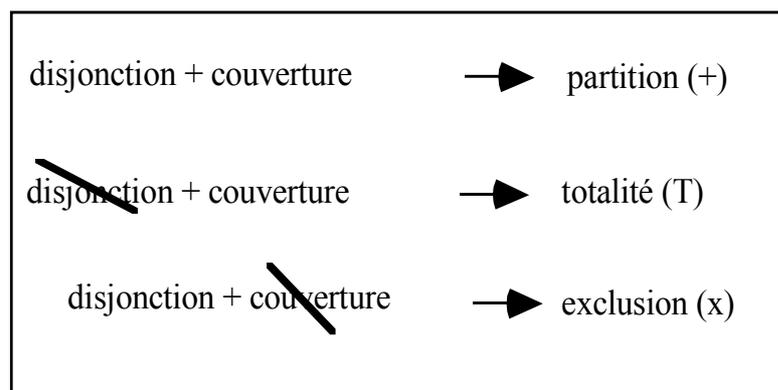


Figure 44 : contraintes d'intégrité entre les sous-types

Contrainte de partition

Un auteur est soit invité, soit accepté, soit refusé.

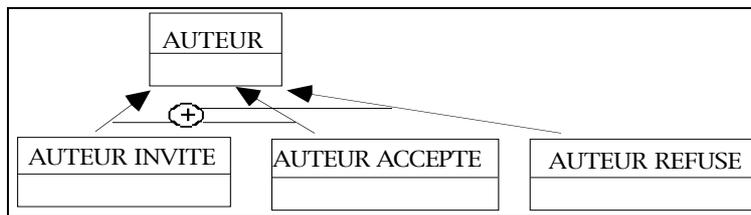


Figure 45 : partition = disjonction + couverture

Contrainte de totalité

Une personne physique peut être à la fois Particulier et Entrepreneur individuel, elle est l'un ou l'autre.

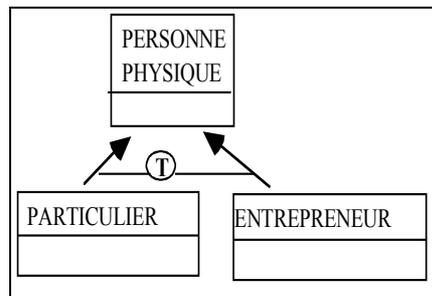


Figure 46 : totalité = couverture et non disjonction

Contrainte d'exclusion

Un contrat ne peut être à la fois un contrat de crédit et un contrat d'épargne, il existe d'autres types de contrats.

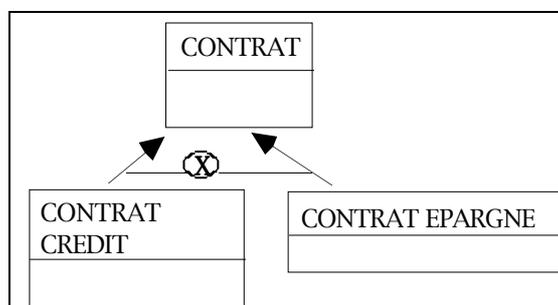


Figure 47 : exclusion = disjonction et non couverture

6.11.3. Contraintes d'intégrité sur les relations

Ici aussi il s'agit de contraintes ensemblistes.

Contrainte de partition (couverture et disjonction)

Une contrainte de partition permet d'exprimer que toutes les occurrences d'un objet (dit objet pivot) impliqués dans deux (ou plus) relations sont présentes dans une et une seule d'entre elles (ou exclusif).

Toute personne est soit résidente en France soit résidente à l'étranger, mais ne peut être les deux.

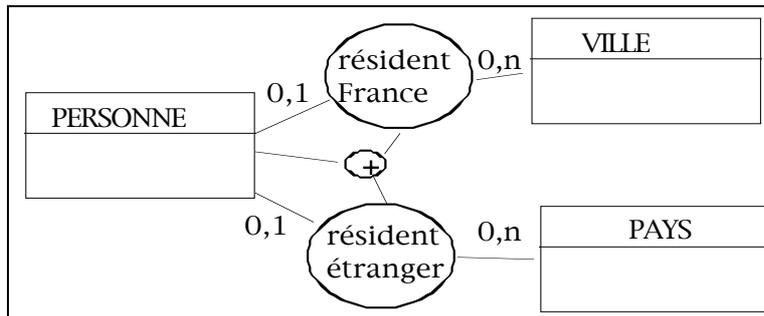


Figure 48 : partition

Contrainte d'exclusion (non couverture et disjonction)

Une contrainte d'exclusion interdit qu'une occurrence d'un objet (dit objet pivot) impliqués dans deux (ou plus) relations soit présente dans deux d'entre elles.

On ne peut être salarié et étudiant, certaines personnes ne sont ni l'un ni l'autre.

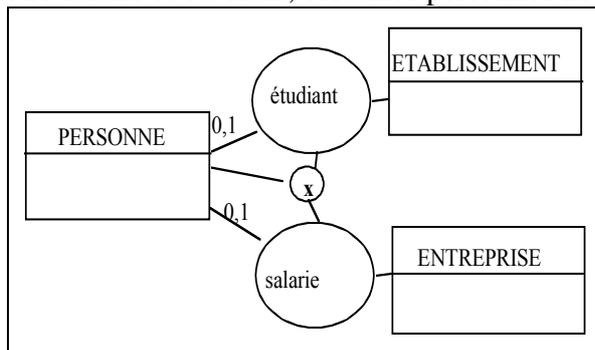


Figure 49 : exclusion

Contrainte de totalité (couverture et non disjonction)

Une contrainte de totalité permet d'exprimer que toutes les occurrences d'un objet (dit objet pivot) impliqués dans deux (ou plus) relations sont présentes dans au moins l'une d'entre elles (ou inclusif).

Toute personne est étudiant dans un établissement, ou salarié dans une entreprise ou les deux à la fois.

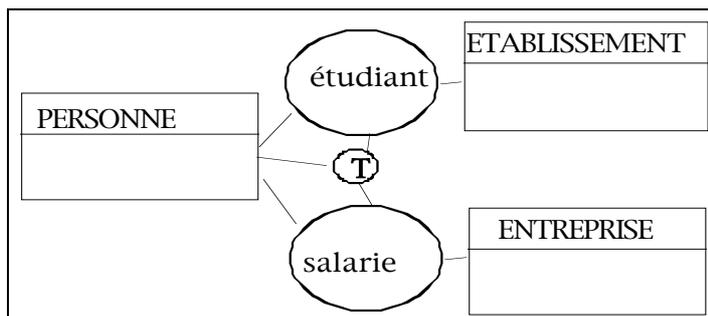


Figure 50 : totalité

Note : Pour les contraintes de partition, d'exclusion et de totalité, si aucun objet pivot n'est mentionné la contrainte porte sur la collection de la relation.

On ne peut être salarié et étudiant du même établissement.

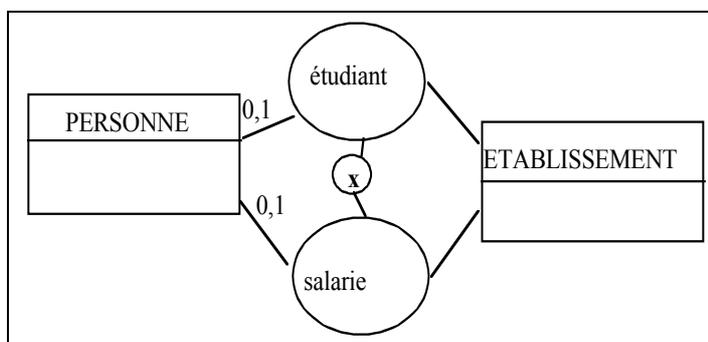


Figure 51 : plusieurs objets pivots

Contrainte d'inclusion

cas 1 : pas d'objet pivot explicitement désigné

Une contrainte d'inclusion permet d'exprimer que l'ensemble des occurrences d'une relation est inclus dans l'ensemble des occurrences d'une autre relation.

Le président du comité d'organisation d'une conférence est choisi parmi les membres du comité d'organisation (de cette même conférence).

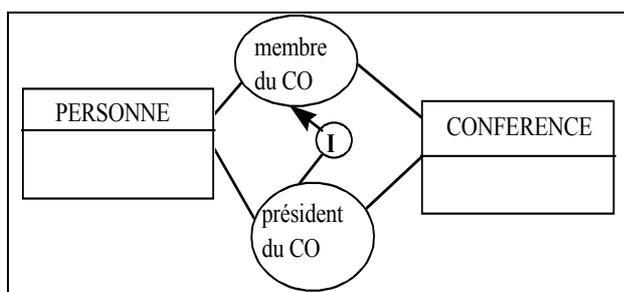


Figure 52 : inclusion sans objet pivot explicite

La relation président du CO est appelée "portée" de la contrainte d'inclusion,
 La relation membre du CO est la "cible" de la contrainte d'inclusion.

cas 2 : un objet pivot

Dans ce cas la contrainte d'inclusion porte sur l'ensemble des occurrences d'un objet (dit objet pivot). L'ensemble des occurrences de l'objet pivot présentes dans une des relations est inclus dans l'ensemble des occurrences de l'objet pivot présentes dans l'autre relation.

Si un employé est rattaché à un service il doit dépendre d'un site de l'entreprise (par contre il peut dépendre d'un site sans être rattaché à un service).

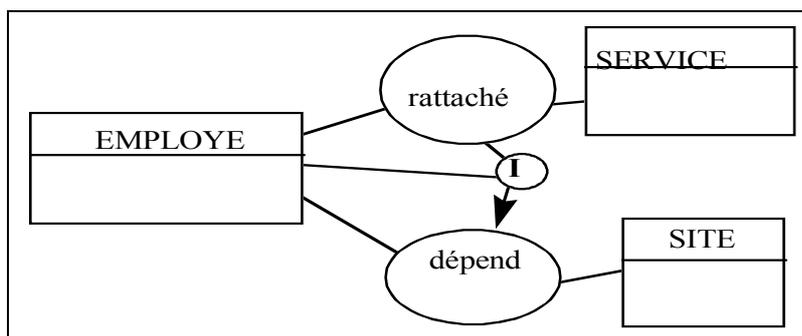


Figure 53 : inclusion avec objet pivot

portée de la CI : "rattaché", cible de la CI : "dépend", objet pivot : EMPLOYE

Note : dans le cas 1 le pivot est constitué des deux objets Personne et Conférence.

Contrainte d'égalité

Une contrainte d'égalité est une combinaison de deux inclusions symétriques.

Tout employé rattaché à un service dépend de l'un des sites de l'entreprise. Inversement tout employé dépendant d'un site doit être rattaché à un service.

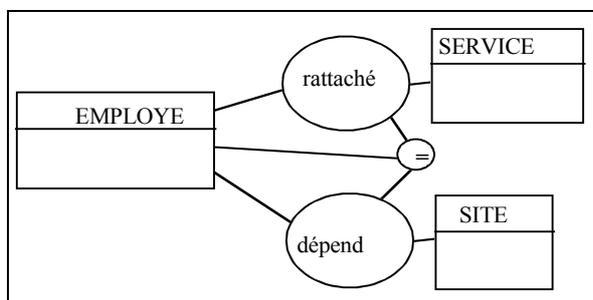


Figure 54 : égalité

Contrainte d'unicité

Une personne à une date donnée ne peut occuper qu'une et une seule fonction.

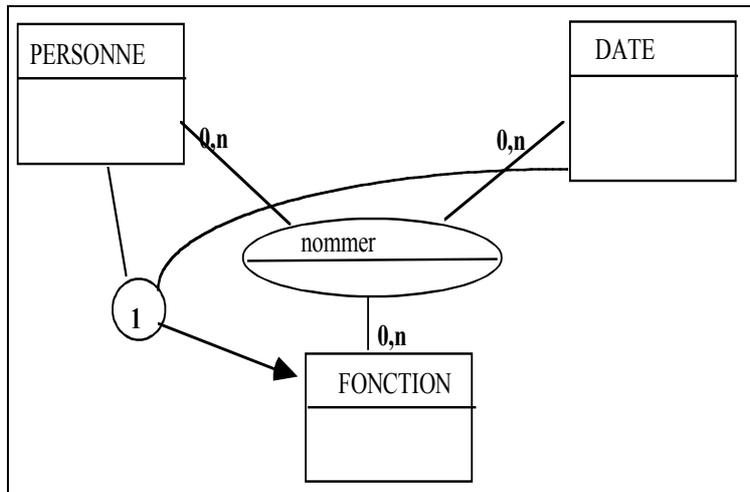


Figure 55 : unicité

Contrainte d'identification

Jusqu'à présent un objet était défini comme ayant une existence et donc une identification propre. Cependant il arrive que certains objets n'ont d'existence que par rapport à un ou plusieurs autres objets. Ses relations avec ces objets peuvent être utilisées pour l'identifier.

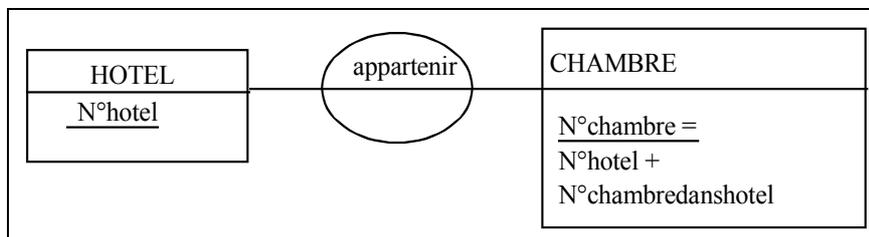


Figure 56 : identification

7. Les Modèles dynamiques

Les modèles précédents ont permis de décrire au niveau conceptuel :

- les aspects structurels par le MCD,
- les aspects fonctionnels par le MFC.

Il s'agit à présent de décrire les aspects comportementaux (dynamiques) du système. Dans Merise la représentation dynamique du système est décrite par le Modèle Conceptuel de Traitement. Merise 2 offre deux modèles complémentaires : le Modèle Conceptuel de Traitements Analytique (MCTA) et les cycles de vie des objets (CVO).

Quel que soit le modèle, il s'agit au niveau conceptuel de décrire l'activité de l'entreprise en éliminant les considérations d'organisation telles que la répartition du travail, le déroulement dans le temps, la nature des tâches et bien sûr les contraintes techniques.

Le formalisme introduit dans Merise 2 (MCTA) se distingue du formalisme classique de Merise (MCT) par les points suivants :

- il met en évidence l'interaction entre les données et les traitements (grâce au CVO),
- il s'appuie sur une définition rigoureuse des événements déclencheurs,
- il permet de représenter le parallélisme de certains traitements,
- il prend en compte le cas où à l'arrivée d'un événement le système ne se trouve pas dans l'état qui permettrait à une opération de se déclencher,
- il distingue les événements déclencheurs des opérations et les ressources (données consultées) nécessaires à l'exécution de ces opérations.

Le paragraphe suivant décrit le MCT originel introduit dans Merise. Nous présenterons ensuite les extensions apportées par le MCTA et enfin nous introduirons les CVO.

8. Modèle Conceptuel des Traitements (MCT)

8.1. Vocabulaire de base

La modélisation des aspects dynamiques repose sur le concept d'événements vus comme des stimulateurs de l'activité. Il s'agit d'obtenir une représentation de l'enchaînement des opérations du système et des conditions de déclenchement de son activité par des stimulations extérieures.

La dynamique du fonctionnement obéit à des règles issues d'un enrichissement des concepts des réseaux de Pétri.

Evénement

C'est la représentation d'un fait nouveau pour le système étudié ; ce fait est porteur d'information. C'est par exemple l'arrivée d'une déclaration de sinistre dans une compagnie d'assurance. Cet événement est porteur d'informations concernant l'assuré, le bien assuré et le sinistre affectant celui-ci.

Opération

C'est la réaction du système, sous forme de traitements, face à l'arrivée d'un événement ou d'un ensemble d'événements. L'arrivée de l'événement "Déclaration de sinistre" déclenche l'opération "vérifier la garantie".

Résultat

C'est la représentation de la réponse du système, générée par une opération. L'opération "vérifier la garantie" produit les résultats "sinistre pris en compte" ou "sinistre rejeté".

Synchronisation

C'est la représentation d'une pré-condition au déclenchement d'une opération. Elle peut faire intervenir plusieurs événements. Le déclenchement de l'opération "rembourser assuré" est conditionné par la présence des événements "rapport d'expertise" ET "facture de réparation".

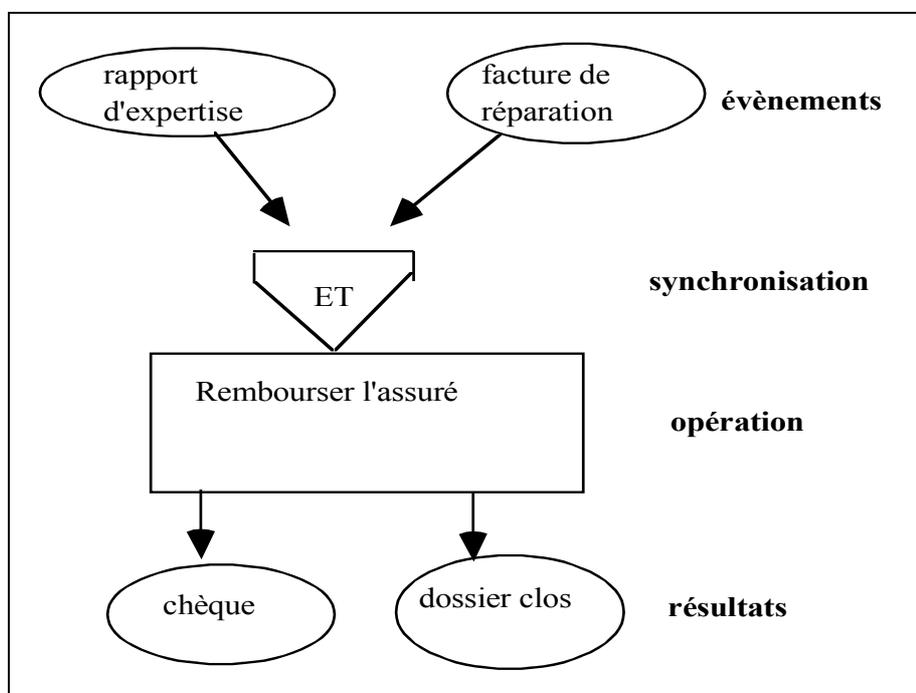


Figure 57 : concepts de base du MCT

8.2. Événement

Un événement est une circonstance portée à la connaissance du système et à laquelle il doit réagir. Pour qu'il y ait événement il faut que certaines conditions soient vérifiées :

- il doit se produire quelque chose (à l'extérieur ou à l'intérieur de l'organisation),
- ce fait doit être perçu par le système (qui devra donc se doter des moyens appropriés de le percevoir)
- ce fait n'intéresse le système que dans la mesure où il est identifié comme un déclencheur possible de son activité.

On distingue plusieurs types d'événements :

- les événements que le système perçoit en provenance de l'extérieur et sur lesquels il n'a aucune maîtrise : les **événements externes** ;
- le résultat d'une opération peut participer en tant qu'événement au déclenchement de l'opération suivante. On parle d'**événement interne**. Ce type d'événement ne peut déclencher l'opération suivante que s'il est combiné avec au moins un autre événement. Si ce n'est pas le cas, l'événement ne doit pas apparaître dans le modèle conceptuel, il s'agit la plupart du temps d'un résultat intermédiaire de l'opération.
- un résultat ne participant pas à la poursuite de l'activité du système et destiné à être émis vers l'extérieur : on parle alors d'**événements résultats** ou simplement de résultats.

8.3. Synchronisation

Il est fréquent de rencontrer des opérations dont le déclenchement est conditionné par plusieurs événements. Il faut alors représenter les conditions d'entrée, c'est à dire 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.

Pour faciliter la lecture il est recommandé d'attacher à chacun des événements concernés un label plus facile à manipuler que le nom de l'événement et qui représente une occurrence de l'événement.

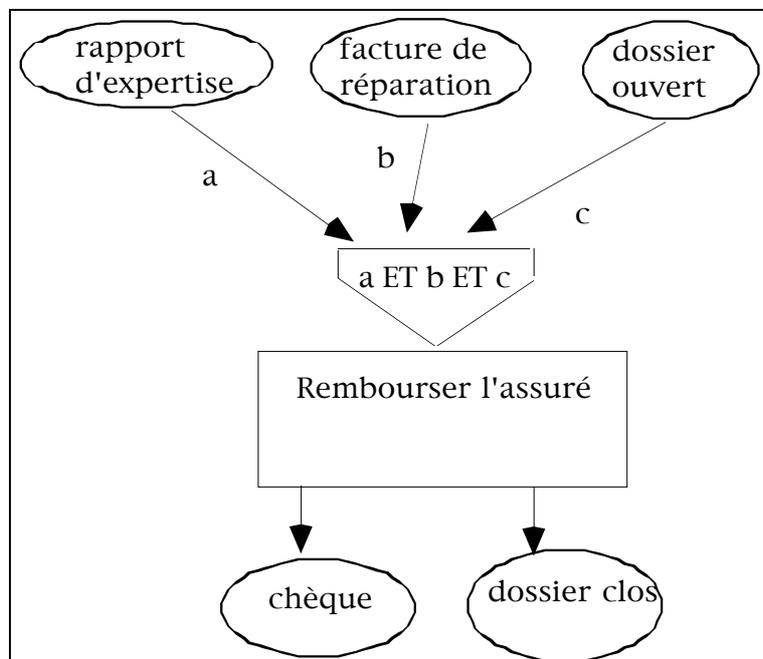


Figure 58 : synchronisation

8.4. Opération

L'arrivée d'un événement déclenche un ensemble de traitements appelé opération.

Une opération ne peut être déclenchée que par l'arrivée de l'événement ou de l'ensemble d'événements qui lui est rattaché. Cet ensemble d'événements peut être considéré comme les ressources nécessaires au bon déroulement de l'opération.

Une opération est une séquence continue d'actions et qui doit s'exécuter sans interruption (principe de non interruptibilité) dès qu'elle est déclenchée. Une opération constitue donc un bloc de traitement, qui ne doit pas souffrir d'aucune interruption durant son exécution (pas d'attente d'événement externe). L'exécution d'une opération produit l'émission d'événements internes (poursuite de l'activité) et/ou de résultats (signaux vers l'extérieur).

Une même opération peut produire plusieurs résultats. Nous avons vu qu'il est possible de représenter les conditions de déclenchement d'une opération. De la même manière, la production de résultats peut être soumise à des conditions de sortie de l'opération appelées **règles d'émission**.

Une opération est donc décrite par :

- sa désignation (un libellé + éventuellement un code),
- les actions élémentaires descriptives des travaux à accomplir; ces actions sont essentiellement des actions sur les données (consultation, mise à jour).
- les événements émis (événement internes ou résultats) et les conditions de ces émissions.

L'opération "vérifier garantie" déclenchée à l'arrivée de l'événement "déclaration de sinistre" génère les résultats : "sinistre rejetée" et "lettre de rejet" ou "sinistre accepté" et "dossier ouvert".

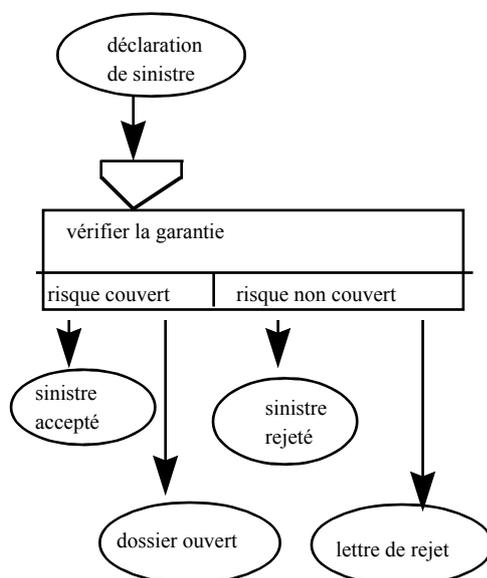


Figure 59 : opération

différents graphismes :

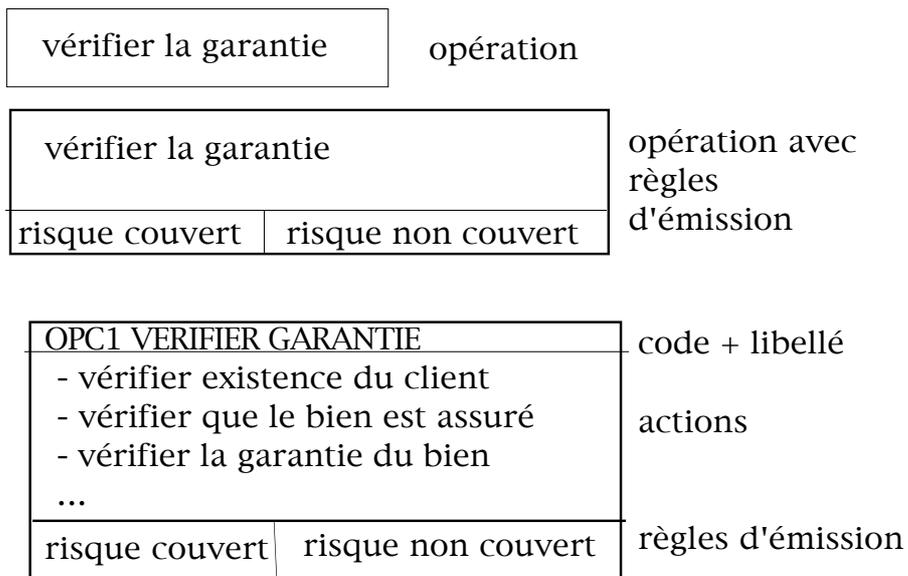


Figure 60 : formalismes graphiques des opérations

8.5. Le processus

Le processus est un enchaînement synchronisé d'opérations qui représente une unité de préoccupation de l'entreprise. Il est propre à un domaine d'activité.

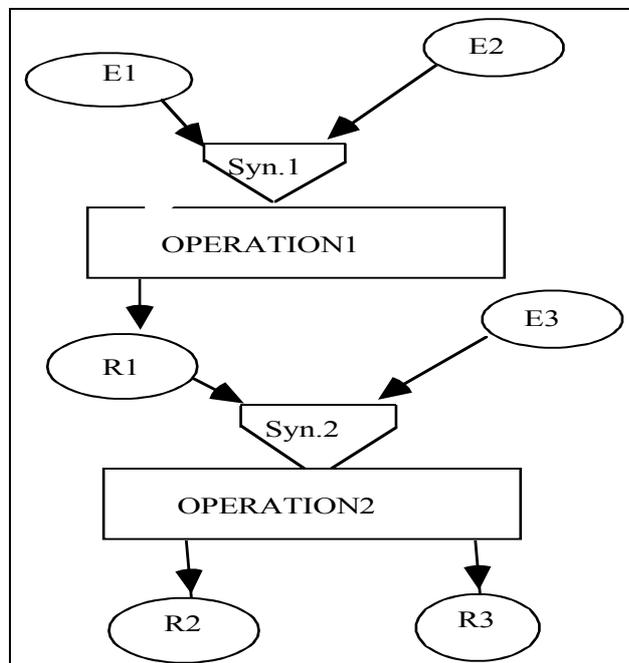


Figure 61 : processus

Exemple de processus

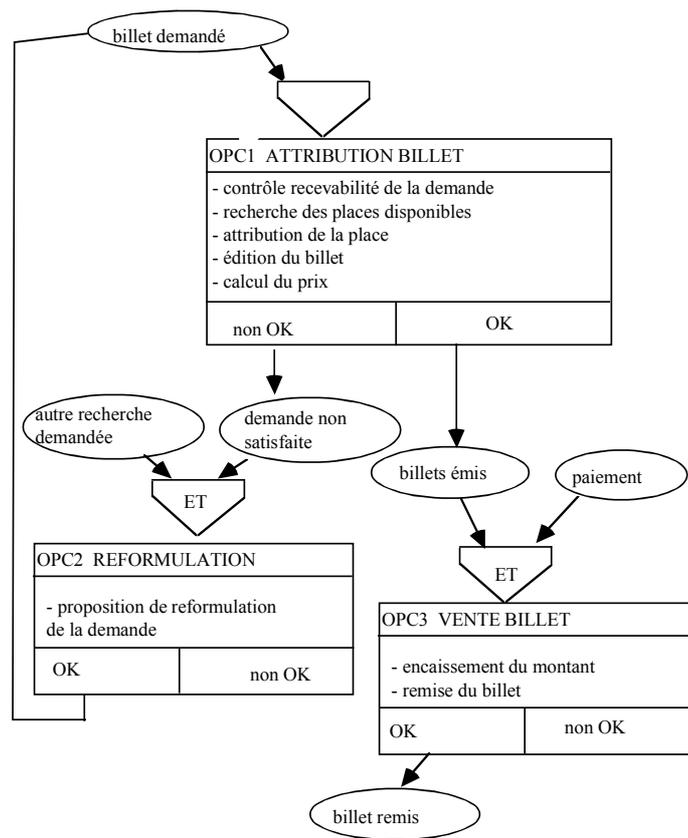
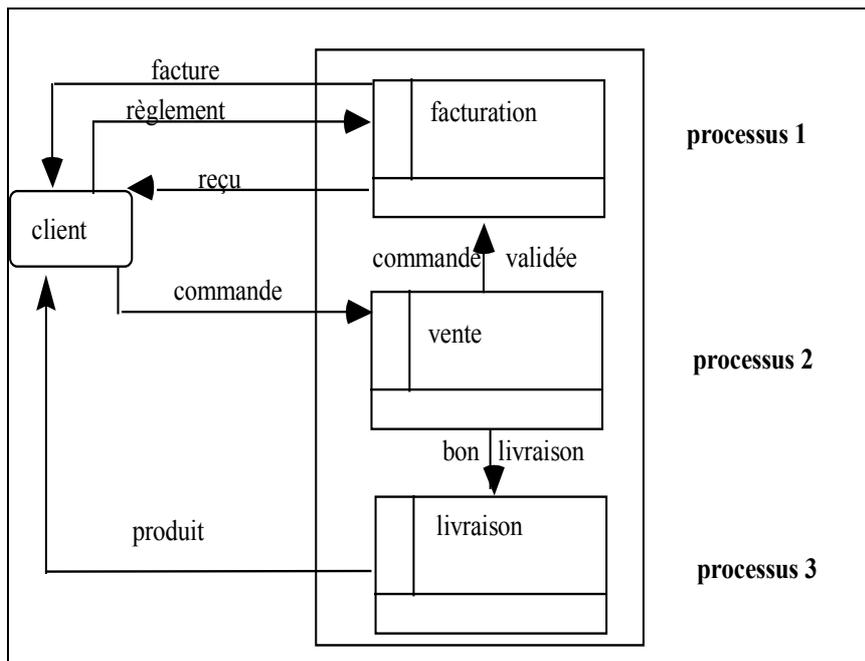


Figure 62 : exemple de processus

Pour décrire un domaine on peut être amené à définir plusieurs processus. Si le domaine est vaste, on le découpe en plusieurs sous-domaines chacun décrit par un processus.

Exemple : commercialisation des produits d'une entreprise.

La gestion des ventes reçoit des commandes puis transmet le double des bons de commandes à la facturation et les bons de livraisons à la gestion des livraisons. La facturation établit des factures, reçoit les règlements et expédie des reçus. La gestion des livraisons livrent les articles aux clients. Nous pouvons établir le diagramme de flux de niveau 1 suivant. Les sous-domaines à distinguer sont les activités vente, facturation et livraison. Chacun de ces sous-domaines est couvert par un processus distinct. Les opérations conceptuelles sont définies à chaque point d'arrivée des événements externes au processus décrit : "commande", "commande validée", "règlement", "bon de livraison".



L'ensemble des événements, opérations et résultats décrivant un processus constitue le modèle conceptuel de traitements de ce processus.

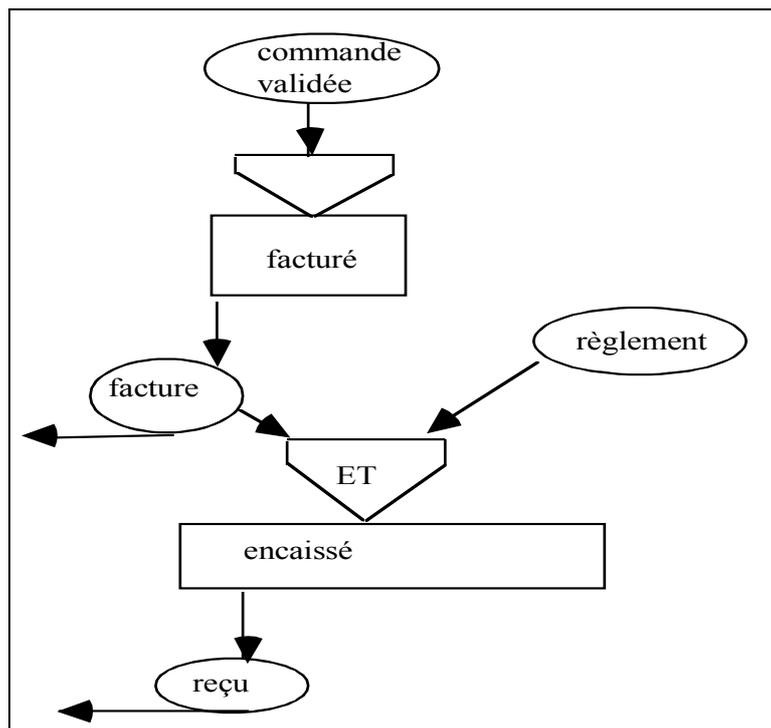


Figure 63 : MCT du processus facturation

On rappelle qu'une opération est interruptible. En particulier, elle ne peut pas s'interrompre dans l'attente d'un événement externe (ou d'une combinaison d'événements interne). Si tel est le cas il convient de décrire une seconde opération s'appuyant sur cet événement en attente.

Considérons par exemple le processus de facturation. Les résultats attendus sont d'une part une facture, d'autre part un reçu après règlement du client. La facturation ne nécessite pas le

paiement, mais l'émission du reçu ne peut s'effectuer qu'après celui-ci. Il y a donc présence de deux opérations : l'une générant la facture, l'autre le reçu.

Remarque : facture est à la fois un événement résultat (transmis à l'extérieur) et un événement interne participant au déclenchement de l'opération suivant. Cette double flèche traduit la règle de gestion imposant la facturation préalable à l'encaissement.

8.6. La consommation

Un événement peut être candidat au déclenchement de plusieurs opérations sachant qu'une occurrence de cet événement ne peut prendre part qu'à une seule de ces opérations.

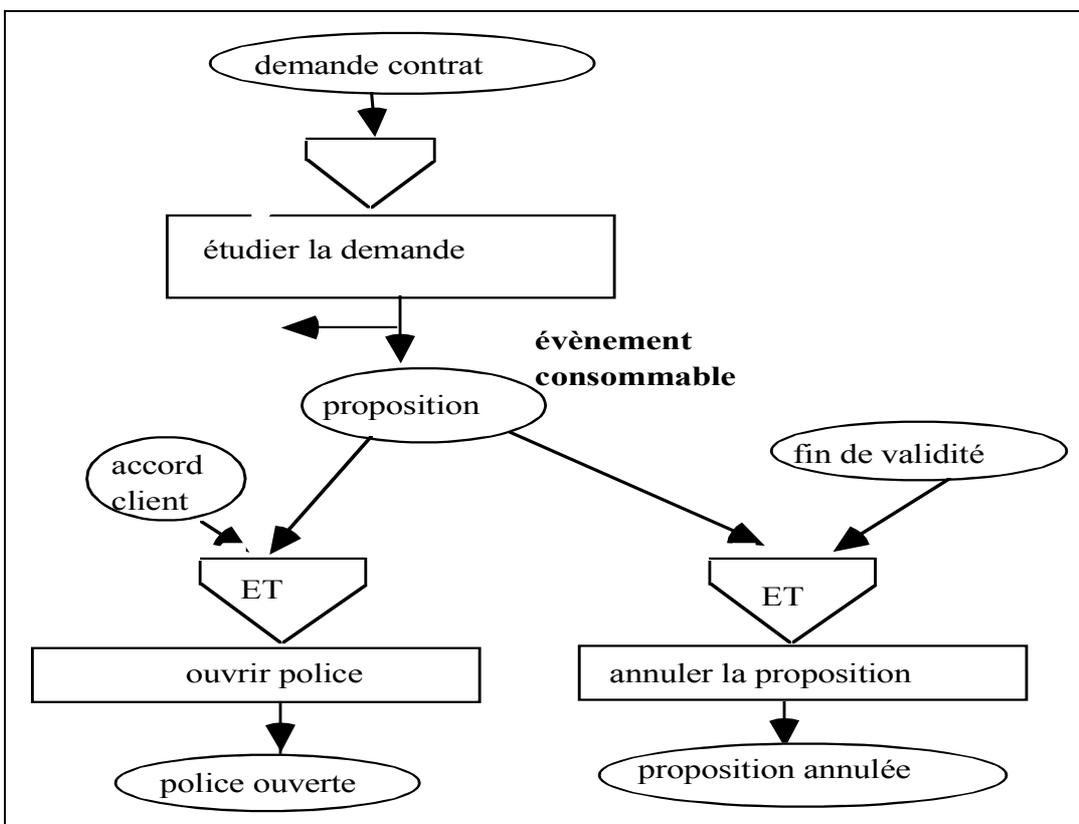


Figure 64 : événement consommable

Exemple : lorsqu'un client désire assurer un bien, il demande à la compagnie de lui faire une proposition. Une fois élaborée, cette proposition est soumise au client et reste valable durant un certain temps au bout duquel faute de réponse du client elle est annulée.

Cette représentation n'est acceptable que si les deux situations ne peuvent avoir lieu simultanément. Dans l'exemple précédent le client ne peut pas simultanément donner son accord et ne pas donner suite à la proposition.

8.6.1. Démarche d'élaboration du MCT

Le plus simple est de partir des modèles de flux. Le point d'arrêt de la décomposition avec le formalisme du MFC correspond au moment où l'activité peut être assimilée à une opération et est donc ininterrompible, mais le dernier niveau utile pour passer au MCT est plutôt le niveau précédent, celui où l'activité peut être assimilée à un processus. Il est donc souvent nécessaire d'aller un niveau trop loin dans le processus de décomposition des MFC pour être capable de déterminer le niveau où il est nécessaire de changer de formalisme.

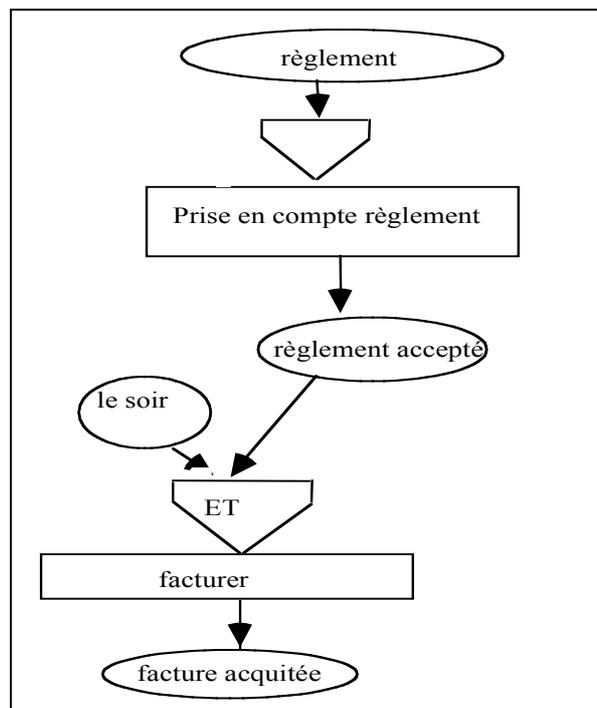
Le modèle de flux ne permet pas d'exprimer la notion de temps et de séquence. On doit alors passer à un raisonnement événementiel et faire intervenir le temps, les règles de synchronisation et les règles d'émission.

Nous avons vu que tous les flux d'information des Modèles Conceptuels de Flux sont transformés en événements dans le MCT. En l'absence d'événements temporels, un flux correspond donc à un événement et une activité correspond à une opération. Par contre, si des événements temporels doivent être pris en compte une activité peut être décomposée en plusieurs opérations conceptuelles : l'activité ininterrompible en terme de flux d'information devient interrompible par le temps.

Reprenons par exemple la figure 19. L'activité "gestion des règlements" est ininterrompible en terme de flux d'information : aucun flux d'information autre que le règlement n'est nécessaire au bon déroulement de cette activité. Supposons à présent l'existence de règles de gestion imposant que :

- les règlements soient immédiatement vérifiés dès leurs arrivées
- les factures acquittées soient envoyées aux clients tous les soirs.

L'événement temporel "tous les soirs" provoque une interruption temporelle durant l'activité "gestion des règlements" qui sera décomposée dans le MCT en deux opérations conceptuelles : la prise en compte des règlements et la facturation.



9. Modèle Conceptuel des Traitements Analytique (MCTA)

La tendance actuelle consiste à développer le couplage traitements-objets, notamment par l'introduction des Cycles de Vie des Objets. Les CVO permettent en effet de rendre compte des changements d'états des principaux objets des modèles de données, au cours de la vie des processus. Ces modifications d'états portées sur le MCTA complètent les opérations conceptuelles.

Nous commencerons par compléter la définition des opérations conceptuelles puis introduirons la notion d'état d'objet, notion qui sera reprise et approfondie lorsque nous présenterons les CVO.

9.1. Opération conceptuelle

L'opération conceptuelle est :

- déclenchée par un ou plusieurs événements,
- met en oeuvre un ensemble de règles conceptuelles formalisées,
- consulte et met à jour des entités de la mémoire permanente par l'intermédiaire d'actions,
- laisse les données du SI dans un état cohérent par rapport aux contraintes d'intégrité de la mémoire permanente,
- ne peut être interrompue par l'attente d'un événement externe ou temporel.

Représentation graphique:

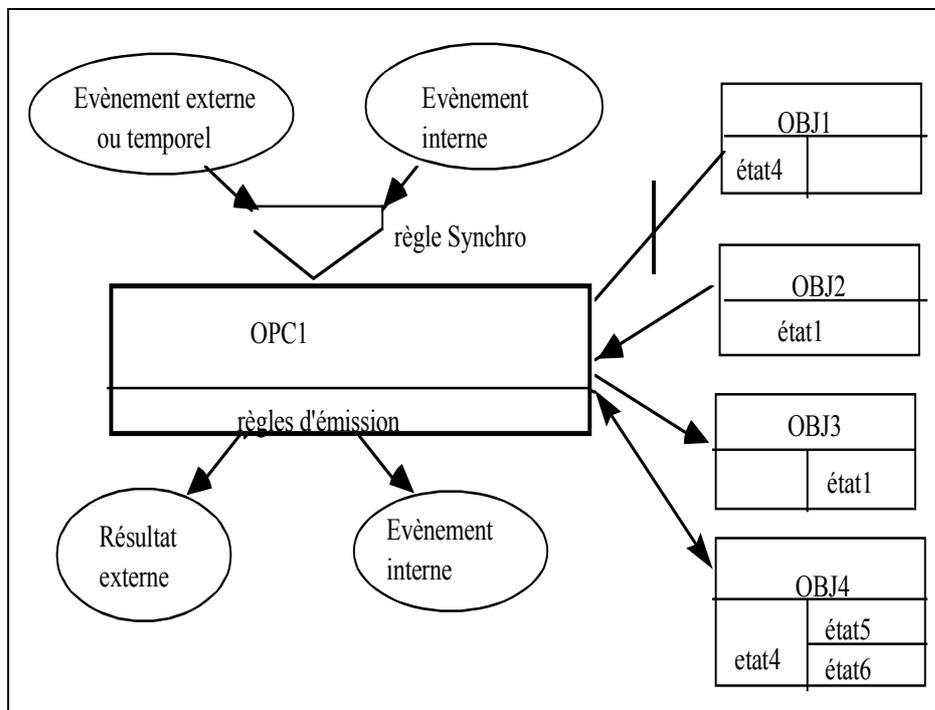


Figure 65 : représentation graphique des opérations dans le MCTA

Exemple intuitif : Nous sommes dans le cadre d'une formation permanente. Une demande d'inscription à un stage déclenche l'opération " enregistrer candidature ". Cette opération :

- s'assure que le stage existe,
- crée une occurrence de l'entité stagiaire dans l'état candidat.

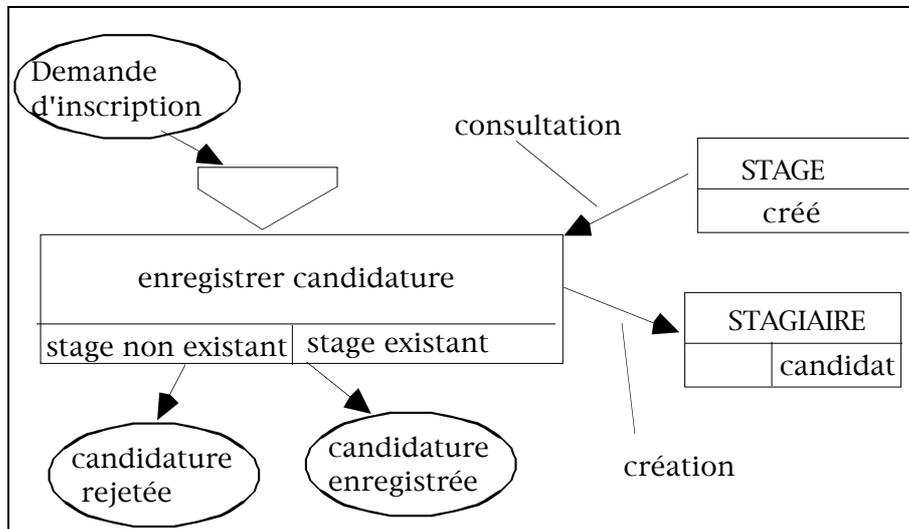


Figure 66 : Opération " enregistrer candidature "

9.2. Etat d'objet

Un état d'objet est un stade transitoire par lequel passe un objet (une entité de la mémoire permanente) au cours de son cycle de vie.

Exemple :

- "à livrer", "en attente", "livrée", "facturée" : états de COMMANDE
- "créé", "complet", "ouvert", "en cours" : états de STAGE
- "candidat", "recalé", "reçu", "inscrit", "en attente", "en formation" : états de STAGIAIRE

Les états des objets sont représentés dans le MCTA pour montrer l'évolution des objets durant l'exécution des opérations conceptuelles.

Représentation graphique dans le MCTA

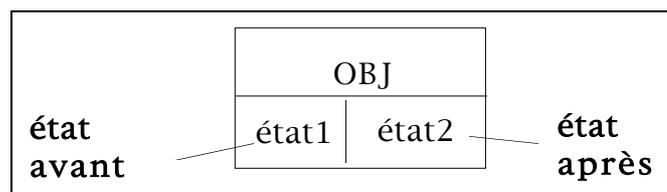


Figure 67 : états des objets avant et après exécution de l'opération

Les différents cas possibles :

création d'une occurrence de l'obj1 dans un état1.	<table border="1"> <tr><td colspan="2">OBJ1</td></tr> <tr><td></td><td>état1</td></tr> </table>	OBJ1			état1	<table border="1"> <tr><td colspan="2">STAGIAIRE</td></tr> <tr><td></td><td>candidat</td></tr> </table>	STAGIAIRE			candidat				
OBJ1														
	état1													
STAGIAIRE														
	candidat													
consultation d'une occurrence de l'obj2 dans un état1	<table border="1"> <tr><td colspan="2">OBJ2</td></tr> <tr><td colspan="2">état1</td></tr> </table>	OBJ2		état1		<table border="1"> <tr><td colspan="2">STAGE</td></tr> <tr><td colspan="2">créé</td></tr> </table>	STAGE		créé					
OBJ2														
état1														
STAGE														
créé														
suppression d'une occurrence de l'obj3 qui se trouve dans un état1	<table border="1"> <tr><td colspan="2">OBJ3</td></tr> <tr><td>état1</td><td></td></tr> </table>	OBJ3		état1		<table border="1"> <tr><td colspan="2">STAGE</td></tr> <tr><td>créé</td><td></td></tr> </table>	STAGE		créé					
OBJ3														
état1														
STAGE														
créé														
modification d'une occurrence de l'obj4. Elle passe de l'état4 à l'état5 ou l'état6.	<table border="1"> <tr><td colspan="2">OBJ4</td></tr> <tr><td>etat4</td><td>état5</td></tr> <tr><td></td><td>état6</td></tr> </table>	OBJ4		etat4	état5		état6	<table border="1"> <tr><td colspan="2">STAGIAIRE</td></tr> <tr><td>candidat</td><td>recalé</td></tr> <tr><td></td><td>reçu</td></tr> </table>	STAGIAIRE		candidat	recalé		reçu
OBJ4														
etat4	état5													
	état6													
STAGIAIRE														
candidat	recalé													
	reçu													

Figure 68 : Evolution des états d'objets

Dans le cas d'une consultation on peut :

- soit ne pas représenter l'état de l'objet, si l'état n'a pas d'importance ;
- soit représenter l'état de l'objet, c'est le cas d'une consultation qui s'assure qu'un objet est bien dans un état donné.

9.3. Action

Une action est une manipulation des données d'un objet ou d'une relation du SI. Elle peut créer / consulter / modifier / supprimer **une et une seule** occurrence d'entité (objet ou relation) de la mémoire permanente. Dans les MCT les actions s'expriment en langue naturelle à l'intérieur des opérations. Dans le MCTA les actions agissant sur les états d'objets sont représentées graphiquement.

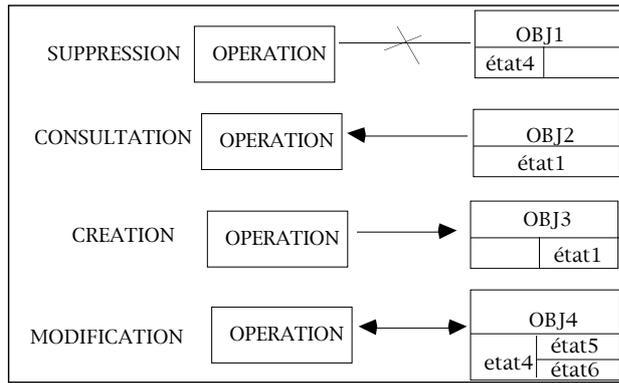


Figure 69 : actions des opérations dans le MCTA

L'ordre des actions d'une opération conceptuelle sera, dans la mesure du possible, présenté de haut en bas. L'opération conceptuelle ENREGISTRER CANDIDATURE est composée de deux actions : une consultation permettant de vérifier que le stage est créé, une création d'un stagiaire dans un état candidat.

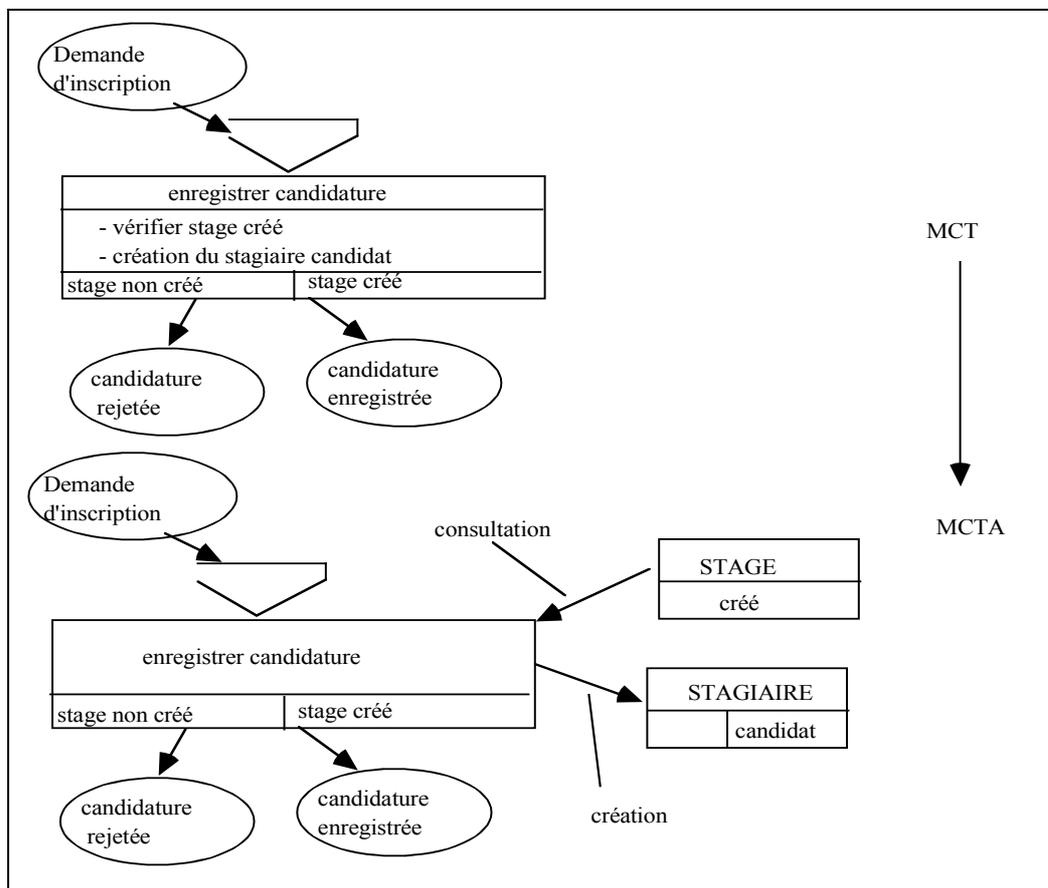


Figure 70 : actions du MCT au MCTA

Une action de consultation peut exprimer une condition de déclenchement sur une mise à jour. C'est le cas ici où l'on souhaiterait exprimer que le stagiaire est créé (état candidat) si le stage existe (état créé). Dans ce cas il s'agit d'exprimer explicitement une condition de déclenchement sur une opération de mise à jour (Figure 71).

Une action peut donc être accompagnée d'une condition de déclenchement qui peut ou non être itérative.

9.4. Condition sur les actions

Condition de Déclenchement

Le déclenchement d'une action peut être soumis à une condition, c'est à dire à une règle qui porte sur l'état de la structure de données au moment où l'événement est constaté.

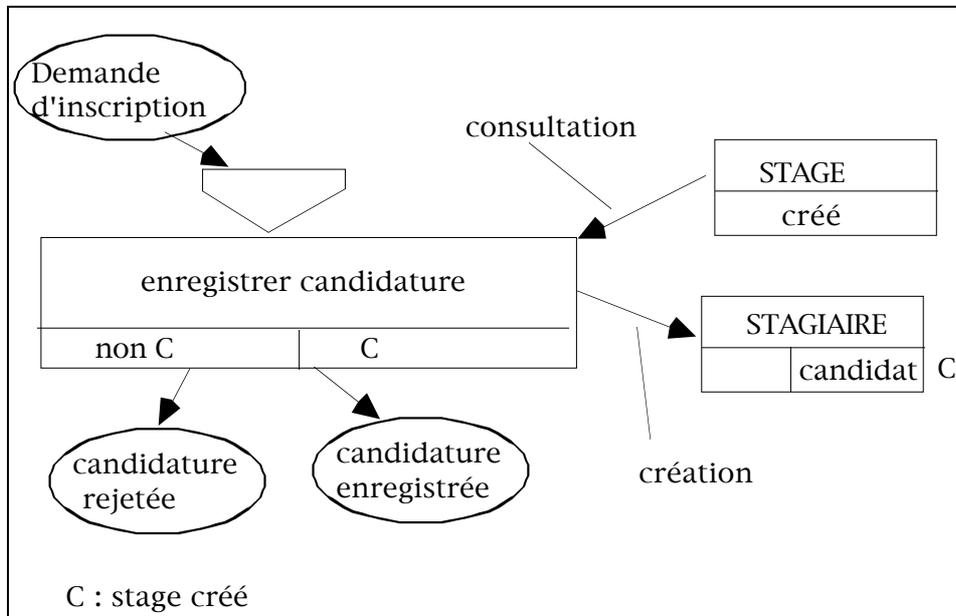
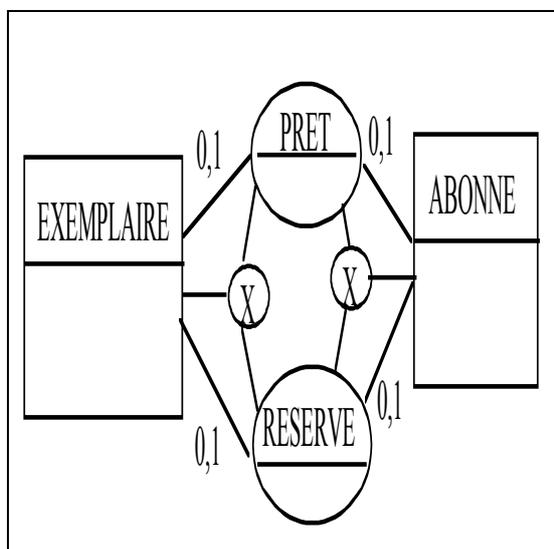


Figure 71 : condition de déclenchement

La condition de déclenchement C permet le respect des contraintes d'intégrité statiques exprimées dans le modèle de données.

Dans l'exemple ci-dessous un exemplaire est soit prêté, soit réservé (il ne peut être les deux). S'il n'est ni l'un ni l'autre il est disponible. Un abonné peut emprunter un livre (il devient actif) ou établir une réservation (il est en attente). Un abonné inactif n'a ni prêt, ni réservation en cours.



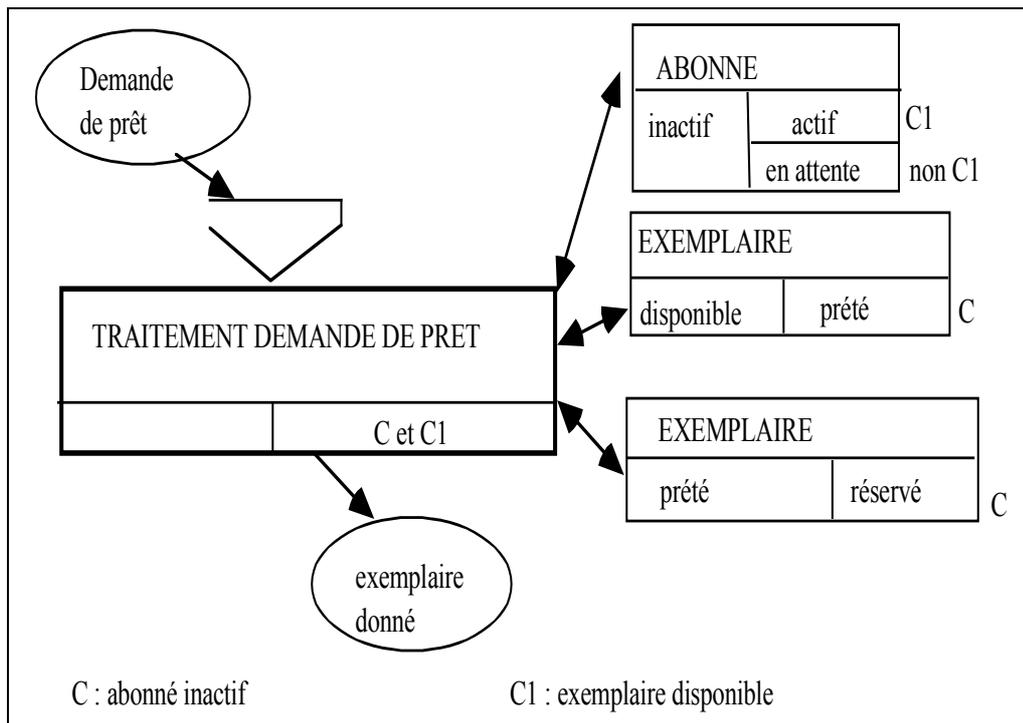


Figure 72 : condition de déclenchement et respects des CI

Condition Itérative

Le déclenchement itératif d'une action peut être contrôlé par une règle qui définit les paramètres de l'itération. Dans le schéma ci-dessous le N signifie que l'opération intervient sur toutes les occurrences de l'objet concerné respectant la condition de déclenchement. On parle d'action collective.

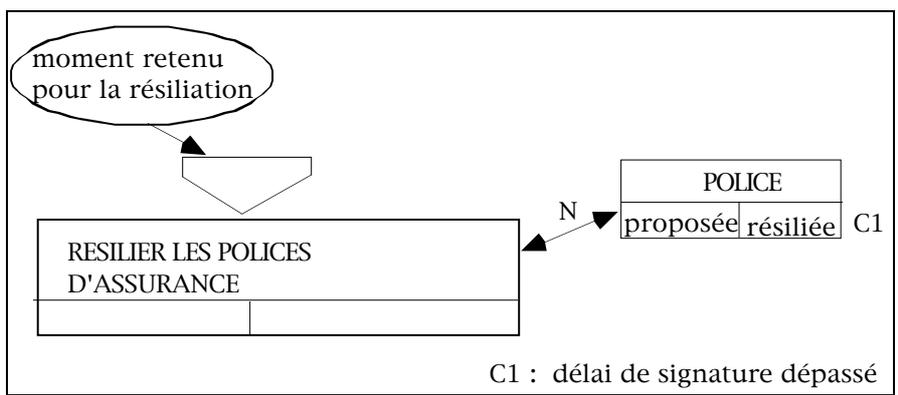


Figure 73 : condition itérative

9.5. Événement interne

Un événement interne correspond à un changement d'état particulier d'objet. Provoqué par une opération, il participe à son tour au déclenchement de l'exécution d'une opération.

- L'événement interne mis en évidence au niveau du MCTA se situe entre deux états cohérents du système. Par exemple dans le MCTA ci-dessous on exprime qu'une commande peut être créée sans être facturée.

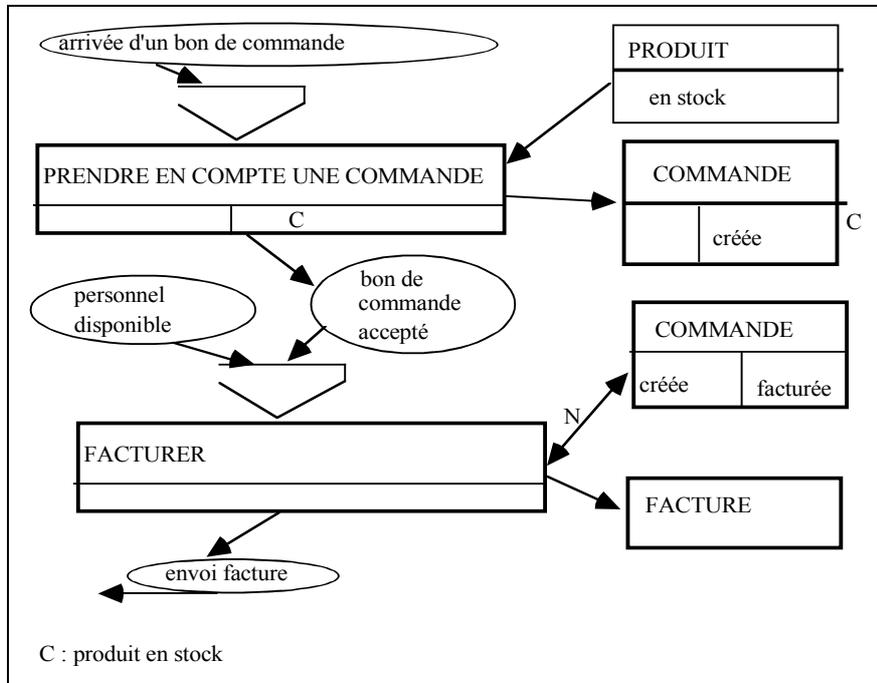


Figure 74 : événement interne déclencheur

- Les ressources informationnelles nécessaires à l'exécution d'une opération, mais non déclenchantes, sont représentées par des actions de consultation et non plus par des événements internes comme dans le MCT (Figure 75).

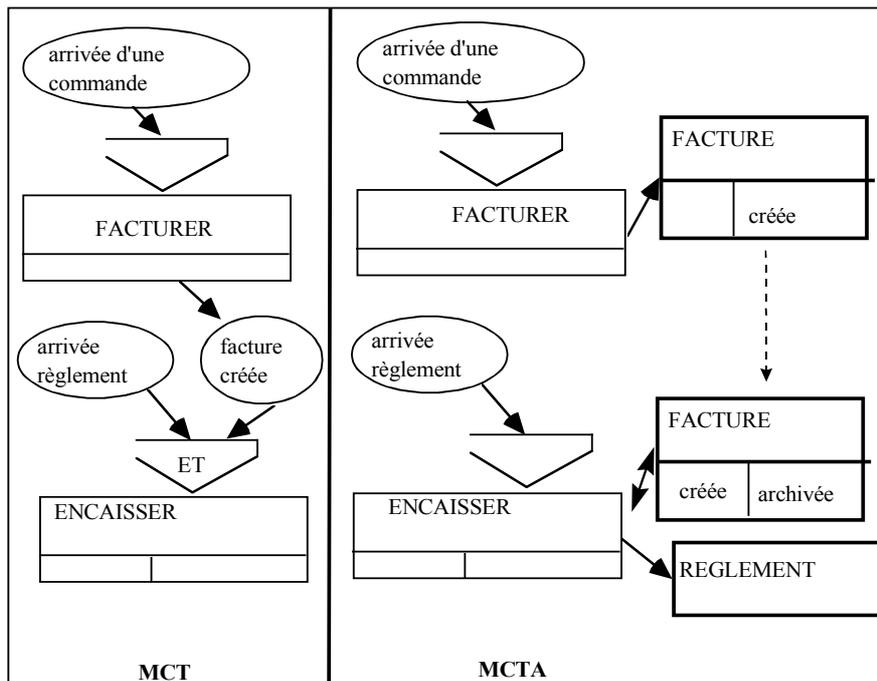


Figure 75 : événement interne non déclencheur

Remarquons que dans l'exemple précédent (Figure 74) l'événement interne "bon de commande accepté" n'est pas remplacé par une opération de consultation. En effet, l'opération Facturer est déclenchée si un membre du personnel est disponible ET au moins un bon de commande a été accepté.

9.7. Passage des modèles de flux au MCTA

La technique présentée dans le chapitre précédent (élaboration du MCT) reste la même. Il s'agit donc de décomposer les activités des modèles de flux jusqu'à ce que l'activité corresponde à une opération conceptuelle ou à un processus.

On peut également passer d'abord par un MCT, puis se focaliser sur les CVO (cf. paragraphe suivant) pour finalement transformer le MCT en MCTA. Ne pas oublier alors de supprimer les événements internes du MCT correspondant à des consultations.

10. Les Cycles de Vie des Objets

10.1. Introduction

Une entité décrite dans le modèle de données n'est pas figée à tout jamais. Elle passe durant sa vie par différents états qui peuvent être constatés à partir des diverses valeurs prises par :

- ses propriétés,
- ou celles des objets qui lui sont liés,
- ou encore par sa participation (ou sa non participation) dans les relations où elle intervient.

Cette succession d'états est représentée par un graphe orienté dont les sommets sont de deux types :

- les divers états de l'objet,
- les événements provoquant le passage d'un état à un autre.

Événement et Transition

La définition de l'événement reste identique. Un événement peut être externe, interne, temporel. Dans le CVO les événements déclenchent la transition d'un état à l'autre.

Un événement peut déclencher plusieurs transitions.

Un événement interne est généré par au moins une transition d'états d'objet.

Une transition correspond au passage d'un objet d'un état dans un autre. Elle est représentée par un double arc orienté, reliant deux états par l'intermédiaire d'un événement.

Représentation graphique et exemple partiel

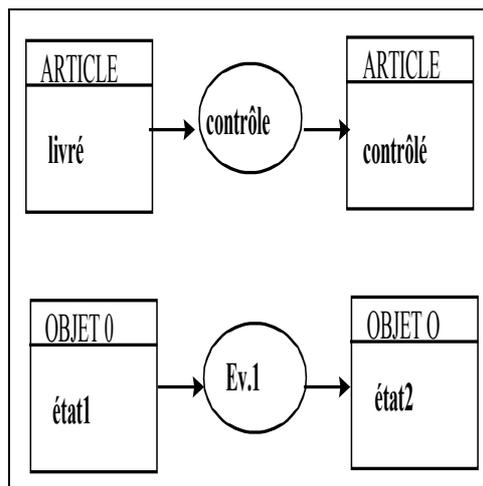


Figure 76 : état, événement et transition dans les CVO

Objectifs

- donner une vision synthétique du cycle de vie des principaux objets,

- compléter le MCD en mettant en évidence des contraintes d'intégrité statiques entre les états d'objet et les propriétés spécifiques aux états,
- mettre en évidence des contraintes d'intégrité dynamiques sur les enchaînement d'états,
- faciliter la construction du MCTA.

Commentaires MCTA / CVO

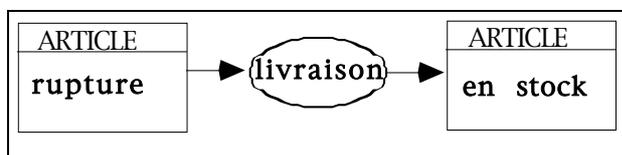
Les deux modèles représentent les aspects dynamique du SI, cependant :

- les CVO modélisent le cycle de vie de chaque objet du système alors que le MCTA modélise le cycle de vie du système.
- le CVO est construit autour de l'objet et décrit l'ensemble des événements affectant l'objet au cours de son cycle de vie. Le MCTA est construit autour des opérations et donne une vision globale des conséquences d'un même événement
- le MCTA donne une vision synthétique de la coordination des événements, le CVO donne une vision synthétique de la coordination entre états.

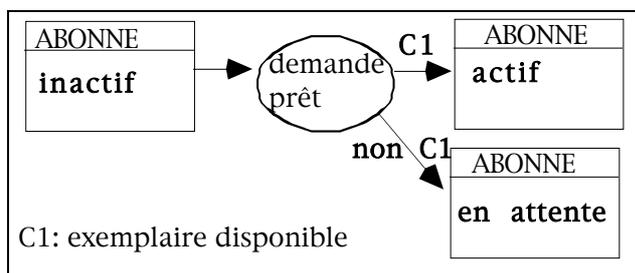
10.2. Les principales transitions

Les trois schémas de base qui peuvent être représentés sur un CVO sont la séquence, l'alternative et l'itération.

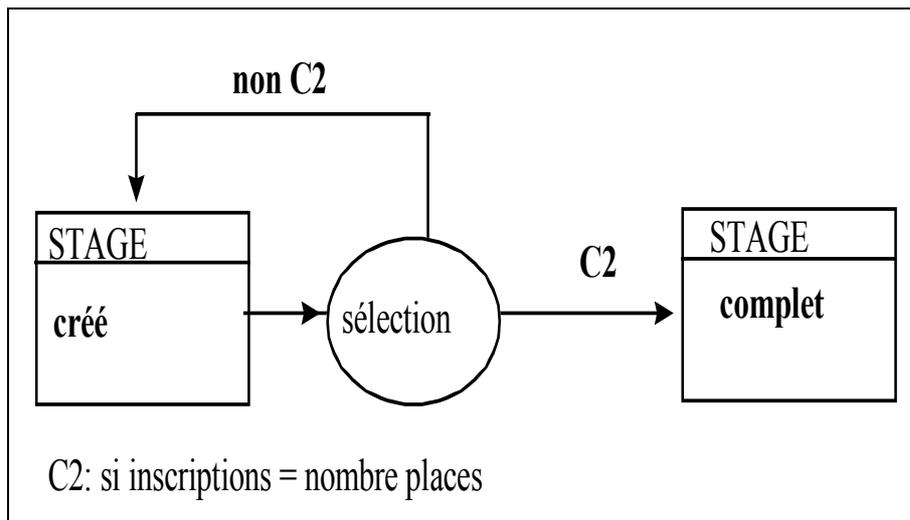
Séquence



Alternative



Itération



En général les cycles de vie des objets sont interconnectés. Lors de la représentation du CVO d'un objet (dit objet principal) on pourra représenter par une flèche pointillée les transitions d'états sur les objets qui lui sont liés.

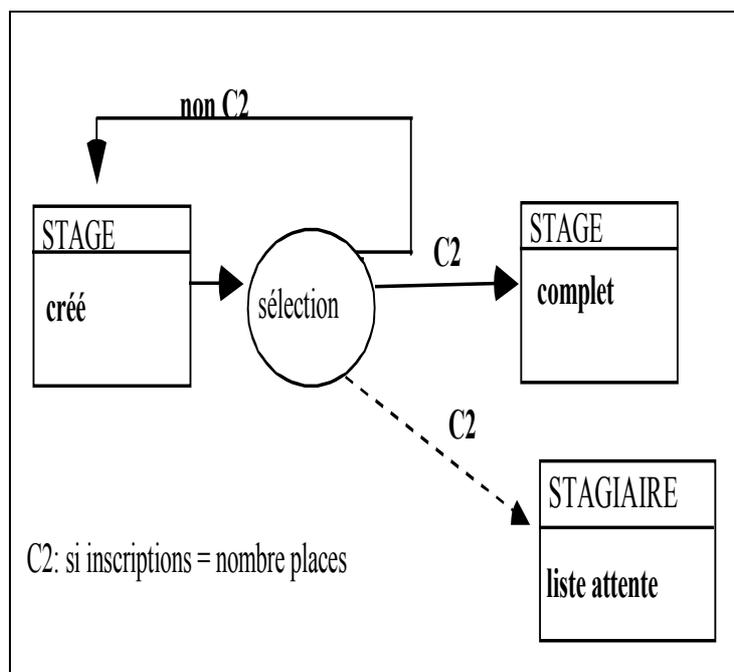


Figure 77 : CVO interconnectés

10.3. Etat d'objet

Nous reprenons la définition introduite précédemment :

Un état d'objet est un stade transitoire par lequel passe un objet (une entité de la mémoire permanente) au cours de son cycle de vie.

Le CVO d'un objet comporte :

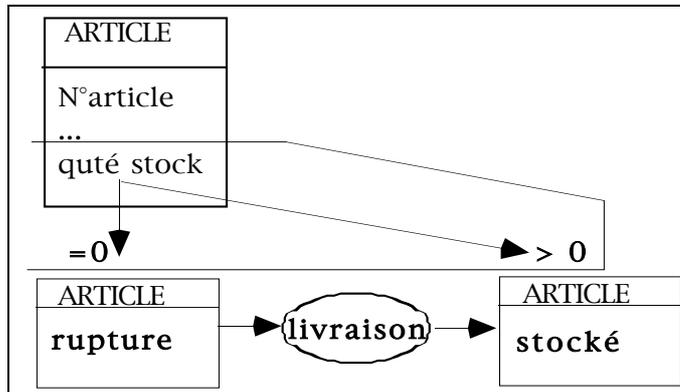
- un ou plusieurs états initiaux,

- un ou plusieurs états intermédiaires,
- un ou plusieurs états finaux.

Les états d'un objets sont identifiés à partir :

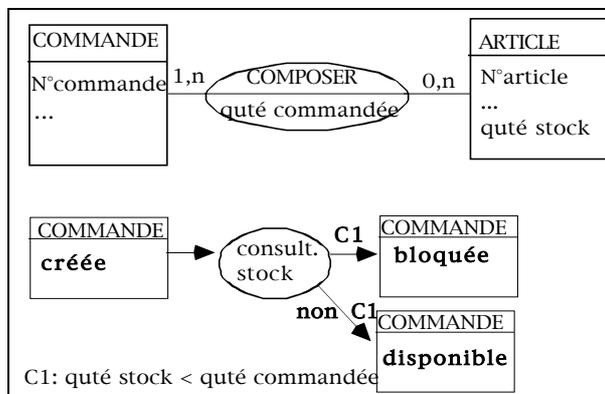
- *de la valeur de propriétés de l'objet*

Un article est mis dans l'état "stocké" si la propriété "quantité en stock" est supérieure à 0.



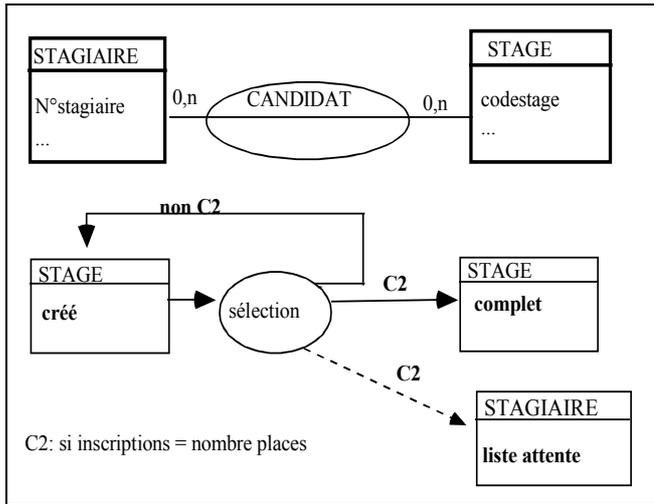
- *de la valeur des propriétés des objets qui lui sont liés*

Une commande est dans l'état "bloquée" si la quantité en stock d'un article commandé est inférieure à la quantité commandée.



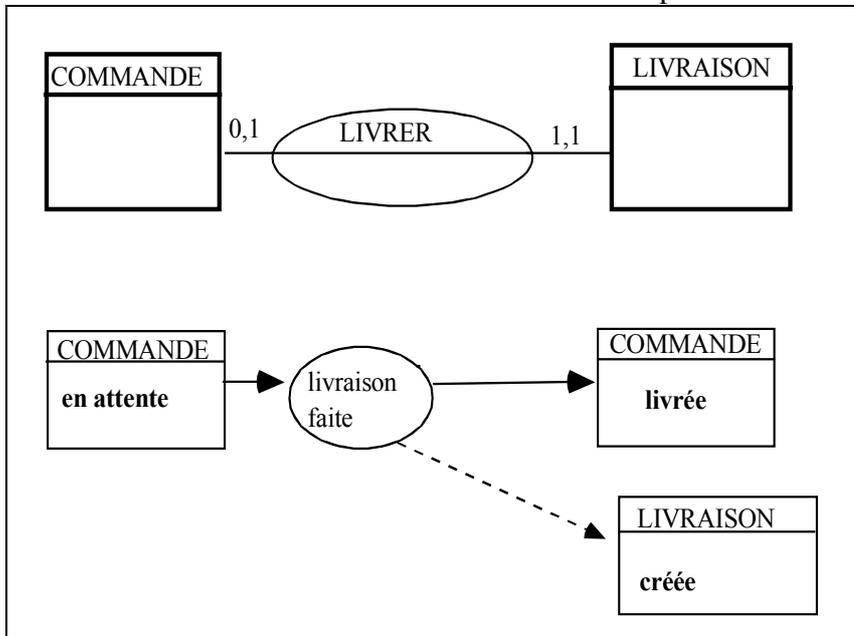
- *de la valeur des états des objets qui lui sont reliés*

Un stagiaire est dans l'état "liste d'attente" si le stage auquel il est candidat est dans l'état "complet".



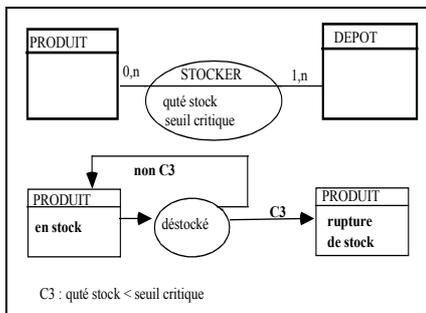
• *de ses relations ou absence de relations avec d'autres objets*

Une commande est dite "en attente" si elle n'intervient pas dans la relation "livrer".



• *de la valeur des propriétés de ses relations avec d'autres objets*

Un produit est dit "en rupture de stock" dans un dépôt si sa quantité en stock est inférieure à un seuil critique.



A chaque état d'objet ou configuration d'états il est donc important d'associer des contraintes d'intégrités qui correspondent à des états cohérents de la structure de données. Ces contraintes d'intégrités seront exprimées avec le CVO des objets.

Etats d'objets et sous-types

C'est généralement le caractère transitoire de l'état d'objet qui le différencie d'un sous-type.

Exemple : HOMME et FEMME sont des sous-types de PERSONNE
 "célibataire", "veuf", "divorcé"... sont des états de l'objet PERSONNE.

Dans un MCD il est préférable d'utiliser la spécialisation par "catégorie" c'est à dire qu'il n'y a pas de transfert d'une occurrence d'un sous-type à un autre.

Il n'est cependant pas interdit de faire de la spécialisation par "état" c'est à dire de permettre des transferts d'occurrence.

10.4. Spécialisation de CVO

On peut représenter le CVO d'un objet à différents niveaux de détail. C'est ainsi que dans un premier temps nous identifierons l'état "en vente" d'un article ce qui correspond à un état "standard" pendant un processus de vente. Cet état peut ensuite être décomposé pour mettre en évidence des états particuliers par lequel passe cet objet durant le même processus. C'est ainsi que nous pourrions identifier l'état "en vente normale" et "en vente soldée".

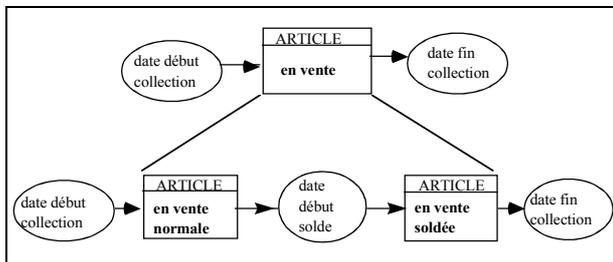


Figure 78 : Spécialisation d'états

10.5. Variables d'état et Structures parallèles

• Variables d'état et structures parallèles

Un objet peut, à un certain moment de sa vie, être dans plusieurs états. Un logement peut être "libre" ou "occupé" tout en étant par ailleurs "en vente", "vendu", "en location" ou "loué". Il peut donc être intéressant de gérer plusieurs variables d'état pour un objet. L'objet admet différents états caractérisés par la valeur de ses différentes variables.

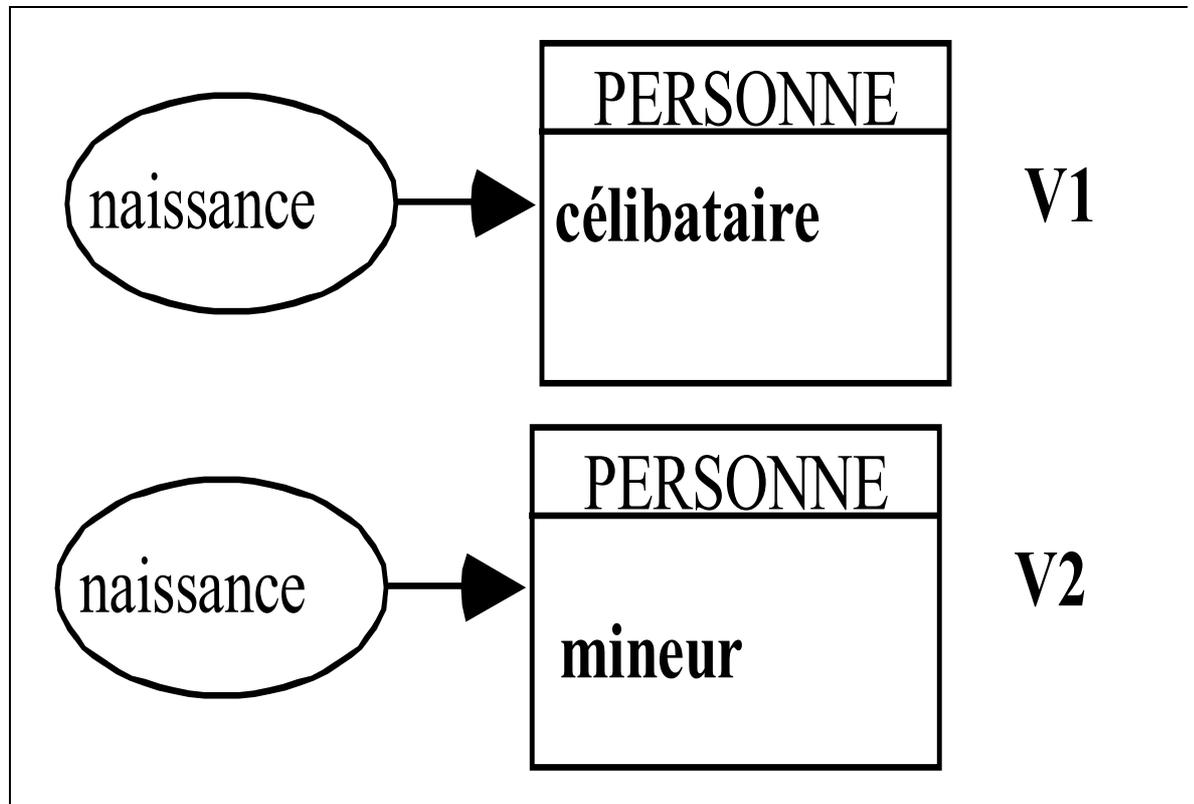
La vie d'une variable d'état peut dépendre partiellement ou totalement de la vie d'une ou plusieurs autres variables d'état de ce même objet. On a alors des structures parallèles dans le CVO.

Exemple : Pour l'objet PERSONNE on s'intéresse à deux variables d'états

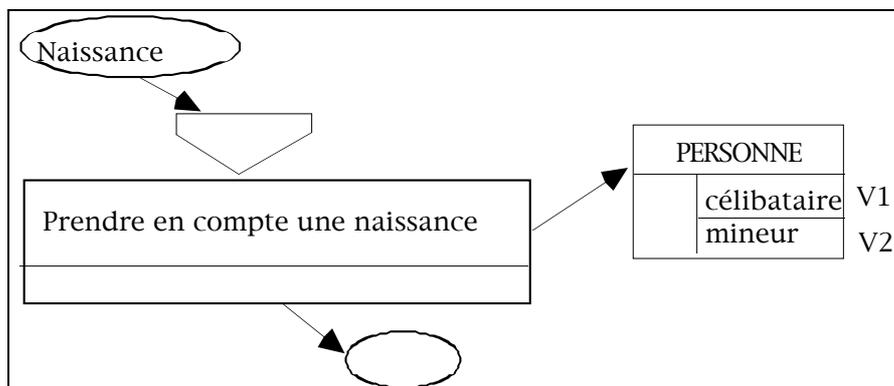
- V1 correspond à l'état matrimonial,

- V2 correspond à l'état civil.

L'événement "naissance" met la personne dans l'état "célibataire" (/V1) et "mineur" (/V2). Nous sommes dans le cas où un même événement fait passer l'objet dans deux états.



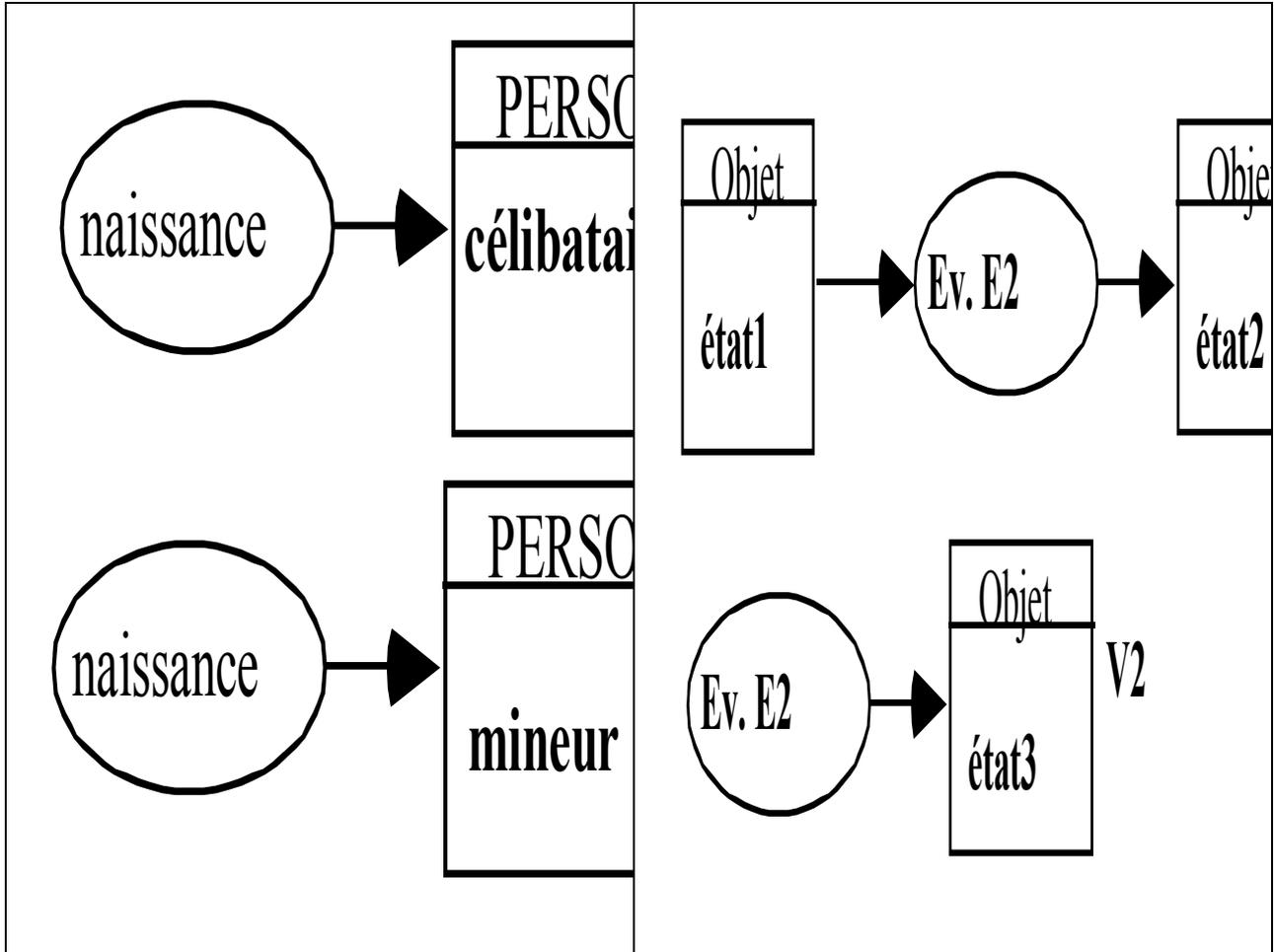
Il est possible au niveau du MCTA de faire apparaître le fait qu'une même opération déclenchée par le même événement modifie plusieurs variables d'état d'un objet.



- Les différents cas de structures parallèle

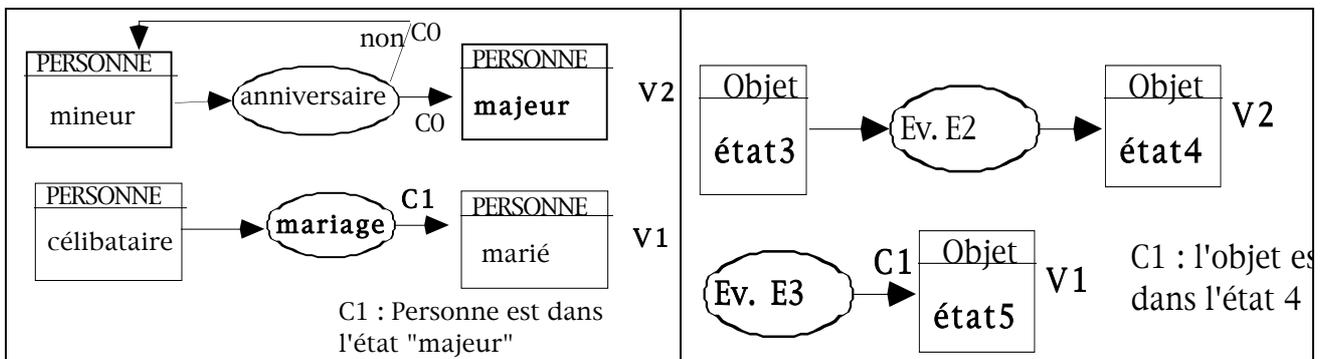
1er cas : un événement commun aux deux variables

C'est le cas vu précédemment ; à l'arrivée d'un événement E2 (naissance) l'objet (personne) passe dans l'état 2 et 3 (célibataire et mineur).



2ème cas : dans le CVO d'une variable, consultation de l'état d'une autre variable.

A l'arrivée de E3 (mariage), l'objet (personne) ne passe dans l'état 5 (marié) que si C1 est vérifiée (si personne est "majeur").



3ème cas : Une transition dans le cycle de vie d'une variable constitue un événement interne qui déclenche une transition sur le cycle de vie d'une autre variable.

Autrement dit la transition état1-état2 constitue un événement interne qui fait passer l'objet A dans un état 3. Intuitivement, le passage de l'état "mineur" à "majeur" vis à vis de la variable "état civil", provoque le passage à l'état "électeur" de la variable "état administratif".

10.6. Vérification et Elaboration des CVO

Règles de vérification

Les règles suivantes sont destinées à vérifier qu'aucun événement ou état n'a été oublié dans un CVO. Elle contribuent également à valider le MCTA en vérifiant sa complétude.

- Pour chaque objet on doit avoir :
 - un événement déclenchant sa création dans un état initial,
 - un événement pour chaque transition possible d'état,
 - un événement qui déclenche sa transition vers un état final ou sa suppression.

Note : la création pourra être représentée par une transition sans état initial, la suppression par une transition sans état final

- Chaque événement doit affecter au moins un objet.
- Chaque objet doit être affecté par au moins un événement.

Enchaînement des modèles (cf. page suivante)

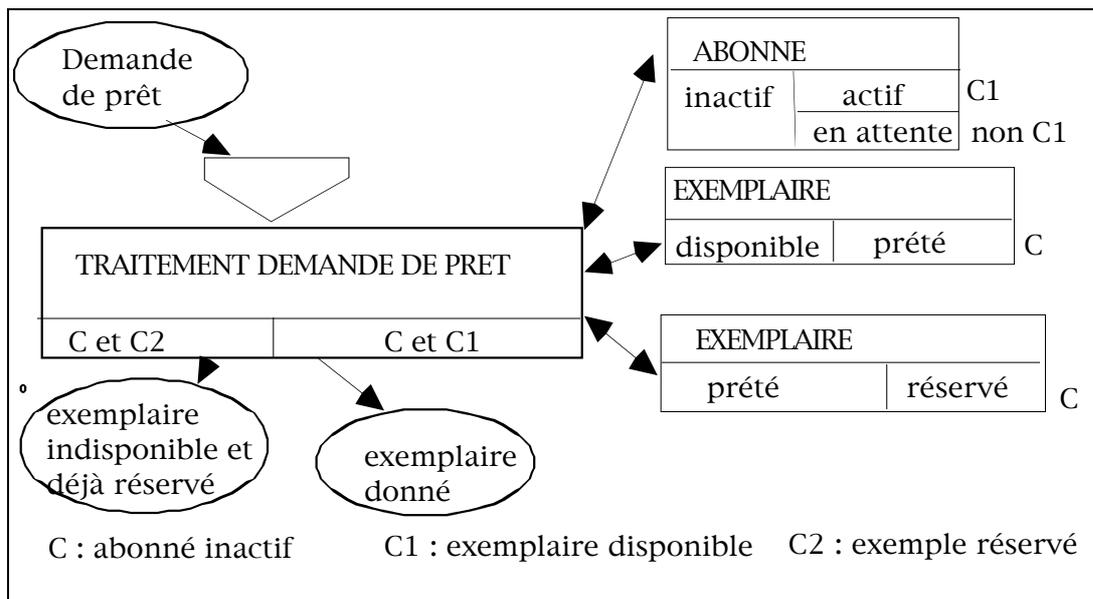
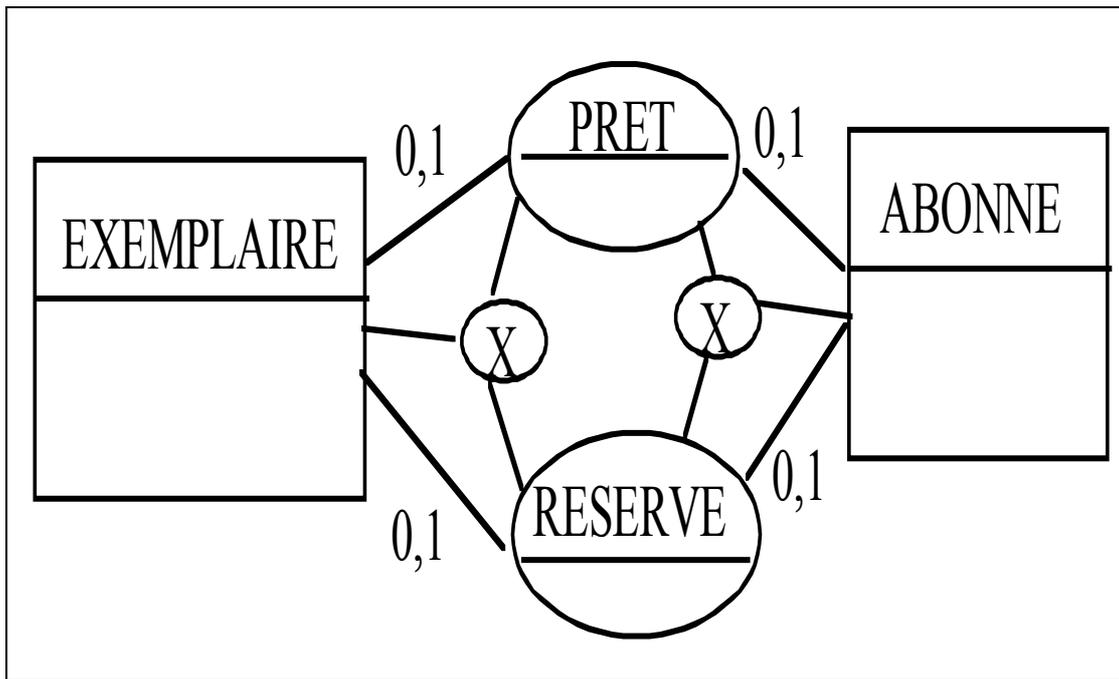
La construction des CVO s'appuient sur le MCTA et sur le MCD. Ce processus est itératif et les CVO permettent à leur tour d'enrichir et de valider le MCTA et le MCD.

Si le système à modéliser comporte de nombreuses règles dynamiques, il est conseillé de commencer par les CVO.

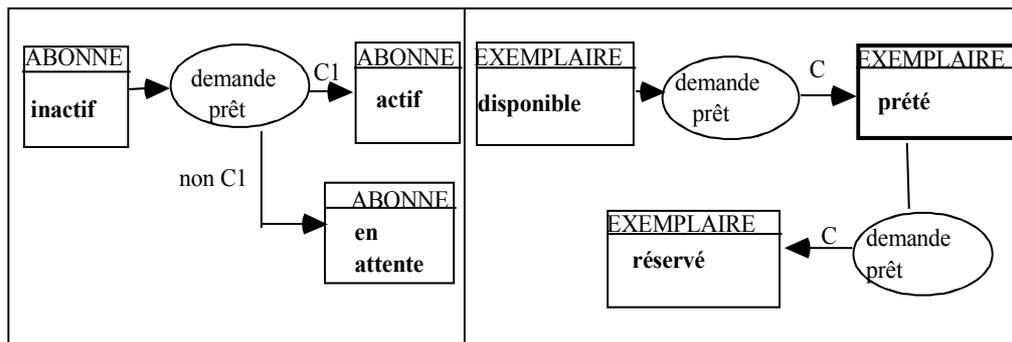
Démarche pratique

1. Choisir un objet au cycle de vie complexe,
2. Commencer par décrire le cycle de vie "standard" de l'objet,
3. Compléter le CVO par les événements et états facultatifs qui ne font pas partie du cycle de vie standard,
4. Définir précisément chaque état en indiquant les contraintes d'intégrité qui le caractérisent,
5. Connecter le graphe aux autres CVO,
6. Construire éventuellement des CVO spécialisés.

Une planche d'illustration : MCD,MCTA,CVO



CVO partiels des objets (ne tient compte que de l'événement demande de prêt)



Abonné inactif = pas de prêt ni de réservation,

Abonné actif = existe prêt,

Abonné en attente = existe réservation,

Exemplaire prêté = existe prêt

Exemplaire réservé = existe réservation

Exemplaire disponible = pas de prêt ni de réservation

11. Bibliographie

Les notes de ce cours sont entièrement issues de :

MERISE/2 : Modèles et Techniques MERISE avancés
Georges Panet, Raymond Letouche
Les éditions d'organisation 1994

De l'autre côté de MERISE, Système d'information et modèles d'entreprise
Yves Tabourier
Les éditions d'organisation 1986

La méthode MERISE- tome 1 : Principes et outils et tome 2 : Démarche et Pratiques
Tardieu, A. Rochfeld, E. Colleti
Les éditions d'organisation 1983

MERISE exercices
Mathelot, H. Annonay, H. Briand, M. Fruchard
Les éditions d'organisation 1990

L'essentiel sur MERISE
Dominique Dionisi
Eyrolles 1994

Parlez-vous MERISE ?
Diviné
Eyrolles

.....