

DI GALLO Frédéric

Méthodologie des systèmes d'information - MERISE

Cours du Cycle Probatoire



A. Lassus – A. Mundubeltz - B. Chaulet
CNAM ANGOULEME 2000-2001

Cnam
CONSERVATOIRE NATIONAL
DES ARTS ET METIERS

METHODOLOGIES DES SYSTEMES D'INFORMATION :

La méthode MERISE

LA METHODE MERISE: INTRODUCTION

I.	LE SYSTEME D'INFORMATION DANS L'ENTREPRISE	5
II.	ARCHITECTURE & CONCEPTION D'UN SYSTEME D'INFORMATION	7
III.	SYSTEME D'INFORMATION ET SYSTEME INFORMATIQUE	8
IV.	CONCLUSION ET OBJECTIF DU COURS.....	9
	4.1) <i>Cycle d'abstraction de conception des S.I.</i>	11
	4.2) <i>Bibliographie :</i>	11

LE MODELE CONCEPTUEL DE LA COMMUNICATION

I.	DEFINITION DE L'ORGANISATION	14
II.	DIAGRAMME DE CONTEXTE	15
III.	DIAGRAMME CONCEPTUEL DE FLUX	15

DESCRIPTION STATIQUE DU SYSTEME D'INFORMATION

I.	LES CONCEPTS DE BASE	18
	1.1) <i>La propriété (ou attribut ou rubrique)</i>	18
	1.2) <i>L'entité ou individu-type</i>	19
	1.3) <i>L'association (ou relation-type)</i>	25
II.	LES CONCEPTS ETENDUS (MERISE 2).....	40
	2.1) <i>Le concept d'héritage</i>	40
	2.2) <i>Les contraintes ensemblistes</i>	43
	2.3) <i>L'identification des occurrences d'entités</i>	53
III.	REGLES DE CONSTRUCTION D'UN M.C.D.....	55
	3.1) <i>modélisation directe</i>	55
	3.2) <i>modélisation par analyse des dép. fonct^{lles}</i>	55

LE MODELE LOGIQUE DES DONNEES

I.	LES REGLES DE PASSAGE DU MCD AU MODELE RELATIONNEL	60
1.1)	<i>Les règles</i>	60
1.2)	<i>Les concepts étendus</i>	63
II.	CREATION D'UNE BD RELATIONNELLES PAR LE LANGAGE SQL	65
2.1)	<i>Déclaration des champs d'une table</i>	67
2.2)	<i>Définition des contraintes d'intégrité</i>	68

MISE EN ŒUVRE AVEC POWERAMC

I.	SAISIE DU MODELE CONCEPTUEL DES DONNEES	77
II.	GENERATION DU MODELE PHYSIQUE	79
III.	GENERATION DE LA BASE DE DONNEES	80

DESCRIPTION DYNAMIQUE DU S.I.

I.	LES CONCEPTS DE BASE	83
1.1)	<i>L'acteur</i>	83
1.2)	<i>L'événement</i>	84
1.3)	<i>L'opération</i>	85
1.4)	<i>La règle d'émission</i>	86
1.5)	<i>La synchronisation</i>	86
1.6)	<i>Représentation graphique</i>	88
1.7)	<i>Extrait du MCT de la gestion des inscriptions</i>	89
II.	FONCTIONNEMENT D'UN MODELE DYNAMIQUE	90
2.1)	<i>Fonctionnement d'un modèle dynamique</i>	90
2.2)	<i>Règles de vérification du fonctionnement</i>	91
III.	REGLES DE CONSTRUCTION D'UN M.C.T.	92
IV.	LE MODELE ORGANISATIONNEL DES TRAITEMENTS	92
4.1)	<i>Le tableau des procédures fonctionnelles</i>	92

METHODOLOGIE – CNAM ANGOULEME 2000-2001

LA METHODE MERISE: INTRODUCTION

La conception d'un système d'information n'est pas évidente car il faut réfléchir à l'ensemble de l'organisation que l'on doit mettre en place. La phase de conception nécessite des méthodes permettant de mettre en place un modèle sur lequel on va s'appuyer. La modélisation consiste à créer une représentation virtuelle d'une réalité de telle façon à faire ressortir les points auxquels on s'intéresse. Ce type de méthode est appelé *analyse*. Il existe plusieurs méthodes d'analyse, la méthode la plus utilisée en France étant la méthode MERISE.

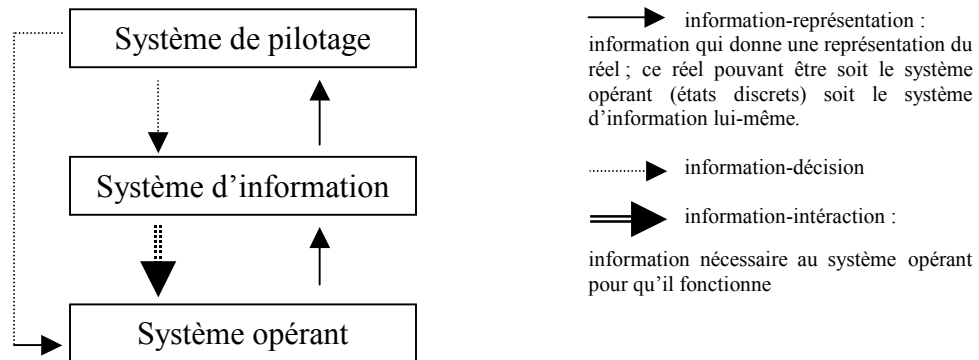
Le but de cette méthode est d'arriver à concevoir un système d'information. La méthode MERISE est basée sur la séparation des données et des traitements à effectuer en plusieurs modèles conceptuels et physiques. La séparation des données et des traitements assure une longévité au modèle. En effet, l'agencement des données n'a pas à être souvent remanié, tandis que les traitements le sont plus fréquemment.

La méthode MERISE date de 1978-1979, et fait suite à une consultation nationale lancée en 1977 par le ministère de l'Industrie dans le but de choisir des sociétés de conseil en informatique afin de définir une méthode de conception de systèmes d'information. Les deux principales sociétés ayant mis au point cette méthode sont le CTI (Centre Technique d'Informatique) chargé de gérer le projet, et le CETE (Centre d'Etudes Techniques de l'Equipement) implanté à Aix-en-provence.

Merise étant une méthode de conception et de développement de système d'information, l'objectif de ce chapitre est d'introduire la notion de système d'information et d'en proposer une description formelle.

I. Le système d'information dans l'entreprise

L'entreprise est un système complexe dans lequel transitent de très nombreux flux d'informations. Sans un dispositif de maîtrise de ces flux, l'entreprise peut très vite être dépassée et ne plus fonctionner avec une qualité de service satisfaisante. L'enjeu de toute entreprise qu'elle soit de négoce, industrielle ou de services consiste donc à mettre en place un système destiné à collecter, mémoriser, traiter et distribuer l'information (avec un temps de réponse suffisamment bref). Ce système d'information assurera le lien entre deux autres systèmes de l'entreprise : le système opérant et le système de pilotage.



- Le système de pilotage décide des actions à conduire sur le système opérant en fonction des objectifs et des politiques de l'entreprise,
- Le système opérant englobe toutes les fonctions liées à l'activité propre de l'entreprise : facturer les clients, régler les salariés, gérer les stocks, ...

Une telle décomposition prend bien en compte :

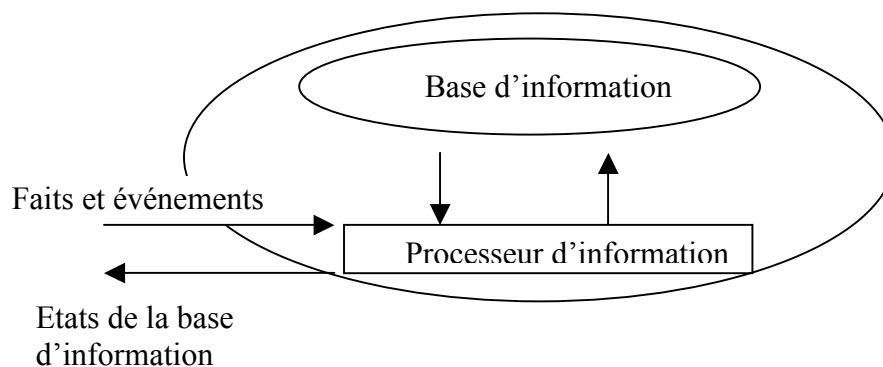
- la différence de besoin en matière d'information des modules opérants et pilotes,
- la nécessité pour le système d'information de ne pas se contenter de transmettre les informations mais d'en changer le niveau de synthèse.

Dans certaines organisations, on peut trouver des formes plus intégrées du système d'information. Cette intégration peut se faire soit au niveau du système opérant, soit au niveau du système de pilotage.

- Un système d'information intégré au système opérant ne décrit plus le fonctionnement du système opérant mais il est intégré à ce fonctionnement. Par exemple dans un système de GPAO (Gestion de Production assistée par Ordinateur), les décisions de pilotage sont directement traduites en des décisions d'exécution de règles incluses dans une gamme opératoire.
- Un système d'information intégré au système de pilotage doit permettre d'engranger les décisions prises lors de diverses situations afin de rendre le pilotage plus intelligent. Ces Systèmes Interactifs d'Aide à la Décision (S.I.A.D) ont une architecture proche de celle des systèmes experts et font donc largement appel pour leur conception aux techniques de l'intelligence artificielle.

II. Architecture & conception d'un système d'information

Le système d'information doit décrire (on dit encore représenter) le plus fidèlement possible le fonctionnement du système opérant. Pour ce faire, il doit intégrer une base d'information dans laquelle seront mémorisés la description des objets, des règles et des contraintes du système opérant. Cette base étant sujette à des évolutions, le système d'information doit être doté d'un mécanisme (appelé processeur d'information) destiné à piloter et à contrôler ces changements. Le schéma suivant synthétise l'architecture d'un système d'information.



Le processeur d'information produit des changements dans la base d'information à la réception d'un message. Un message contient des informations et exprime une commande décrivant l'action à entreprendre dans la base d'information. Le processeur d'information interprète la commande et effectue le changement en respectant les contraintes et les règles.

Si le message exprime une recherche sur le contenu de la base d'information, le processeur interprète la commande et émet un message rendant compte du contenu actuel de la base d'information. Dans tous les cas, l'environnement a besoin de connaître si la commande a été acceptée ou refusée. Le processeur émet, à cet effet, un message vers l'environnement.

Relativement à la conception d'un système d'information, l'architecture présentée ci-dessus induit une double conception :

- celle de la base d'information (aspect statique)
- celle du processeur de traitement (aspect dynamique)

Pour aider le concepteur dans ces deux tâches, la méthode Merise propose un ensemble de formalismes et de règles destinées à modéliser de manière indépendante les données et les traitements du système d'information. Ces modèles ne sont qu'une base de réflexion pour le concepteur et un moyen de communication entre les divers acteurs du système d'information dans l'entreprise. Seul la validation de l'ensemble se fera en commun.

III. Système d'information et système informatique

Parmi les informations qui appartiennent au système d'information, certaines doivent ou peuvent faire l'objet d'un traitement automatisé grâce aux outils informatiques. Pour assurer la cohérence du système d'information, la méthode Merise propose une démarche d'informatisation comportant les étapes suivantes :

- **le schéma directeur** : dont le rôle est de définir, de manière globale, la politique d'organisation et d'automatisation du système d'information. Pour ce faire, il est nécessaire de répertorier l'ensemble des applications informatiques existantes à modifier et à développer. Pour rendre contrôlable et modulable ce développement, il est nécessaire de découper le système d'information en sous-ensembles homogènes et relativement indépendant. Ces sous-ensembles sont appelés domaines. *Par exemple, on peut trouver le domaine « Approvisionnement », le domaine « Personnel ».* Les résultats attendus à la fin de cette étape sont une définition précise des domaines, une planification du développement de chaque domaine et un plan détaillé, année par année, des applications qui doivent être réalisées.

- **l'étude préalable par domaine**: qui doit aboutir à une présentation générale du futur système de gestion (modèles des données et des traitements) en indiquant les principales novations par rapport au système actuel, les moyens matériels à mettre en œuvre, les bilans coût – avantage. Cette étude est réalisée en 4 phases :

- une **phase de recueil** qui a pour objectif d'analyser l'existant afin de cerner les dysfonctionnements et les obsolescences les plus frappantes du système actuel.
- une **phase de conception** qui a pour objectif de formaliser et hiérarchiser les orientations nouvelles en fonction des critiques formulées sur le système actuel et d'autre part des politiques et des objectifs de la direction générale. Cela revient à modéliser le futur système avec une vue pertinente de l'ensemble.
- une **phase d'organisation** dont l'objectif est de définir le système futur au niveau organisationnel: qui fait quoi ?
- une **phase d'appréciation** dont le rôle est d'établir les coûts et les délais des solutions définies ainsi que d'organiser la mise en œuvre de la réalisation. A cet effet un découpage en projets est effectué.

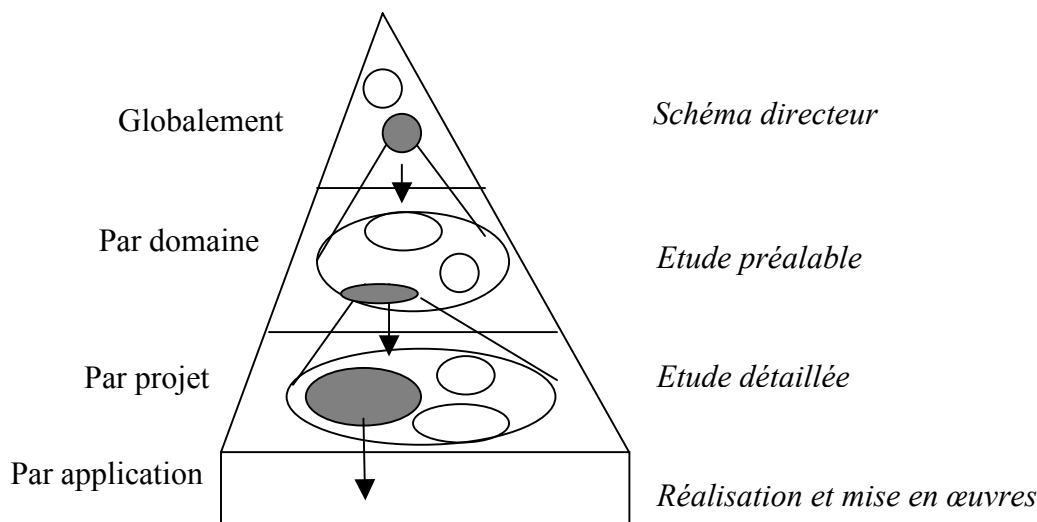
- **l'étude détaillée par projet** qui consiste d'une part à affiner les solutions conçues lors de l'étude préalable et d'autre part à rédiger, pour chaque procédure à mettre en œuvre, un dossier de spécifications détaillé décrivant les supports (maquettes d'états ou d'écran) ainsi que les algorithmes associés aux règles de gestion... A l'issue de cette étude, il est possible de définir le cahier des charges utilisateurs qui constitue la base de l'engagement que prend le concepteur vis à vis des utilisateurs. Le fonctionnement détaillé du futur système, du point de vue de l'utilisateur, y est entièrement spécifié.

- **la réalisation** dont l'objectif est l'obtention des programmes fonctionnant sur un jeu d'essais approuvés par les utilisateurs.

- **la mise en œuvre** qui se traduit par un changement de responsabilité : l'équipe de réalisation va en effet transférer la responsabilité du produit à l'utilisateur. Cette étape intègre en particulier la formation des utilisateurs. Après une période d'exploitation de quelques mois, la recette définitive de l'application est prononcée.

- **la maintenance** qui consiste à faire évoluer les applications en fonction des besoins des utilisateurs, de l'environnement et des progrès technologiques.

Le schéma suivant, extrait de l'ouvrage « *La méthode Merise* » reprend les étapes décrites ci-dessus.



Cette démarche lourde et parfois complexe est adaptée à l'automatisation de « gros systèmes d'information ». Pour des informatisations plus modestes, elle peut être perçue comme un carcan, et il convient donc de l'adapter afin de retenir uniquement les concepts et/ou les étapes appropriées aux besoins.

IV. Conclusion et objectif du cours

Même si la méthode MERISE étant, avant tout, une méthode de conception de systèmes d'information, et non de systèmes informatiques, il apparaît aujourd'hui que les systèmes d'information sont largement gérés par des applications informatiques. Les modèles MERISE doivent donc être utilisés pour faciliter le développement de ces applications en s'appuyant sur les technologies logicielles actuelles telles que les bases de données relationnelles et/ou l'architecture client-serveur.

De plus, il apparaît que les méthodes traditionnelles, composées d'étapes menées séquentiellement depuis l'analyse du besoin jusqu'à la recette, présentent l'inconvénient d'être rigides et peu réactives. Ainsi, le temps écoulé entre les spécifications et la phase de livraison est parfois tellement important que les besoins ont changé de nature. Pour pallier ces défauts, il faut envisager des démarches qui impliquent beaucoup plus l'utilisateur dans le processus global d'informatisation et qui procèdent par affinements successifs. Ainsi, une démarche basée sur des méthodes traditionnelles, comme MERISE pour l'aspect conceptuel, et plus modernes, comme le RAD pour produire des prototypes, pourrait s'avérer être un compromis avantageux pour la conception d'applications informatiques.

Ce cours s'inscrit dans cette logique : il ne détaillera donc pas les étapes de la méthode Merise dans le processus d'informatisation, mais sera axé sur les formalismes et concepts de Merise utiles aux descriptions statique et dynamique du système d'information à automatiser. Relativement à ces descriptions (encore appelées modèles) la méthode Merise préconise 3 niveaux d'abstraction :

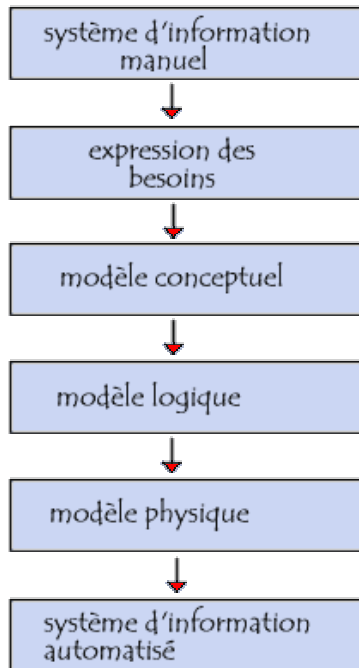
- le **niveau conceptuel** qui décrit la statique et la dynamique du système d'information en se préoccupant uniquement du point de vue du gestionnaire.
- le **niveau organisationnel** décrit la nature des ressources qui sont utilisées pour supporter la description statique et dynamique du système d'information. Ces ressources peuvent être humaines et/ou matérielles et logicielles.
- le **niveau opérationnel** dans lequel on choisit les techniques d'implantation du système d'information (données et traitements)

Du fait de ce découpage (qui a été introduit pour faciliter l'analyse d'un problème) seul le premier niveau est réellement indépendant de toute considération technologique : logicielle ou matérielle. *Par exemple, si les données du futur système d'information doivent être gérées par un SGBD, c'est au niveau organisationnel que le choix du type du SGBD (relationnel, réseau ou objets) devra être effectué.* La description statique du système d'information à ce niveau sera donc basée sur l'organisation des bases relationnelles, ou réseau, ou objets. Le troisième niveau est encore plus dépendant de l'aspect technologique puisqu'il cherchera à optimiser l'implantation. Il suppose donc une connaissance très pointue de l'architecture et des fonctions du SGBD qui gèrera le système d'information.

L'étude des technologies logicielles, telles que les types de SGBD ou encore l'architecture client-serveur, sortant du cadre de ce cours, celui-ci se focalisera sur le niveau conceptuel tant au niveau des données que des traitements. L'apprentissage des formalismes associés à ce niveau suffit à illustrer la richesse, la puissance et parfois même les faiblesses des formalismes Merise en général et, donne ainsi une bonne idée des principaux aspects de la méthode.

Enfin, l'utilisation de l'Atelier de Génie Logiciel AMC*Designer permettra de découvrir comment Merise a été intégré à un outil de conception ainsi que son apport dans le développement d'une application client-serveur.

4.1) Cycle d'abstraction de conception des S.I.



La conception du système d'information se fait par étapes, afin d'aboutir à un système d'information fonctionnel reflétant une réalité physique. Il s'agit donc de valider une à une chacune des étapes en prenant en compte les résultats de la phase précédente. D'autre part, les données étant séparées des traitements, il faut vérifier la concordance entre données et traitement afin de vérifier que toutes les données nécessaires aux traitements sont présentes et qu'il n'y a pas de données superflues.

Cette succession d'étapes est appelée *cycle d'abstraction pour la conception des systèmes d'information*:

L'expression des besoins aboutit au **MCC** (Modèle conceptuel de la communication) qui définit les flux d'informations à prendre compte.

L'étape suivante consiste à mettre au point le **MCD** (Modèle conceptuel des données) et le **MCT** (Modèle conceptuel des traitements) décrivant les règles et les contraintes à prendre en compte.

Le modèle organisationnel consiste à définir le **MLD** (Modèle logique des données) qui représente un choix logiciel pour le système d'information et le **MOT** (Modèle organisationnel des traitements) décrivant les contraintes dues à l'environnement (organisationnel, spatial et temporel).

Enfin, le modèle physique reflète un choix matériel pour le système d'information.

Niveau	Statique (données)	Dynamique (traitements)	
Conceptuel	MCD	MCT	Indépendant du système: <i>QUOI ?</i>
Organisationnel ou logique	MLD (<i>OU ?</i>)	MOT (<i>QUI ? QUAND ?</i>)	Choix du SGBD: <i>QUI ? QUAND ? OU ?</i>
Opérationnel ou physique	MPD	MOPT	Haute connaissance du SGBD: <i>COMMENT ?</i>

4.2) Bibliographie :

La méthode Merise : H. Tardieu, A. Rochfeld, R. Coletti aux Ed. d'organisation
AMC*Designer : Mise en œuvre de merise – Gilles GUEJ aux Editions Eyrolles
www.commentcamarche.net: La méthode Merise.

LA METHODE MERISE :

Le Modèle Conceptuel de la Communication

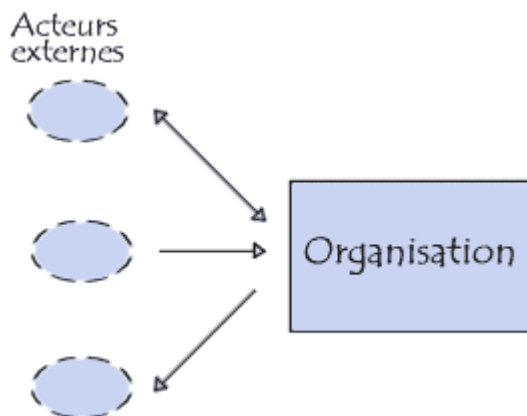
<p style="text-align: center;">LE MODELE CONCEPTUEL DE LA COMMUNICATION</p>
--

I.	DEFINITION DE L'ORGANISATION	14
II.	DIAGRAMME DE CONTEXTE	15
III.	DIAGRAMME CONCEPTUEL DE FLUX	15

METHODOLOGIE – CNAM ANGOULEME 2000-2001

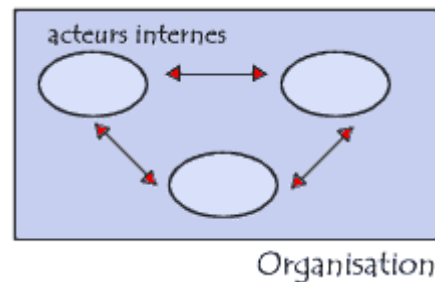
LE MODELE CONCEPTUEL DE LA COMMUNICATION

I. Définition de l'organisation



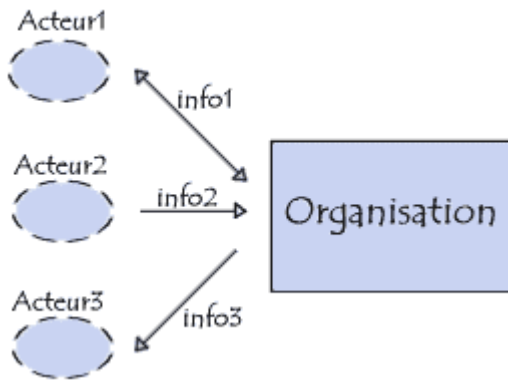
La première étape de ce modèle est d'arriver à isoler le système en le délimitant. Il s'agit donc de définir le système et les éléments externes avec lesquels il échange des flux d'information. Ces éléments extérieurs sont appelés **acteurs externes** (ou partenaires).

La seconde étape consiste à découper l'organisation en entités appelées *acteurs internes* (ou domaines). Lorsque les domaines d'une organisation sont trop importants, ils peuvent être décomposés eux-mêmes en *sous-domaines*.



La dernière étape est l'analyse des flux d'information, c'est-à-dire la définition des *processus*.

II. Diagramme de contexte



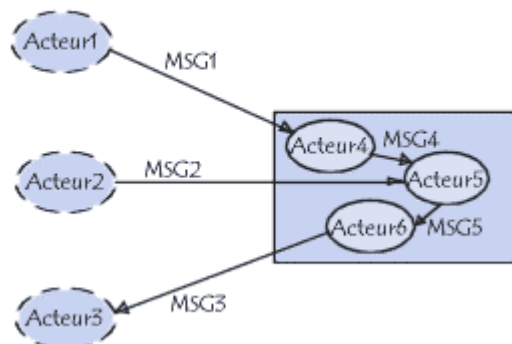
Le diagramme de contexte a pour but de représenter les flux d'informations entre l'organisation et les acteurs externes selon une représentation standard dans laquelle chaque objet port un nom:

- l'organisation est représentée par un rectangle
- les acteurs externes sont représentés par des ellipses en pointillés
- les flux d'information sont représentés par des flèches dont l'orientation désigne le sens du flux d'information

III. Diagramme conceptuel de flux

Ce diagramme (appelé aussi **modèle conceptuel de la communication**) permet de compléter le diagramme de contexte en décomposant l'organisation en une série d'acteurs internes. Dans ce diagramme la représentation standard est la suivante:

- Les acteurs internes sont représentés par des ellipses,
- Les messages internes sont représentés par des flèches



LA METHODE MERISE :

Description statique du système d'information

DESCRIPTION STATIQUE DU SYSTEME D'INFORMATION

I.	LES CONCEPTS DE BASE	18
1.1)	<i>La propriété (ou attribut ou rubrique)</i>	18
1.2)	<i>L'entité ou individu-type</i>	19
a)	<i>Définition</i>	19
b)	<i>Les identifiants</i>	20
c)	<i>Occurrence d'entité ou individu</i>	21
d)	<i>Notion de dépendance fonctionnelle directe</i>	21
Exercice n°1	<i>(Acquis : Propriété et Entité)</i>	22
1.3)	<i>L'association (ou relation-type)</i>	25
a)	<i>Définition</i>	25
b)	<i>Occurrence d'association</i>	26
c)	<i>Cardinalité</i>	26
Exercice n°2	<i>(Acquis : propriété, entité et association)</i>	27
d)	<i>Caractéristiques d'une association</i>	31
e)	<i>L'association porteuse</i>	32
Auto-évaluation n° 1:	33
f)	<i>Les associations transitives</i>	35
g)	<i>Les associations réflexives</i>	35
Exercice n°3	<i>(Acquis : Concepts de base uniquement)</i>	36
Auto-évaluation n° 2:	37
II.	LES CONCEPTS ETENDUS (MERISE 2).....	40
2.1)	<i>Le concept d'héritage</i>	40
Exercice n° 5	<i>(Acquis : concepts de base + sous-typage)</i>	41
2.2)	<i>Les contraintes ensemblistes</i>	43
a)	<i>La contrainte d'inclusion</i>	45
b)	<i>La contrainte de totalité</i>	47
c)	<i>La contrainte d'exclusion</i>	48
d)	<i>La contrainte d'égalité</i>	50
Exercice n° 6	<i>(Acquis : base + sous-typage + contraintes ensemblistes)</i>	51
2.3)	<i>L'identification des occurrences d'entités</i>	53
III.	REGLES DE CONSTRUCTION D'UN M.C.D.....	55
3.1)	<i>modélisation directe</i>	55
3.2)	<i>modélisation par analyse des dép. fonct^{lles}</i>	55

METHODOLOGIE – CNAM ANGOULEME 2000-2001

DESCRIPTION STATIQUE DU SYSTEME D'INFORMATION

Le modèle conceptuel des données est une représentation statique du système d'information de l'entreprise qui met en évidence sa sémantique. Il a pour but d'écrire de façon formelle les données qui seront utilisées par le système d'information. Il s'agit donc d'une représentation des données, facilement compréhensible. Cet aspect recouvre les mots qui décrivent le système ainsi que les liens existants entre ces mots. Le formalisme adopté par la méthode Merise pour réaliser cette description est basé sur les concepts « entité-association ».

I. Les concepts de base

1.1) La propriété (ou attribut ou rubrique)

- La propriété est une information élémentaire, c'est-à-dire non déductible d'autres informations, qui présente un intérêt pour le domaine étudié. *Par exemple, si l'on considère le domaine de gestion des commandes d'une société de vente par correspondance, les données : « référence article », « désignation article », « prix unitaire HT », « taux de TVA » sont des propriétés pertinentes pour ce domaine. La donnée « prix unitaire TTC » n'est, d'après la définition, pas une propriété car ses valeurs peuvent être retrouvées à partir des propriétés « prix unitaire HT » et « taux de TVA ».*
- Chaque valeur prise par une propriété est appelée occurrence. *Des occurrences de la rubrique « désignation article » sont par exemple : « râteau », « bêche », « scie », ...*
- Une propriété est dite simple ou encore atomique si chacune des valeurs qu'elle regroupe n'est pas décomposable. *La propriété « Adresse », dont des exemples d'occurrences sont donnés ci-dessous, n'est pas élémentaire car elle peut être décomposée en trois propriétés : la rue, le code postal et la ville.*

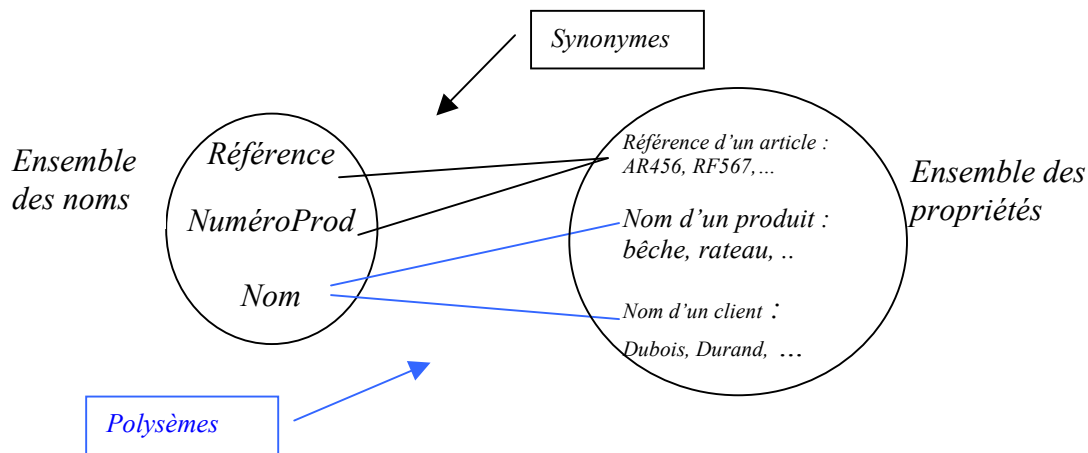
<i>Adresse</i>
<i>310, rue de la gare 16000 Angoulême</i>
<i>45, avenue de la Plage 17000 La Rochelle</i>

La décomposition d'une propriété en propriétés plus simples ne doit pas être systématique et doit surtout tenir compte de son exploitation dans le système. Si cette exploitation est toujours globale, l'atomisation n'est pas nécessaire, dans les autres cas il faut procéder à l'isolement de chacune des composantes de la propriété et donc introduire de nouvelles propriétés.

Propriété	Occurrences
Prénom	J-Philippe, Laurent, Jean

- Une propriété paramètre est une propriété qui, à un instant donné, contient une seule valeur. *Un des exemples les plus classiques pour illustrer les paramètres est la rubrique « ValeurEuro ».*

Dans le modèle conceptuel des données figurent toutes les propriétés, identifiées par un nom, qui présentent un intérêt pour le domaine à étudier. Ce nom doit être le plus explicite possible : à sa seule lecture on doit donc pouvoir se faire une idée de ce que représente la propriété. En outre, l'identification de chaque propriété consiste à garantir une bijection entre l'ensemble des noms et l'ensemble des propriétés à gérer. On devra donc **exclure** les synonymes qui correspondent à deux noms différents pour identifier la même propriété et les polysèmes qui représentent deux propriétés différentes ayant le même nom.



Enfin, le principe de non-redondance impose que chaque propriété, correctement identifiée, n'apparaisse qu'une seule fois dans le modèle.

1.2) L'entité ou individu-type

a) Définition

Une entité est la représentation d'un élément matériel ou immatériel ayant un rôle dans le système que l'on désire décrire. On appelle **classe d'entité** un ensemble composé d'entités de même type, c'est-à-dire dont la définition est la même. Le classement des entités au sein d'une classe s'appelle *classification* (ou *abstraction*). Une entité est une *instanciation* de la classe. Chaque entité est composée de propriétés, données élémentaires permettant de la décrire.

Prenons par exemple une Ford fiesta, une Renault Laguna et une Peugeot 306. Il s'agit de 3 entités faisant partie d'une classe d'entité que l'on pourrait appeler voiture. La Ford Fiesta est donc une instanciation de la classe voiture. Chaque entité peut posséder les propriétés couleur, année et modèle.

Au premier abord, on peut définir l'entité comme étant un regroupement bien pensé, donc sensé, de plusieurs propriétés. *Par exemple, on considère l'entité ARTICLE qui regroupe les propriétés : Référence, Désignation et PrixUnitaireHT.* Le droit d'entrée d'une propriété dans une entité est soumis à d'autres facteurs que le bon sens, et ce sont ces facteurs que l'on va étudier.

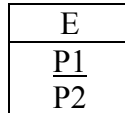


Les classes d'entités sont représentées par un rectangle. Ce rectangle est séparé en deux champs:

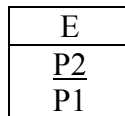
- le champ du haut contient le libellé. Ce libellé est généralement une abbréviation pour une raison de simplification de l'écriture. Il s'agit par contre de vérifier qu'à chaque classe d'entité correspond un et un seul libellé, et réciproquement.
- le champ du bas contient la liste des propriétés de la classe d'entité.

Considérons deux propriétés P1 et P2. La création d'une entité E regroupant ces deux seules propriétés n'est envisageable que si l'une des deux conditions suivantes est satisfaite :

- à toute valeur de la propriété P1 doit correspondre au plus une valeur de la propriété P2. Ce fait traduit l'existence d'une dépendance fonctionnelle monovaluée entre P1 et P2 notée : $P1 \rightarrow P2$. On dit encore que P1 détermine P2. P1 est alors rubrique identifiante de l'entité E. La représentation graphique de l'entité E a la forme suivante :



- ou à toute valeur de la rubrique P2 doit correspondre au plus une valeur de la rubrique P1. P2 est alors en dépendance fonctionnelle avec P1 et l'entité E doit être représentée ainsi :



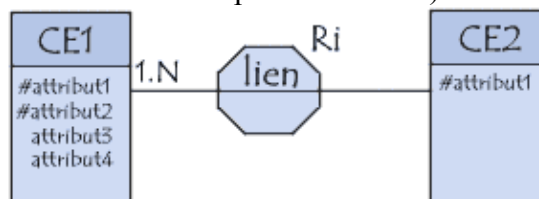
L'entité suivante, qui peut être considérée comme un regroupement sensé, n'est pas correcte car il n'y a pas dépendance fonctionnelle entre la rubrique « CodePostal » et la rubrique « Commune ». Ainsi au code postal « 16600 » correspond plusieurs communes : « Mornac », « Magnac », ..



b) Les identifiants

Un identifiant est un ensemble de propriétés (une ou plusieurs) permettant de désigner une et une seule entité. La définition originale est la suivante: **L'identifiant est une propriété particulière d'un objet telle qu'il n'existe pas deux occurrences de cet objet pour lesquelles cette propriété pourrait prendre une même valeur.**

Les attributs d'une classe d'entité permettant de désigner de façon unique chaque instance de cette entité sont appelé identifiant absolu. Le modèle conceptuel des données propose de souligner les identifiants (parfois de les faire précéder d'un #).



Ainsi, chaque classe d'entité doit posséder au moins un attribut identifiant, et l'ensemble de ses attributs identifiants doivent être renseignés à la création de l'entité.

c) Occurrence d'entité ou individu

D'après la définition d'une entité, on sait que la connaissance d'une valeur de la rubrique identifiante détermine la connaissance des valeurs des autres rubriques de l'entité. L'ensemble de ces valeurs est appelé occurrence d'entité. Le tableau suivant présente des exemples d'occurrences de l'entité *ARTICLE*.

<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="text-align: center;"><i>ARTICLE</i></td></tr> <tr><td><i>Référence</i></td></tr> <tr><td><i>Désignation</i></td></tr> <tr><td><i>PrixUnitaireHT</i></td></tr> </table>	<i>ARTICLE</i>	<i>Référence</i>	<i>Désignation</i>	<i>PrixUnitaireHT</i>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="text-align: center;"><i>134ER</i></td></tr> <tr><td><i>Rateau</i></td></tr> <tr><td><i>150 F</i></td></tr> </table>	<i>134ER</i>	<i>Rateau</i>	<i>150 F</i>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="text-align: center;"><i>354TY</i></td></tr> <tr><td><i>Bêche</i></td></tr> <tr><td><i>68,50 F</i></td></tr> </table>	<i>354TY</i>	<i>Bêche</i>	<i>68,50 F</i>
<i>ARTICLE</i>												
<i>Référence</i>												
<i>Désignation</i>												
<i>PrixUnitaireHT</i>												
<i>134ER</i>												
<i>Rateau</i>												
<i>150 F</i>												
<i>354TY</i>												
<i>Bêche</i>												
<i>68,50 F</i>												
	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="text-align: center;"><i>452GT</i></td></tr> <tr><td><i>Scie</i></td></tr> <tr><td><i>45 F</i></td></tr> </table>	<i>452GT</i>	<i>Scie</i>	<i>45 F</i>								
<i>452GT</i>												
<i>Scie</i>												
<i>45 F</i>												

d) Notion de dépendance fonctionnelle directe

Considérons l'entité suivante et quelques une de ses occurrences :

<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="text-align: center;"><i>ARTICLE</i></td></tr> <tr><td><i>Référence</i></td></tr> <tr><td><i>Désignation</i></td></tr> <tr><td><i>PrixUnitaireHT</i></td></tr> <tr><td><i>NoCatégorie</i></td></tr> <tr><td><i>LibelléCatégorie</i></td></tr> </table>	<i>ARTICLE</i>	<i>Référence</i>	<i>Désignation</i>	<i>PrixUnitaireHT</i>	<i>NoCatégorie</i>	<i>LibelléCatégorie</i>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="text-align: center;"><i>134ER</i></td></tr> <tr><td><i>Rateau</i></td></tr> <tr><td><i>150 F</i></td></tr> <tr><td><i>A</i></td></tr> <tr><td><i>Jardinage</i></td></tr> </table>	<i>134ER</i>	<i>Rateau</i>	<i>150 F</i>	<i>A</i>	<i>Jardinage</i>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="text-align: center;"><i>354TY</i></td></tr> <tr><td><i>Bêche</i></td></tr> <tr><td><i>68,50 F</i></td></tr> <tr><td><i>A</i></td></tr> <tr><td><i>Jardinage</i></td></tr> </table>	<i>354TY</i>	<i>Bêche</i>	<i>68,50 F</i>	<i>A</i>	<i>Jardinage</i>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="text-align: center;"><i>452GT</i></td></tr> <tr><td><i>Scie</i></td></tr> <tr><td><i>45F</i></td></tr> <tr><td><i>B</i></td></tr> <tr><td><i>Bricolage</i></td></tr> </table>	<i>452GT</i>	<i>Scie</i>	<i>45F</i>	<i>B</i>	<i>Bricolage</i>
<i>ARTICLE</i>																								
<i>Référence</i>																								
<i>Désignation</i>																								
<i>PrixUnitaireHT</i>																								
<i>NoCatégorie</i>																								
<i>LibelléCatégorie</i>																								
<i>134ER</i>																								
<i>Rateau</i>																								
<i>150 F</i>																								
<i>A</i>																								
<i>Jardinage</i>																								
<i>354TY</i>																								
<i>Bêche</i>																								
<i>68,50 F</i>																								
<i>A</i>																								
<i>Jardinage</i>																								
<i>452GT</i>																								
<i>Scie</i>																								
<i>45F</i>																								
<i>B</i>																								
<i>Bricolage</i>																								

Cette entité est juste mais elle implique une redondance d'information relative à la catégorie. L'association entre le numéro de la catégorie et son libellé est en effet répétée dans chaque occurrence de l'entité *ARTICLE*.

Pour supprimer de telles redondances, on devra veiller à ce que toute dépendance fonctionnelle entre la propriété identifiante de l'entité et une propriété non identifiante de l'entité soit directe. **Une dépendance fonctionnelle monovaluée $x \rightarrow y$ est directe s'il n'existe pas de propriété z telle que : $x \rightarrow z$ et $z \rightarrow y$.**

Dans l'exemple précédent la dépendance fonctionnelle *Référence* \rightarrow *LibelléCatégorie* n'est pas directe car il existe la propriété *NoCatégorie* telle que :

Référence \rightarrow *NoCatégorie* et *NoCatégorie* \rightarrow *LibelléCatégorie*

Exercice n°1 (Acquis : Propriété et Entité)

La société Azur-Hebdo consacre l'essentiel de son activité à l'édition et à la distribution d'un journal spécialisé dans les petites annonces et la publicité dans le Sud-est de la France. La parution du journal est hebdomadaire et sa distribution est assurée uniquement dans les départements des Alpes maritimes et du Var.

La tarification d'une annonce est fournie ci-dessous :

- Première semaine de parution : tarif pour 5 lignes au plus :

Rubrique de l'annonce	Prix
Emploi	50 F
Bourse aux affaires	40 F
Tout ce qui roule	55 F
Immobilier	55 F
Contacts	75 F
Loisirs	50 F

- Options

Prix de la ligne supplémentaire : 50 F

Domiciliation (pour préserver l'anonymat du client) : 80 F

Semaines supplémentaires : des réductions sont accordées selon le tableau ci-dessous :

Période	Pourcentage de réduction *
2 ^{ème} semaine	20%
3 ^{ème} semaine et suivantes	40 %

*Ces réductions sont applicables au prix de base de la première semaine.

Le tableau ci-dessous répertorie un ensemble de données qui se rapporte à la gestion des annonces (La liste est triée sur le nom de la donnée).

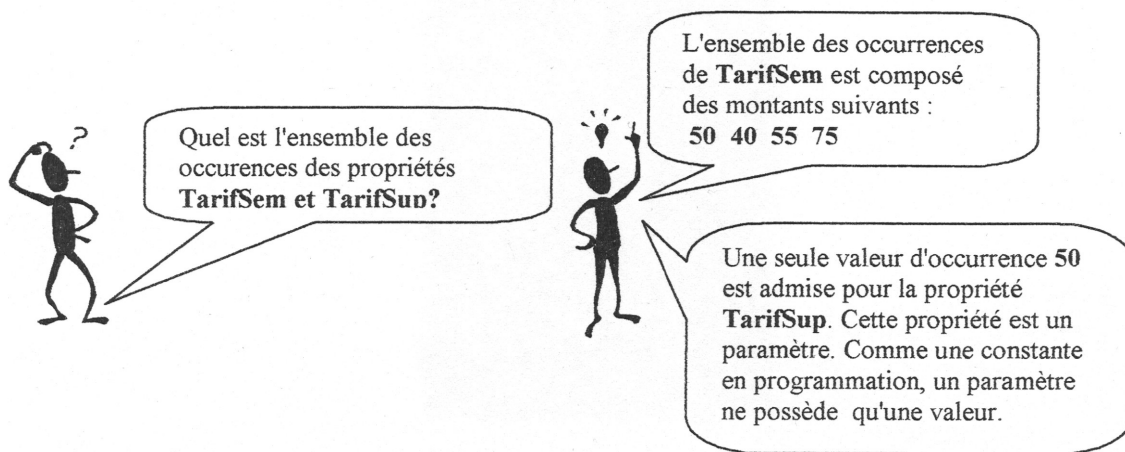
Nom	Signification
CP	Code postal du client qui dépose l'annonce
DateRedac	Date de dépôt de l'annonce
Dom	Domiciliation de l'annonce déposée(oui/non)
NbLignes	Nombre de lignes de l'annonce déposée
NbSem	Nombre de semaines de parution de l'annonce déposée
Nom	Nom du client qui dépose l'annonce
NumAnn	Numéro qui identifie chaque annonce déposée
NumCli	Numéro qui identifie chaque client qui dépose une annonce
Prénom	Prénom du client qui dépose l'annonce
Prix	Prix de l'annonce déposée
Rub	Rubrique de l'annonce déposée : Emploi, Contacts
Rue	Première partie de l'adresse du client qui dépose une annonce
TarifDom	Tarif de la domiciliation
TarifSup	Tarif de la ligne supplémentaire
TarifPrem	Tarif de la première semaine
Texte	Texte de l'annonce
Ville	Ville de l'adresse du client qui dépose l'annonce

1. Indiquer, parmi les données ci-dessus, celles qui peuvent être qualifiées de propriétés.

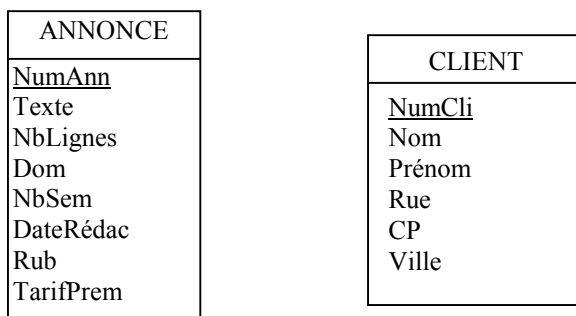
Parmi la liste des données, on recherche les informations non déductibles d'autres informations et qui ont un intérêt pour le domaine étudié. Or, parmi cette liste, **deux** données seulement **ne peuvent pas être considérées comme des propriétés**:

- NbLignes: donnée qui peut être déduite de la propriété Texte et éventuellement d'une propriété paramètre fournissant le nombre de caractères par ligne,
- **Prix** : donnée calculée à partir des propriétés TarifDom, TarifSup, TarifPrem.

2. En vous basant sur les éléments de tarification, fournir l'ensemble des occurrences des propriétés : TarifPrem et TarifSup. Parmi ces deux propriétés laquelle est une propriété paramètre ?



3. On considère l'ébauche du modèle conceptuel de données suivante :



3.1 On considère l'événement suivant :

Figeac Claire qui habite : 72, Avenue de la gare 05000 NICE dépose le 14/02/2001 l'annonce ci-après :

Annonce n° 465

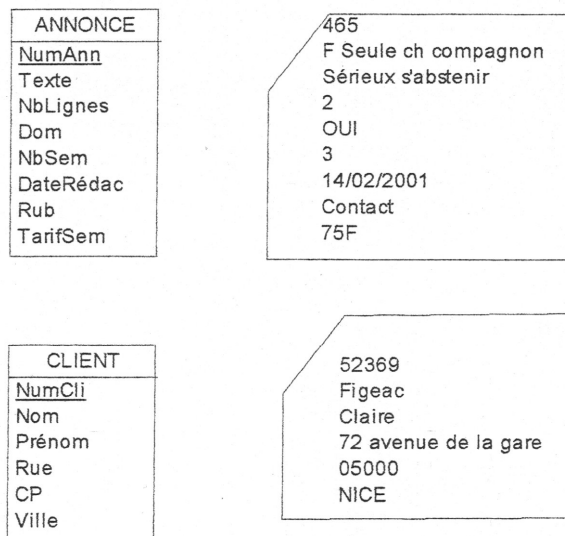
F seule ch compagnon

Sérieux s'abstenir

Ecrire sous référence 52369 au journal

(Nombre de semaines de parution : 3)

En limitant le système d'information à cette annonce, fournir les occurrences de l'entité CLIENT et ANNONCE.



3.2 Les dépendances fonctionnelles issues de la conception de l'entité ANNONCE sont-elles toutes directes ? justifier votre réponse.

La dépendance fonctionnelle NumAnn ---> TarifPrem n'est pas une dépendance fonctionnelle directe. Elle peut, en effet, être retrouvée par transitivité grâce aux deux dépendances fonctionnelles élémentaires suivantes: NumAnn --->Rub

Rub--->TarifSem

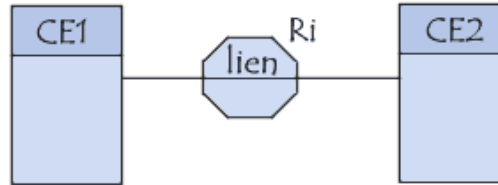
Pour corriger ce MCD, il faut supprimer la dépendance fonctionnelle directe tout en maintenant les deux dépendances fonctionnelles NumAnn ---> Rub et Rub ---> TarifPrem.

L'idée est de créer une nouvelle entité RUBRIQUE dont Rub est la propriété identifiante. Pour garder la dépendance fonctionnelle NumAnn ---> Rub il faut relier les entités Rubrique et Annonce par des associations.

1.3) L'association (ou relation-type)

a) Définition

Une association (appelée aussi parfois *relation*) est un lien sémantique entre plusieurs entités. Une classe de relation contient donc toutes les relations de même type (qui relient donc des entités appartenant à des mêmes classes d'entité).

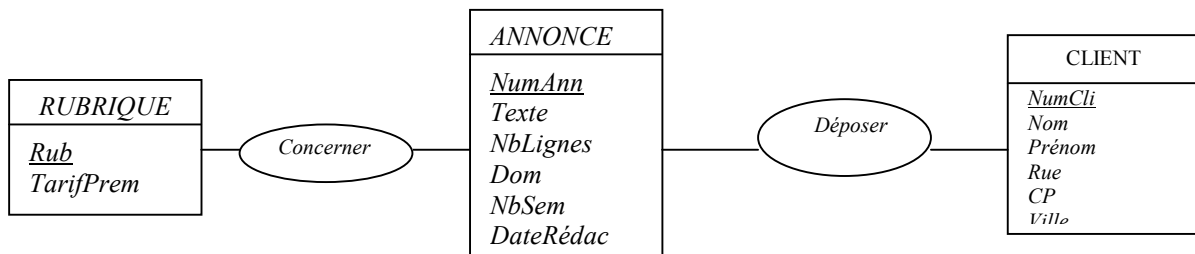


Une classe de relation peut lier plus de deux classes d'entité. Voici les dénominations des classes de relation selon le nombre d'intervenants:

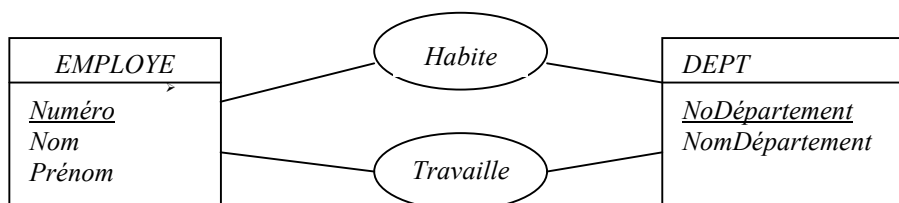
- une classe de relation **réursive** (ou *réflexive*) relie la même classe d'entité
- une classe de relation **binaire** relie deux classes d'entité
- une classe de relation **ternaire** relie trois classes d'entité

Une classe de relation **n-aire** relie n classes d'entité. Les classes de relations sont représentées par des hexagones (parfois des ellipses) dont l'intitulé décrit le type de relation qui relie les classes d'entité (généralement un verbe). On définit pour chaque classe de relation un identificateur de la forme R_i permettant de désigner de façon unique la classe de relation à laquelle il est associé. On peut éventuellement ajouter des propriétés aux classes de relation.

Par exemple, dans le modèle conceptuel de données relatif à la gestion des annonces on introduit les deux associations « Déposer » et « Concerner » afin d'exprimer les réalités suivantes : un client dépose une annonce et une annonce concerne une rubrique.



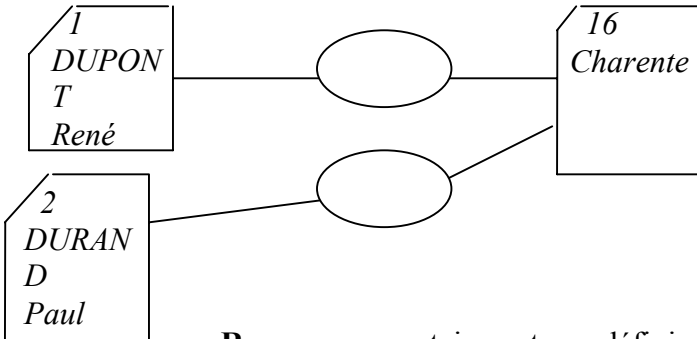
Il peut y avoir, entre deux mêmes entités, plusieurs associations qui représentent chacune des réalités différentes. Dans le modèle conceptuel des données ci-dessous, l'association *Habite* indique le département dans lequel l'employé habite et l'association *Travaille* fournit le département dans lequel il travaille. Pour certains employés ces deux départements peuvent être identiques.



L'ensemble des entités participant à une association est appelé collection de cette association. La collection de l'association *HABITE* est formée des entités : *EMPLOYE* et *DEPT*.

b) Occurrence d'association

Une occurrence d'association est un lien particulier qui relie deux occurrences d'entités. *Le schéma ci-dessous présente deux exemples d'occurrences de l'association « Habite ».*



Remarque : certains auteurs définissent l'identifiant d'une association comme étant la concaténation des identifiants des entités qui participent à l'association.

c) Cardinalité

Les cardinalités permettent de caractériser le lien qui existe entre une entité et la relation à laquelle elle est reliée. La cardinalité d'une relation est composé d'un couple comportant une borne maximale et une borne minimale, intervalle dans lequel la cardinalité d'une entité peut prendre sa valeur:

- la borne minimale (généralement 0 ou 1) décrit le nombre minimum de fois qu'une entité peut participer à une relation
- la borne maximale (généralement 1 ou n) décrit le nombre maximum de fois qu'une entité peut participer à une relation

Un couple de cardinalités placé entre une entité E et une association A représente le nombre minimal et maximal d'occurrences de l'association A qui peuvent être « ancrées » à une occurrence de l'association E. Le tableau ci-après récapitule les valeurs que peut prendre ce couple.

	Pour chaque occurrence de E, le modèle admet : <ul style="list-style-type: none"> - soit l'absence de lien - soit la présence d'un seul lien
	Pour chaque occurrence de E le modèle admet la présence d'un et un seul lien
	Pour chaque occurrence de E le modèle admet la présence d'un seul ou de plusieurs liens
	Pour chaque occurrence de E le modèle admet : <ul style="list-style-type: none"> - soit l'absence de lien - soit la présence de plusieurs liens

Remarque : dans certaines situations, la lettre n peut être remplacée par une valeur.

Exercice n°2 (Acquis : propriété, entité et association)

Le système d'information étudié concerne l'activité de gestion des locations saisonnières d'une agence immobilière. Une analyse de l'existant a permis de dégager les entités suivantes :

Entité	Objectif	Propriétés
PROPRIETAIRE	Regroupe toutes les informations relatives aux propriétaires d'appartements	NumPropriétaire Nom Prénom Adresse1 Adresse2 CodePostal Ville NumTel1 NumTel2 E-mail Cacumulé
APPARTEMENT	Regroupe toutes les informations des appartements meublés mis à la location	NumLocation Catégorie : 1, 2, ou 3 étoiles Type : T2, T3, T4 NbPersonnes AdresseLocation Photo Equipements
LOCATAIRE	Regroupe toutes les informations sur les locataires qui ont effectué au moins une location par l'intermédiaire de l'agence	NumLocataire NomLocataire PrénomLocataire Adresse1Locataire Adresse2Locataire CodePostalLocataire VilleLocataire NumTel1Locataire NumTel2Locataire E-mailLocataire
CONTRAT	Regroupe toutes les informations relatives à une location qui va avoir lieu ou qui a actuellement lieu. Une location s'étend éventuellement sur plusieurs semaines consécutives.	NumContrat Etat : réservé, confirmé, soldé DateCréation DateDébut DateFin
TARIF	Regroupe les informations liées à la tarification	CodeTarif PrixSemHS (prix semaine haute saison) PrixSemBS (prix semaine basse saison)

1. Pourquoi l'information CAcumulé de l'entité PROPRIETAIRE est-elle une propriété ?

Bien que pouvant être une donnée calculée par le système d'information, il est préférable d'avoir le chiffre d'affaire cumulé en propriété de manière à pouvoir accéder plus rapidement à l'information (allègement des traitements). C'est une "dénormalisation".

2. La propriété Equipements est destinée à décrire les principaux équipements de l'appartement : téléviseur, lave-vaisselle, ... Quels sont les inconvénients liés à une telle propriété ?

Le fait de regrouper toutes les caractéristiques des équipements dans une même propriété ne permettra pas de faire des traitements précis sur ces derniers. *Par exemple, il sera impossible de lister tous les appartements de type T2 qui possède un lave-vaisselle.*

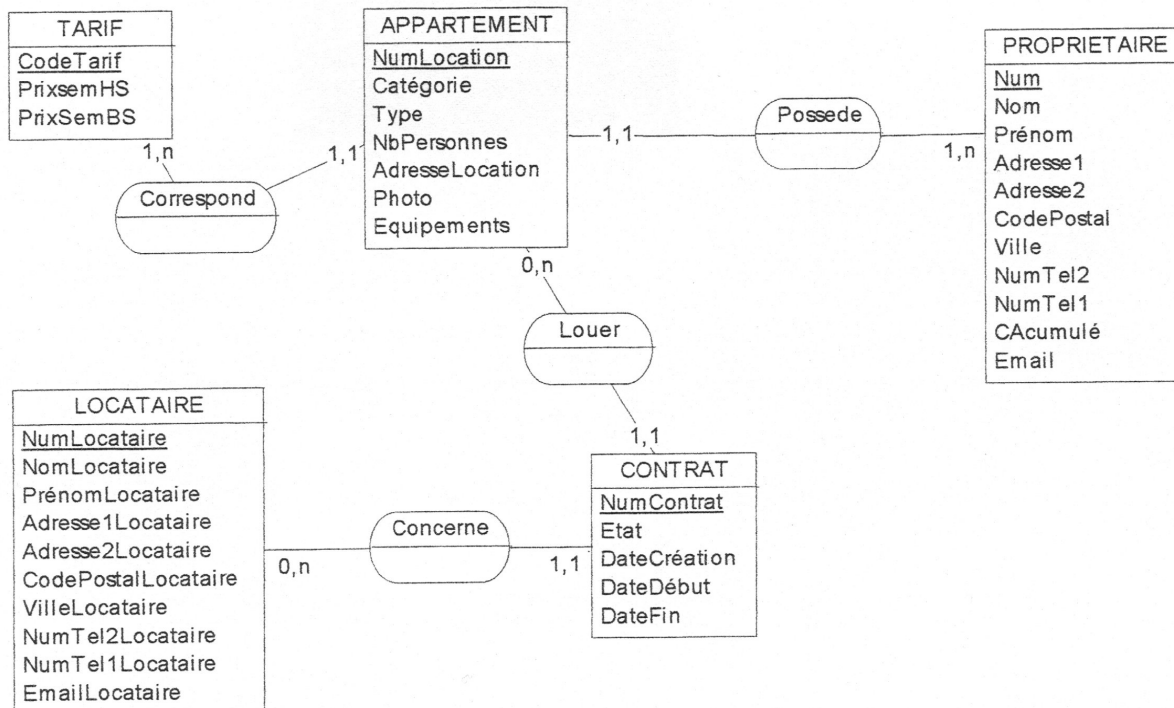
L'isolement des différents composants peut être réalisé de deux manières;

- Introduire dans l'entité APPARTEMENT des propriétés booléennes telles que Téléviseur, Lave-vaisselle etc.
- Créer une entité TYPE-EQUIPEMENT et mettre en place une association (m-n) entre l'entité APPARTEMENT et l'entité TYPE-EQUIPEMENT.

3. Présenter le modèle conceptuel des données décrivant ce système d'information en tenant compte des règles de gestion suivantes :

- La notion de co-propriété ne doit pas être prise en compte ce qui revient à dire que tout appartement appartient à un et un seul propriétaire.
- A tout appartement correspond un code tarif

Seules les noms des entités figureront sur le modèle.



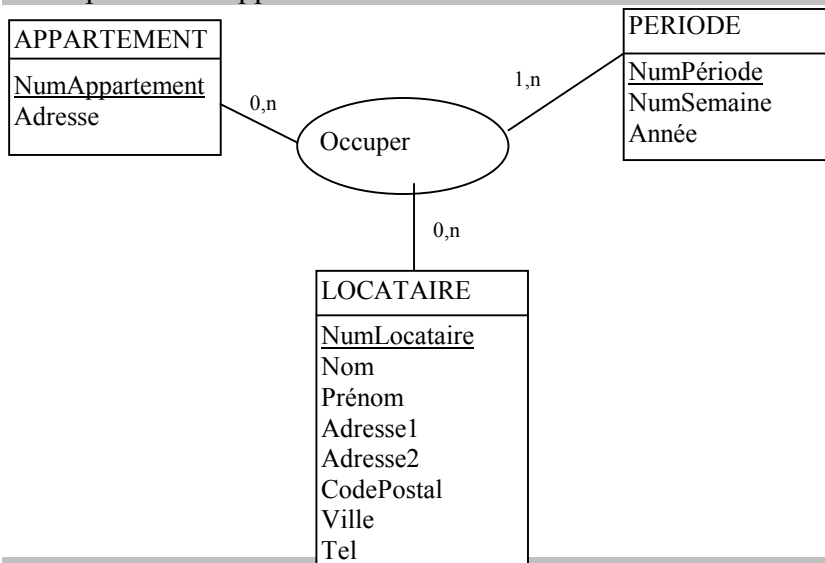
4. On restreint le domaine étudié à la gestion des locations des appartements possédés par M. X. Les entités recensées sont données ci-dessous :

Entité	Objectif	Propriétés
APPARTEMENT	Regroupe toutes les informations relatives aux appartements de M. X	<u>NumAppartement</u> Adresse
PERIODE	Cette entité admet une occurrence par semaine réservée ou occupée	<u>NumPériode</u> NumSemaine Année
LOCATAIRE	Regroupe toutes les informations sur le locataire	NumLocataire Nom Prénom Adresse1 Adresse2 CodePostal Ville Tel

Pour une semaine donnée, un appartement de M. X peut être :

- soit réservé ou occupé par un locataire
- soit libre
- soit indisponible (ce cas correspond à l'occupation de l'appartement par M.X)

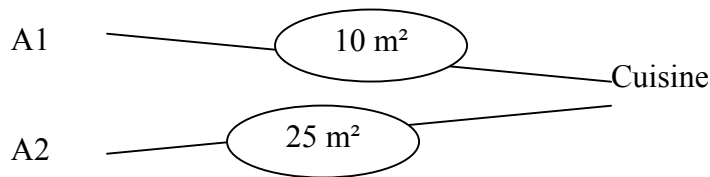
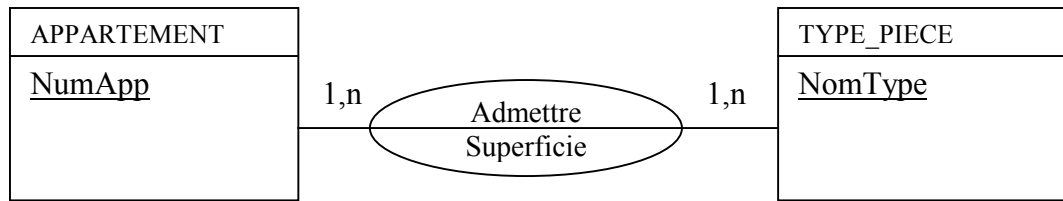
Discuter la proposition de modélisation suivante qui est destinée à représenter l'occupation des appartements de M. X :



Cette représentation n'est pas satisfaisante car elle admet que le locataire 2205 occupe l'appartement n°1 pour la période 3 et que le locataire 1903 occupe également l'appartement n°1 pour la période 3. Le système d'information accepte qu'un appartement soit loué pour la même période à deux locataires différents.

Il faut pouvoir représenter le fait que pour un couple d'occurrence APPARTEMENT-PERIODE, on n'admette qu'une et une seule occurrence de LOCATAIRE.

5. On souhaite décrire pour chaque appartement les différentes pièces qui le composent ainsi que leur superficie. *Par exemple : l'appartement n° 345 possède une kitchenette de 4 m2, une salle de bains de 4 m2, un séjour de 20 m2 et une terrasse de 5m2.*
 Enrichir le modèle conceptuel afin de représenter une telle réalité



d) Caractéristiques d'une association

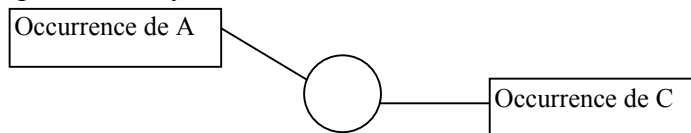
➤ **La dimension d'une association**

La dimension d'une association indique le nombre d'entités participant à l'association. Les dimensions les plus courantes sont 2 (association binaire) et 3 (association ternaire) :

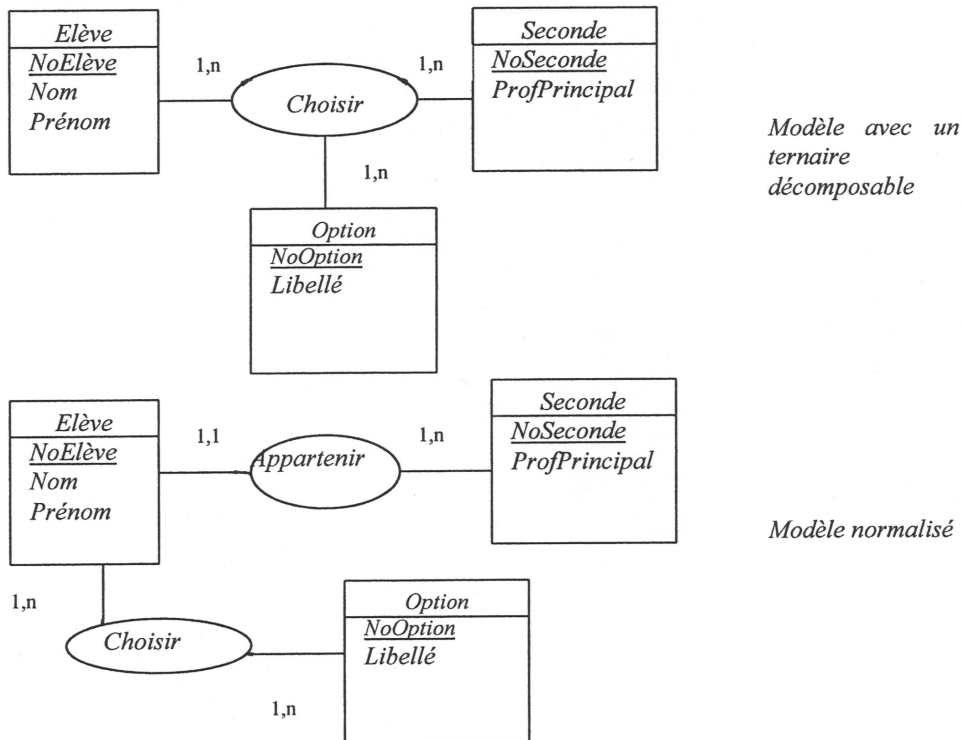
- L'association binaire exprime la présence de liens sémantiques entre les occurrences d'une entité A et les occurrences d'une entité B. *L'association « Habite » entre les EMPLOYE et HABITE est binaire.*
- L'association ternaire exprime la présence de liens sémantiques entre les occurrences de 3 entités.

Remarques :

- Toute occurrence d'une association de dimension n doit être reliée à n occurrences d'entités. *Par exemple, pour une association ternaire dans laquelle participent trois entités « A », « B » et « C », toute occurrence doit être reliée à 3 occurrences des entités respectives A, B et C. On ne peut donc pas avoir une occurrence à 2 pattes de la forme ci-dessous.*



- L'opération de décomposition consiste à éclater une relation de dimension n en une ou plusieurs associations de dimension moindre sans perte de sémantique. *Dans l'exemple ci-dessous, qui se rapporte à la gestion des options en classe de seconde, l'association ternaire du premier modèle doit être éclatée en deux associations binaires afin d'aboutir au second modèle.*



Les associations figurant dans le modèle conceptuel des données devront être non décomposables. Pour ce faire on recherchera en priorité les associations binaires puis les ternaires.

➤ **La fonctionnalité d'une association binaire**

Les différents types de fonctionnalités d'une association binaire A définie entre deux entités E1 et E2 sont les suivants :

- un à un (**1 – 1**) caractérisé par la cardinalité **maximum** égale à 1 sur les deux segments « E1 -- A » et « E2 – A »,
- un à plusieurs (**1 – n**) caractérisé par la cardinalité **maximum** égale à 1 sur l'un des segments et la cardinalité maximum égale à n sur l'autre,
- plusieurs à plusieurs (**m-n**) caractérisé par la cardinalité **maximum** égale à n sur les deux segments « E1 – A » et « E2 – A ».

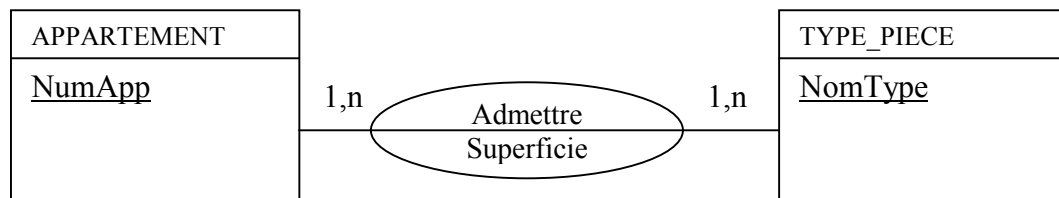
➤ **Les notions de partialité et de totalité**

Une association binaire A définie entre les entités E1 et E2 est totale si elle est caractérisée par la cardinalité minimale égale à un sur les deux segments « E1 – A » et « E2 – A ».

Une association binaire A définie entre les entités E1 et E2 est partielle si elle est caractérisée par la cardinalité minimum égale à 0 sur l'un des segments.

e) L'association porteuse

Les propriétés qui dépendent fonctionnellement de plusieurs identifiants d'entités sont portées par les associations entre ces entités. C'est une dépendance fonctionnelle multi-attributs au niveau de la source. *Dans le système d'information de l'exercice n° 2 la propriété Superficie est portée par l'association qui relie l'entité APPARTEMENT à l'entité TYPE_PIECE car pour un appartement et une pièce on a une et une seule superficie. Par exemple pour l'appartement 134 et la pièce de type kitchenette on a la superficie : 3 m².*



Le droit d'entrée d'une propriété P dans une association reliant n entités est donc soumis à l'existence de la dépendance fonctionnelle suivante : $I_1, \dots, I_n \rightarrow R$
 où I_1, I_2, \dots représentent l'identifiant de chacune des entités qui participent à l'association.

Pour éviter toute redondance, on s'assurera en outre que la dépendance fonctionnelle est élémentaire. Les associations porteuses sont donc toujours de type (**m-n**). On dit qu'une propriété est en dépendance fonctionnelle élémentaire avec une liste de rubriques LR :

- si elle est fonctionnellement dépendante de LR,
- si elle n'est pas fonctionnellement dépendante d'une sous-liste de LR.

*La dépendance fonctionnelle suivante : NoAppart, NomType, NumPropriétaire → Superficie n'est pas élémentaire car il existe la sous-liste NumAppart, NomType telle que :
 NumAppart, NomType → Superficie.*

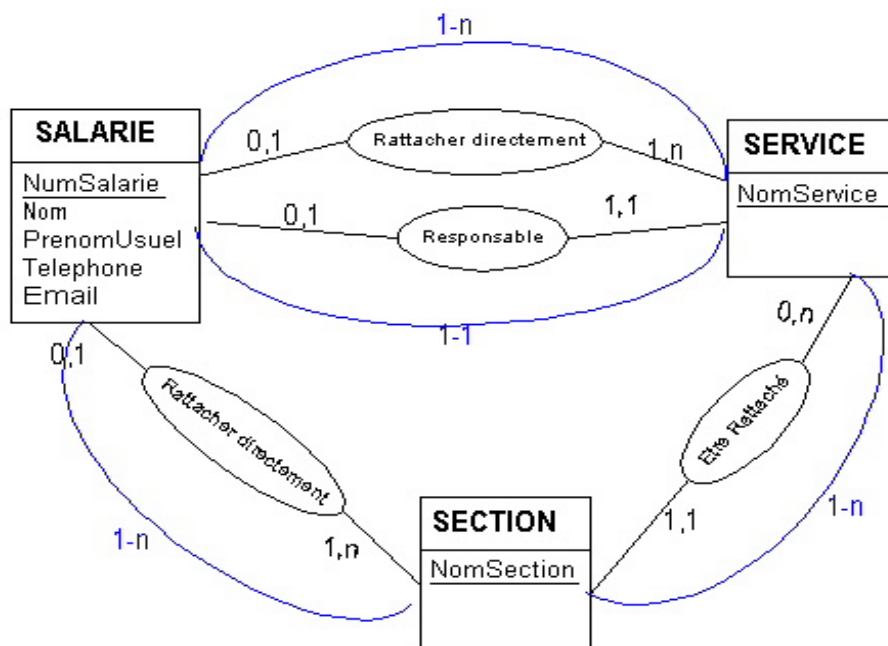
Auto-évaluation n° 1:

Pour faciliter les contacts entre ses différents employés, la société X a confié à son service informatique la réalisation d'un annuaire électronique. L'une des spécifications de cette application précise qu'elle devra être accessible à l'ensemble du personnel par l'intermédiaire de l'INTRANET de l'entreprise.

Un extrait de l'entretien avec le demandeur de l'application M. D est reproduit ci-dessous :

L'analyste : Comment sont identifiés les salariés de la société ?
M. D : Chaque employé de la société est identifié par un numéro et appartient à un service caractérisé par un nom. Comme exemples je peux vous citer le service comptabilité ou le service production.
L'analyste : Peut-il y avoir deux services qui portent le même nom ?
M. D : Non.
L'analyste : Quelles sont les données relatives à un salarié que vous souhaitez déposer sur l'annuaire
M.D : son nom, son prénom, ses coordonnées téléphonique et INTERNET ainsi que le service et la section auquel il est rattaché. Je précise que certains services sont découpés en sections, elles aussi identifiés par un nom. Par exemple, le service informatique comprend les sections études et production. Le service comptabilité n'admet pas de section.
L'analyste : Tout salarié a-t-il un poste téléphonique ?
M.D : non, certains employés n'ont pas encore de poste téléphonique. Pour les autres ils en ont un et un seul. C'est un numéro interne composé de 4 chiffres. Le mien est par exemple 48 14.
L'analyste : Je suppose que tous les salariés n'ont pas une adresse électronique.
M.D : Sur notre INTRANET tous les employés ont une adresse professionnelle qui leur permet d'envoyer ou de recevoir du courrier électronique relatif à leur activité.
L'analyste : Revenons sur les sections. Une section peut-elle concerner plusieurs services ?
M. D : Non, une section concerne un et un seul service.
L'analyste : En terme de traitements quelles sont vos attentes ?
M.D : Ma première attente serait bien évidemment de retrouver très rapidement les coordonnées d'un employé. Dans le cas où l'employé n'a pas de numéro de téléphone, il serait souhaitable d'afficher le nom ainsi que le numéro de téléphone du responsable du service auquel il appartient. Je précise qu'il y a un seul responsable par service.

1. Proposer un modèle conceptuel des données pour la conception de cette application
2. Indiquer pour chacune des associations son type : 1 - 1, 1-n ou m-n



Notons qu'une association est partielle si une des cardinalité est à 0. Donc ici, elles sont toutes partielles.

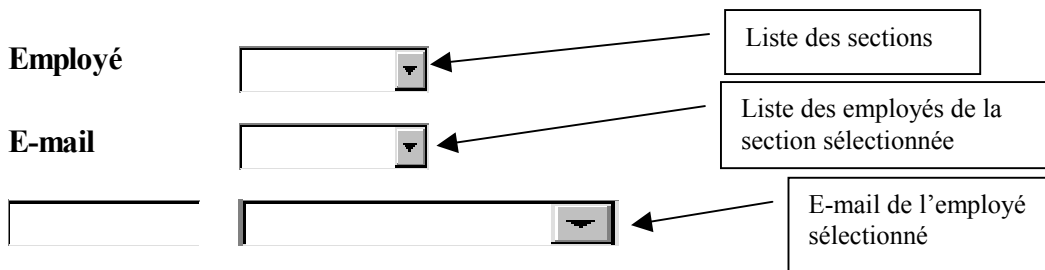
3. Lister toutes les incohérences acceptées par votre MCD

- Un salarié peut n'appartenir à rien (ni section, ni service),
- Un salarié peut être responsable d'un autre service que celui dont il fait partie.

Il faut donc mettre en place des contraintes applicatives:

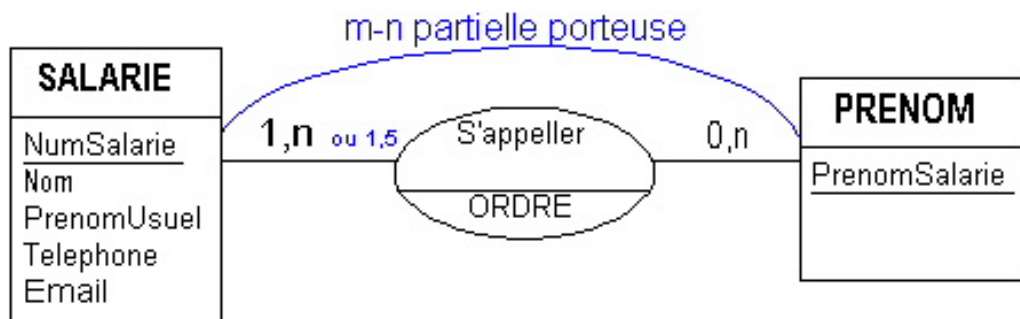
- Un responsable d'un service doit être "Rattaché directement" au même service,
- Tout salarié est rattaché soit à une section, soit à un service.

4. Faire apparaître sur le MCD le chemin emprunté lors de la mise œuvre du traitement suivant :



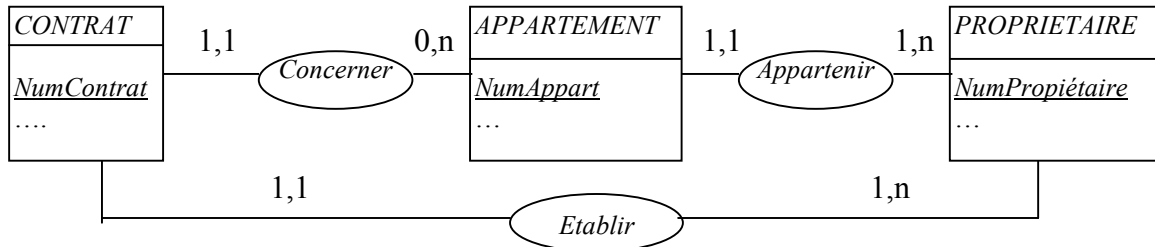
- Section: On prend toutes les occurrences de l'entité SECTION,
- Employé: Cardinalité 1,n pour avoir les occurrences de SALARIE concernées,
- E-mail: Cardinalité 0,1 pour avoir la propriété du SALARIE concerné.

5. On souhaite modifier la modélisation de façon à permettre la mémorisation de tous les prénoms d'un employé. Certains, aux parents forts imaginatifs, admettent jusqu'à 5 prénoms ! Proposer une solution qui permet l'enregistrement non seulement des prénoms d'un employé mais aussi de leur ordre dans l'état civil.



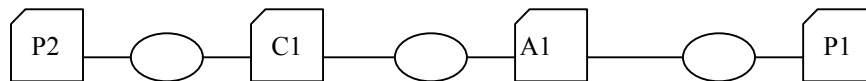
f) Les associations transitives

Considérons le modèle suivant :



L'association binaire qui relie l'entité « CONTRAT » et l'entité « PROPRIETAIRE » doit être ôtée du modèle car on peut retrouver le propriétaire à partir des associations « Concerner » et « Appartenir ». Il s'agit d'une association transitive. Seules les dépendances fonctionnelles directes entre identifiants d'entités devront donner lieu à des association binaires de type (1-n).

On veillera à supprimer les dépendances transitives pour ne pas surcharger le MCD car on risque de faire des incohérences comme:



g) Les associations réflexives

L'association réflexive est une association binaire qui relie une entité à elle-même. Une occurrence de l'association établit donc un lien entre une occurrence de l'entité et une autre occurrence de cette même entité. Dans le cas d'une association non symétrique, on doit faire porter le rôle sur chacun des segments, *comme l'illustre l'exemple suivant*.

<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p>PERSONNE</p> <p>Nom Prénom</p> </div> <p>Parent de 0,n</p> <div style="border: 1px solid black; border-radius: 50%; width: 100px; height: 30px; margin: 0 auto; display: flex; align-items: center; justify-content: center;">Parenté</div> <p>0,n Enfant de</p>	
<p><u>Remarque</u> : on suppose que deux personnes ne portent pas le même nom.</p>	<p>Le lien « Parent de » est représenté au moyen de segments en pointillés.</p>

Exercice n°3 (Acquis : Concepts de base uniquement)

On considère l'univers du discours suivant :

« Le personnel du service informatique de la MAAF peut être réparti en deux catégories :

- le personnel interne qui regroupe les employés rémunérés par la MAAF,
- le personnel extérieur qui englobe des salariés de différentes SSII

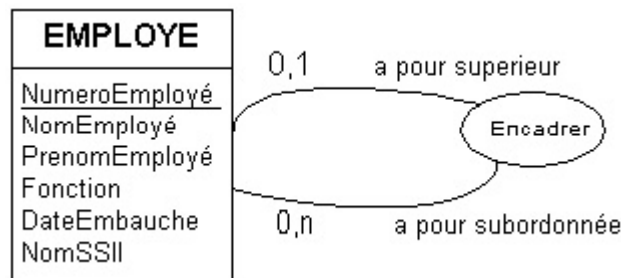
Excepté le directeur du service, chaque employé (extérieur ou interne) est encadré par un supérieur hiérarchique qui est obligatoirement un salarié interne. »

Relativement à la gestion du personnel du service informatique, on retient la liste des données suivantes :

NumeroEmployé	numéro qui identifie chaque employé travaillant au service informatique
NomEmployé	nom de l'employé
PrénomEmployé	prénom de l'employé
Fonction	fonction occupée par l'employé : Analyste-programmeur, chef de projet, ...
DateEmbauche	date de l'embauche. Propriété uniquement définie pour les salariés internes
NomSSII	nom de la société de services. Cette propriété est uniquement définie pour les salariés extérieurs

Proposer un modèle conceptuel de données décrivant cette réalité et permettant de répondre à des requêtes telles que :

- Quel est le nom du supérieur hiérarchique de M.X?
- Quelle est la liste des salariés internes?
- Quelle est la liste des employés ayant une fonction d'encadrement?



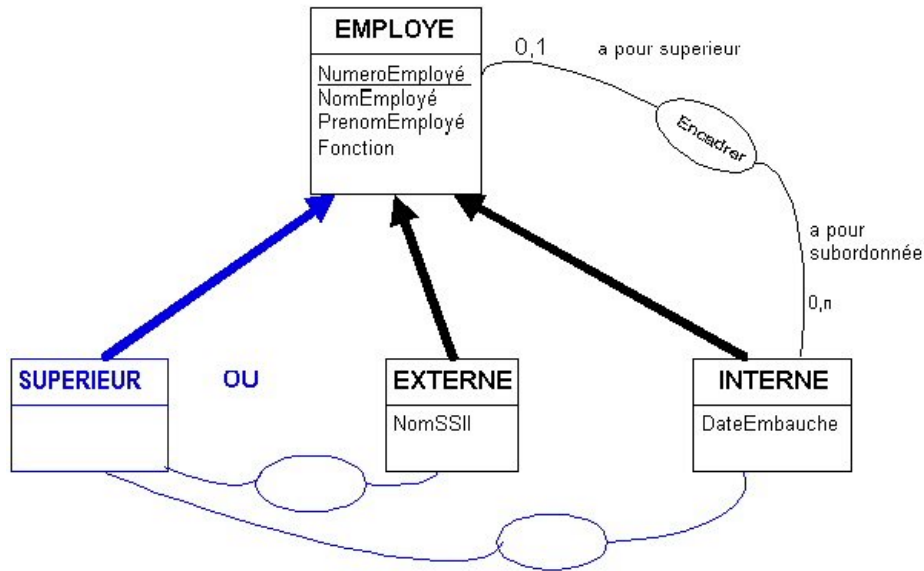
Contrainte applicative: Un employé doit avoir un supérieur "interne".

Contrainte d'exclusion: sur DateEmbauche et NomSSII.

Niveau sémantique: la date d'embauche n'a pas de sens pour un employé externe (idem pour NomSSII et un interne).

Considération physique: optimisation mémoire si NomSSII vide.

Solution par l'utilisation de Merise 2:



Auto-évaluation n° 2:

Dans le cadre de l'automatisation du suivi annuel de formation du personnel de la société x, le responsable des ressources humaines a défini pour chaque poste de l'entreprise, les compétences requises. Par exemple, le poste «infographiste » requiert les compétences: créativité, connaissances techniques et aptitudes relationnelles. Outre cette définition de poste, chaque salarié réalise en fin d'année un bilan de compétences destiné à connaître pour chaque compétence exigée par son poste son niveau actuel. Le tableau suivant récapitule les compétences du salarié n°1325.

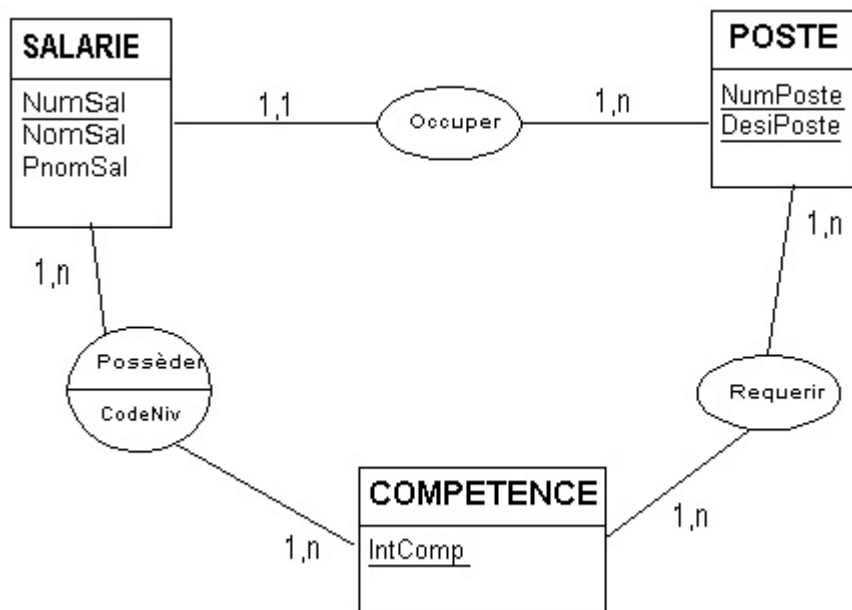
N° salarié: 1325	Poste: infographiste		
Nom: Dubois	Prénom: Michel		
Intitulé des compétences	Niveau actuel		
	A	B	C
Connaissances techniques	X		
Créativité		X	
Aptitudes relationnelles		X	

Les niveaux de compétence sont codés ainsi:

- A : compétence confirmée
- B : compétence à renforcer
- C : compétence en cours d'acquisition

1. Proposer un modèle conceptuel des données permettant de structurer les propriétés figurant dans le tableau suivant:

Nom propriété	Définition
NumSal	Numéro du salarié
NomSal	Nom du salarié
PnomSal	Prénom du salarié
NumPoste	Numéro du poste
DésiPoste	Désignation du poste
<u>IntComp</u>	Intitulé de la compétence
<u>CodeNiv</u>	Code du niveau: A, B ou C



Notons que le couple (NumSal, IntComp) détermine un et un seul CodeNiv.

2. Indiquer les incohérences éventuellement admises par la modélisation proposée.

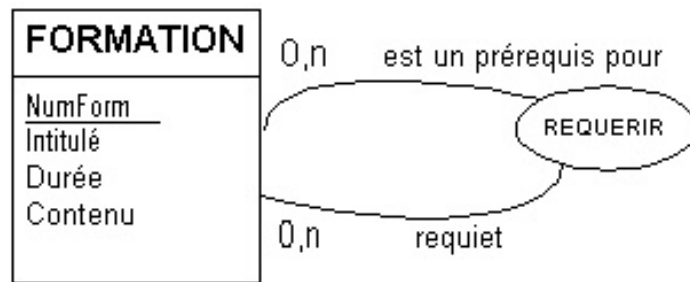
Un salarié peut occuper un poste sans posséder une compétence requise alors qu'il devrait l'avoir avec un certain niveau.

3. Afin de permettre à chaque employé d'évoluer dans son poste ou sur un autre poste, l'entreprise X propose différentes formations. Un extrait du catalogue est présenté ci-dessous:

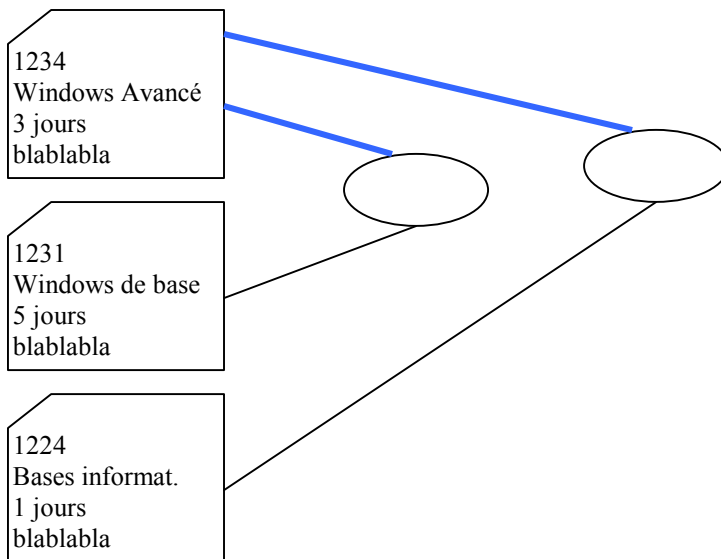
Formation n° 1231
 Intitulé : WINDOWS de base
 Durée: 5 jours
 Contenu : blablabla
 Formation(s) prérequis :

Formation n° 1234
 Intitulé: WINDOWS Avancé
 Durée: 3 jours
 Contenu: blablabla
 Formations pré-requis: 1231, 1224

Proposer un modèle conceptuel des données permettant d'accueillir l'ensemble des données contenues dans le catalogue de formations.



Exemple:

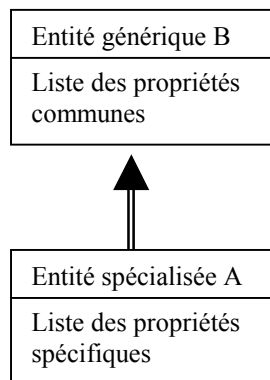


II. Les concepts étendus (MERISE 2)

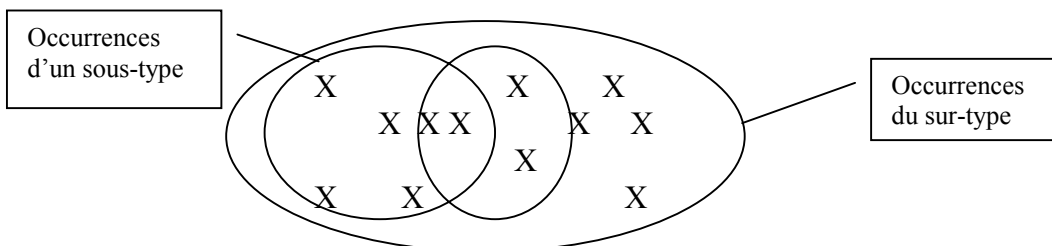
Le modèle entité-association retenu par la méthode Merise date des années 70. Or les concepts de ce modèle peuvent s'avérer insuffisants pour modéliser certaines situations ou contraintes et l'on est obligé dans ce cas d'ajouter des commentaires pour en faire mention. Les extensions au modèle individuel remédie aux faiblesses du formalisme de base.

2.1) Le concept d'héritage

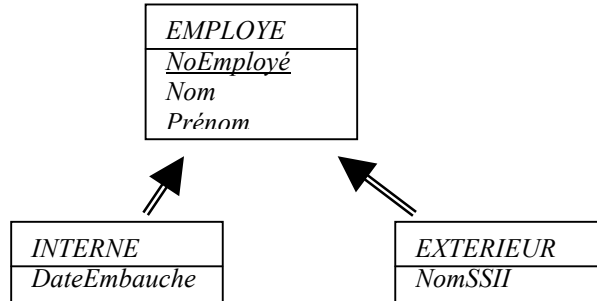
Quand le concepteur s'aperçoit que plusieurs entités, proches mais distinctes, partagent un ensemble de caractéristiques, il doit mettre en œuvre un processus de création d'entités génériques (ou entités sur-types) et d'entités spécialisées (ou entités sous-types) appelé «héritage». Ce concept qui permet de représenter le lien «est-un» ou «IS-A» entre deux entités A et B (une occurrence de A est une occurrence de B) est représenté graphiquement par une flèche double allant de A vers B.



On dit qu'il y a **héritage simple** quand un sous-type n'a qu'un seul sur-type. Dans ce cas, toutes les occurrences du sous-type sont en même temps des occurrences de son sur-type. Cela n'implique pas que toutes les occurrences du sur-type soient des occurrences de l'un des sous-types. Le schéma suivant illustre l'inclusion des ensembles d'occurrences des sous-types dans l'ensemble des occurrences du sur-type.



Le sous-type hérite de toutes les propriétés de son sur-type y compris de son identifiant. *Ce mécanisme qui correspond à l'héritage par spécialisation est utilisé dans le modèle suivant pour décrire partiellement l'univers du discours de l'exercice n° 3.*



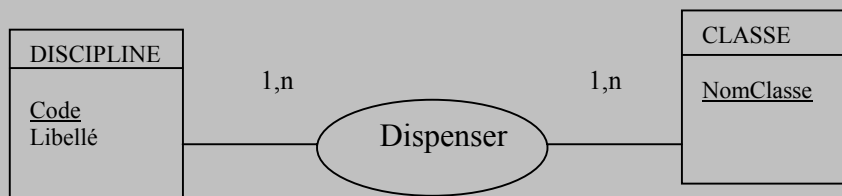
Le sous typage est une orientation vers le monde "objet".

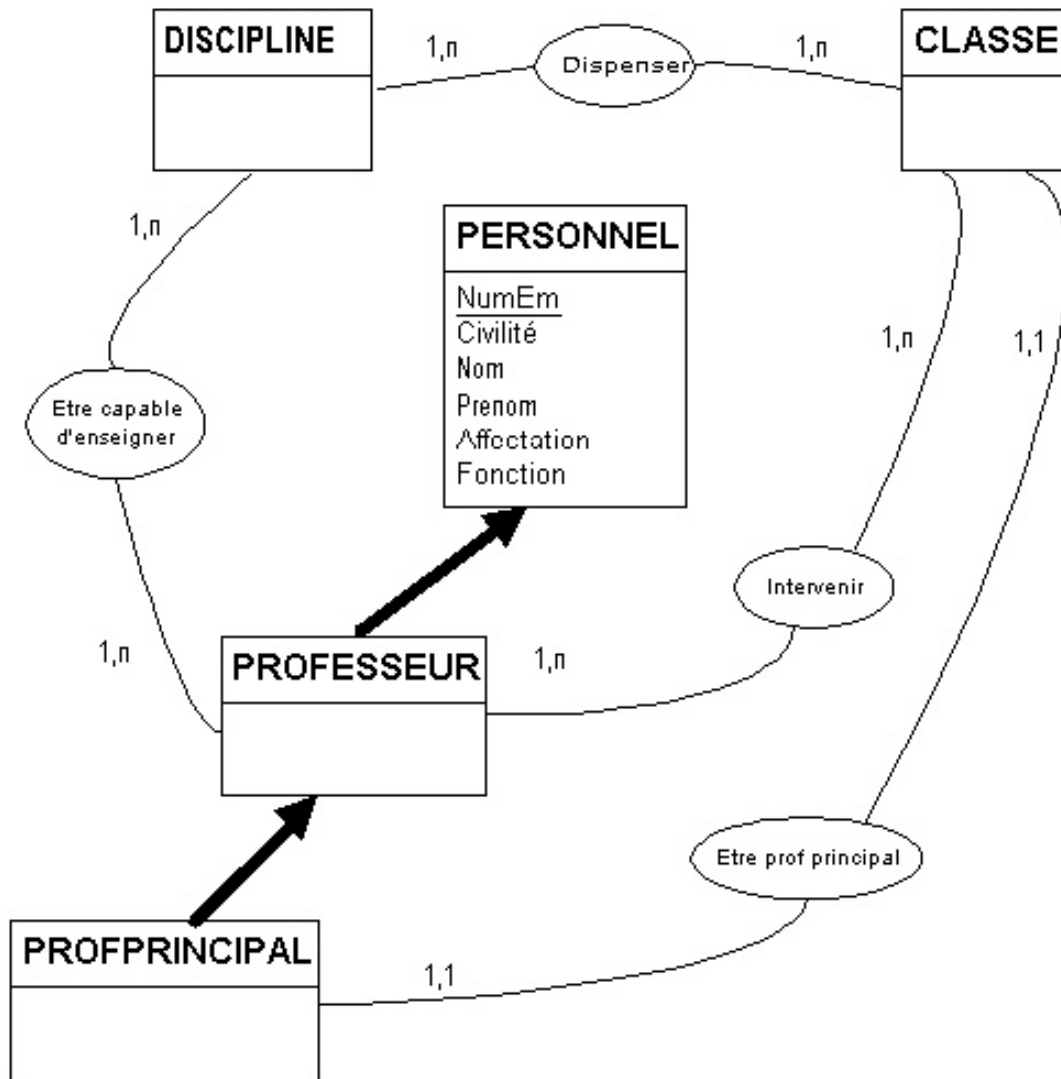
Exercice n° 5 (Acquis : concepts de base + sous-typage)

On souhaite représenter le système d'information relatif à la composition du personnel intervenant dans un lycée. Cette description devra fournir pour chaque personnel du lycée : administratif, professeur, surveillant, etc. un numéro qui permet de l'identifier (NUMEN ?) sans ambiguïté ainsi que les données signalétiques suivantes :

- civilité,
- nom,
- prénom,
- date d'affectation dans le lycée,
- fonction : professeur, proviseur, surveillant, CPE, ...

Parmi ces personnels, la représentation devra donner des renseignements supplémentaires sur les professeurs notamment la ou les discipline(s) qu'ils sont capables d'enseigner ainsi que les classes qu'il ont en charge. La notion de professeur principal devra être aussi modélisée. Compléter le modèle conceptuel des données suivant et mentionner sous forme de commentaires les contraintes qui sont non exprimables par les concepts de base.





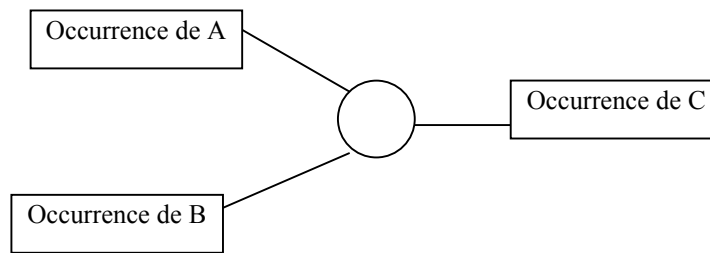
Contraintes:

- On ne sait pas ce que le professeur enseigne à sa classe.
- Le professeur doit enseigner à la classe dont il est principal.

On remarquera qu'ici, Ens(Classe) est incompatible avec Ens(Intervenir). Par contre, Ens(ProfPrincipal) est compatible avec Ens(Intervenir).

2.2) Les contraintes ensemblistes

Ces formalismes vont permettre d'exprimer des contraintes sur des ensembles d'occurrences d'entités ou d'associations. On peut considérer que c'est le plus grand apport de Merise 2. Si la notion d'ensemble d'occurrences d'une entité ne pose aucun problème, il convient d'apporter certaines précisions sur l'ensemble des occurrences d'une association. Considérons pour cela une association ternaire reliant trois entités A, B et C. Une occurrence de cette association est un lien « tri-pattes » qui relie une occurrence de l'entité A, une occurrence de l'entité B et une occurrence de l'entité C .



En désignant par a1, b1 et c1 les valeurs des rubriques identifiantes de ces 3 occurrences d'entités, on peut matérialiser l'occurrence de l'association par le triplet : (a1, b1, c1). L'ensemble des occurrences d'une association peut donc être représenté par l'ensemble des triplets issus de la présence de liens entre les occurrences des entités A, B et C. Le tableau ci-dessous présente des exemples d'occurrences des associations « Travailler » et « Habiter ».

Modèle conceptuel des données	Occurrences de l'association TRAVAILLER	Occurrences de l'association HABITER
	(« Dupont »,16) (« Dubois »,16) (« Durand »,17) (« Laforet »,17)	(« Dupont »,16) (« Dubois »,17) (« Durand »,16) (« Laforet »,86)
<p><i>Remarque : on suppose qu'il n'y a pas deux professeurs qui portent le même nom</i></p>		

A partir de l'ensemble des occurrences d'une association, il est possible de construire d'autres ensembles en supprimant, dans chaque nuplet, la participation d'une ou plusieurs entités. Dans l'exemple ci-dessus, si l'on considère uniquement l'entité DEPT (et donc on ne se préoccupe plus de l'entité PROF), l'ensemble des occurrences de l'association Travailler, limité à l'entité DPT, devient composé des éléments 16 et 17.

- (« Dupont »,16)
- (« Dubois »,16)
- (« Durand »,17)
- (« Laforet »,17)

...

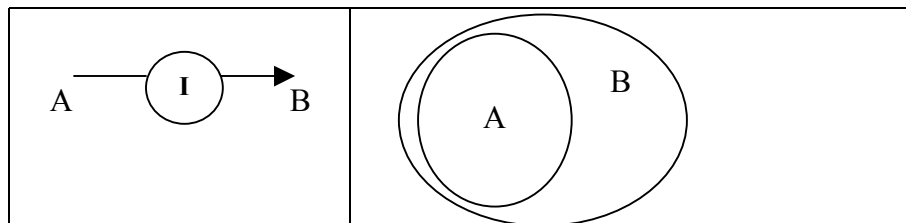
On appelle pivot la (ou les) entité(s) que l'on retient pour former ces nouveaux ensembles. Pour simplifier les futures définitions, l'ensemble des occurrences d'un pivot participant à une association sera désormais référencé **Ens(Pivot,Association)**. En gardant la même logique, la référence **Ens(Entité)** désignera l'ensemble des occurrences de l'entité citée et la référence **Ens(Association)** l'ensemble des occurrences de l'association.

Pour les associations réflexives, il est possible de définir, à partir de l'ensemble des occurrences, deux nouveaux ensembles, notés **Ens(Rôle, Association)** issus chacun des rôles portés par les deux segments. *En reprenant l'exemple fournit au paragraphe 1.3-g, les occurrences de l'association Parenté sont : (« Dubois », « Durand ») et (« Durand », « Delage »). Le premier composant du couple désigne le père, le second l'enfant. L'ensemble Ens(Enfant, Parenté) regroupe donc les valeurs « Durand », « Delage ».*

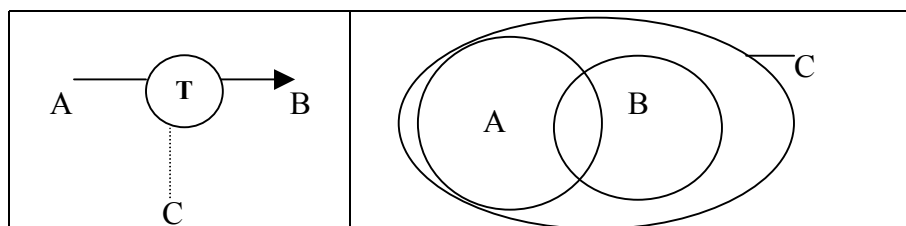
Quelque soit leur origine, les ensembles sur lesquels portent une contrainte ensembliste doivent être compatibles c'est-à-dire que leurs éléments doivent avoir des structures (en terme de rubriques) identiques. *Par exemple on peut définir une contrainte ensembliste entre Ens(TRAVAILLER) et Ens(HABITER). Par contre, il n'est pas possible de définir une contrainte ensembliste entre Ens(PROF) et Ens(TRAVAILLER) car leurs éléments n'ont pas la même structure.*

Les contraintes ensemblistes qu'il est possible d'exprimer au moyen des concepts étendus sont présentées de façon générale ci-dessous :

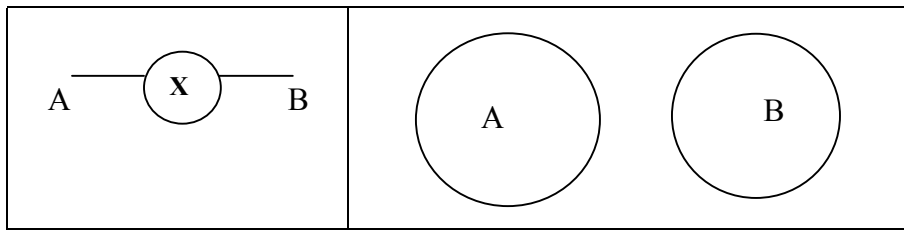
- **La contrainte d'inclusion** (notée **I**) entre deux ensembles A et B impose que l'ensemble A soit inclus dans l'ensemble B. L'inclusion n'étant pas symétrique, le formalisme associé à cette contrainte met en valeur, parmi les deux ensembles, celui qui doit être inclus dans l'autre.



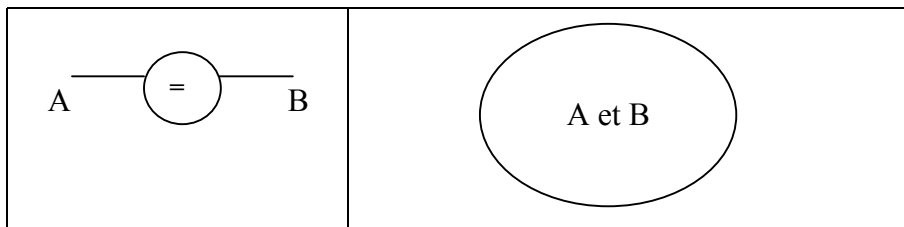
- **La contrainte de totalité** (notée **T**) porte sur trois ensembles A, B et C et impose que l'union de A et de B soit égale à C. Les trois ensembles ne jouant pas le même rôle, le formalisme prévoit un moyen de discerner les ensembles sur lesquels l'union sera réalisée et l'ensemble qui doit être égal à cette union.



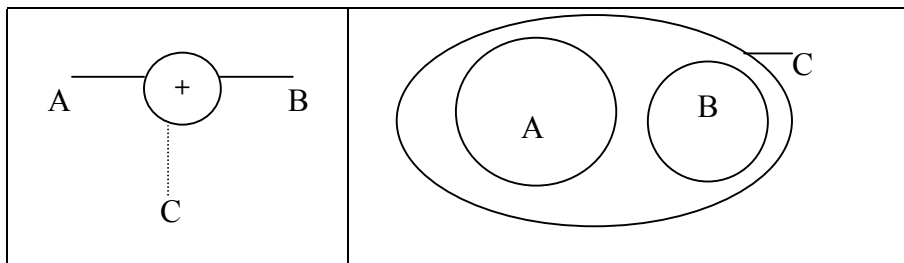
- **La contrainte d'exclusion** (notée X) entre deux ensembles A et B impose que l'intersection entre A et B soit vide.



- **La contrainte d'égalité** (notée =) entre deux ensemble A et B impose que l'ensemble A soit égal à l'ensemble B. Elle traduit en fait les deux inclusions suivantes : A inclus dans B et B inclus dans A.



- **La contrainte du ou exclusif** (notée +) porte sur trois ensembles et est la combinaison d'une exclusion et d'une totalité. Elle traduit le fait que l'intersection entre A et B est vide et que l'union de A et B est égale à l'ensemble C.



Les contraintes d'égalité et du « ou exclusif » étant dérivées respectivement des contraintes d'inclusion, de totalité et d'exclusion, seules ces trois dernières font l'objet d'une description plus précise dans les paragraphes qui suivent.

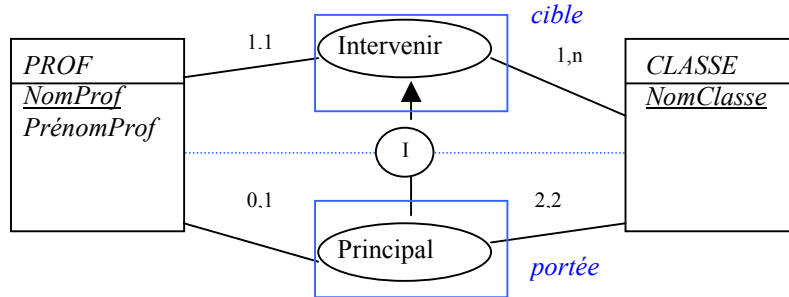
a) La contrainte d'inclusion

La contrainte d'inclusion s'exprime entre une ou plusieurs associations sources appelées **portée** et une seule association destination appelée **cible**. Si le pivot n'est pas précisé graphiquement (au moyen de traits en pointillés), il est nécessaire de le déterminer en appliquant la règle suivante : il est constitué des entités communes aux associations de la portée et de la cible. La contrainte d'inclusion garantit alors que $Ens(\text{Pivot}, \text{Portée})$ est inclus dans $Ens(\text{Pivot}, \text{Cible})$. *Dans l'exemple suivant le pivot est implicite et correspond aux entités ELEVE et CLASSE. On a donc les deux égalités suivantes :*

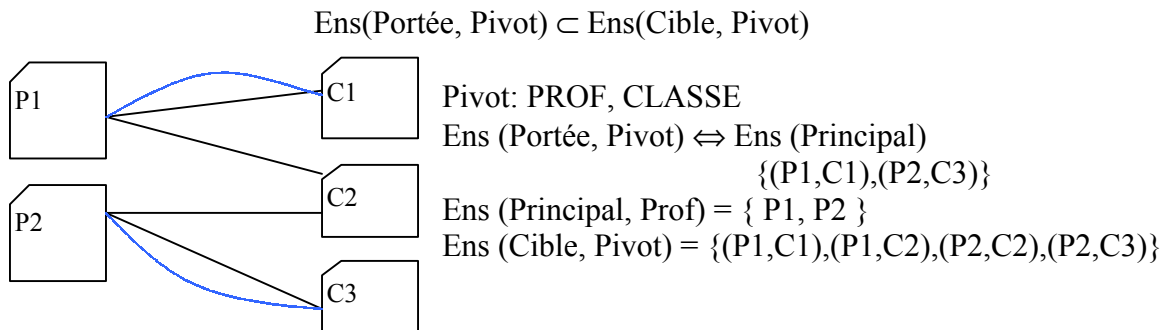
$$Ens(\text{Pivot}, \text{Portée}) = Ens(\text{Principal})$$

$$Ens(\text{Pivot}, \text{Cible}) = Ens(\text{Intervenir})$$

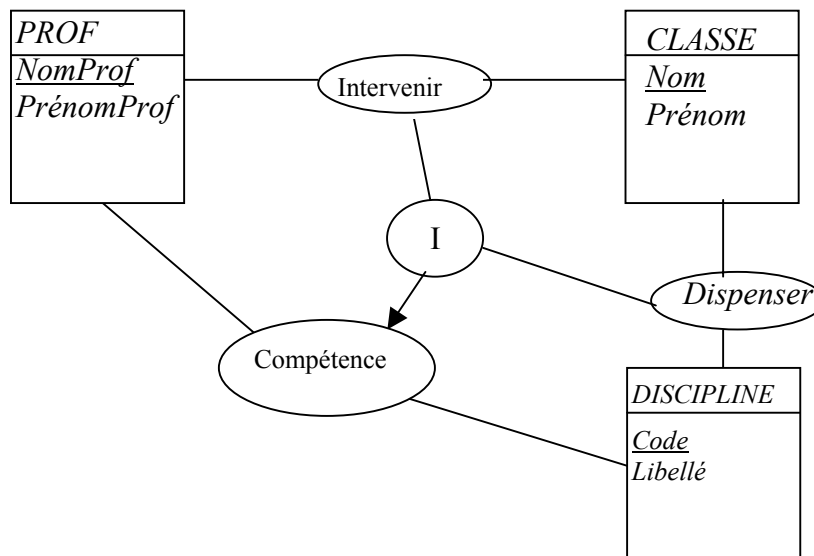
La contrainte d'inclusion assure que tout professeur principal d'une classe intervient dans la classe.

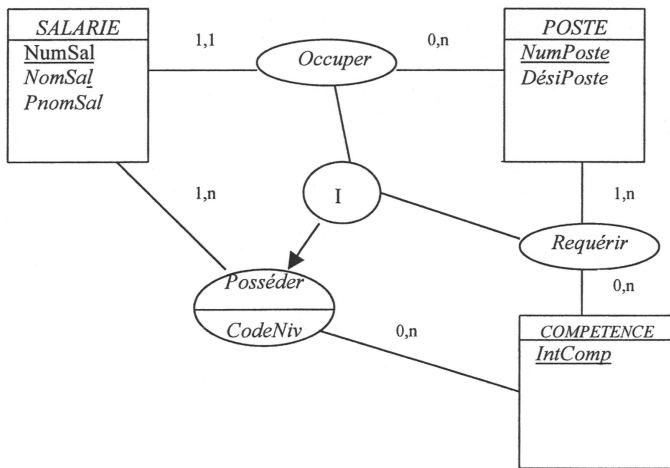


On note le pivot explicite par des pointillés, alors que le pivot implicite est formé par les entités communes à la portée et à la cible).

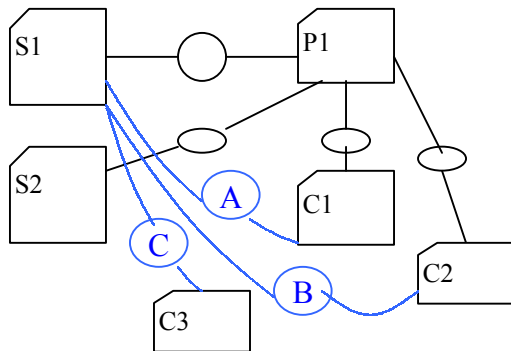


La cible est toujours simple, mais la portée peut être composée. Dans le cas où la portée est composée (c'est-à-dire formée de plusieurs associations), la détermination de $Ens(\text{Pivot}, \text{Portée})$ impose des opérations de jointure comme le montre les exemples suivants :





Le pivot implicite est formé par les entités : SALARIE et COMPETENCE. La portée est formée par les deux associations : Occuper et Requirir. Pour trouver l'ensemble $Ens(Pivot, Portée)$, il faut pour chaque Salarié, récupérer les compétences qu'il requiert. Tous les couples (Salarié, Compétence) trouvés en empruntant ce chemin forment l'ensemble $Ens(Pivot, Portée)$ qui doit être inclus dans l'ensemble $Ens(Posséder)$.

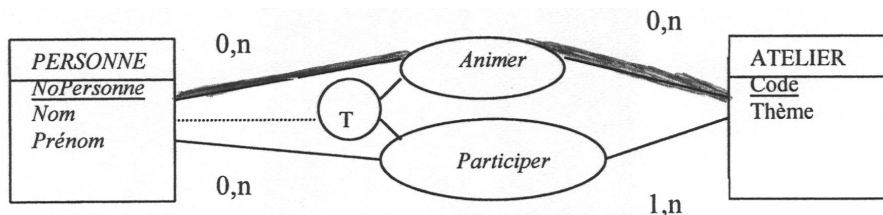


$Ens(Occuper) = \{(S1, P1), (S2, P1)\}$
 $Ens(Occuper, Salarié) = \{S1, S2\}$
association, pivot
 $Ens(Requirir, Compétence) = \{C1, C2\}$
 $Ens(Portée, Pivot) = (occuper-requirir, salarié-compétence)$
 $\{(S1, C1), (S1, C2), (S2, C1), (S2, C2)\}$
 $Ens(Cible, Pivot) = \{(S1, C1), (S1, C2), (S1, C3)\}$

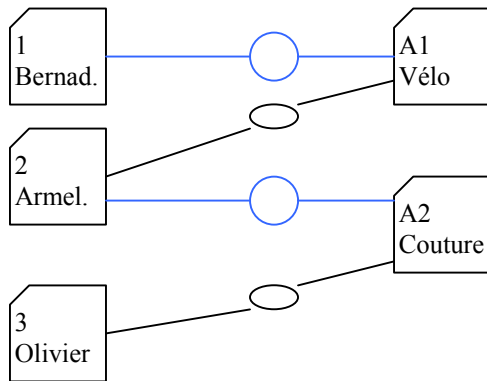
b) La contrainte de totalité

➤ **La contrainte de totalité entre associations**

Cette contrainte s'exprime entre n associations ($n \geq 2$) et impose que le pivot implicite ou explicite soit composé d'une seule entité. La contrainte garantit que l'union de tous les ensembles $Ens(Pivot, Association_i)$ (avec i compris entre 1 et n) soit égal à $Ens(Pivot)$. Dans l'exemple qui suit la contrainte de totalité impose que toute personne est soit animateur d'un atelier, soit participant soit les deux.



$Ens(AssociationA, Pivot) \cup Ens(AssociationB, Pivot) = Ens(Entité Pivot)$

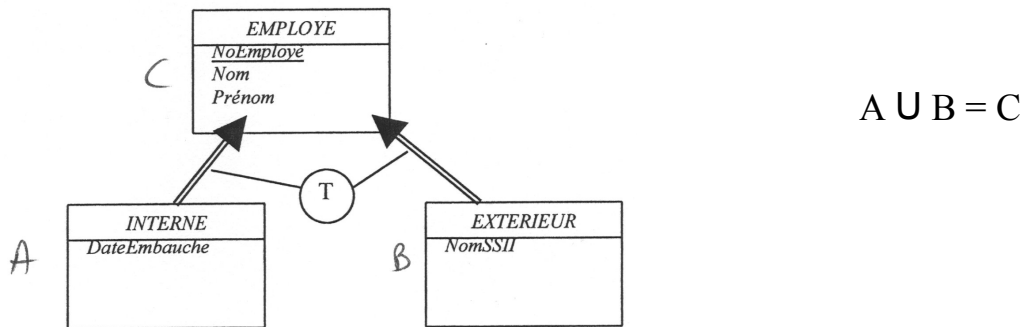


$Ens(Animer) = \{(1,A1), (2A2)\}$
 $Ens(Animer, Personne) = \{1,2\}$
 $Ens(Participer, Personne) = \{2,3\}$

$Ens(Participer, Personne) \cup Ens(Animer, Personne) = \{1,2,3\}$

➤ **La contrainte de totalité entre sous-types**

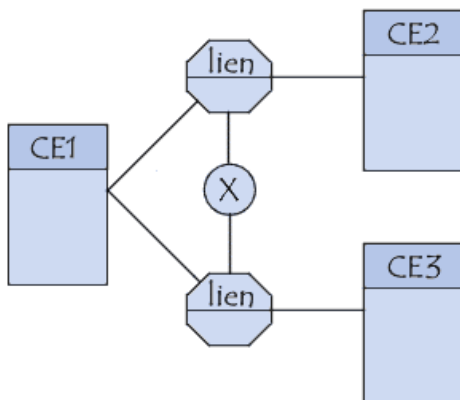
Cette contrainte s'exprime entre n sous-types ($n \geq 2$). Elle traduit le fait que l'union des ensembles $Ens(SousType_i)$ (avec i compris entre 1 et n) soit égale à $Ens(SurType)$. Toute occurrence du sur-type correspond donc à une occurrence d'un ou de plusieurs sous-types. Par exemple pour modéliser le fait que tous les employés du service informatique de la MAAF sont soit des internes soit des extérieurs on placerait une contrainte de totalité entre les deux sous-types.



Il est important de remarquer que cette contrainte n'interdit pas qu'un salarié interne soit aussi un extérieur. Pour refuser cette possibilité il sera nécessaire de renforcer cette contrainte par l'exclusion.

c) **La contrainte d'exclusion**

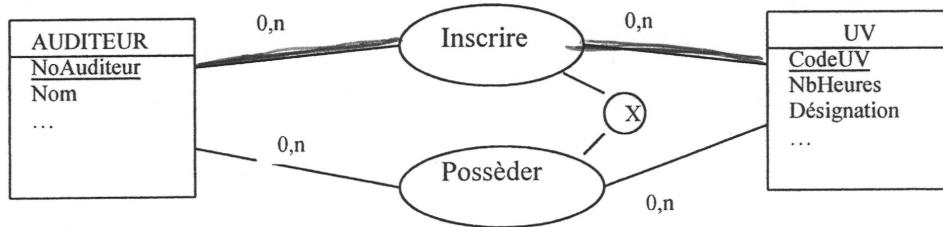
➤ **La contrainte d'exclusion entre associations**



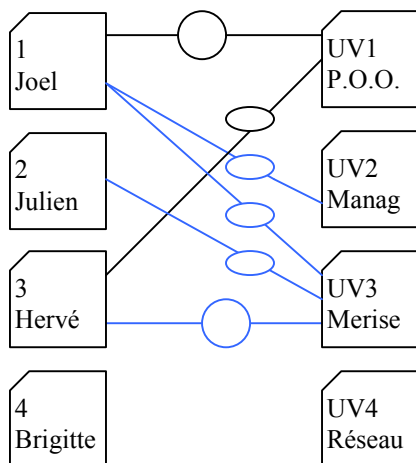
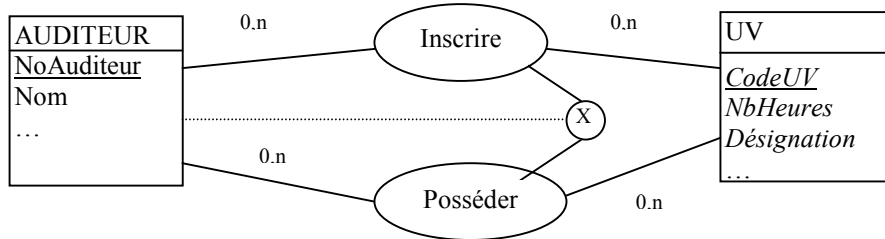
La contrainte d'exclusion sur relation exprime le fait que **deux occurrences de classes d'entité ne peuvent pas participer simultanément à une même classe de relation.**

Elle est représenté par un "X" reliant deux classes de relation.

Cette contrainte s'exprime entre n associations ($n \geq 2$). Si le pivot n'est pas précisé graphiquement le pivot implicite est composé des entités communes aux différentes associations en jeu. La contrainte d'exclusion garantit que l'intersection des Ens (Pivot, Association i) (avec i compris entre 1 et n) est vide. Dans le modèle suivant, la contrainte d'exclusion permet d'exprimer qu'un auditeur ne peut, à la fois, être inscrit à une UV et la posséder. Le pivot implicite est composé des entités AUDITEUR et UV.



Le modèle suivant, qui fait apparaître un pivot explicite, interdit qu'un auditeur apparaisse à la fois dans une occurrence de l'association Inscrire et dans une occurrence de l'association Posséder. Cela signifie que tout auditeur doit soit être inscrit à un ensemble d'UV soit posséder un ensemble d'UV mais pas les deux à la fois. Une telle modélisation n'a bien évidemment pas de sens.



Pivot: Auditeur et UV

Ens (Posséder, Pivot) = {(1,UV1), (3,UV1)}

Ens(Inscrire,Pivot) = {(1,UV2),(1,UV3),(2,UV3),(3,UV3)}

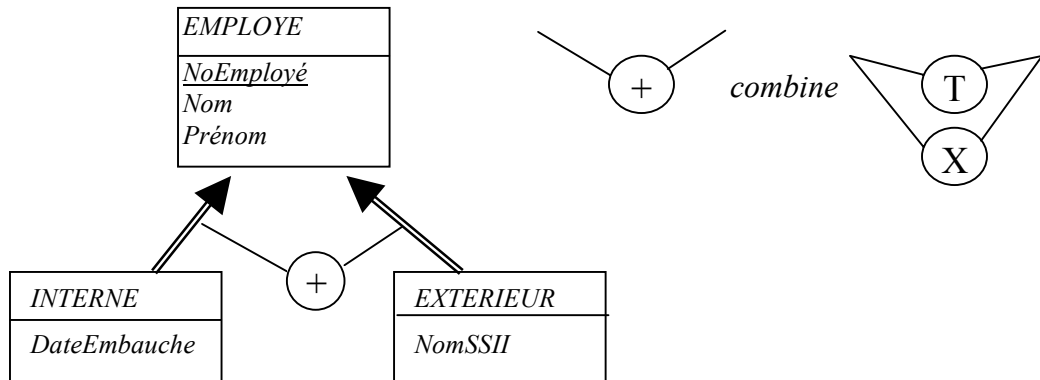
Pivot: Auditeur

Ens (Inscrire, Pivot) = { 1, 2, 3 }

Ens (Posséder, Pivot) = { 1, 3 }

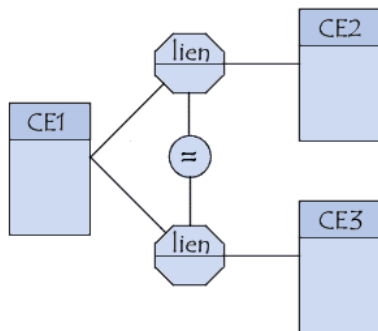
➤ La contrainte d'exclusion entre sous-types

Cette contrainte s'exprime entre n sous-types ($n \geq 2$) et traduit le fait que l'intersection des $\text{Ens}(\text{SousTypes}_i)$ (avec i compris entre 1 et n) est vide ce qui revient à dire que toute occurrence de l'entité sur-type ne peut participer qu'à l'un ou l'autre (ou aucun) des sous-types de la contrainte. Dans l'exemple relatif à la modélisation du personnel du service informatique, un employé est soit un extérieur soit un salarié interne mais il ne peut pas être les deux à la fois.



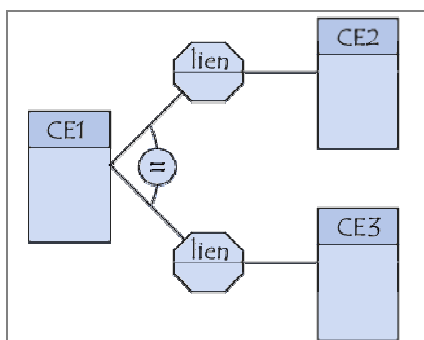
d) La contrainte d'égalité

➤ La contrainte d'égalité entre associations



La contrainte d'égalité entre associations exprime le fait qu'**une occurrence de classe d'entité participant à une classe d'association, participe obligatoirement à l'autre classe d'association, et réciproquement**. Il s'agit donc d'une contrainte de sous-ensemble bidirectionnelle. Elle est représenté par un signe "=" reliant deux classes d'association. Cette contrainte peut faire intervenir plusieurs occurrences de classes d'entité, auquel cas une occurrence de classe d'entité participant à une classe de relation doit participer aux n classes d'association.

➤ La contrainte d'égalité entre sous-types



La contrainte d'égalité entre sous-types exprime le fait qu'**une entité participant à une classe d'association, participe obligatoirement à l'autre association, et réciproquement**. Il s'agit donc d'une contrainte de sous-ensemble bidirectionnelle. Elle est représenté par un signe "=" reliant deux classes d'entités. Cette contrainte peut faire intervenir plusieurs associations, auquel cas une entité participant à une association doit participer aux n associations.

Exercice n° 6 (Acquis : base + sous-typage + contraintes ensemblistes)

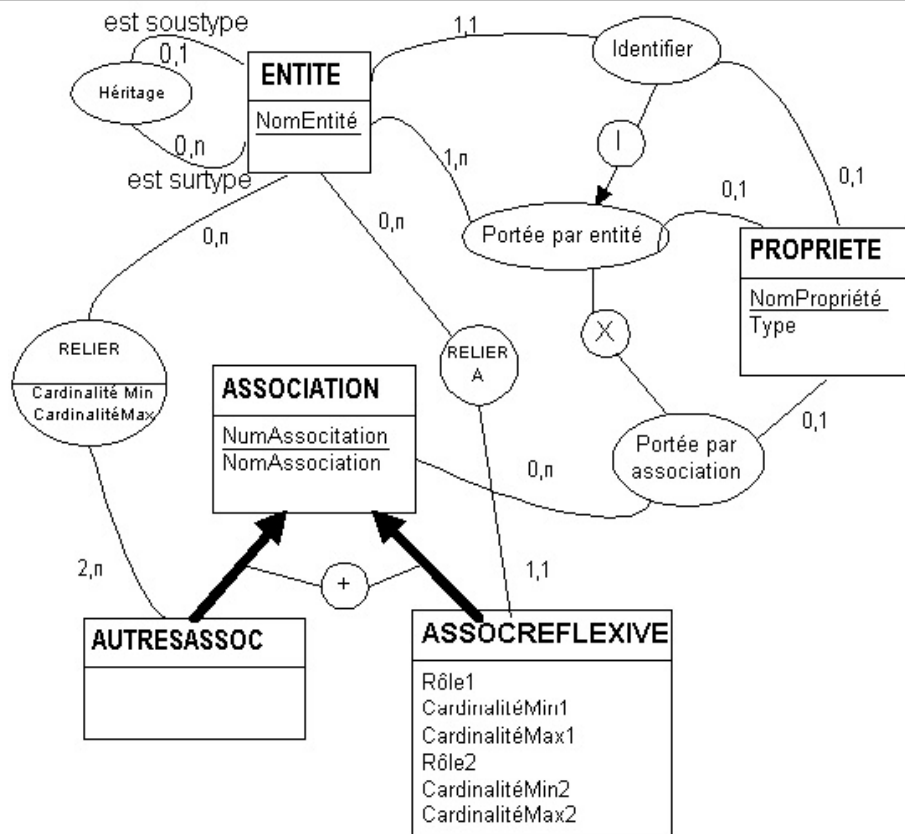
Cet exercice consiste à représenter toutes les notions qui se rapporte au modèle conceptuel des données au moyen des formalismes de ce modèle. On parle de méta-modèle. Les propriétés à structurer sont présentées dans la liste qui suit :

Propriétés	Définition
NomPropriété	Nom qui identifie une propriété
TypePropriété	Type de la propriété : alphabétique, numérique, ..
NumAssociation	Numéro qui identifie chaque association
NomAssociation	Nom de l'association
NomEntité	Nom qui identifie chaque entité
CardinalitéMin	Valeur minimale d'une cardinalité
CardinalitéMax	Valeur maximale d'une cardinalité
Rôle	Rôle porté par un segment d'une association réflexive

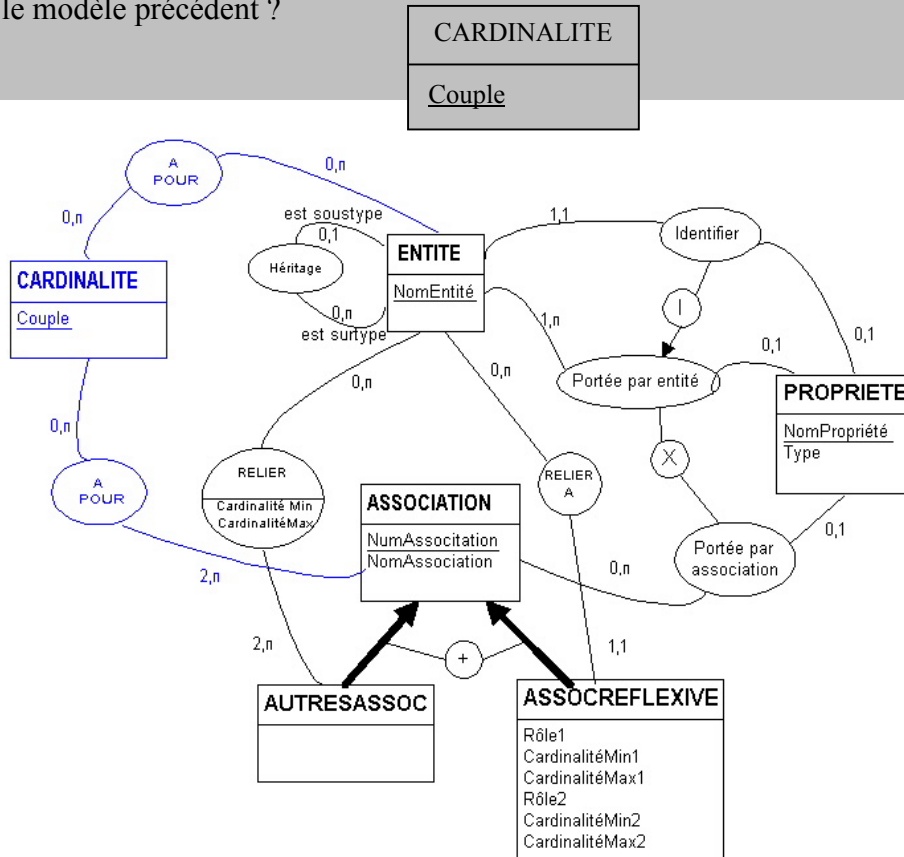
Remarque : la notion de contrainte n'est pas modélisée

1. Concevoir un MCD qui permet de répondre aux requêtes suivantes :

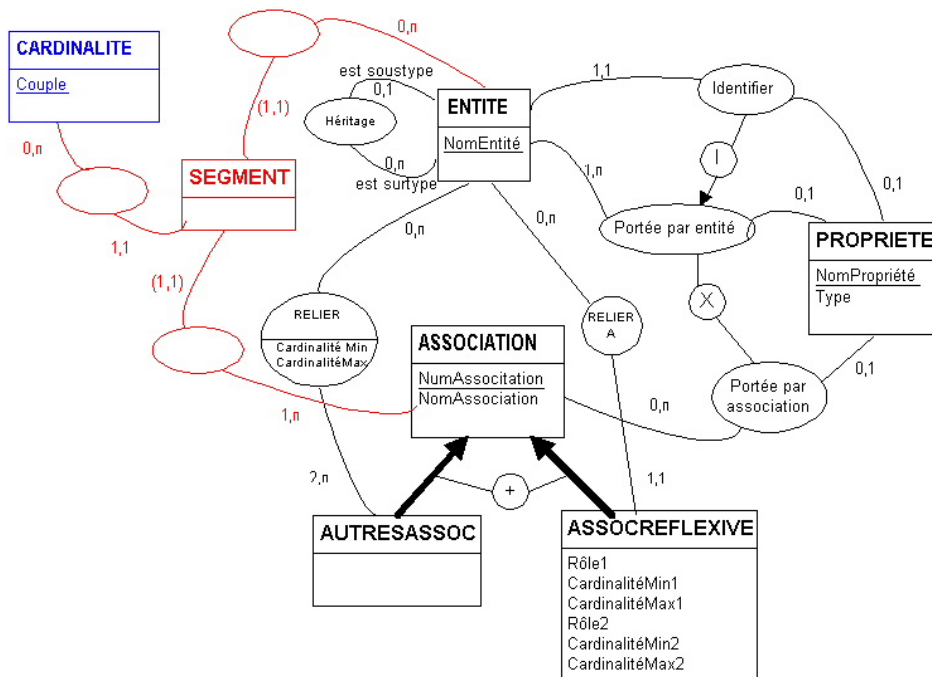
- Quelle est la liste des propriétés d'une entité?
- Quel est le nom de la propriété identifiante d'une entité?
- Quelle est la liste des entités sous-types d'une entité?
- Quelle est la liste des entités, participant à une association, avec pour chaque segment, reliant une association à une entité, le couple de cardinalités?
- Quelle est la liste des associations avec éventuellement les propriétés portées?
- Quels sont les rôles associés aux segments d'une association réflexive?



2. On introduit l'entité **CARDINALITE** ci-dessous destinée à mémoriser les 4 couples possibles de cardinalités : (0,1), (1,1), (0,n) et (1,n). Quel problème pose l'insertion de cette entité dans le modèle précédent ?



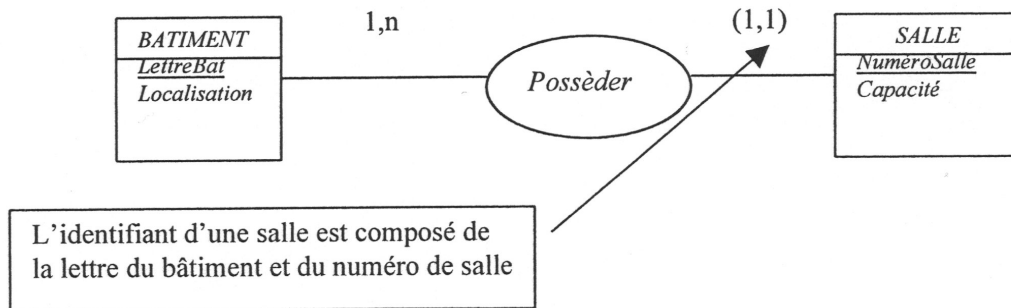
Le couple entité-relation n'est plus respecté mais on évite ainsi des cardinalités absurdes (comme 0,0 ou 20,10). On peut introduire aussi une pseudo entité **SEGMENT** avec un identifiant relatif (1,1) qui est la traduction du couple (NomEntité, NumAssociation).



2.3) L'identification des occurrences d'entités

Les extensions à l'identification des entités permettent d'accepter plusieurs façon d'identifier une entité et suppriment ainsi les identifiants artificiels, introduits uniquement pour respecter la définition d'une entité. Merise 2 admet les deux types d'identifiants suivants :

- **l'identifiant absolu** constitué de une ou plusieurs propriétés de l'entité. Dans la représentation graphique celles-ci sont soulignées. *On pourra ainsi identifier une personne au moyen des trois propriétés : nom, prénom et date de naissance.*
- **l'identifiant relatif** constitué de propriétés de l'entité et/ou au moins de l'identifiant d'une autre entité reliée par une association 1-n ou 1-1. *L'exemple suivant illustre une telle notion et introduit sa représentation graphique.*

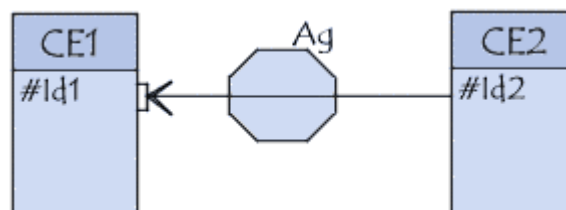


Remarque : l'association entre les deux entités doit être stable, c'est-à-dire qu'une fois un lien établi entre deux occurrences, celui-ci ne doit plus être modifié dans le temps.

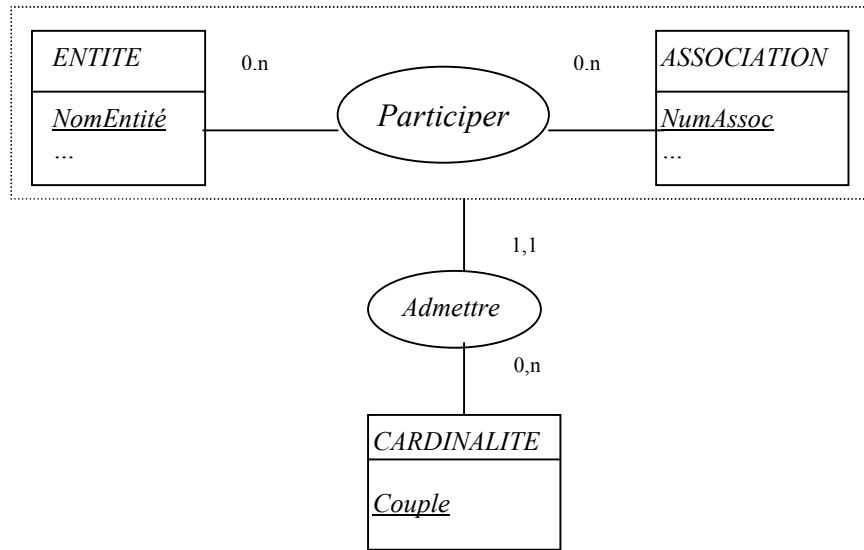
La notion d'identifiant relatif permet aussi d'exprimer un lien entre une association et une ou plusieurs entités. Certains auteurs appellent une telle association pseudo-entité ou agrégation. Lorsqu'un identifiant est constitué uniquement d'attributs intrinsèques à une entité, c'est-à-dire ne faisant référence à aucune autre entité, on le nomme **identifiant absolu**. Les entités comportant des identifiants absolus peuvent être définis indépendamment des autres occurrences d'entités, on dit que ces entités sont indépendantes. Certaines entités ne peuvent toutefois être identifiées que par l'intermédiaire d'autres entités, c'est la raison pour laquelle on parle d'**identification relative**. On parlera par exemple de *la 4^{ème} porte au 2^{ème} étage du bâtiment B* au lieu de dire la porte n°3451... Ainsi, l'**agrégation** (appelée aussi identification relative) permet de spécifier qu'une entité est nécessaire pour en identifier une autre.

- la classe d'entité permettant d'identifier est appelé classe d'entité agrégante
- la classe d'entité identifiée est appelée classe d'entité agrégée

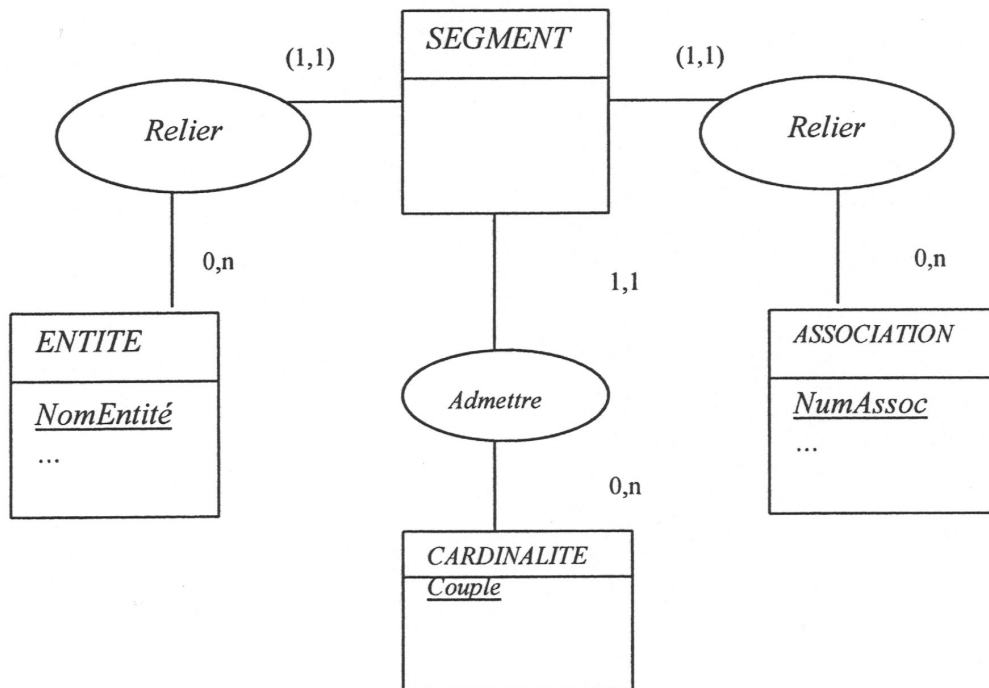
La représentation de ce type de relation est la suivante:



Dans l'exemple qui suit on souhaite établir un lien entre l'association participer et l'entité CARDINALITE.



Pour se dispenser de cette pseudo-entité, il suffit de transformer l'association en une entité et de lui associer un identifiant relatif composé des identifiants des entités reliées par des association de fonctionnalité (1-n) ou (1-1). L'exemple ci-dessous pourrait donc être transformé ainsi :



III. Règles de construction d'un M.C.D.

Pour la construction du modèle conceptuel, beaucoup de méthodes ont été mises en place mais aucune ne donne réellement satisfaction. On peut cependant les répartir en deux catégories :

3.1) modélisation directe

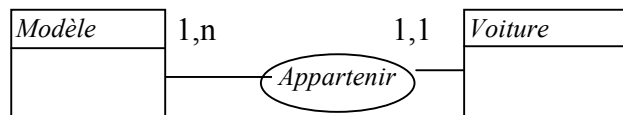
Elle consiste à identifier, à partir d'une description exprimée en langage naturel, les entités et les associations en appliquant les règles suivantes :

- les noms deviennent des entités
- les verbes deviennent des associations

L'exemple suivant qui illustre ce propos est bien trop simple pour que cette méthode conduise à des résultats satisfaisants sur un système d'information de taille plus importante.

Une voiture appartient à un modèle particulier. Les noms sont : « voiture », « modèle ». Le verbe est : « appartient à »

Ce qui donne la modélisation



Le modèle obtenu par cette méthode est très loin de la représentation optimale et il sera nécessaire d'appliquer une phase de validation et de normalisation (élimination des situations qui induisent des redondances) pour aboutir à une solution satisfaisante.

3.2) modélisation par analyse des dép. fonct^{lles}

Cette méthode consiste à identifier en premier lieu toutes les propriétés du système d'information à analyser. Cette étape aboutit au dictionnaire des données épuré qui devra comporter ni synonyme, ni polysème, ni donnée calculée. Pour faciliter la conception ultérieure des bases de données, il est recommandé de définir pour chaque donnée du dictionnaire son domaine. Le domaine d'une donnée est l'ensemble des valeurs que peut prendre cette donnée. Il peut être :

- étendu: il correspond alors au type d'une donnée : Numérique, alphabétique, etc.
- restreint: on l'exprime alors au moyen d'une liste ou d'un intervalle. Par exemple, pour la rubrique « Sexe », le domaine sera la liste de valeurs « F », « M ».

La seconde étape réside dans la recherche des dépendances fonctionnelles entre les propriétés recensées à la première étape. Pour mener de façon méthodique ce travail, on construit une matrice des dépendances fonctionnelles admettant une ligne et une colonne par propriétés du dictionnaire. Un « 1 » placé à l'intersection de la ligne i et de la colonne j indique la présence d'une dépendance fonctionnelle entre la propriété Pj et Pi (Pj → Pi).

L'exemple suivant illustre cette technique sur l'exemple classique de gestion des commandes d'une entreprise.

	1	2	3	4	5	6	7	8
1 NoCommande	1							
2 DateCommande	1	1						
3 NoClient	1		1					
4 NomClient	1		1	1				
5 RefProduit					1			
6 Designation					1	1		
7 PrixUnitaire					1		1	
8 QtéCommandée								1

NoClient → NomClient

Cette première phase met en évidence deux types de propriétés :

- les **propriétés identifiantes** repérées dans la matrice par des colonnes qui comportent au moins deux « 1 ». Dans l'exemple ci-dessus les propriétés 1, 3 et 5 sont sources de dépendances fonctionnelles et joueront donc le rôle d'identifiant d'entités dans le modèle conceptuel.
- les **propriétés** qui ne sont **destination d'aucune dépendance** fonctionnelle et qui ont donc, dans la matrice des dépendances fonctionnelles, leur ligne vide. Dans l'exemple ci-dessus, les propriétés *TauxTVA* et *QtéCommandée* ont ces caractéristiques. Parmi ces propriétés il convient alors de distinguer :
 - les **propriétés paramètres** telles que le *taux de TVA*
 - les **autres propriétés** pour lesquelles on doit rechercher les dépendances fonctionnelles ayant des sources multi-attributs qui permettent de les atteindre.

La source de ces dépendances sera constituée d'un sous-ensemble des rubriques identifiantes repérées à l'étape précédente. Ainsi, la propriété *QtéCommandée*, est déterminée à partir d'un numéro de commande et d'une référence produit ce qui revient à écrire la dépendance fonctionnelle suivante : **RefProduit, NoCommande → QtéCommandée**

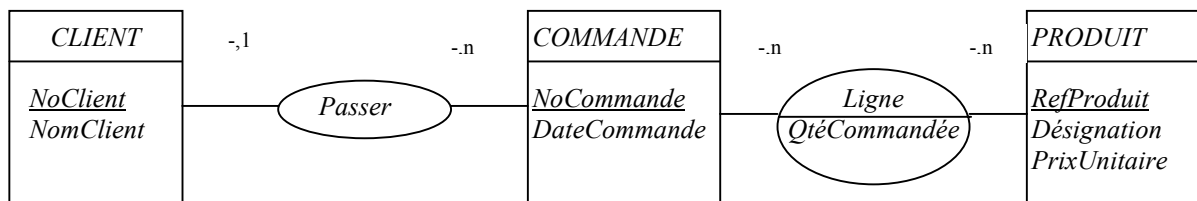
Ces nouvelles dépendances fonctionnelles sont alors ajoutées à la matrice des dépendances fonctionnelles comme le montre la figure suivante.

	<u>1</u>	2	<u>3</u>	4	<u>5</u>	6	7	8	1,5
1 NoCommande	1								
2 DateCommande	1	1							
3 NoClient	1		1						
4 NomClient	1		1	1					
5 RefProduit					1				
6 Designation					1	1			
7 PrixUnitaire					1		1		
8 QtéCommandée								1	1

Certaines des dépendances fonctionnelles mentionnées dans cette matrice sont « parasites » car elles peuvent être déduites d'autres dépendances fonctionnelles par application des propriétés remarquables telles que la réflexivité ou la transitivité. Il faut donc les éliminer pour obtenir l'ensemble minimal des dépendances fonctionnelles qui représente la même information. Si l'on désigne par F l'ensemble initial des dépendances fonctionnelles, l'ensemble obtenu, noté F[^], après élimination des dépendances parasites est appelé couverture minimale de F et peut être obtenu par application d'un algorithme.

C'est encore un procédé algorithmique qui permet d'aboutir, à partir de la matrice dépouillée de toute dépendance inutile, au modèle conceptuel des données. Ainsi toute propriété identifiante donne naissance à une entité dont le contenu sera formée des propriétés avec lesquelles elle est en dépendance. Les propriétés atteintes par des dépendances fonctionnelles multi-attributs seront intégrées à des associations porteuses (m-n) reliant les entités dont les identifiants sont spécifiées dans la source.

Enfin, les dépendances entre identifiants se matérialiseront par la présence d'une association (1-n). Le modèle ci-dessous résulte de l'application de ces différentes règles. Ce dernier devra alors être complété notamment au niveau des cardinalités minimales afin de prendre en compte toutes les règles de gestion. Les associations non porteuses de type m-n ...



Cette méthode est très lourde dans sa mise en œuvre dès que le nombre de propriétés devient important. De plus, tout lien sémantique devenant une dépendance fonctionnelle, elle est très réductrice d'un point de vue sémantique, et ne permet pas de mettre en évidence des situations telles que plusieurs associations portant des réalités différentes entre deux mêmes entités.

Il semble que la bonne approche de construction d'un modèle conceptuel des données soit un compromis entre la méthode directe, qui laisse une large part à l'intuition et la méthode basée sur l'étude des dépendances fonctionnelles. Quelle que soit la technique utilisée, le modèle doit être vérifié, normalisé et enrichi de toutes les concepts étendus pour représenter le plus fidèlement possible l'univers du discours.

**LA METHODE
MERISE :

LE MODELE
LOGIQUE
DES DONNEES**

LE MODELE LOGIQUE DES DONNEES

I.	LES REGLES DE PASSAGE DU MCD AU MODELE RELATIONNEL	60
1.1)	<i>Les règles</i>	60
	<i>Si l'on reprend l'exercice 2:</i>	63
1.2)	<i>Les concepts étendus</i>	63
a)	<i>Table sur-type et disparition des sous-types</i>	63
b)	<i>Table sous-types et disparition du sur-type</i>	63
c)	<i>Table sur-type et sous-types</i>	64
II.	CREATION D'UNE BD RELATIONNELLES PAR LE LANGAGE SQL.....	65
2.1)	<i>Déclaration des champs d'une table</i>	67
2.2)	<i>Définition des contraintes d'intégrité</i>	68
a)	<i>L'intégrité de relation</i>	69
b)	<i>L'intégrité de valorisation</i>	69
c)	<i>L'intégrité de domaine</i>	70
2.2-4	<i>L'intégrité référentielle</i>	71
	<i>Auto-évaluation n° 3: Le modèle logique des données</i>	72

METHODOLOGIE – CNAM ANGOULEME 2000-2001

LE MODELE LOGIQUE DES DONNEES

La description conceptuelle a permis de représenter le plus fidèlement possible les réalités de l'univers à informatiser. Mais cette représentation ne peut pas être directement manipulée et acceptée par un système informatique. Il est donc nécessaire de passer du niveau conceptuel à second un niveau plus proche des capacités des systèmes informatiques. Ce niveau, appelé niveau logique, consiste à choisir l'un des trois modèles suivants :

- modèle hiérarchique (années 80),
- modèle réseau,
- ou modèle relationnel

Chacun de ces modèles repose sur des techniques d'organisation des données particulières que des logiciels seront capables de gérer. *Par exemple, dans le modèle relationnel l'unique structure d'accueil des données est la relation qui peut être assimilée à un tableau de la forme suivante :*

<i>Propriété1</i>	<i>Propriété2</i>	...	<i>Propriété n</i>
<i>Valeur1</i>	<i>Valeur2</i>	...	
<i>Valeur1'</i>	<i>Valeur2'</i>	...	

Des SGBD relationnels, tels que ORACLE ou SQL Server, permettent à un utilisateur d'exploiter les données ainsi mémorisées.

Dans ce chapitre, seul le modèle relationnel sera étudié. Il est conseillé de se référer au cours de bases de données afin de revoir tous les concepts qui se rapportent à ce modèle et qui ne sont pas rappelés dans le présent chapitre.

I. Les règles de passage du MCD au modèle relationnel

Ces règles sont de type algorithmique et peuvent donc être mises en œuvre par des outils de génie logiciel (*PowerAMC par exemple*). La traduction des concepts de base du modèle conceptuel est régie par les trois règles suivantes :

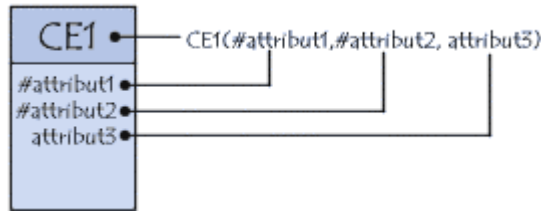
1.1) Les règles

- **Toute entité devient une relation.** L'identifiant de l'entité devient clé primaire de la relation.

E1
<u>P1</u>
P2
P3

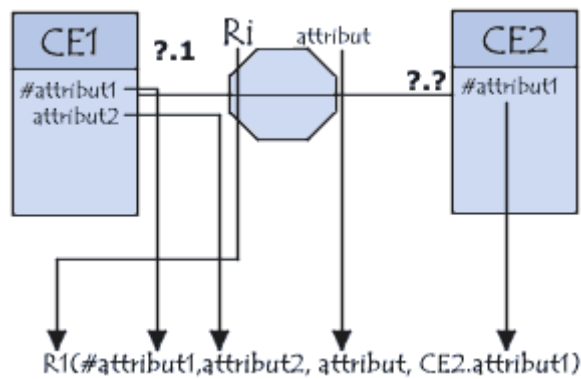
E1(P1,P2,P3, ...)

C'est à dire que chaque classe d'entité du modèle conceptuel devient une table dans le modèle logique. Les identifiants de la classe d'entité sont appelé *clés de la table*, tandis que les attributs standards deviennent des attributs de la table, c'est-à-dire des colonnes.



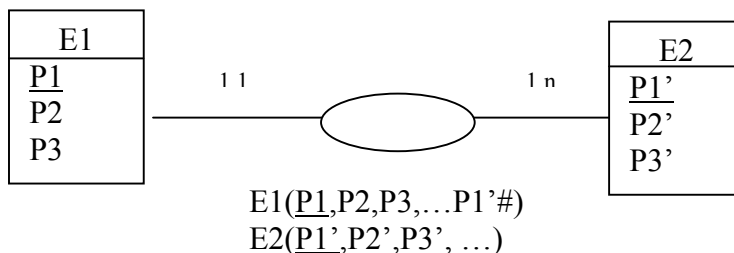
Le passage du modèle conceptuel au modèle logique au niveau des classes de relation se fait selon les cardinalités des classes d'entité participant à la relation:

- si une des classes d'entités possède une cardinalité faible: la table aura comme attributs, les attributs de la classe ayant une cardinalité faible, puis le (ou les) attribut(s) de relation et enfin les attributs de la seconde classe précédé du nom de la classe
- si les deux classes d'entités possèdent une cardinalité forte: la table aura comme attributs, les attributs des deux classes de relation précédés des noms des classes respectives, puis le (ou les) attribut(s) de relation

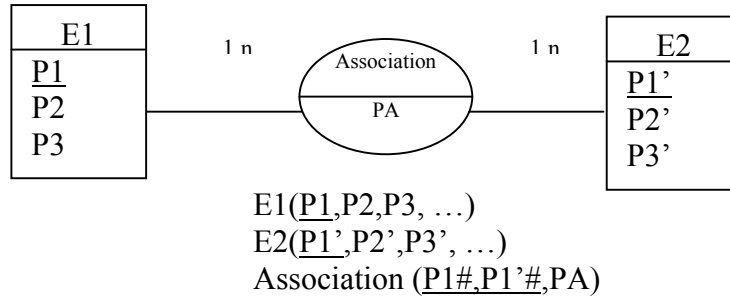


- **Toute association binaire de type (1-n) ou (1-1)** est caractérisée par l'existence d'une dépendance fonctionnelle entre l'identifiant de l'entité reliée par le segment portant la cardinalité 1,1 ou 0,1 et l'autre entité. Dans le schéma ci-dessous on a la dépendance fonctionnelle suivante : $P1 \rightarrow P1'$

Une telle association entraîne l'intégration de l'identifiant de l'entité but de la dépendance fonctionnelle dans la relation associée à l'entité source de la dépendance fonctionnelle. La propriété ainsi dupliquée devient clé étrangère dans la relation source et est marquée par un #. Une clé étrangère dans une relation est une propriété qui est clé primaire dans une autre relation.

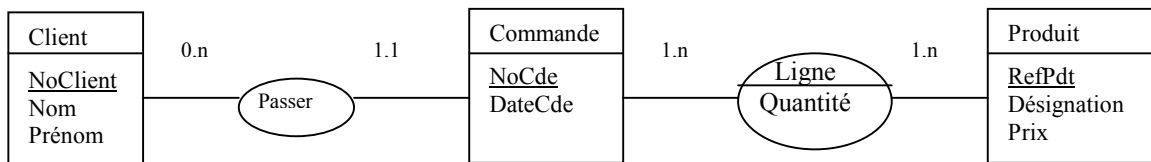


- **Toute association de type (m-n) devient une relation qui hérite** des identifiants des entités participants à la relation. Si l'association est porteuse, la relation sera complétée par la liste des propriétés portées.



Remarque : dans la relation nommée « Association », P1 et P1' sont suffixées d'un # (elles sont clés étrangères) car elles sont clés primaires respectivement dans les relations E1 et E2.

L'exemple ci-dessous illustre l'application de ces trois règles sur le modèle conceptuel classique de gestion des commandes suivant :



Modèle relationnel textuel :

Client(NoClient, Nom, Prénom)
Commande(NoCde, DateCde, NoClient#)
Produit(RefPdt, Désignation, Prix)
Ligne(NoCde#, RefPdt#, Qté)

<u>NoClient</u>	Nom	Prénom
1	Lassus	Annick
2	Mundubeltz	Armelle
3	Chaulet	Bernadette

<u>NoCde</u>	DateCde	NoClient#
100	14/04/2001	2
101	14/04/2001	1

<u>RefPdt</u>	Désignation	Prix
VE45	Vélo	1500 F
VE32	Kit 2 roues arrières	300 F
VE21	Kit éclairage	150 F

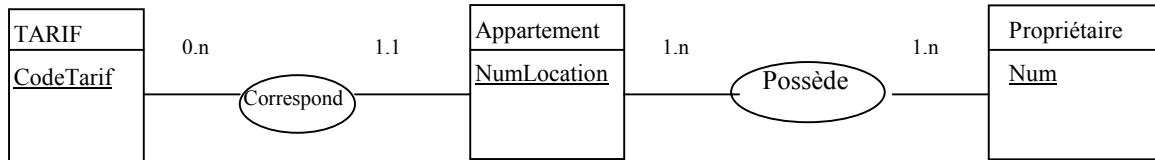
<u>NoCde#</u>	<u>RefPdt#</u>	Qté
100	VE45	1
100	VE32	1
101	VE21	2

On peut remarquer, que dans le modèle relationnel textuel les cardinalités minimales portées par le modèle conceptuel sont perdues. Celles-ci seront en fait représentées au moyen des contraintes de valorisation étudiées au paragraphe 2.2.

➤ **Traduction d'une classe d'agrégation**

Dans le cas de la présence d'une classe d'agrégation, la classe d'entité agrégée a comme attributs supplémentaires les attributs de la classe d'entité agrégeante

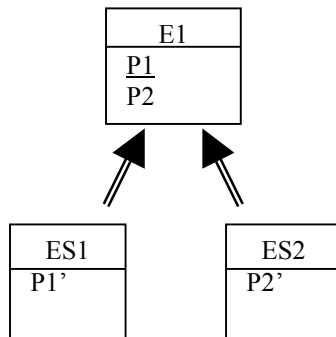
Si l'on reprend l'exercice 2:



Tarif (CodeTarif, PrixSemHS, PrixSemBS)
 Appartement (NumLocation, ..., CodeTarif#, Num#)
 Propriétaire (Num, Nom, ...)

1.2) Les concepts étendus

En ce qui concerne les concepts étendus, mis à part la notion d'identifiant relatif, leur implantation en relationnel n'est pas directement réalisable, car il est impossible de représenter les contraintes ensemblistes. Il faudra donc mettre en place, au niveau des traitements, des dispositifs pour garantir toutes les contraintes qu'ils expriment. La suite de ce paragraphe présente trois possibilités de traduction du concept d'héritage rappelé par le schéma ci-dessous :



a) Table sur-type et disparition des sous-types

Première possibilité : intégration des sous-types dans la relation sur-type (les sous-types disparaissent). Avec un tel principe les propriétés spécifiques à chacun des sous-types ne seront pas valorisées pour certaines occurrences de la relation sur-type.

E1 (P1, P2, P1', P2')

b) Table sous-types et disparition du sur-type

Seconde possibilité : intégration des propriétés figurant dans le sur-type dans tous les sous-types (le sur-type disparaît). Cette solution entraîne une redondance importante des données du sur-type si il n'y a pas exclusion entre les sous-types.

ES1 (P1, P2, P1')
 ES2 (P1, P2, P2')

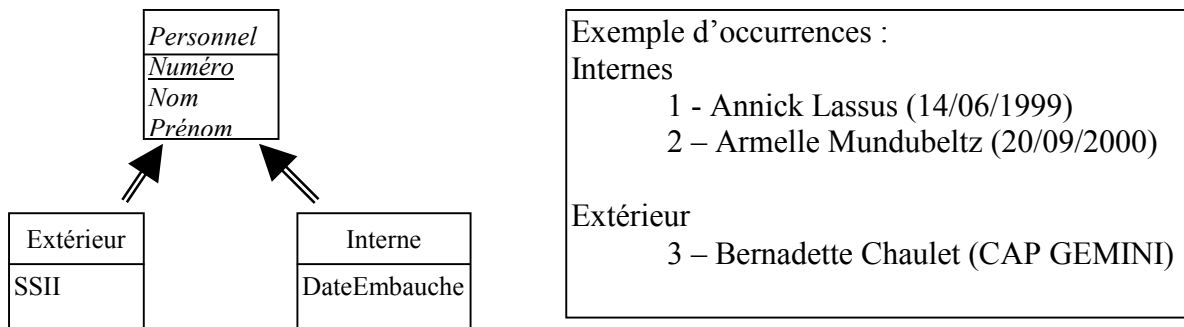
c) Table sur-type et sous-types

Troisième et dernière possibilité : conservation de l'entité sur-type et des entités sous-types. Dans chacune des relations sous-types, l'identifiant de l'entité sur-type est intégré. Il est à la fois clé primaire de la relation et clé étrangère par rapport à l'entité sur-type.

- E1 (P1, P2)
- ES1 (P1#, P1')
- ES2 (P1#, P2')

Il est important de noter que quelque soit la solution adoptée, toute la puissance portée par le concept d'héritage est perdue dans le modèle relationnel.

L'exemple ci-dessous illustre ces trois possibilités pour le modèle conceptuel décrivant la composition du service informatique de la MAAF.



Première possibilité : *PERSONNEL* (Numéro, Nom, Prénom, SSII, DateEmbauche)

<i>PERSONNEL</i>				
<i>Numéro</i>	<i>Nom</i>	<i>Prénom</i>	<i>SSII</i>	<i>DateEmbauche</i>
1	Lassus	Annick		14/06/1999
2	Mundubeltz	Armelle		20/09/2000
3	Chaulet	Bernadette	CAP GEMINI	

Seconde possibilité : *EXTERIEUR*(Numéro, Nom, Prénom, SSII)
INTERNE (Numéro, Nom, Prénom, DateEmbauche)

<i>EXTERIEUR</i>			
<i>Numéro</i>	<i>Nom</i>	<i>Prénom</i>	<i>SSII</i>
3	Chaulet	Bernadette	CAP ..

<i>INTERNE</i>			
<i>Numéro</i>	<i>Nom</i>	<i>Prénom</i>	<i>DateEmbauche</i>
1	Lassus	Annick	14/06/1999
2	Mundubeltz	Armelle	20/09/2000

Troisième possibilité : *PERSONNEL* (Numéro, Nom, Prénom)
EXTERIEUR (Numéro#, SSII)
INTERNE (Numéro#, DateEmbauche)

EXTERIEUR

<i>Numéro</i>	<i>SSII</i>
3	CAP..

INTERNE

<i>Numéro</i>	<i>DateEmbauche</i>
1	14/06/1999
2	20/09/2000

PERSONNEL

<i>Numéro</i>	<i>Nom</i>	<i>Prénom</i>
1	Lassus	Annick
2	Mundubeltz	Armelle
3	Chaulet	Bernadette

II. Création d'une BD relationnelles par le langage SQL

L'objectif de ce paragraphe est d'apprendre à transmettre à un S.G.B.D. R. (Système de Gestion de Bases de Données Relationnelles) la structure d'une base de données relationnelle. Cette déclaration consiste :

- à **indiquer les champs** (encore appelés colonnes ou attributs) qui composent chacune des tables du modèle relationnel. Dans cette première étape, la définition d'un champ se limitera à la donnée de son nom et de son type ; ces deux informations étant issues du dictionnaire des données établi lors de la phase d'analyse conceptuelle.
- à **implémenter des contraintes d'intégrité** destinées à garantir la cohérence des données mémorisées ; dans cette seconde étape seront mises en place (entre autres) les notions de clé primaire et de clé étrangère représentées sur le modèle relationnel textuel respectivement par un soulignement et un #.

Considérons par exemple le schéma relationnel suivant :

COMMANDE(NoCommande, DateCommande, DateLivraison, NoClient#)
 CLIENT (NoClient, NomClient, TypeClient)

Rubriques	Type
NoCommande	Entier
DateCommande	Date
DateLivraison	Date
NoClient	Entier
NomClient	Chaîne de 25 caractères max
TypeClient	Chaîne de 15 caractères max, 2 valeurs possibles : PARTICULIER ou PROFESSIONNEL

La première étape de la déclaration de cette base consiste à transmettre au SGBD R les informations suivantes :

Table CLIENT		Table COMMANDE	
NoClient	Entier	NoCommande	Entier
NomClient	Chaîne de 25 cars max	DateCommande	Date
TypeClient	Chaîne de 15 cars max	DateLivraison	Date
		NoClient	Entier

Si on limitait la création de cette base à cette étape, le SGBD R pourrait accepter les lignes suivantes :

CLIENT			COMMANDE			
NoClient	NomClient	TypeClient	N°Commande	DateCommande	DateLivraison	NoClient
1		PROFESSIONNEL	101	15/12/1999		3
2	DURAND	PARTICULIER	102	17/12/1999	16/12/1999	1
2	DUPONT	ARTISAN	103	17/12/1999	22/12/1999	1

Or, seule la ligne associée à la commande 103 est acceptable. Pour empêcher toutes ces incohérences, il est nécessaire d'enrichir la définition de la base par la mise en place des contraintes d'intégrité suivantes :

<i>Contraintes</i>	<i>Conséquences</i>
Intégrité de relation (encore appelée intégrité d'entité) sur les champs NoClient de la table CLIENT et NoCommande de la table COMMANDE	Cette contrainte garantit des valeurs uniques et obligatoires pour les champs NoClient et NoCommande. Elle permet donc d'implanter la notion de clé primaire.
Intégrité de domaine : - sur le champ TypeClient avec la règle : = « PARTICULIER » ou « PROFESSIONNEL » - sur la table COMMANDE avec la règle : DateLivraison > DateCommande	Pour le champ TypeClient, seules les valeurs « PARTICULIER » ou « PROFESSIONNEL » seront tolérées. Pour la table COMMANDE, seules les lignes présentant une date de livraison supérieure à la date de commande seront tolérées
Intégrité de valorisation sur les champs NomClient et TypeClient de la table CLIENT et sur les champs DateCommande et NoClient de la table COMMANDE	Cette contrainte impose la présence d'une valeur pour les champs cités.
Intégrité référentielle sur le champ NoClient de la table COMMANDE	Cette contrainte impose que toute valeur du champ NoClient de la table COMMANDE appartienne à l'ensemble des valeurs du champ NoClient de la table CLIENT.

Le moyen le plus normalisé pour transmettre la structure d'une base à un SGBD R est l'utilisation du **langage S.Q.L.** (Structured Query Language). Certains SGBD R (comme ACCESS) sont dotés d'une interface graphique qui permettent de créer une base en s'affranchissant des contraintes syntaxiques du langage.

S.Q.L. est un langage de requête normalisé utilisé pour la gestion d'une base de données relationnelle ; le terme de gestion est très large puisqu'il englobe aussi bien l'administration de la base de données que son exploitation. En fonction de leur finalité, les ordres S.Q.L. sont regroupés dans différentes catégories, les plus connues étant le **L.D.D.** (Langage de Définition des Données) et le **L.M.D.** (Langage de Manipulation des Données). Ainsi pour une table, les ordres permettant de créer, de modifier ou de détruire sa structure appartiendront au L.D.D. alors que les ordres permettant d'interroger ou de modifier son contenu (ajout de nouvelles lignes, destruction de lignes existantes, ...) seront catalogués dans le L.M.D.

Enfin, malgré le fait que le langage S.Q.L. soit normalisé, les syntaxes des instructions diffèrent sensiblement d'un S.G.B.D. à l'autre. Les paragraphes qui suivent présentent les syntaxes reconnues par le SGBD **ORACLE**.

2.1) Déclaration des champs d'une table

L'ordre S.Q.L. « Create table » permet de créer une table. Limitée à la définition des champs de la table, la syntaxe de cet ordre a la forme suivante :

```
CREATE TABLE NomTable
    (NomChamp Type,
    NomChamp Type,
    ..... )
```

Il existe différents types pouvant être répertoriés dans trois catégories :

- Les types caractères
 - **CHAR(n)** : chaîne de caractères de longueur fixe avec n compris dans l'intervalle [1 ; 16383]
 - **VARCHAR(n)** : chaîne de longueur variable ; n représente la longueur maximale de la chaîne
- Les types numériques
 - **SMALLINT** : entier signé sur 2 octets compris entre - 32768 et 32 767
 - **INTEGER** : entier compris entre - 2^{E-31} et 2^{E 31}
 - **NUMBER(n[,d])** : réel à n positions : chiffres, signe et marque décimale ; d représente le nombre de chiffres après la virgule et est facultatif
- Les types dates
 - **DATE** : ce type intègre la date et l'heure

Les types BOOLEEN et Numéro Automatique ne sont pas gérés par la plupart des SGBD R.

Les ordres S.Q.L. permettant de définir les champs des tables CLIENT et COMMANDE sont fournis ci-dessous :

```
CREATE TABLE Client
  (NoClient Integer,
  NomClient Varchar(25),
  TypeClient VarChar(15) )

CREATE TABLE Commande
  (NoCommande Integer,
  DateCommande Date,
  DateLivraison Date,
  NoClient Integer)
```

2.2) Définition des contraintes d'intégrité

En langage S.Q.L., la définition des contraintes d'intégrité consiste à enrichir l'ordre Create table par des clauses Constraint NomDeContrainte TypeContrainte.

L'ordre de création de la table Client aura la forme suivante :

```
CREATE TABLE Client
  (NoClient Integer
  NomClient VARCHAR(25)
  CONSTRAINT NN_NomClient NOT NULL,
  TypeClient VARCHAR(15)
  CONSTRAINT NN_TypeClient NOT NULL
  CONSTRAINT PK_Client PRIMARY KEY (NoClient),
  CONSTRAINT CK_Client CHECK (TypeClient = « PARTICULIER » or
  TypeClient = « PROFESSIONNEL)
```

Contrainte de domaine attachée au champ TypeClient

Contrainte de valorisation attachée au champ NomClient

Contrainte d'entité attachée au champ NoClient

Pour nommer les contraintes, il est d'usage de suivre la norme indiquée ci-dessous ; dans une même base de données on ne peut pas avoir deux contraintes qui portent le même nom.

AbréviationTypeContrainte_NomTable

Les abréviations les plus fréquemment utilisées pour chaque type de contraintes sont :

PK Intégrité d'entité (Primary Key : clé primaire)

CK Intégrité de domaine (Check : contrôle)

FK Intégrité référentielle (Foreign Key : clé étrangère)

PK_Client est le nom associé à la contrainte d'entité attachée de la table client.

Remarque : sous le logiciel PowerAMC, il est possible de définir la façon de nommer les contraintes dans des fichiers systèmes ayant l'extension « .def » (Access95.def par exemple).

a) L'intégrité de relation

Attachée à un (ou plusieurs) champ(s) d'une table, cette contrainte garantit des valeurs uniques pour ce (ou cette combinaison de) champ(s). Elle permet donc d'implanter la notion de **clé primaire**.

Syntaxe : **CONSTRAINT PK_NomTable PRIMARY KEY (ListesDesChampsClés)**

L'exemple suivant illustre cette contrainte pour la définition de la clé primaire NoClient dans la table Client.

```
CREATE TABLE Client
(NoClient Integer
NomClient Varchar(25) NOT NULL
TypeClient Varchar(15) NOT NULL
CONSTRAINT PK_Client PRIMARY KEY (NoClient)
)
```

La clé de la table Client est NoClient

b) L'intégrité de valorisation

Attachée à un champ cette contrainte permet d'imposer la saisie d'une valeur pour ce champ. Il n'est pas utile de définir cette contrainte pour un champ clé primaire car l'intégrité d'entité inclut l'intégrité de valorisation.

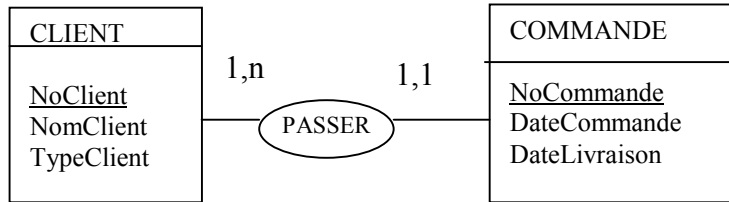
Syntaxe : **NomChamp Type NOT NULL**

Dans la table Client, pour imposer la saisie d'une valeur dans les champs NomClient et TypeClient on écrira l'ordre SQL suivant :

```
CREATE TABLE Client
(NoClient Integer
NomClient Varchar(25) NOT NULL
TypeClient Varchar(15) NOT NULL
CONSTRAINT PK_Client PRIMARY KEY (NoClient)
)
```

Le Nom du client ainsi que le Type du client devront être renseignés

L'intégrité de valorisation définie sur un champ clé étrangère permet de représenter la cardinalité minimale 1 d'une association de type (1-1) ou (1-n). *Ainsi pour indiquer qu'à toute commande est associée un client (ce qui revient à traduire la cardinalité minimale indiquée en gras sur le modèle conceptuel ci-dessous),*



On définit une contrainte NOT NULL sur le champ NoClient de la table COMMANDE :

```

CREATE TABLE Commande
(NoCommande Integer
 DateCommande Date NOT NULL,
 DateLivraison Date,
 NoClient Integer NOT NULL,
 CONSTRAINT PK_Commande PRIMARY KEY (NoCommande),
 )
    
```

c) L'intégrité de domaine

Cette contrainte permet d'exprimer une condition qui porte sur un ou plusieurs champs de la table. Syntaxe : **CONSTRAINT CK_NomTable CHECK** (Condition portant sur un ou plusieurs champs)

Dans la table Client pour attacher la contrainte de domaine : = PARTICULIER ou = PROFESSIONNEL au champ TypeClient on écrira l'ordre suivant :

```

CREATE TABLE Client
(NoClient Integer
 NomClient VARCHAR2(25) NOT NULL
 TypeClient VARCHAR2(15) NOT NULL
 CONSTRAINT PK_Client PRIMARY KEY (NoClient),
 CONSTRAINT CK_Client CHECK (TypeClient = « PARTICULIER » OR TypeClient =
 « PROFESSIONNEL »)
    
```

Le TypeClient sera renseigné et égal à PROFESSIONNEL ou PARTICULIER

Dans la table Commande pour exprimer la contrainte garantissant la cohérence entre la date de commande et la date de livraison on écrira l'ordre suivant :

```

CREATE TABLE Commande
(NoCommande Integer
 DateCommande Date NOT NULL
 DateLivraison Date,
 NoClient Integer,
 CONSTRAINT PK_Commande PRIMARY KEY (NoCommande)
 CONSTRAINT CK_Commande CHECK
 (DateLivraison IS NULL
 OR (DateLivraison IS NOT NULL AND DateCommande < DateLivraison))
    
```

Soit la Date Livraison est nulle, soit elle ne l'est pas et dans ce cas elle doit être supérieure à la DateCommande

2.2-4 L'intégrité référentielle

Attachée à un champ "clé étrangère" cette contrainte garantit que toute valeur prise par ce champ appartient à l'ensemble des valeurs de la clé primaire.

Syntaxe : **CONSTRAINT FK_NomChamp FOREIGN KEY (NomChamp) REFERENCES**
NomTable (NomChamp) [ON DELETE CASCADE]

L'option ON DELETE CASCADE permet de supprimer une ligne d'une table ainsi que tous les lignes liées dans une autre table.

Si l'on suppose que le contenu des tables CLIENT et COMMANDE est limité aux lignes suivantes :

CLIENT			COMMANDE			
NoClient	NomClient	TypeClient	N°Commande	DateCommande	DateLivraison	NoClient
1	DUBOIS	PROFESSIONNEL	101	15/12/1999		3
2	DELAGE	PARTICULIER	102	17/12/1999	18/12/1999	1
3	DUPONT	PROFESSIONNEL	103	17/12/1999	22/12/1999	1

La suppression du client Numéro 1 dans la table CLIENT pourrait :

- soit entraîner la suppression des commandes 102 et 103 dans la table COMMANDE si l'option ON DELETE CASCADE est mentionnée,
- soit entraîner un refus de suppression de ce client si l'option ON DELETE CASCADE n'est pas mentionnée ; dans ce cas il faut préalablement supprimer les commandes 102 et 103 dans la table COMMANDE pour ensuite pouvoir supprimer le client Numéro 1.

Dans la table COMMANDE pour définir le champ NoClient en tant que clé étrangère on écrira l'ordre SQL suivant :

```
CREATE TABLE Commande
  (NoCommande Integer
  DateCommande Date
    CONSTRAINT NN_DateCommande NOT NULL,
  DateLivraison Date,
  NoClient Integer NOT NULL,
  CONSTRAINT PK_Commande PRIMARY KEY (NoCommande),
  CONSTRAINT CK_Commande CHECK
    (DateLivraison IS NULL
    OR (DateLivraison IS NOT NULL AND DateCommande < DateLivraison),
  CONSTRAINT FK_Commande FOREIGN KEY (NoClient) REFERENCES
  CLIENT(N°Client) ON DELETE CASCADE,
  )
```

Le N°Client est défini comme clé étrangère de la table Commande. Elle fait référence à la clé primaire de la table CLIENT

Auto-évaluation n° 3: Le modèle logique des données

La LEPOL (Ligue de Protection des Oiseaux en Limousin) est une association qui a comme objectif la protection et l'étude des oiseaux en Limousin. Elle lance des campagnes de sensibilisation et édite des revues consacrées à la défense des oiseaux. En début d'année, la ligue envoie à chaque abonné une demande de renouvellement de cotisation (annexe 1). Lors du retour du bulletin, elle souhaite enregistrer dans sa base d'information toutes les informations qui y sont consignées. Cette base d'information devra en outre mémoriser les revues auxquelles sont abonnés les membres. Un état tel que celui présenté à l'annexe 2 pourra ainsi être édité. Chaque revue est identifiée par un numéro et est caractérisé par un prix ainsi qu'un nombre de numéros par an.

Enfin, pour financer les opérations qu'elle mène, la LEPOL recueille des fonds auprès des membres de l'association. Elle leur envoie plusieurs fois par an des bulletins de souscription leur permettant de verser une somme dont le montant est à leur convenance (annexe 3). Il arrive qu'un adhérent effectue dans la même année plusieurs versements pour la même action. La base d'information devra conserver l'ensemble de ces souscriptions.

Annexe 1 (bulletin d'adhésion)

LiguE de Protection des Oiseaux en Limousin
12, rue Sébastien Faure
87 000 Limoges

BULLETIN D'ADHESION

Nom Prénom :
Adresse :
.....

demande a être accepté comme membre actif : No adhérent :
Fait à le
Signature :

Ci-joint un chèque bancaire, chèque postal, mandat-lettre à l'ordre de LEPOL (rayer la mention inutile)

Tarifs 2001

No de cotisation	Libellé	Prix	Choix (cocher la case)	Nom - prénom ou numéro du conjoint
1	Ordinaire	150 F		
2	Conjoint	100 F		
3	Scolaire ou personne en difficulté	70 F		
4	Bienfaiteur	500 F		

Remarque :

- Seuls les anciens membres inscrivent leur numéro d'adhérent
- Dans un couple d'adhérents, chaque personne remplit un bulletin d'adhésion et reçoit un numéro d'adhérent. Une personne paye la cotisation ordinaire, l'autre la cotisation conjoint. Cette dernière doit alors préciser dans la dernière colonne du tableau les nom et prénom ou le numéro de son conjoint.
- Chaque membre ne cotise qu'un seul type de cotisation

Annexe 2 (état récapitulatif des abonnés)

N° membre	Nom	Prénom	Adresse	Ville	Nom revue	abonnement termineéau n°
158	Moine	Olivier	14, rue des rapaces	87 000 Limoges	Oiseau Mag	7
					La caille	78
					La chevêche	5
196	Bellon	Elodie	12, impasse des roseaux	19100 Brive	La caille	82
....						

Annexe 3 (demande de souscription)

LiguE de Protection des Oiseaux en Limousin
12, rue Sébastien Faure
87 000 Limoges

BULLETIN D'ADHESION

Nom Prénom :
Adresse :
N° adhérent :

souhaite que la somme versée d'un montant de soit affectée à l'une des actions suivantes (entourer l'action)

- Protection des oiseaux
- Protection des milieux humides
- Achat de milieux sensibles
- Acquisition de matériels

Ci-joint un chèque bancaire, chèque postal, mandat-lettre à l'ordre de LEPOL (rayer la mention inutile)

Fait à le
Signature :

1. Construire le modèle conceptuel des données correspondant au système décrit

2. Ecrire le modèle relationnel textuel associé

3. Quelles étapes faudrait-il suivre sous l'atelier de génie logiciel PowerAMC pour créer la base de données

4. On souhaite distinguer parmi les membres de l'association, les membres fondateurs des autres membres. Seuls les membres fondateurs sont dispensés du paiement d'une cotisation. Proposer une modélisation permettant de prendre en compte cette réalité

LA METHODE MERISE :

**Mise en œuvre
avec l'A.G.L.
AMC*DESIGNOR**

MISE EN ŒUVRE AVEC POWERAMC

I.	SAISIE DU MODELE CONCEPTUEL DES DONNEES	77
II.	GENERATION DU MODELE PHYSIQUE	79
III.	GENERATION DE LA BASE DE DONNEES.....	80

METHODOLOGIE – CNAM ANGOULEME 2000-2001

MISE EN ŒUVRE AVEC POWERAMC

PowerAMC (éditeur :Sybase) créé par la société SDP en 1989, est un outil graphique (sous Windows) de conception de systèmes d'information. Il permet de mettre en œuvre la méthode Merise en facilitant les étapes de création de bases de données et de développement d'applications. Il se compose de différents modules, les principaux étant :

- **le module « Données »** qui offre les fonctions nécessaires à l'élaboration du Modèle Conceptuel de Données et du Modèle Physique de Données Merise. Ce dernier modèle correspond à un schéma relationnel destiné à être géré par un SGBD Relationnel précis.
- **le module « Traitements »** utile pour représenter les différents modèles de traitements Merise.
- **le module « Développeur »** capable de générer des prototypes d'applications pour PowerBuilder ou VisualBasic. Un tel module favorise la mise en œuvre de démarche de projets de type RAD (Rapid Application Development).

Seule l'étude du module « Données » sera abordée dans ce cours. L'utilisation la plus classique de celui-ci consiste, à partir d'un MCD, à créer automatiquement la base de données relationnelles. Pour ce faire, il suffit de suivre les trois étapes suivantes...

I. Saisie du modèle conceptuel des données

(tableaux extraits de l'ouvrage AMC*Designor et Merise)

Entité

Symbole	Définition	Exemple
<div style="border: 1px solid black; padding: 5px; width: fit-content;">Ent_1</div>	objet de gestion pour lequel on souhaite conserver des informations	<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 5px; width: 40%;"> Client No client Nom client Prénom client </div> <div style="border: 1px solid black; padding: 5px; width: 40%;"> Commande No commande Date commande </div> </div>

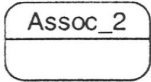
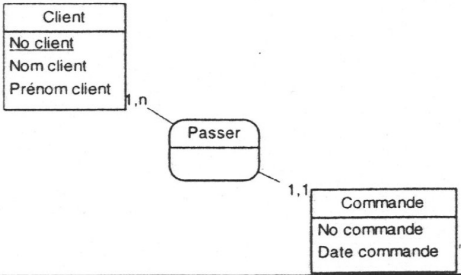
Propriété

Symbole	Définition	Exemple
aucun	information rattachée à une entité	nom client, prénom client, date de commande...

Propriété identifiante

Symbole	Définition	Exemple
propriété soulignée	propriété identifiant tout élément d'entité	

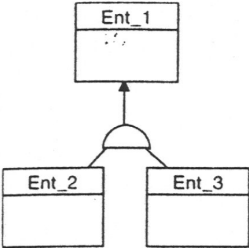
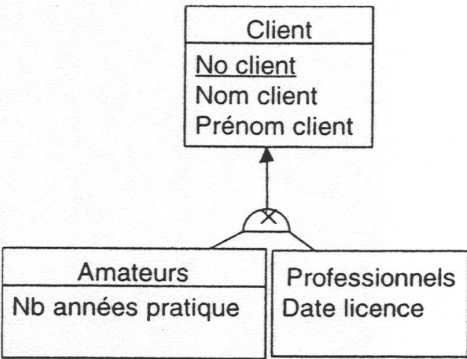
Association

Symbole	Définition	Exemple
	traduit l'existence d'une relation entre entités	

Cardinalités

Symbole	Définition	Exemple
couples de valeurs 1,1 1,N 0,1 0,N	nombre MINIMAL et MAXIMAL de fois qu'un élément de l'entité est associé aux éléments de l'association	

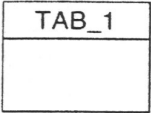
Héritage

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

II. Génération du modèle physique

Il est destiné à être implanté sur un SGBD R donné (Option « Générer le modèle physique » du menu « Dictionnaire »)

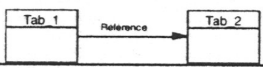
Table

Symbole	Origine conceptuelle	Définition
	entité	Structure physique, composée de lignes et de colonnes, permettant de stocker des informations.

Domaine

Symbole	Origine conceptuelle	Définition
aucun	domaine	type de données générique.

Référence

Symbole	Origine conceptuelle	Définition
	association 1-1 ou 1-N	relation entre deux tables.

Clé primaire

Symbole	Origine conceptuelle	Définition
colonne typée PK colonne soulignée dans la table	identifiant de l'entité	colonne ou groupe de colonnes prenant une valeur unique dans la table et déterminant une seule ligne dans la table.

Clé étrangère

Symbole	Origine conceptuelle	Définition
colonne typée FK	se déduit de l'association	colonne ou groupes de colonnes correspondant à la clé primaire de la table père. Définit ainsi une jointure entre tables.

III. Génération de la base de données

On la génère par production d'un script SQL ou par création directe (Option « Générer la base de données » du menu « SGBD »)

Voyons l'exemple de la création directe d'une base Access concernant le modèle sur les clients, produits et les commandes. Le logiciel exécute les commandes SQL comme ceci:

```

create table CLIENT
( NOCLIENT      Long   not null,
  NOM            Text(25) null,
  PRENOM        Text(25) null,
  constraint PK_CLIENT primary key (NOCLIENT)
)

create table PRODUIT
( REFPDT        Long     not null,
  DESIGNATION   Text(25) null,
  PRIX          Currency not null,
  constraint PK_PRODUIT primary key (REFPDT)
)

create table COMMANDE
( NOCDE         Long     not null,
  NOCLIENT     Long     not null,
  DATECDE      DateTime null,
  constraint PK_COMMANDE primary key (NOCDE),
  constraint FK_COMMANDE_PASSER_CLIENT foreign key (NOCLIENT)
  references CLIENT (NOCLIENT)
)

create index PASSER_FK on COMMANDE (NOCLIENT asc)

create table LIGNE
( NOCDE         Long     not null,
  REFPDT        Long     not null,
  QTE           Long     null,
  constraint PK_LIGNE primary key (NOCDE, REFPDT),
  constraint FK_LIGNE_LIEN_18_COMMANDE foreign key (NOCDE)
  references COMMANDE (NOCDE),
  constraint FK_LIGNE_LIEN_19_PRODUIIT foreign key (REFPDT)
  references PRODUIT (REFPDT)
)

create index LIEN_18_FK on LIGNE (NOCDE asc)

create index LIEN_19_FK on LIGNE (REFPDT asc)

```


LA METHODE MERISE :

Description dynamique du système d'information

DESCRIPTION DYNAMIQUE DU S.I.

I.	LES CONCEPTS DE BASE	83
1.1)	<i>L'acteur</i>	83
1.2)	<i>L'événement</i>	84
a)	<i>Occurrences d'événements</i>	85
b)	<i>Participation et cardinalité d'un événement</i>	85
1.3)	<i>L'opération</i>	85
1.4)	<i>La règle d'émission</i>	86
1.5)	<i>La synchronisation</i>	86
1.6)	<i>Représentation graphique</i>	88
1.7)	<i>Extrait du MCT de la gestion des inscriptions</i>	89
II.	FONCTIONNEMENT D'UN MODELE DYNAMIQUE	90
2.1)	<i>Fonctionnement d'un modèle dynamique</i>	90
2.2)	<i>Règles de vérification du fonctionnement</i>	91
III.	REGLES DE CONSTRUCTION D'UN M.C.T.....	92
IV.	LE MODELE ORGANISATIONNEL DES TRAITEMENTS	92
4.1)	<i>Le tableau des procédures fonctionnelles</i>	92
	<i>Exercice n°7:</i>	93
	<i>Exercice n° 8:</i>	95
	<i>Exercice n° 9 – Synthèse</i>	97

METHODOLOGIE – CNAM ANGOULEME 2000-2001

DESCRIPTION DYNAMIQUE DU S.I.

Comme il a été dit dans le chapitre d'introduction, tout système d'information est composé d'une base d'information et d'un processeur d'information qui représentent respectivement sa statique et sa dynamique. A l'instar du Modèle Conceptuel des Données (MCD) qui schématise les données du système d'information, le Modèle Conceptuel des Traitements (MCT) décrit les traitements et plus précisément toutes les activités découlant des échanges entre le domaine étudié et le monde extérieur. Il exprime donc ce que fait le domaine sans se poser le problème de savoir qui le fait, quand et comment.

I. Les concepts de base

1.1) L'acteur

Un acteur est une personne morale ou physique capable d'émettre ou de recevoir des informations. *Par exemple, l'élève de terminale qui souhaite s'inscrire à un DEUG préparé par la faculté de Bordeaux est un acteur du domaine « Gestion des inscriptions » de cette faculté.* On distingue deux types d'acteurs :

- les acteurs internes qui appartiennent au système d'information étudié. *Pour le domaine cité ci-dessus, le service des inscriptions ou le service comptabilité de la faculté de Bordeaux sont des acteurs internes.*
- les acteurs externes qui n'appartiennent pas au système d'information mais qui sont l'origine ou la destination de flux d'informations reçus ou émanant du système d'information. L'élève de terminale qui effectue une demande de pré-inscription à la faculté de Bordeaux est un exemple d'acteur externe.

Dans le Modèle Conceptuel de Traitements, seuls les acteurs externes sont modélisés ; d'une part parce qu'on ne cherche qu'à formaliser les traitements découlant d'interactions avec l'environnement et que d'autre part, on ne cherche pas à connaître les acteurs internes qui réalisent les traitements en question.

Remarque : pour certains Ateliers de Génie Logiciel (AGL), la notion d'acteur est implicite : ils n'apparaissent donc pas graphiquement dans les différents modèles de traitements produits.

1.2) L'événement

L'événement matérialise un fait, qui en se produisant, doit déclencher une réaction du système. Plus précisément cette notion recouvre deux aspects :

- le fait qui survient et sa perception. *La décision d'un élève de terminale de s'inscrire à la faculté de Bordeaux illustre cet aspect.*
- le compte rendu de cette perception faite auprès du système d'information. *Ainsi, dans le cas de la pré-inscription, c'est le remplissage du dossier qui constitue le compte rendu du souhait de l'élève.*

Seul le second aspect est pris en compte dans la dynamique du système d'information et correspond à la définition d'événement. Du fait de cette restriction l'événement (au sens du modèle conceptuel des traitements) est porteur d'informations qui doivent être obligatoirement digérées par le système d'information sans quoi il ne répondrait pas à ses objectifs.

Parmi les événements, on distingue les événements externes et les événements internes :

- les événements déclencheurs externes sont des événements émis par un acteur externe. *Par exemple le dépôt d'un dossier de pré-inscription est un événement externe déclenché par un futur bachelier souhaitant intégrer un DEUG à la faculté de Bordeaux.*
- les événements internes sont des événements qui surviennent lorsqu'une opération se termine. *Ce peut être par exemple l'acceptation de la pré-inscription après vérification du contenu du dossier.* Un événement interne n'a lieu d'être que si le compte rendu de la fin d'une opération doit être soit suivi d'une nouvelle réaction du système d'information, soit de l'émission d'un message vers l'environnement.

Certains événements externes sont liés au temps. Par exemple, pour déclencher un traitement en début d'année civil, on introduira l'événement « Début d'année ». L'événement « Date actuelle est JJ/MM/AAAA » permettra d'exécuter un traitement à une date donnée.

Dans le MCT, chaque événement est identifié au moyen d'un libellé générique tel que « Dépôt d'un dossier de pré-inscription ». Compte tenu de ce qui vient d'être dit, cet intitulé est très insuffisant pour décrire l'événement car il ne fait pas apparaître les données du compte rendu associé à l'événement. *Par exemple, le dépôt d'un dossier de pré-inscription apporte de nombreuses informations telles que l'état civil de l'élève qui effectue le dépôt, des données sur sa scolarité actuelle, le DEUG qu'il souhaite intégrer, etc.* Dans le cas où le nombre d'informations contenues dans le message associé à l'événement est peu important il est recommandé de les citer en annexe du MCT. Dans le cas contraire, il sera utile de dégager les principales entités figurant dans le compte rendu. Le terme « entité » ne fait pas référence au modèle conceptuel des données ; il est pris dans son sens très général pour désigner tout objet abstrait ou concret caractérisé par un ensemble de propriétés. *Par exemple, l'événement « Dépôt de dossier de pré-inscription » sera détaillé ainsi : « Dépôt du dossier de pré-inscription de l'élève E de la terminale T à la formation F ». Ce texte, qui constitue le message associé à l'événement, fait intervenir trois « entités » : Elève, Terminale et FormationSup.*

Notons enfin que l'intitulé générique étant parfois long, on lui associe, sur le MCT un alias afin de le référencer plus facilement. Les alias seront codés ainsi : « ext » ou « int » pour indiquer le type de l'événement suivi d'un numéro séquentiel. *Par exemple « ext1 » désignera l'événement « Dépôt d'un dossier de pré-inscription ».*

a) Occurrences d'événements

L'occurrence d'un événement correspond à la réalisation effective d'un événement. *Par exemple, le dépôt du dossier de pré-inscription de M. Girard de terminale ES pour le DEUG MASS, constitue une occurrence de l'événement déclencheur « Dépôt du dossier de pré-inscription ».* Deux occurrences d'un même événement peuvent être distinguées soit par des valeurs de propriétés (ou d'entités) différentes, soit si les valeurs sont identiques, par le moment précis où l'événement s'est produit.

La notion d'occurrence d'événements n'est, en général, pas modélisée, par contre la capacité d'un événement, qui est le nombre maximum d'occurrences acceptées par le processeur d'information, et la fréquence d'apparition des occurrences le sont. *Ainsi, on peut fixer à 3000 le nombre maximum d'occurrences acceptées par le processeur pour l'événement « Dépôt du dossier de pré-inscription ».*

b) Participation et cardinalité d'un événement

La participation d'un événement définit le nombre d'occurrences différentes nécessaires au lancement de l'opération. *Dans le processus de gestion des inscriptions, le traitement du dossier déposé doit être déclenché à chaque apparition d'une occurrence de l'événement « Dépôt d'un dossier de pré-inscription ».* La participation de l'événement au traitement est donc égal à 1.

La cardinalité d'un événement est le nombre d'occurrences identiques d'un événement résultat. Si l'on associe à l'événement interne « Carte étudiant éditée » la cardinalité 2, cela impliquera une émission en deux exemplaires de la carte d'étudiant. Si ces deux caractéristiques ne sont pas précisées sur le MCT, elles prennent la valeur 1 par défaut.

1.3) L'opération

La réponse à l'arrivée d'un événement est le déclenchement d'un ensemble de traitements appelé opération. *Le traitement d'enregistrement d'une pré-inscription est une opération déclenchée lors du dépôt de dossier de pré-inscription*

Lors de son exécution une opération ne peut pas être interrompue par l'attente d'un événement externe. L'exécution d'une opération se ramène à l'exécution d'actions élémentaires effectuées sur la base d'informations à partir des données portées par le ou les événement(s) déclencheur(s). Ces actions élémentaires portent sur des occurrences d'entités ou d'associations du modèle conceptuel des données et peuvent appartenir à l'un des quatre types suivants :

- insertion
- la modification
- l'effacement
- recherche

La logique d'enchaînement des actions élémentaires n'est pas toujours séquentielle et peut faire intervenir des structures alternatives (Si .. Alors ... Sinon) ou itératives (Tant que ..., Répéter ..., Pour ...).

Sur le MCT, une opération est identifiée par un libellé et peut être décrite, de manière détaillée, en annexe du MCT, en présentant la logique algorithmique du déclenchement des actions élémentaires. *Par exemple l'opération d'enregistrement du dossier de préinscription pourrait être détaillé ainsi :*

- création d'une occurrence de l'entité « Préinscrit »
- création d'une occurrence de l'association « Demande » (reliant l'entité « Préinscrit » à l'entité « Formation »)
- ...

Il est possible d'associer à une opération une durée qui représente le temps maximal qui lui est alloué pour qu'elle s'exécute.

1.4) La règle d'émission

La production effective d'une ou de plusieurs occurrences d'un événement interne est soumise à une règle d'émission, c'est-à-dire à une proposition logique qui s'applique au contenu de la base d'information après exécution de l'opération. L'événement est produit si la proposition logique est vraie. A l'issue de l'enregistrement d'un dossier de pré-inscription deux cas peuvent se présenter :

- soit le dossier est complet et une occurrence de l'événement « Pré-inscription de l'élève E à la formation F réalisée le JJ/MM/AAAA » est émise
- soit le dossier est incomplet (certaines propriétés du MCD n'ont pas été renseignées) et une occurrence de l'événement « Dossier D Mis en attente le JJ/MM/AAAA » est produite.

Si la plupart des règles d'émission sont basées sur une structure alternative et donne donc lieu à une seule occurrence d'événement interne, certaines peuvent intégrer une structure itérative de type « Pour – Tout » afin de produire n occurrences d'un événement interne. *Par exemple pour envoyer en début d'année des lettres de renouvellement d'adhésion, on introduira la règle d'émission suivante :*

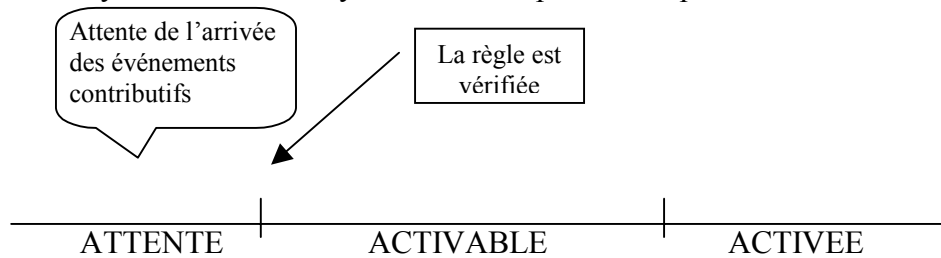
*Pour tout adhérent A enregistré dans la base d'informations
créer une occurrence de int1 (Renouvellement d'adhésion envoyé à A le JJ/MM/AAAA)
Fin pour*

1.5) La synchronisation

La synchronisation d'une opération est composée de deux éléments :

- d'une part la liste des événement (internes ou externes) qui doivent être arrivés avant de déclencher l'opération.
- et d'autre part la règle sous forme d'une proposition logique qui précise de quelle manière les événements participent au déclenchement de l'opération.

Le cycle de vie d'une synchronisation peut être représenté ainsi :



Remarque : on parle de fonctionnement asynchrone lorsque les états ACTIVABLE et ACTIVEE ne font qu'un.

Pour des raisons de lisibilité ce sont les alias des événements participant à la synchronisation qui sont mentionnés, sur le MCT, dans l'expression logique de la synchronisation. *Par exemple la condition ext1 et ext2 signifie que la synchronisation sera activable lorsque :*

- le nombre d'occurrences de l'événement ext1 sera égal à la participation de ext1
- et le nombre d'occurrences de l'événement ext2 sera égal à la participation de ext2

A cette proposition logique sont associées des conditions locales qui permettent de préciser, lorsque plusieurs occurrences d'un événement sont présentes comment choisir celles qui participera effectivement à la synchronisation. Les conditions locales portent obligatoirement sur les valeurs des propriétés ou des entités associées aux messages des événements à synchroniser. Une synchronisation ne peut pas consulter la base d'informations.

Par exemple, pour modéliser le déclenchement de la mise à jour d'un dossier incomplet suite à la réception des pièces manquantes, on introduira une synchronisation admettant en entrée les deux événements suivants :

Événement int2 : « Dossier D Mis en attente le JJ/MM/AAAA »

Événement ext2: « Réception des pièces manquantes du dossier D »

Proposition logique : int2 et ext2

Conditions locales : int2.D = ext2.D

Lorsque l'expression logique de la synchronisation est vérifiée, l'opération est déclenchée et toutes les occurrences d'événements qui ont permis ce déclenchement sont alors consommées par l'opération. Par contre si elle n'est pas vérifiée, les occurrences d'événement restent en attente. Quand une opération est déclenchée par un seul événement, la synchronisation est facultative.

Comme pour les règles d'émission, les conditions locales d'une synchronisation peuvent comporter une structure itérative de type « Pour tout ». Cette possibilité permet de traiter (on dit encore consommer) n occurrences d'un événement contributif à la synchronisation. *La synchronisation définie ci-dessous permet de lancer la suppression de tous les dossiers mis en attente et pour lesquels les pièces manquantes n'ont pas été fournies dans un délai de 8 jours.*

Événement int2 : « Dossier D mis en attente le JJ/MM/AAAA »

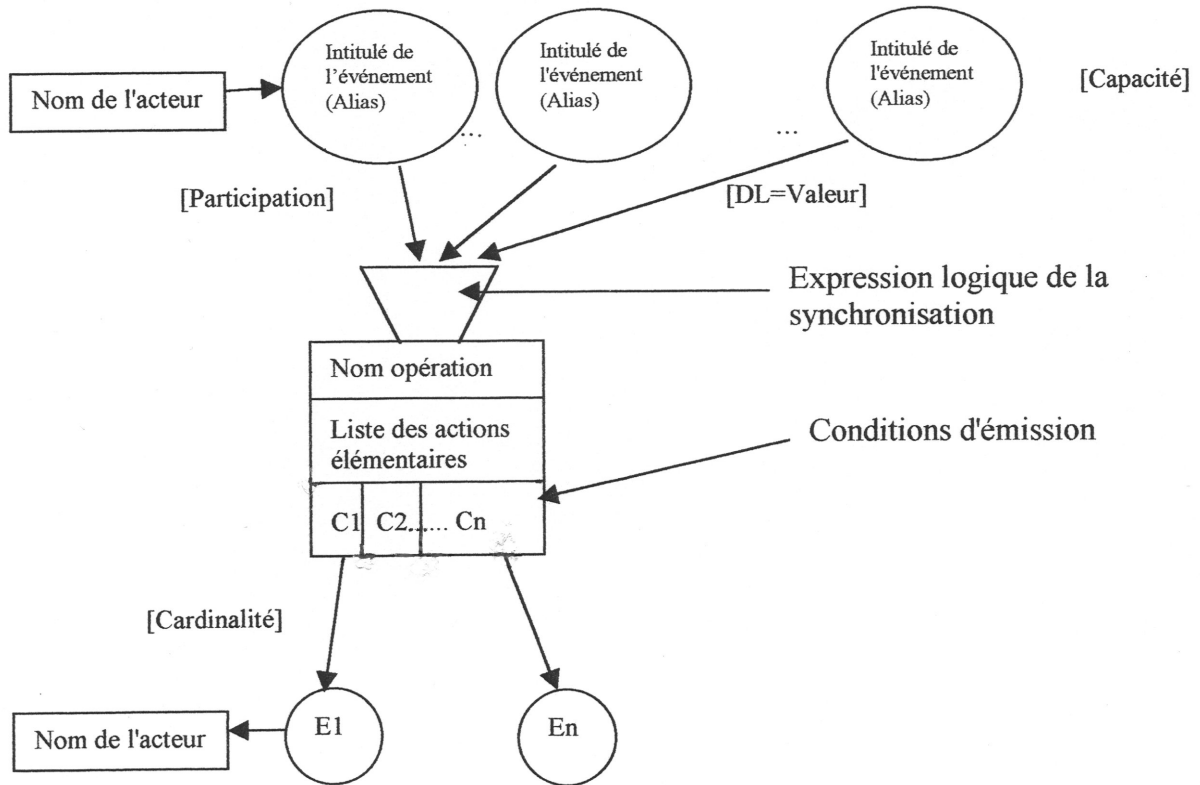
Événement ext3 : « la date actuelle est JJ/MM/AAAA »

Proposition logique : int2 et ext3

Conditions locales : Pour tout int2 ayant

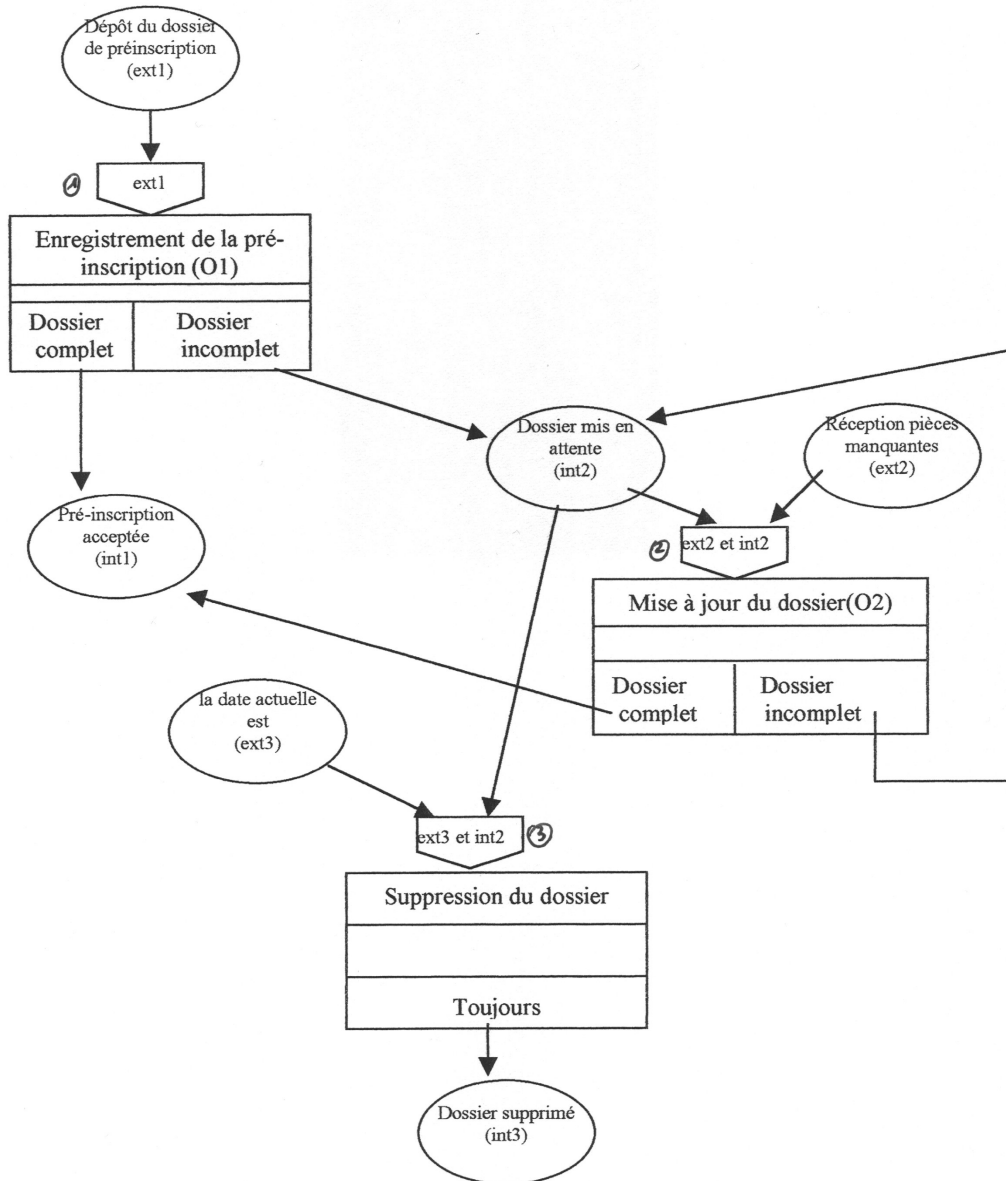
int2.JJ/MM/AAAA = ext3.JJ/MM/AAAA

1.6) Représentation graphique



Remarque: Le MCT conditionne complètement l'interface graphique du S.I.

1.7) Extrait du MCT de la gestion des inscriptions



Événement ext1 : Dépôt du dossier de pré-inscription de l'élève E pour la formation F

Événement ext2 : Réception des pièces manquantes du dossier n° D

Événement ext3 : La date actuelle est JJ/MM/AAAA

Événement int1 : Dossier de pré inscription n° D accepté le JJ/MM/AAAA

Événement int2 : Dossier de pré-inscription n° D mis en attente le JJ/MM/AAAA

Événement int3 : Avis de suppression du dossier n° D émis le JJ/MM/AAAA

Synchronisation : Si ext2 et int2 Avec $ext2.D = int2.D$

Synchronisation : Si int2 et ext3 Avec Pour tout int2 ayant

$$int2.JJ/MM/AAAA + 8 = ext3.JJ/MM/AAAA$$

Remarque : ce MCT fait apparaître un cycle caractérisé par une opération qui admet comme événement contributif un événement dont elle déclenche elle-même l'émission.

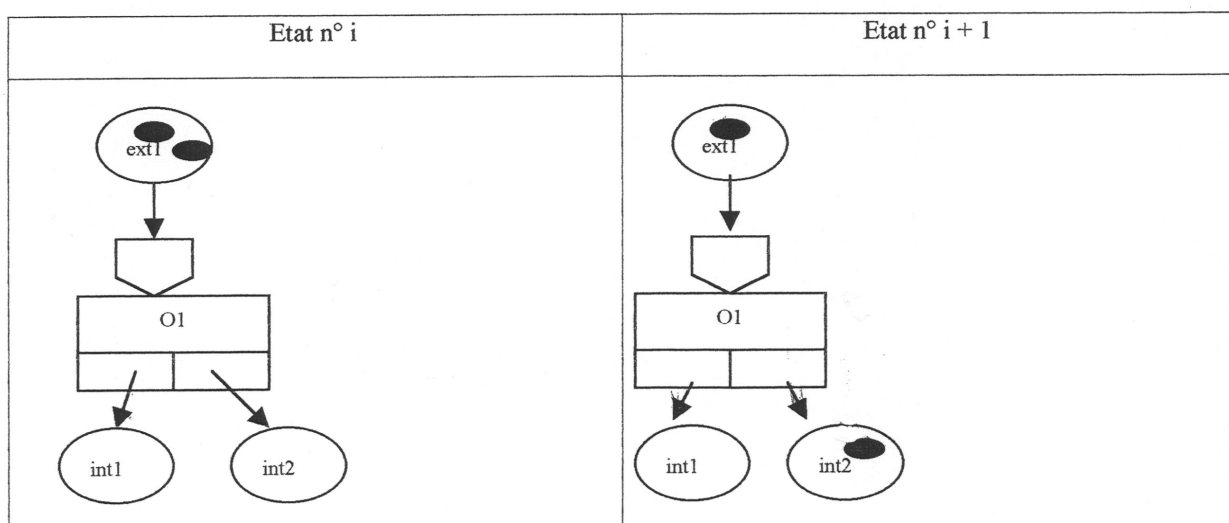
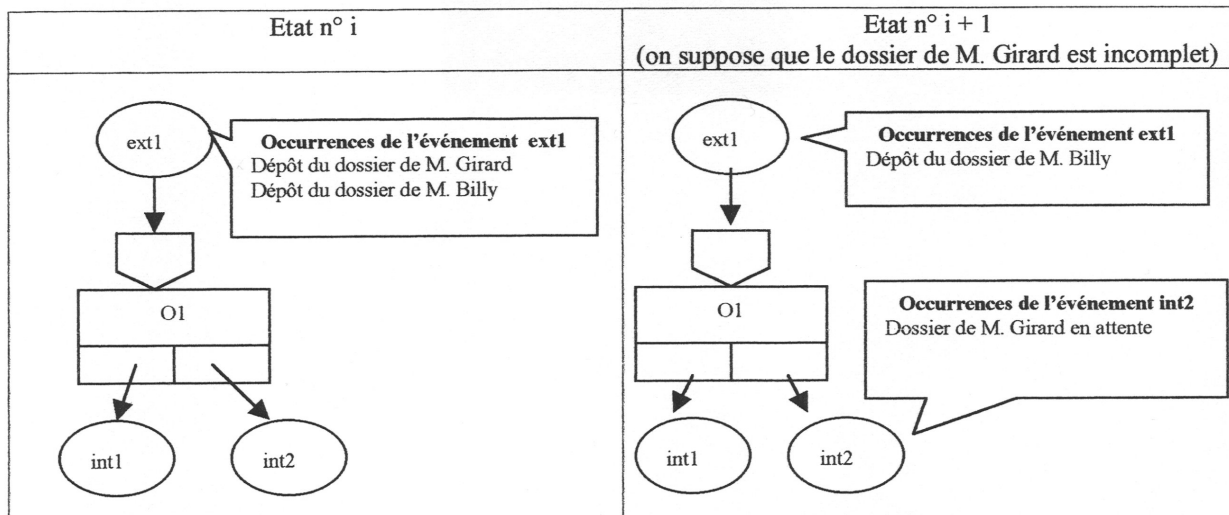
II. Fonctionnement d'un modèle dynamique

2.1) Fonctionnement d'un modèle dynamique

L'arrivée d'un événement externe dans le système d'information provoque l'apparition d'une occurrence nouvelle pour cet événement. On appelle jeton cette occurrence d'événement. Une synchronisation, lorsqu'elle est en attente, devient activable, lorsque la proposition logique associée et les conditions locales deviennent vraies par l'arrivée d'un nouveau jeton. Lorsque la synchronisation est activée, il y a consommation d'un ou de plusieurs jetons par événement qui a contribué à rendre vrai le prédicat et les conditions locales de synchronisation.

La synchronisation déclenche le démarrage de l'opération qui s'exécute et qui provoque l'apparition d'un ou de plusieurs jetons supplémentaires dans tous les événements en sortie de l'opération pour lesquels la règle d'émission est vérifiée.

Les tableaux qui suivent illustrent ce principe de consommation en premier lieu en terme d'occurrences d'événements et ensuite avec les jetons :

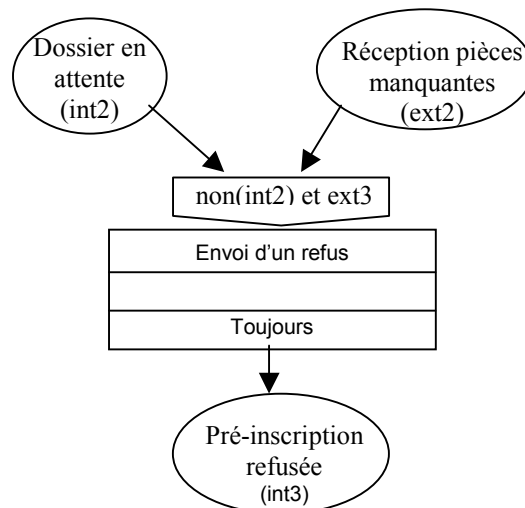


2.2) Règles de vérification du fonctionnement

Compte tenu du principe de fonctionnement exposé précédemment, un modèle dynamique admet un ensemble d'états qui se matérialisent au moyen de jetons répartis dans ses différents événements. Pour que le modèle fonctionne correctement, différentes règles relatives à la consommation de ces jetons devront être vérifiées. On devra par exemple s'assurer que des jetons ne s'accumulent pas dans un événement. Une telle situation signifierait que le système ne remplit pas sa fonction de consommation ou de traitement.

Dans le processus de pré-inscription, il peut y avoir accumulation de jetons dans l'événement « Réception de pièces manquantes » dès lors que le dossier mis en attente a été supprimé du fait du délai de 8 jours dépassé. C'est un choix du gestionnaire d'ignorer ou de traiter ces jetons. Si on décide, par exemple, d'envoyer un avis à l'élève afin de l'informer que ses pièces sont arrivées trop tardivement, on devra enrichir le MCT précédent par le diagramme suivant :

(ce traitement suppose que l'événement « pièces manquantes » soit porteur de suffisamment d'informations pour effectuer l'envoi).

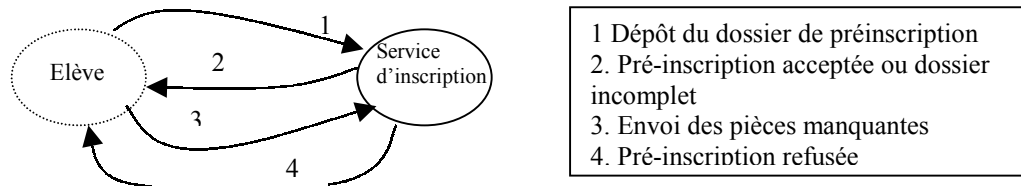


Remarque : il existe de nombreuses autres règles de vérification de fonctionnement détaillées dans l'ouvrage « La méthode Merise – principes et outils ».

III. Règles de construction d'un M.C.T.

Comme pour le modèle conceptuel des données, il n'existe pas de méthode algorithmique permettant d'aboutir à un modèle conceptuel des traitements. Si la présentation de ses concepts peut en effet être entièrement formalisée et explicitée, leur assemblage pour résoudre un problème donné exige des qualités d'analyse et de réflexion que seule l'expérience peut accroître. Il existe cependant des outils ou des démarches d'aide à la conception d'un MCT. Ainsi certains auteurs préconisent, pour faciliter la conception d'un MCT, l'élaboration d'un Modèle Conceptuel de Communication (MCC).

Ce modèle consiste à recenser la liste de tous les acteurs intervenants dans le système d'information et à schématiser les flux d'information qu'ils échangent. Les acteurs internes - ceux qui appartiennent au système d'information - sont représentés dans des cercles en trait plein, les acteurs externes au système d'information sont représentés dans des cercles en pointillés. Les flux d'information sont schématisés par des arcs entre acteurs. *Le schéma ci-après représente les flux inhérents à la pré-inscription à la faculté de Bordeaux.*



Ce diagramme met en évidence les événements externes du MCT. Chaque flux échangé d'un acteur externe vers un acteur interne devient en effet un événement déclencheur externe. Les flux à destination d'un acteur externe deviendront des événements internes. Les opérations ainsi que leurs événements résultats ne sont pas aisément déductibles de ce schéma.

IV. Le modèle organisationnel des traitements

Le modèle organisationnel des traitements s'attache à décrire les propriétés des traitements non traitées par le modèle conceptuel des traitements, c'est-à-dire: le temps, les ressources, le lieu. Le modèle organisationnel des traitements consiste donc à représenter le modèle conceptuel des traitements dans un tableau dont les colonnes sont la durée, le lieu, les responsables et ressources nécessaires à une action.

4.1) Le tableau des procédures fonctionnelles

La première étape du modèle organisationnel des traitements consiste à découper les opérations en **procédures fonctionnelles**, une succession de traitements déclenchée par un événement. Il s'agit donc d'associer dans un tableau: les procédures fonctionnelles, l'heure de début et de fin, le lieu du poste de travail, le responsable du poste de travail, les ressources du poste de travail.

Procédure	temps		poste de travail		
	début	durée	lieu	responsable	ressources

Exercice n°7:

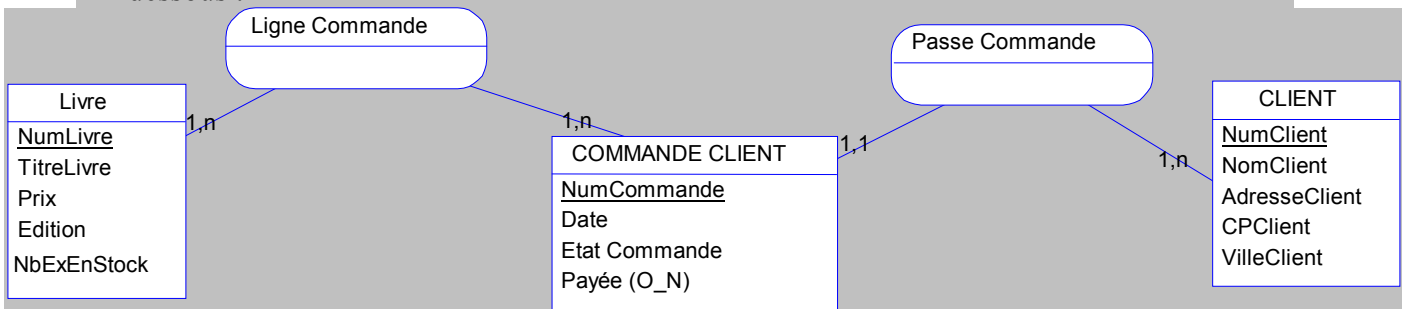
Claire Figeac, a envoyé à la librairie TrouveTout la lettre suivante :

Monsieur,
 Pouvez vous m'envoyer 15 exemplaires de l'ouvrage "La méthode Merise " aux Editions Organisation
 Claire FIGEAC
 Avenue de la Gare
 05000 NICE

1. Déduire de cette occurrence d'événement, le message associé à l'événement de libellé « Commande d'un ouvrage »

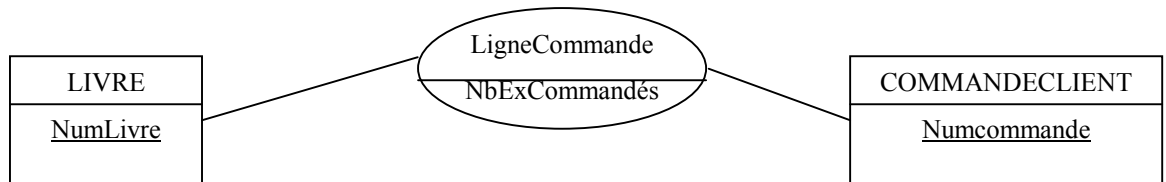
Commande de N exemplaire de l'ouvrage O par le client C.

2. Un extrait du MCD de la base d'information de la librairie TrouveTout est présenté ci dessous :



Le concepteur du MCD a oublié de modéliser le nombre d'exemplaires commandés d'un livre. Modifier le MCD de manière à rajouter la propriété NbExCommandés.

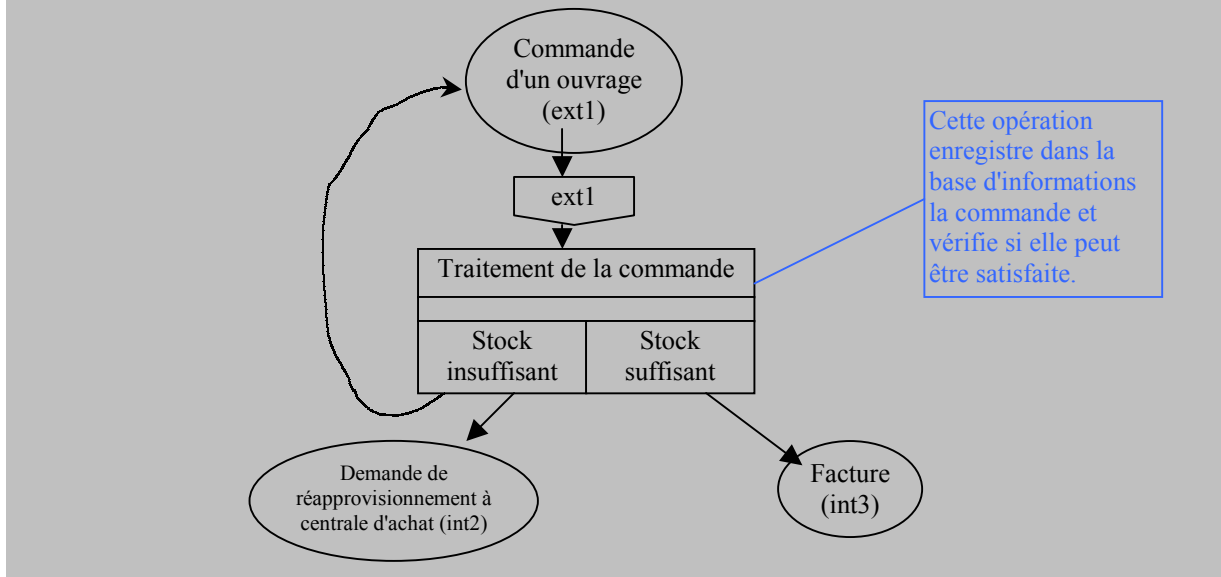
Il suffit de transformer l'association "Ligne Commande" en association porteuse avec la propriété NbExCommandés:



3. Lors de la réception d'une commande de plusieurs exemplaires d'un ouvrage, deux cas peuvent se présenter :

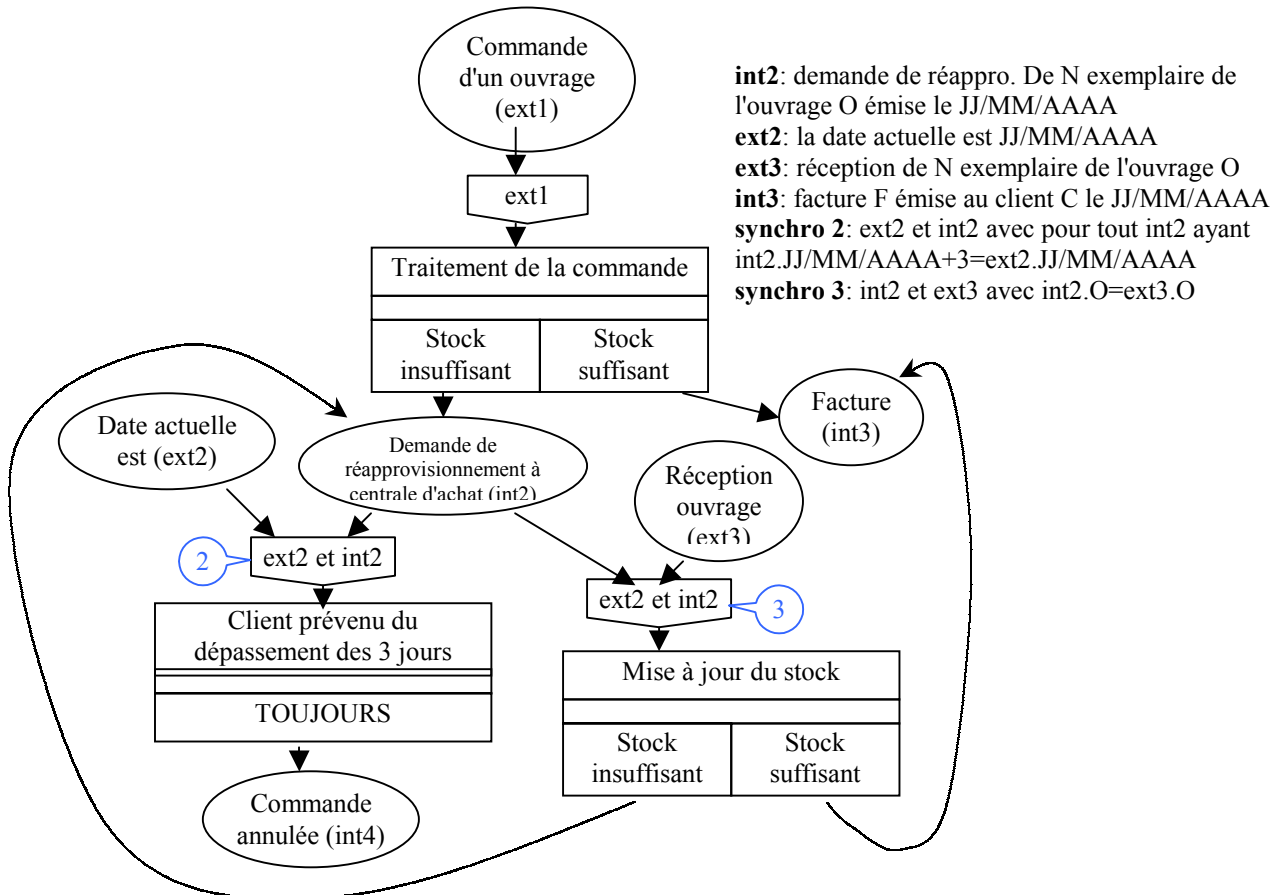
- soit le stock est supérieur au nombre d'exemplaires commandés : dans ce cas une facture est établie et l'envoi des ouvrages est réalisé
- soit le stock est insuffisant : une demande de réapprovisionnement est alors envoyée à la centrale d'achats qui expédie, à TrouveTout, les ouvrages demandés dans un délai maximum de 3 jours. Au-delà de cette limite, le client doit être prévenu que sa commande ne peut pas être satisfaite.

3.1 Expliquer pourquoi le MCT suivant ne peut pas convenir pour représenter le processus de traitement d'une commande.



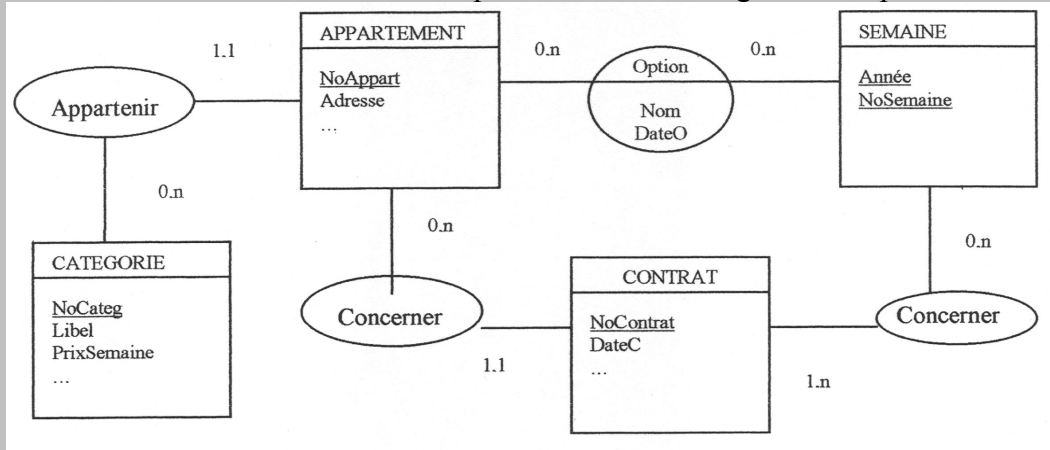
- Ce MCT ne tient pas compte du délai maximum de trois jours après lesquels le client doit être prévenu.

3.2 Proposer un MCT garantissant le traitement de toute commande, c'est-à-dire aboutissant soit à l'édition d'une facture, soit à l'émission d'un avis d'impossibilité de livraison.



Exercice n° 8

On souhaite modéliser le processus de gestion des locations saisonnières de l'agence immobilière I. Un extrait du modèle conceptuel associé à cette gestion est présenté ici :



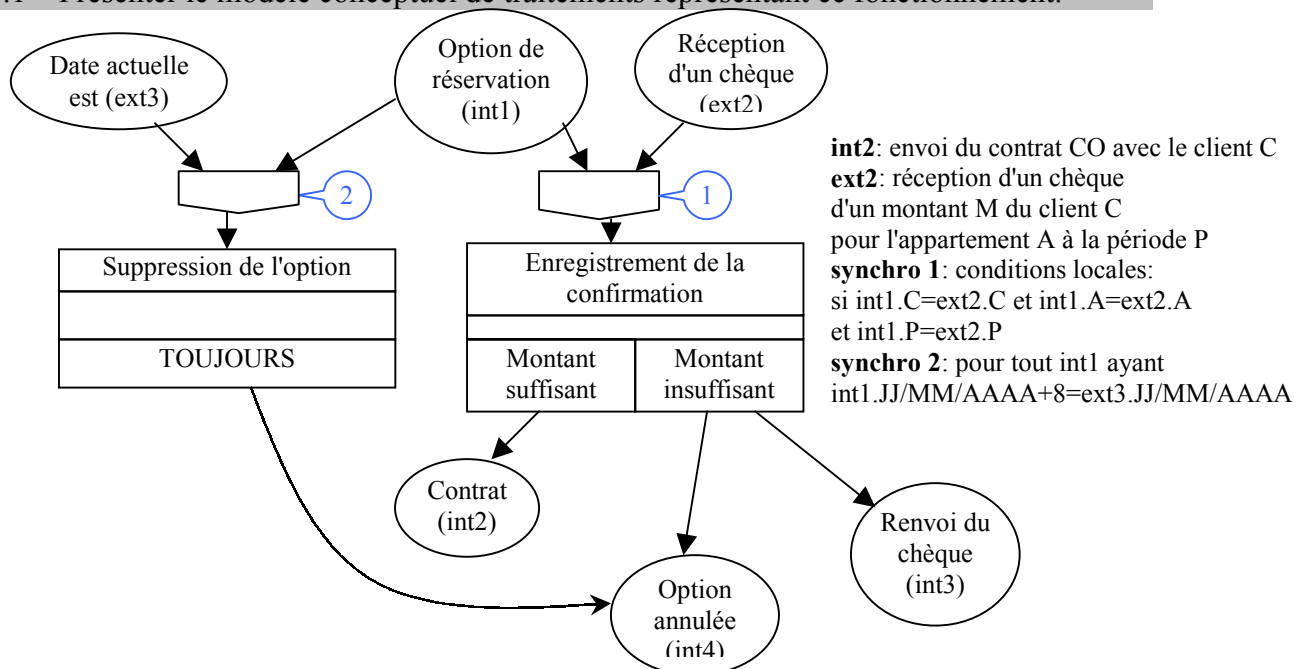
L'association porteuse « Option » permet de mémoriser le nom du client qui a pris une option de réservation sur un appartement et une semaine donnés ainsi que la date à laquelle il a procédé à cette option.

1. On considère l'événement interne « int1 » défini ci-dessous :

int1 : « Option de réservation du client de nom N pour l'appartement A et période P réalisée le JJ/MM/AAAA. » (la période est une suite contiguë de semaines)

Modéliser le fait que le client dispose de 8 jours pour confirmer sa réservation en envoyant un chèque correspondant à 25 % du montant total de la location. Son option sera levée dans les deux cas suivants : l'agence I n'a reçue aucune confirmation durant les 8 jours; le montant envoyé est inférieur au montant dû. Dans ce cas le chèque est renvoyé à l'adresse figurant sur celui-ci. Dans le cas où il y a confirmation dans les délais, un contrat est créé et envoyé au client.

1.1 Présenter le modèle conceptuel de traitements représentant ce fonctionnement.



1.2 Indiquer, pour chaque opération figurant dans le MCT, les entités et les associations concernées ainsi que le type d'action effectuée :

C pour création,
L pour lecture,
M pour modification
S pour suppression

On a simplement deux opérations:

- Pour la Suppression: S d'une occurrence (ou plusieurs) de l'association OPTION,
- Pour la Confirmation: L de APPARTEMENT (donc L de CATEGORIE)
C de CONTRAT (donc C de CONCERNE 1, 2)
S de OPTION

2. Le contrat reçu par le client précise que le montant de la location doit être réglé entièrement un mois avant le début de la location. Si tel n'était pas le cas, un rappel est envoyé au client qui dispose alors de 8 jours pour régulariser sa situation. A l'issue de ces 8 jours si aucun règlement n'est parvenu, l'agence immobilière envoie un avis d'annulation au client et encaisse bien évidemment les arrhes. Proposer une modélisation de cette gestion. On suppose que l'on dispose de l'événement interne « int3 » suivant : « Début de location du contrat C prévu le JJ/MM/AAAA ».

Exercice n° 9 – Synthèse

➤ Première partie : modélisation des données

La société DOMELEC est spécialisée dans la fabrication et la commercialisation d'appareils électroménagers à usage domestique. Elle comprend plusieurs sites, aussi appelés « unités de production » répartis sur le territoire métropolitain. Chaque unité de production est identifiée par un code et possède un nom en rapport avec sa localisation (exemple : l'unité « Bourgogne »). Une unité de production prend en charge au moins une ligne de produits. Chaque unité de production héberge plusieurs équipes de production.

Une ligne de produits est une famille de produits qui possèdent la même fonction générale. La fabrication d'une ligne de produits est confiée à l'une des unités de production. Chaque produit n'appartient qu'à une seule ligne de produits. On trouvera par exemple :

Ligne de produits		Produits	
Référence	Nom de la ligne	Numéro	Nom du produit
L01	Nettoyeur	1	Aspirateur traîneau 1 000 W
		2	Aspirateur Traîneau 1 400 W
		3	Aspirateur bidon eau/poussière 1 400 W
		4	Lave-vitre
		5	...
L02	Frigorifique	1	Réfrigérateur 1 porte 150 litres
		2	Réfrigérateur combiné 2 portes
		3	Congélateur bac 1000 litres
		4	...

L'identifiant d'un produit est composé de l'identifiant de la ligne de produits à laquelle il appartient et d'un numéro séquentiel. Pour chaque produit, il est important de mémoriser un nom explicite, sa date de première mise en fabrication et un descriptif détaillé.

Une équipe de production est composée exclusivement de salariés de l'entreprise. Une équipe travaille sur une seule ligne de produits. Chaque équipe est identifiée par un numéro d'équipe. Elle est encadrée par un chef d'équipe appelé superviseur. La date de prise en charge de l'équipe par son superviseur doit être mémorisée. Les autres membres de l'équipe sont qualifiés d'opérateurs.

Pour un opérateur il faut connaître la date d'affectation à son poste de travail actuel, sa formation d'origine, décrite sous forme d'un commentaire, ainsi que le plus haut diplôme obtenu par l'opérateur, s'il en possède un.

Le superviseur peut diriger plusieurs équipes au sein de la même unité de production. Pour un superviseur, les informations à prendre en compte sont : la date de première affectation en qualité de chef d'équipe dans l'entreprise et son origine (recrutement externe ou promotion interne).

Un opérateur fait partie d'une seule équipe. Il est affecté à un seul poste de travail. On ne s'intéresse pas aux précédentes affectations de cet opérateur (équipe ou poste de travail).

Le même poste de travail peut être mobilisé pour fabriquer différents produits d'une même ligne de produits car, en général, les techniques de fabrication des produits d'une ligne sont semblables. La fabrication d'un produit requiert plusieurs postes de travail.

Chaque poste est défini par un numéro, un nom et un libellé explicatif. On souhaite également connaître la date d'installation du poste, la date de sa dernière vérification et l'intervalle de temps maximal entre deux vérifications.

La maîtrise de l'utilisation d'un poste de travail par un opérateur peut être décomposé en compétences élémentaires. Une compétence est un savoir-faire élémentaire sur un poste de travail. Elle peut être commune à plusieurs postes. Chaque compétence possède un code et un libellé.

Une habilitation est un ensemble de compétences liées à un poste de travail. Pour chaque poste, il existe quatre niveaux hiérarchisés d'habilitation qui correspondent à des degrés de maîtrise de ce poste. Chaque habilitation est identifiée par le poste qu'elle concerne et son niveau (de 1 à 4). Un libellé décrit chaque habilitation.

Exemple :

Poste numéro 354 : Installation système refroidissement et protection isotherme

Niveau et libellé de l'habilitation

Code et libellé de compétence

1. Mettre en route et pratiquer le poste

- C005 – Mise sous tension et hors tension du poste
- C012 – Test de fonctionnement du poste de soudure
- C025 – Test de fonctionnement de l'unité de remplissage du gaz
- C118 – Test du compresseur à installer
- C009 – Connaissances de l'ordre des opérations sur le poste

2. Assurer la production d'un appareil réfrigéré

- C259 – Mise en place du circuit de refroidissement
- C384 – Mise en place du compresseur
- C201 – Soudure des différents éléments
- C882 – Remplissage du gaz sous pression
- C449 – Mise en place de l'isolation

3. Assurer la quantité et la qualité

- C419 – Maîtrise de chaque étape de production
- C127 – Exécution des protocoles de test
- C222 – Intervention en cas de dysfonctionnement

4. Maîtriser totalement le poste et transmettre ses compétences

- C843 – Production en quantité et qualité
- C992 – Formation des utilisateurs sur ce poste

Travail à faire

Présenter le schéma conceptuel de données correspondant au domaine décrit. Ce schéma devra prendre en compte l'ensemble des règles de gestion relatives aux unités de production, aux produits et lignes de produits, aux équipes, aux salariés, superviseurs et opérateurs, aux postes de travail, ainsi qu'aux compétences et habilitations.

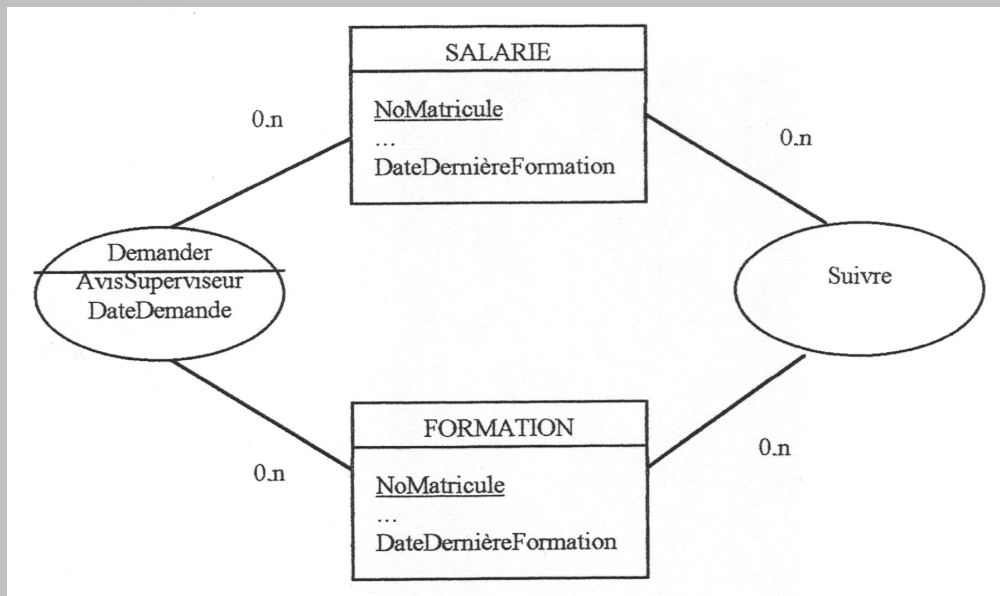
➤ **Seconde partie : la modélisation des traitements.**

La formation est l'un des instruments du management. Son but est d'améliorer l'implication des salariés dans le processus de production, en leur permettant de progresser dans la maîtrise des postes de travail. Un plan annuel de formation (PAF) paraît au mois de décembre. Il s'agit d'un catalogue présentant les formations de l'année à venir. On y trouve un bref descriptif qui indique le libellé, l'objectif général, la date de début et la date de fin de chaque formation.

Pour participer à une formation, un salarié remplit une demande adressée au centre de formation. Le centre étudie alors cette demande et l'enregistre si elle est recevable. Dans le cas contraire, un refus motivé est transmis au salarié.

Le nombre des participants à une formation est limité en fonction d'impératifs définis par les formateurs. Il existe également un seuil minimum de participants en dessous duquel une formation prévue dans le PAF ne sera pas réalisée. Quinze jours avant le début de la formation, les demandes enregistrées sont examinées et les candidats retenus sont définitivement inscrits. Lorsque le nombre de demandes est supérieur au nombre de places, les participants sont sélectionnés en fonction de critères multiples (avis du superviseur, date de dernière formation, ...). A l'inverse, lorsque le nombre de demandes n'atteint pas le seuil minimum prévu, la formation est annulée.

La schématisation conceptuelle des données relatives au domaine « Formation » est la suivante :



Travail à faire

Présenter de manière détaillée, la « synchronisation-opération » qui se rapporte à l'examen des demandes quinze jours avant le début de la formation. Les événements en entrée et sortie sont donnés dans le tableau ci-dessous :

Événements en entrée :

ext1 : la date actuelle est JJ/MM/AAAA

int1 : la formation F doit débuter le JJ/MM/AAAA

Événements en sortie :

int2 : la formation F est annulée

int3 : le salarié S va suivre la formation F

Pour la synchronisation on précisera le prédicat ainsi que les conditions locales.

Pour l'opération les association et entités concernées ainsi que le type d'action effectuée :

C pour création,

L pour lecture,

M pour modification

S pour suppression