MERISE

Modélisation de Systèmes d'Information

Pierre Gérard

IUT de Villetaneuse - Université de Paris 13

DUT Informatique 2ème année
2004/2005

LaTeX

Cycle de vie

« La qualité du processus de fabrication est garante de la qualité du produit »

- Pour obtenir un logiciel de qualité, il faut en maîtriser le processus d'élaboration
  - La vie d'un logiciel est composée de différentes étapes
  - La succession de ces étapes forme le cycle de vie du logiciel
  - Il faut contrôler la succession de ces différentes étapes

Etude de faisabilité

- Déterminer si le développement proposé vaut la peine d'être mis en œuvre, compte tenu de attentes et de la difficulté de développement
  - Etude de marché : Déterminer s'il existe un marché potentiel pour le produit.

Spécification

- Déterminer les fonctionnalités que doit posséder le logiciel
  - Collecte des exigences : obtenir de l'utilisateur ses exigences pour le logiciel
  - Analyse du domaine : déterminer les tâches et les structures qui se répètent dans le problème

Organisation du projet

- Déterminer comment on va développer le logiciel
  - Analyse des coûts : établir une estimation du prix du projet
  - Planification : établir un calendrier de développement
  - Assurance qualité du logiciel : déterminer les actions qui permettront de s'assurer de la qualité du produit fini
  - Répartition des tâches : hiérarchiser les tâches et sous-tâches nécessaires au développement du logiciel

Conception

- Déterminer la façon dont dont le logiciel fournit les différentes fonctionnalités recherchées
  - Conception générale
    - Conception architecturale : déterminer la structure du système
    - Conception des interfaces : déterminer la façon dont les différentes parties du système agissent entre elles
  - Conception détaillée : déterminer les algorithmes pour les différentes parties du système

Implémentation

- Ecrire le logiciel

Tests

- Essayer le logiciel sur des données d'exemple pour s'assurer qu'il fonctionne correctement
  - Tests unitaires : faire tester les parties du logiciel par leurs développeurs
  - Tests d'intégration : tester pendant l'intégration
  - Tests de validation : pour acceptation par l'acheteur
  - *Tests système :* tester dans un environnement proche de l'environnement de production
  - *Tests Alpha :* faire tester par le client sur le site de développement
  - *Tests Bêta :* faire tester par le client sur le site de production
  - *Tests de régression :* enregistrer les résultats des tests et les comparer à ceux des anciennes versions pour vérifier si la nouvelle n'en a pas dégradé d'autres

Livraison

- Fournir au client une solution logicielle qui fonctionne correctement
  - Installation : rendre le logiciel opérationnel sur le site du client
  - Formation : enseigner aux utilisateurs à se servir du logiciel
  - Assistance : répondre aux questions des utilisateurs

Maintenance

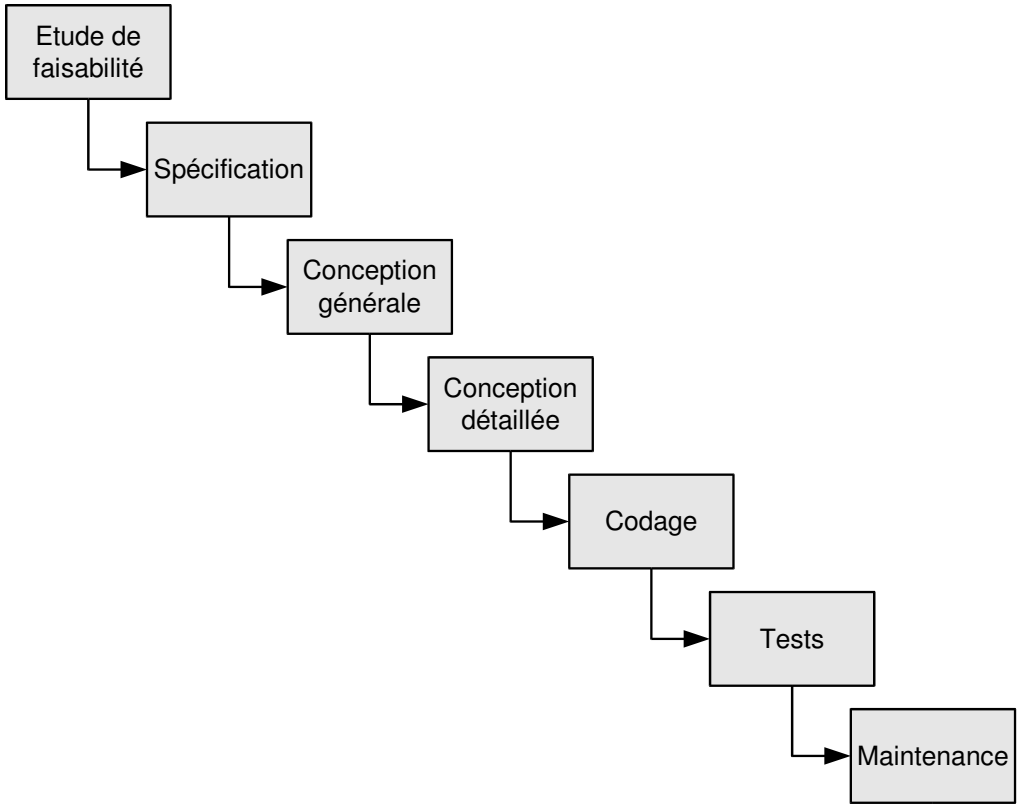
- Mettre à jour et améliorer le logiciel pour assurer sa pérénité
- Pour limiter le temps et les coûts de maintenance, il faut porter ses efforts sur les étapes antérieures

	Répartition effort dév.	Origine des erreurs	Coût de la maintenance
Définition des besoins	6%	56%	82%
Conception	5%	27%	13%
Codage	7%	7%	1%
Intégration Tests	15%	10%	4%
Maintenance	67%		

Modèles linéaires et incrémentaux

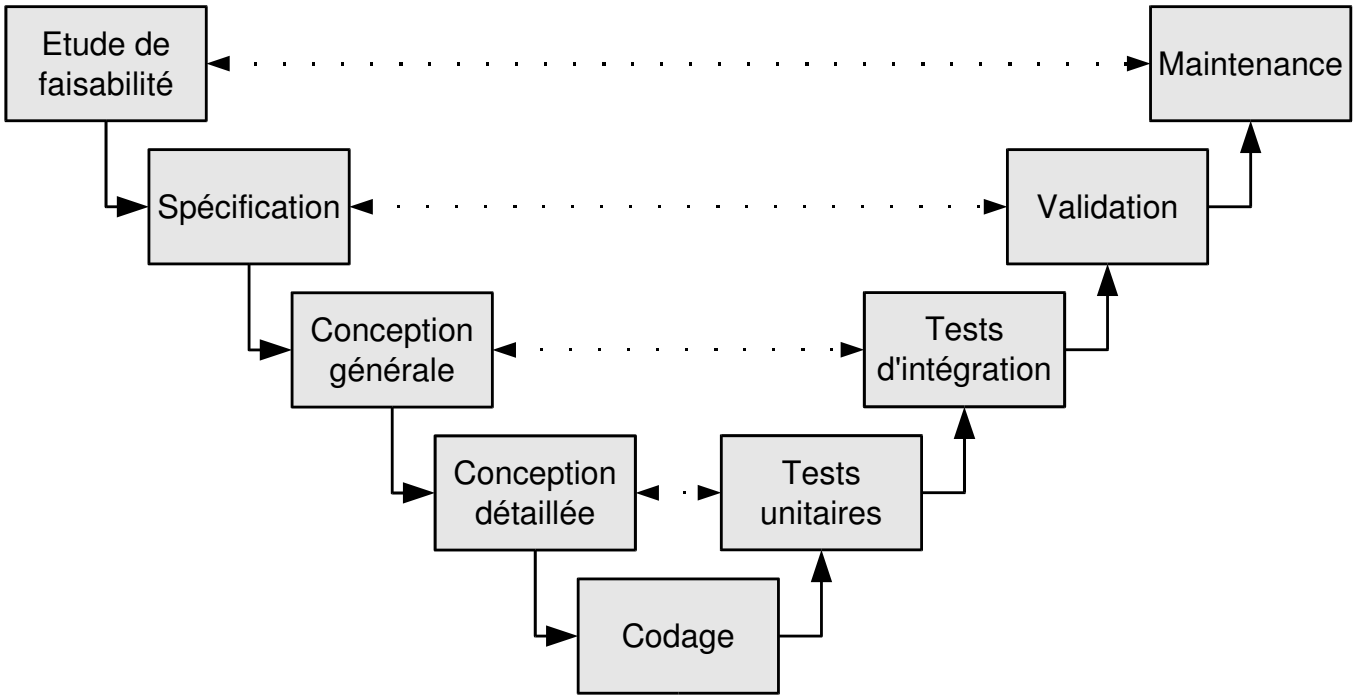
- Modèles linéaires
  - cascade
  - modèle en V
  - ...
- Modèles non linéaires
  - prototypage
  - modèles incrémentaux
  - modèle en spirale
  - ...

Le cycle de vie en « Cascade »



- Adapté pour des projets de petite taille, et dont le domaine est bien maîtrisé

Le cycle de vie en « V »



- Adapté pour des projets dont le domaine est bien maîtrisé

Le prototypage

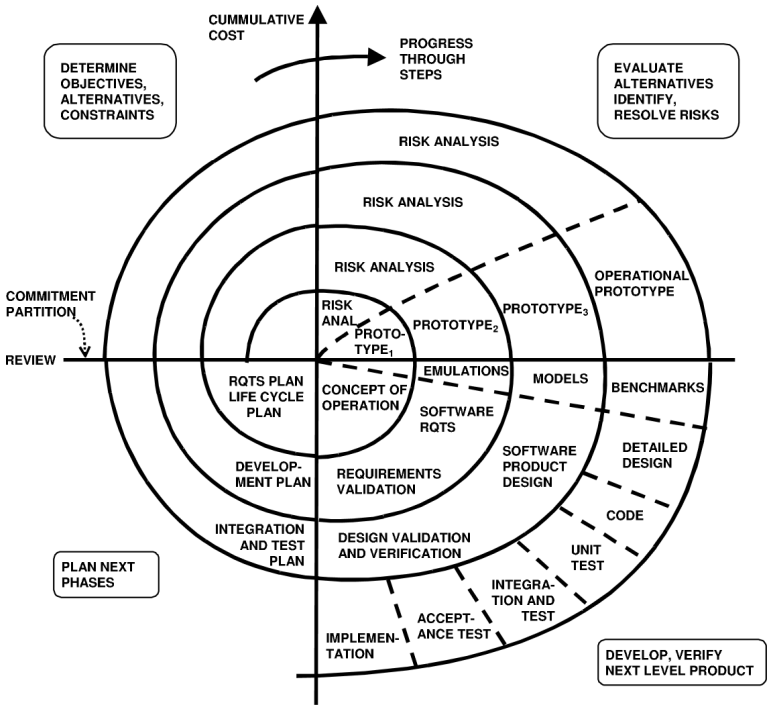
- Prototype : version d'essai du logiciel
  - Pour tester les différents concepts et exigences
  - Pour montrer aux clients les fonctions que l'on veut mettre en œuvre
- Lorsque le client a donnné son accord, le développement suit souvent un cycle de vie linéaire
- Avantages : Les efforts consacrés au développement d'un prototype sont le plus souvent compensés par ceux gagnés à ne pas développer de fonctions inutiles

Le modèle incrémental de Parnas

1. Concevoir et livrer au client un sous-ensemble minimal et fonctionnel du système
2. Procéder par ajouts d'incréments minimaux jusqu'à la fin du processus de développement
3. Avantages : Meilleure intégration du client dans la boucle, produit conforme à ses attentes

Le modèle en Sprirale de Boehm

- Un modèle mixte
- A chaque cycle, recommencer :
  1. Consultation du client
  2. Analyse des risques
  3. Conception
  4. Implémentation
  5. Tests
  6. Planification du prochain cycle



- Avantages : meilleure maîtrise des risques, mais nécessite une (très) grande expérience

Méthode : une démarche et un formalisme

- Démarche : succession d'étapes pour
  - Mieux maîtriser le déroulement d'un projet
  - Meilleure visibilité pour les utilisateurs sur certains résultats intermédiaires et garantir que le résultat final sera celui attendu
- Formalisme défini par:
  - Un langage formel
  - Un langage semi-formel généralement graphique
  - Un langage naturel
- Fonction :
  - Représenter le monde réel tel qu'il est perçu par le concepteur
  - Outil de communication entre informaticiens et utilisateurs
  - Constitué par un ensemble de modèles permettant d'assurer une bonne compréhension des besoins des utilisateurs

Modèles

- Représentation abstraite de la réalité qui exclut certains détails du monde réel
- Permet de réduire la complexité d'un phénomène en éliminant les détails qui n'influencent pas son comportement significatif
- Reflète ce que le concepteur croit important pour la compréhension et la prédiction du phénomène modélisé, les limites du phénomène modélisé dépendent des objectifs du modèle

MERISE

Méthode d'Etude et de Réalisation Informatique pour les Systèmes d'Entreprise

- Méthode Eprouvée pour Retarder Indéfiniment la Sortie des Etudes
- MEthode pour Rassembler les Idées Sans Effort
  - Sutout lorsqu'on utilise un AGL

Approche Données / Traitements

- Pour étudier et développer l'informatique d'une organisation, il est nécessaire de connaître:
  - comment elle réagit à une sollicitation externe
  - quelle est la structure des informations qu'elle utilise
- MERISE modélise cette connaissance de manière duale :
  - Modèles des Traitements (réaction aux événements...)
  - Modèles des Données (vocabulaire de la structure...)
  - Les 2 aspects sont complémentaires, synchronisés et validés entre eux

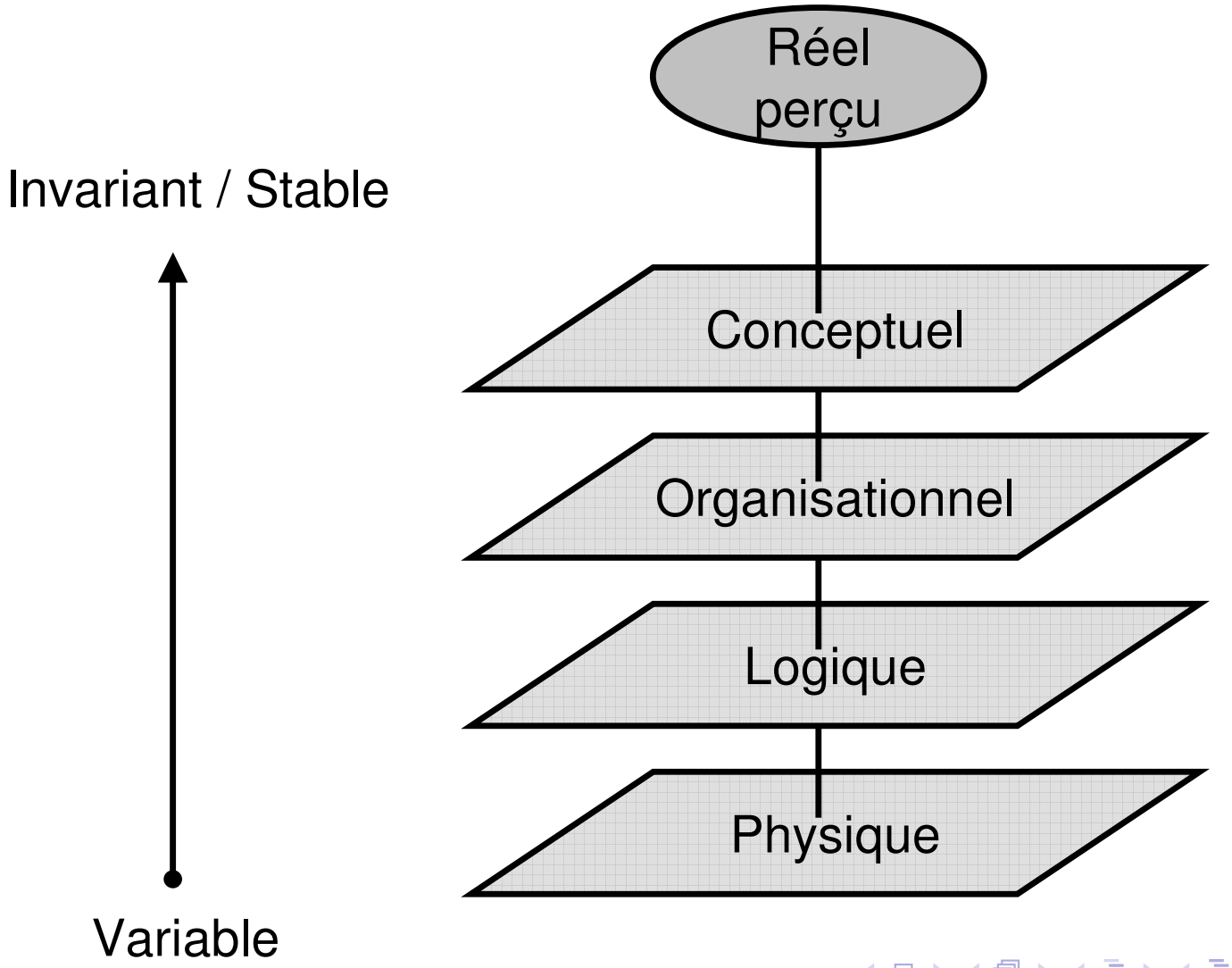
Niveaux d'abstraction

- Pour chacun des problèmes de modélisation (données / traitements)
  - Procéder de manière progressive...
  - ... du plus stable au plus technique

Niveaux d'abstraction

- Niveau Conceptuel
  - Ce qu'il faut faire
  - Quoi ?
- Niveau Organisationnel
  - La manière de faire
  - Pour les traitements
- Niveau Logique
  - Choix des moyens et ressources
  - Pour les données
- Niveau Physique
  - Les moyens de le faire
  - Comment ?

Niveaux d'abstraction



Exemples de niveaux d'abstraction

- Conceptuel
  - Le client effectue une demande de service à la compagnie pour assurer son véhicule. Cette dernière lui propose un devis
- Organisationnel
  - Un client effectue une demande de service à l'agence de son choix, par courrier, pour assurer un véhicule. Un agent de service concerné, si le client est fiable (consultation d'un fichier central inter assurances), prend contact par téléphone pour une visite à domicile (après 17 heures) afin d'examiner plus précisément ses besoins et établir un devis
- Physique
  - Le fichier central inter assurances est accessible par internet. Les agences sont connectées au siège de la compagnie par liaison ADSL. Chaque agence dispose de micro-ordinateurs de type PC et peut traiter ses données en local grâce au SGBD Access

Le niveau Conceptuel

- Exprime les choix fondamentaux de gestion, les objectifs de l'organisation
- Décrit les invariants de l'organisation
  - Le métier de l'organisation
- Définit
  - Des activités
  - Des choix de gestion
  - Des informations
- Indépendamment
  - Des aspects organisationnels
  - Des aspects techniques de mise en oeuvre
- Du point de vue
  - Des traitements: objectif, résultat, règle de gestion, enchaînement
  - Des données: signification, structure, liens

Le niveau Organisationnel

- Exprime les choix organisationnels de ressources humaines et matérielles
- Définit:
  - La répartition géographique et fonctionnelle des sites de travail (du point de vue des données et des traitements)
  - Le mode de fonctionnement : temps réel ou temps différé
  - La répartition du travail homme/machine (degré et type d'automatisation)
  - Les postes de travail et leur affectation,
  - La volumétrie des données
  - La sécurité des données
- Indépendamment des moyens de traitement et de stockage de données actuels ou futurs
- Les opérations conceptuelles vont être décomposées au niveau organisationnel en une ou plusieurs opérations organisationnelles

Le niveau Logique

- Exprime la forme que doit prendre l'outil informatique pour être adapté à l'utilisateur, à son poste de travail
- Indépendamment de l'informatique spécifique, des langages de programmation ou de gestion des données
- Introduit la notion d'outils en tant que fonction réutilisable
- Décrit
  - Le schéma de la base de données (relationnel, hiérarchique ou réseau), cad les caractéristiques du mode de gestion des données
  - La répartition des D sur les différentes unités de stockage
  - Les volumes par unité de stockage
  - L'optimisation des coûts induits par le mode de gestion

Le niveau Physique

- Traduit les choix techniques et la prise en compte de leurs spécificités
- Répond aux besoins des utilisateurs sur les aspects logiciels et matériels.
- Définit complètement:
  - Les fichiers, les programmes
  - L'implantation physique des données et des traitements
  - Les ressources à utiliser
  - Les modalités de fonctionnement

Les modèles au niveau Conceptuel

- Le Modèle Conceptuel des Données (MCD)
  - Description des données et des relations en termes de
    - Entité ou Individu
    - Relation ou Association
    - Propriétés ou d'Attributs
- Le Modèle Conceptuel des Traitements (MCT)
  - Description de la partie dynamique du SI en termes de
    - Processus
    - Opérations

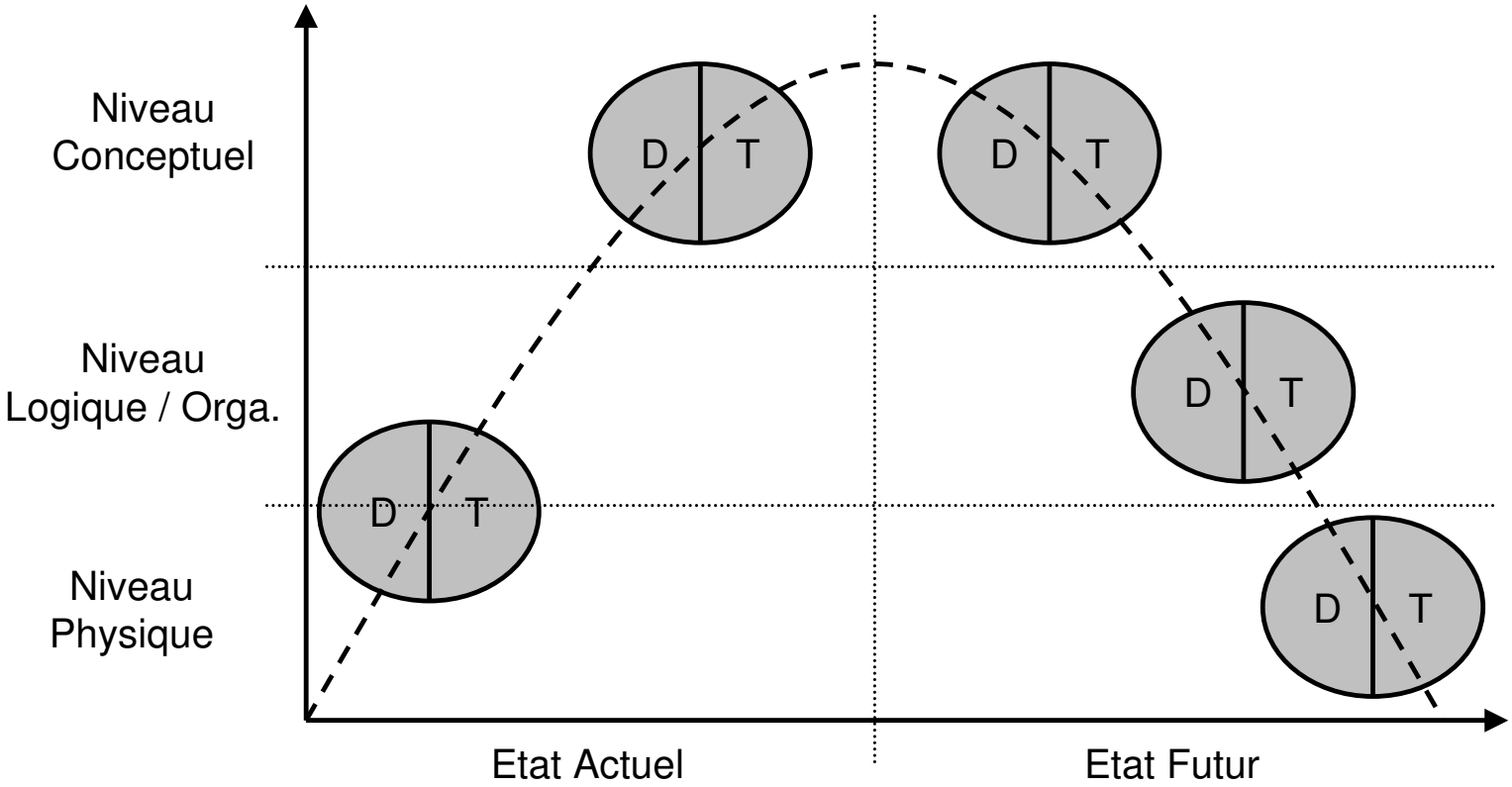
Les modèles aux niveaux Organisationnel et Logique

- Le Modèle logique de donnée (MLD)
  - Le modèle « CODASYL » si une orientation base de données réseau est choisie
  - Le modèle « relationnel » si une orientation base de données relationnelle est choisie
  - Le modèle « hiérarchique »
- Le Modèle Organisationnel des Traitement (MOT)
  - Permet de représenter par procédure les phases et les tâches effectuées par chaque poste de travail

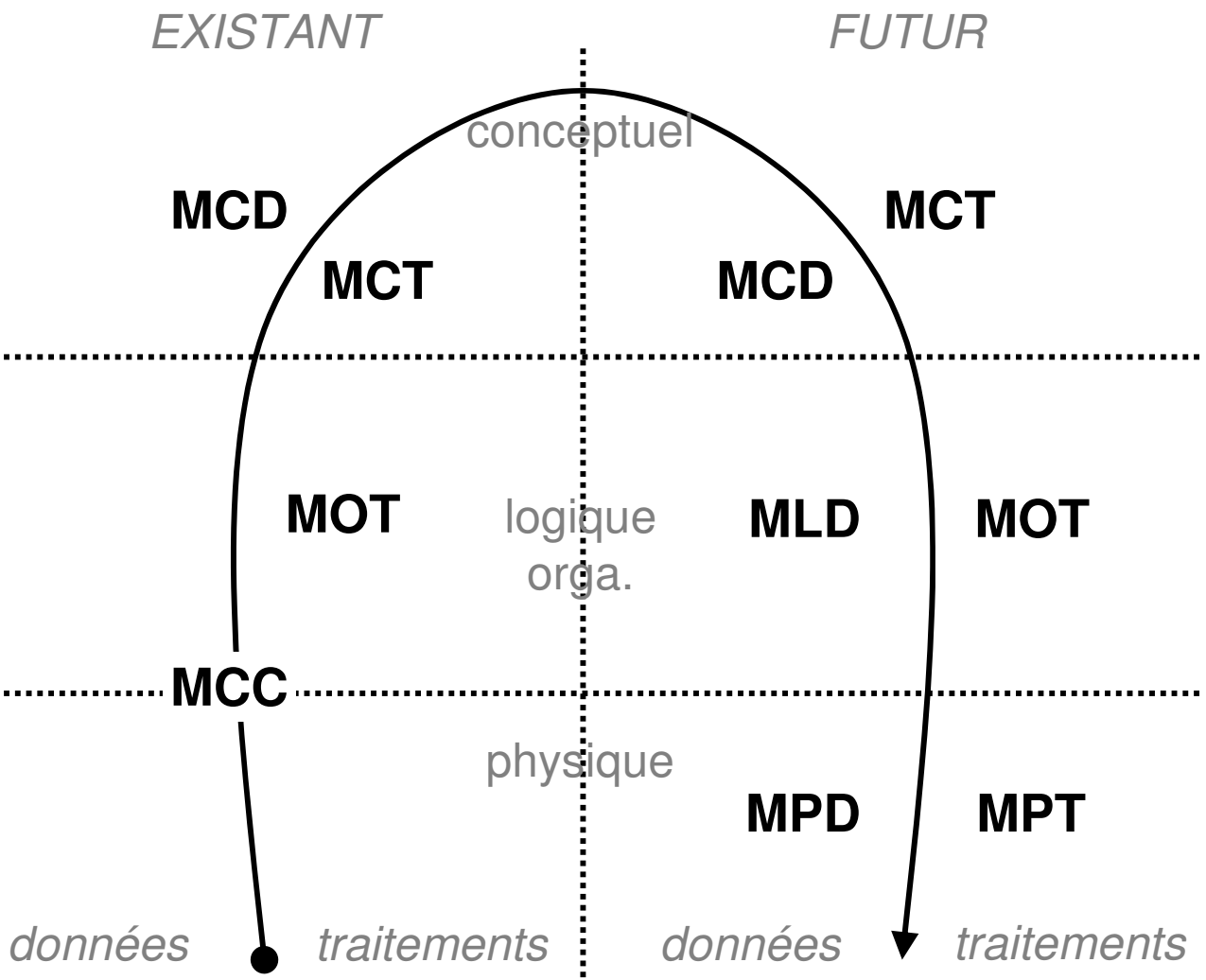
Les Modèles au niveau Physique

- Le Modèle Physique des Données (MPD)
  - Spécifie les organisations physiques de données
- Le Modèle Physique des Traitements (MPT)
  - Décrit les traitements réalisés pour chaque transaction (temps réel) ou chaque unité de traitement (temps différé)

Processus de développement



Modèles successifs produits



Organisation du projet

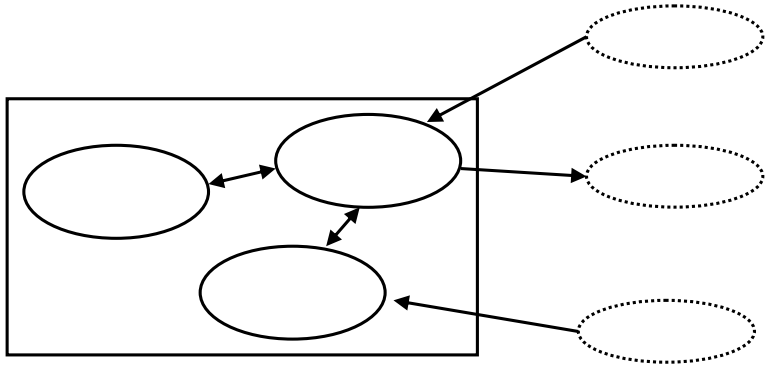
- Par groupe de 5 étudiants : analyse complète du cas proposé
- Pour chaque scéance de TD
  - Conception du modèle demandé pour la scéance en question
  - A la fin de chaque scéance, l'enseignant collecte votre travail
  - Au début de chaqué scéance
    - L'enseignant vous rend le travail de la scéance précédente corrigé
    - Vous prenez en compte les corrections pour les étapes ultérieures
- La note finale est la somme des notes partielles obtenues à chaque scéance

Echéancier

- Semaine :
  1. Compte rendu d'entretiens et MCC
  2. MCT
  3. VED pour chaque opération
  4. MCD en $3^{ème}$ forme normale
  5. MOT
  6. MPD
  7. Génération d'une base de données
  8. Synthèse

Modèle Conceptuel de Communication (MCC)

- Représente, au niveau conceptuel, les échanges d'information entre les acteurs



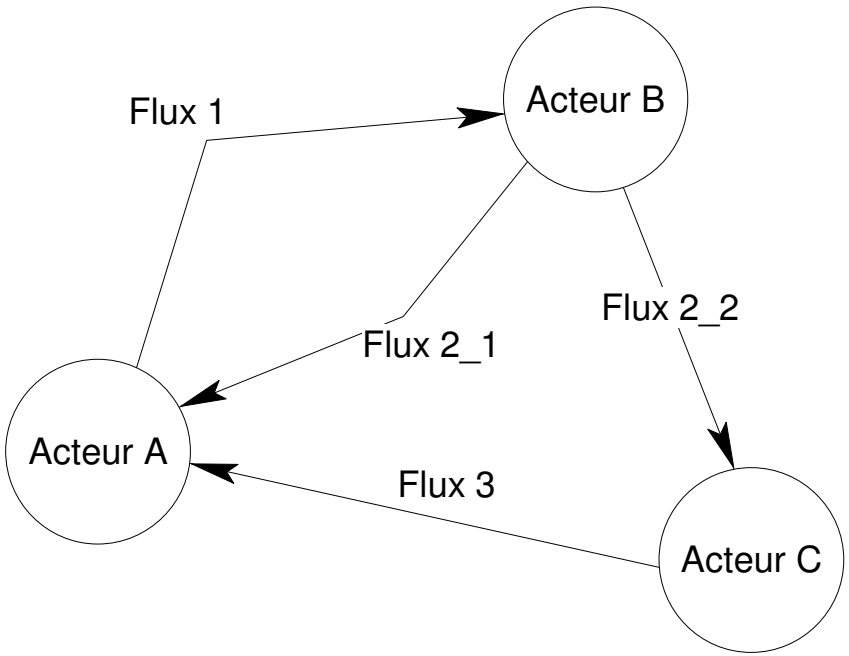
- Première étape d'une étude de l'existant, pour modéliser les habitudes de travail dans l'organisation concernée
  - Délimiter le domaine étudié
  - Réduire la complexité en identifiant des sous problèmes traités individuellement
  - Identifier les acteurs externes et internes
  - Modéliser les échanges d'informations entre les différents acteurs

Acteurs

- Représenté par un cercle libellé par le nom de l'acteur
- L'acteur représente une unité active intervenant dans le fonctionnement d'un système opérant. Il peut
  - Etre stimulé par des flux d'information
  - Transformer et émettre des flux d'information
- Un acteur « fait quelque chose », il est actif
  - Ex : Service comptabilité, Guichet ...
- Un acteur est un rôle plutôt qu'une personne physique (« Direction » et pas « Jean-Claude »)
  - Il peut être pertinent de modéliser séparément deux fonctions assumées par une même personne physique
- On distingue les acteurs internes et externes

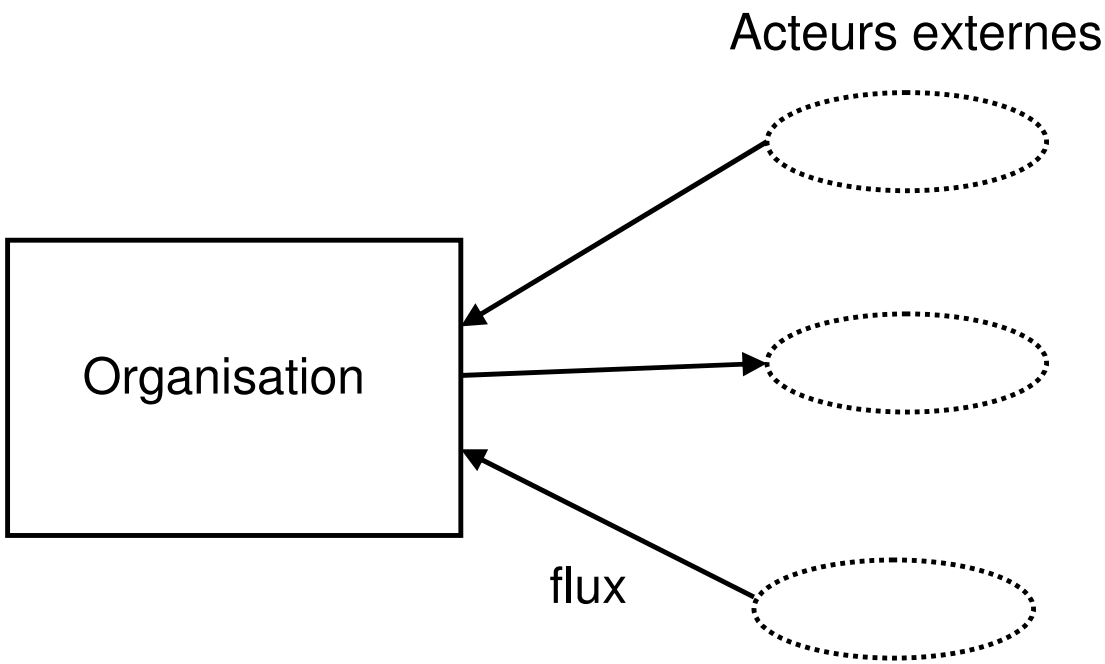
Flux d'information

- Représenté par une flèche entre deux acteurs, étiquetée par le nom du flux
- Echange d'informations entre deux acteurs
  - Ex : documents, appels téléphoniques, données informatiques



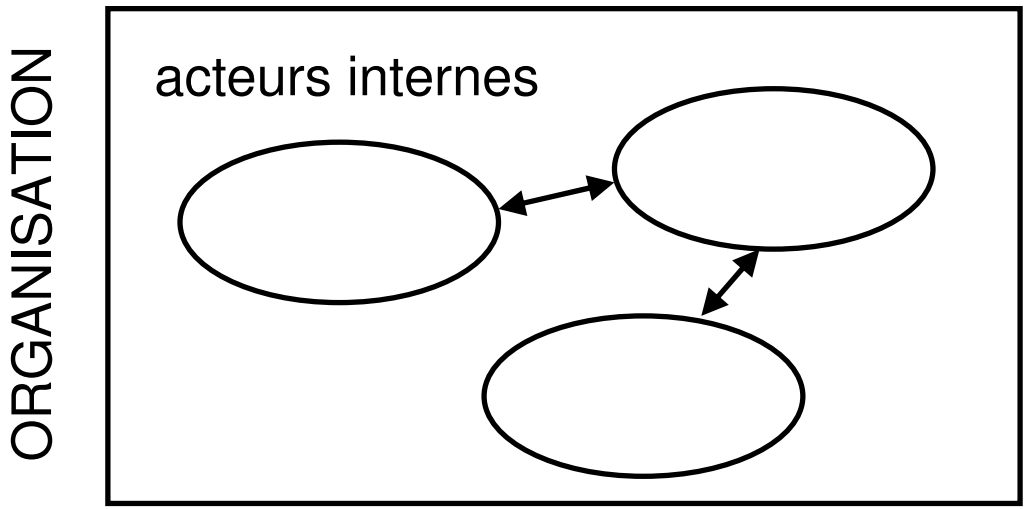
Acteurs externes

- Eléments externes avec lesquels le système échange des flux d'information
  - Ex : clients, fournisseurs...



Acteurs internes

- Acteurs faisant partie du système d'information étudié
  - Ex : guichet, service informatique...
- Si le système est complexe, on peut considérer un acteur interne comme un sous-domaine et détailler ce sous-domaine dans un nouveau MCC



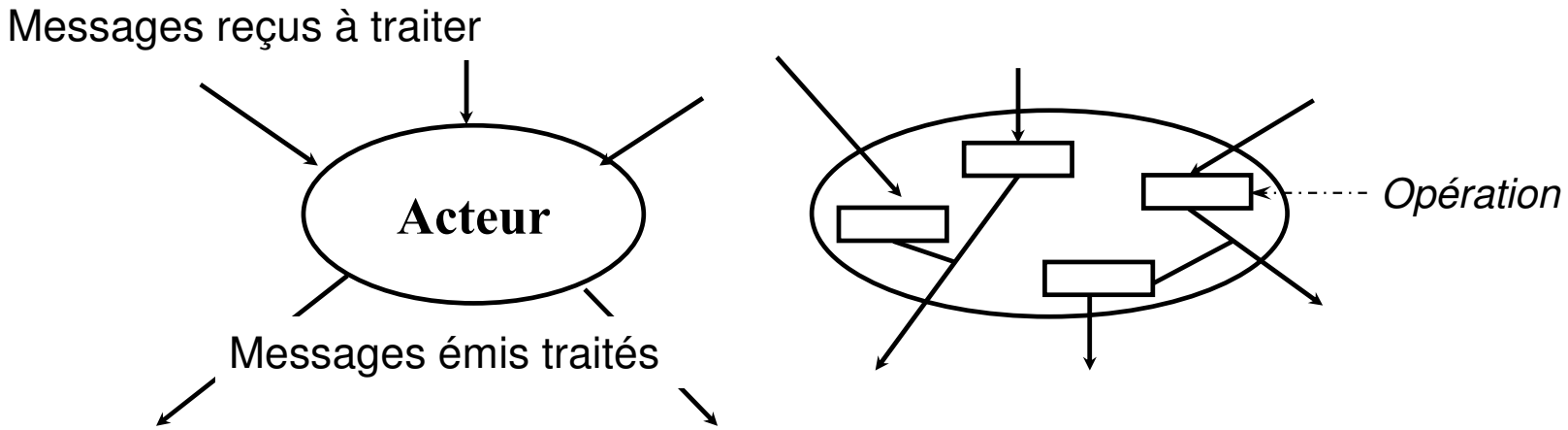
Modèle Conceptuel des Traitements (MCT)

- Représente formellement les activités exercées par le domaine (à la base de la connaissance du SI)
- Repose sur la prise en compte des échanges (flux) du domaine avec son environnement
- S'effectue en faisant abstraction de l'organisation et des choix technologiques

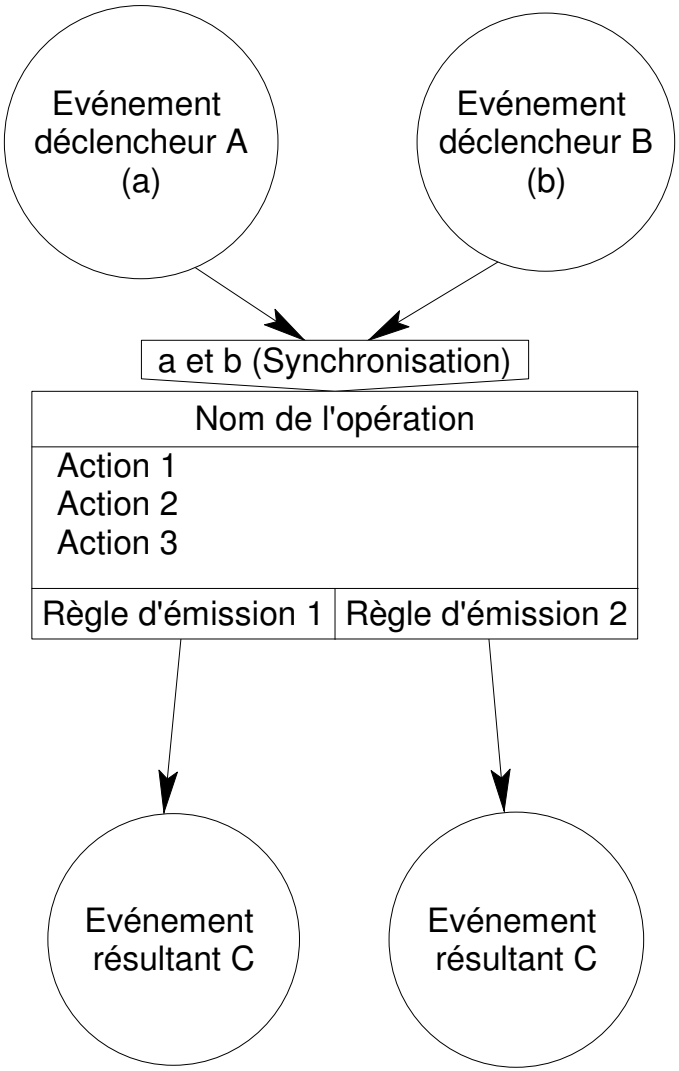
La définition des interactions du domaine avec son environnement prime sur la manière dont on assurera ces activités

MCC et MCT

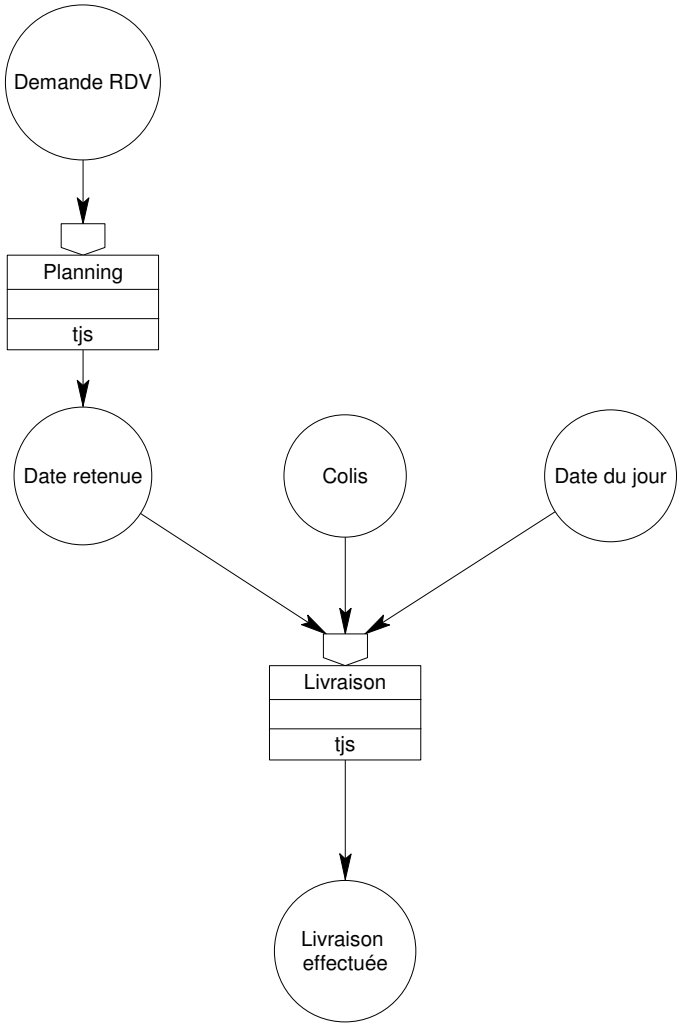
- Le MCT est un « zoom » sur le MCC
  - Dans les MCC, on représente les messages échangés entre acteurs
  - Dans les MCT, on représente comment un acteur de l'organisation réagit quand il reçoit ce message et quelle opération il effectue



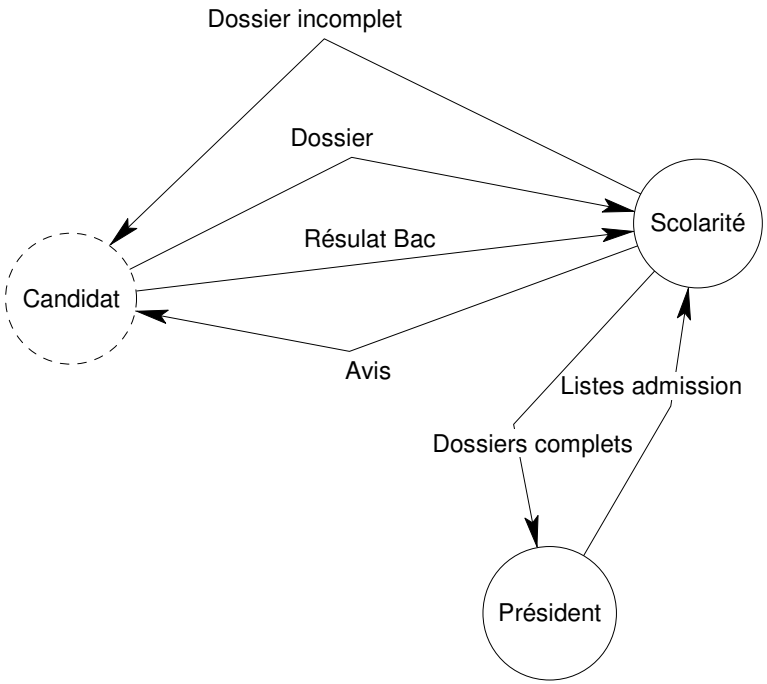
Modèle de MCT

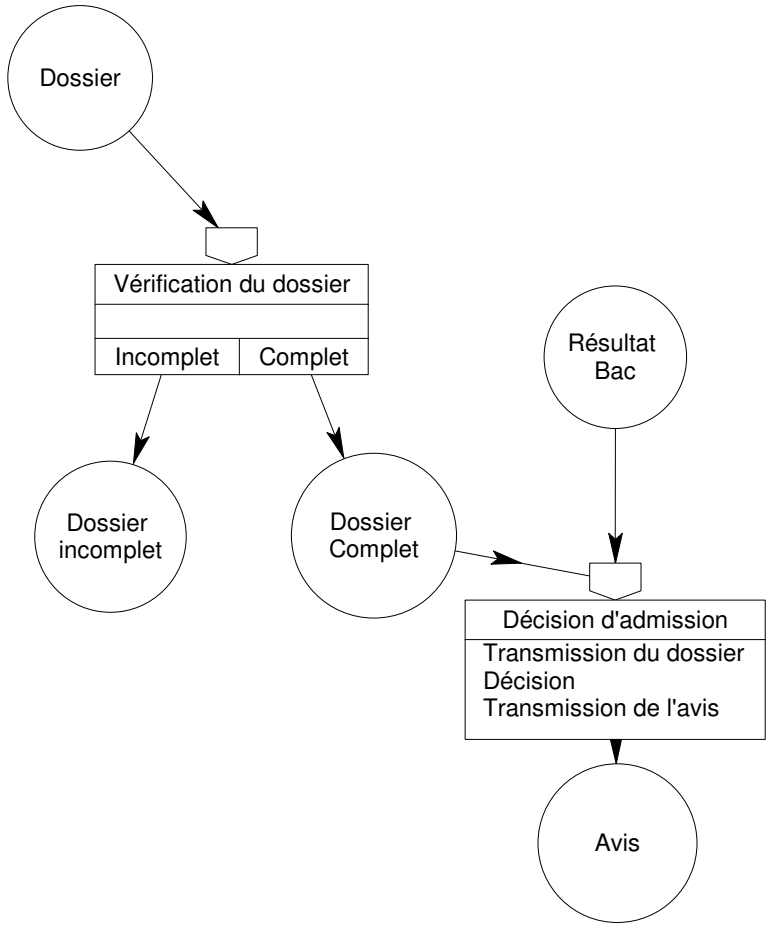


Exemple de MCT



Passage du MCC au MCT

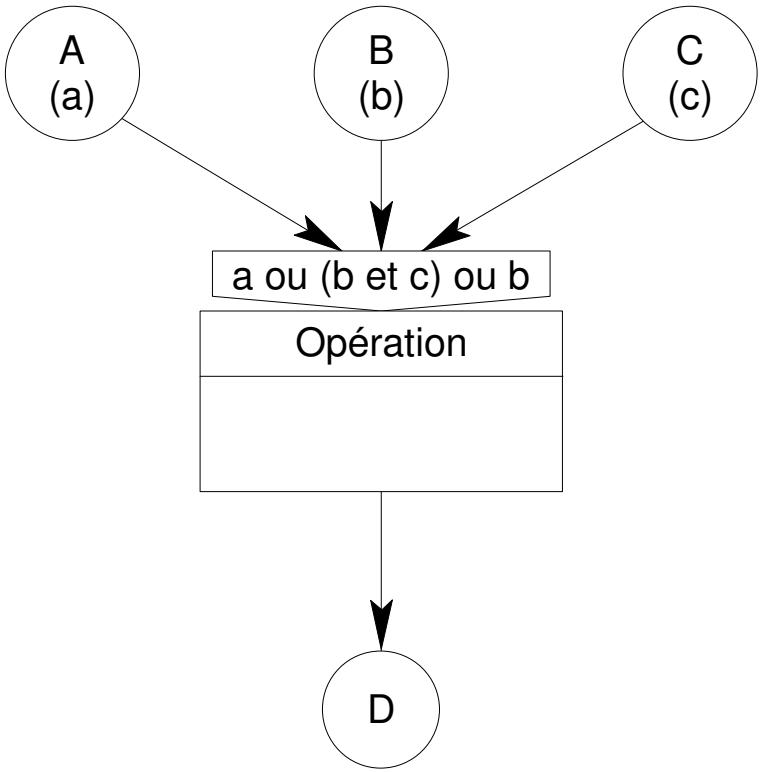




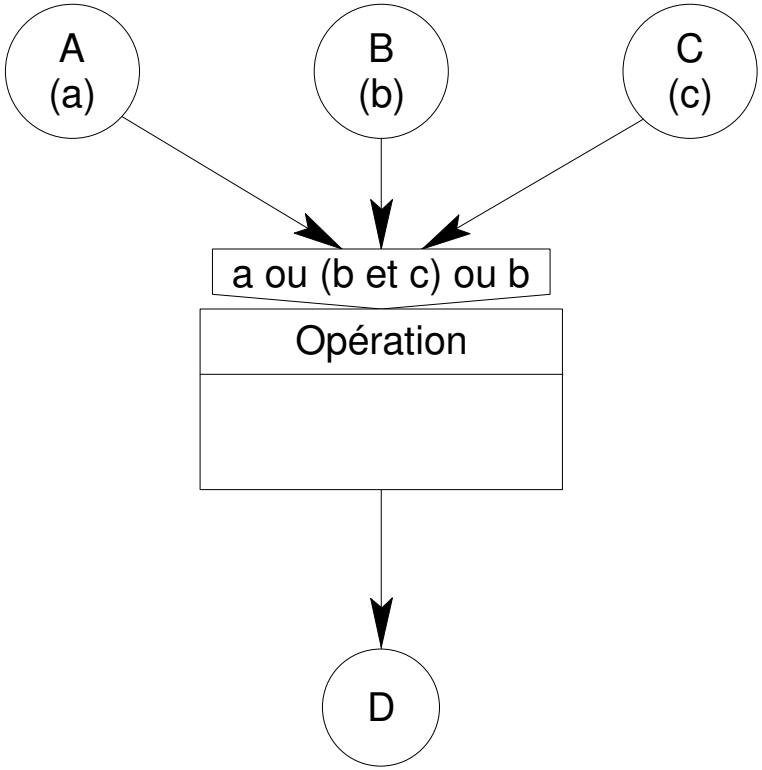
Erreurs de modélisation fréquentes

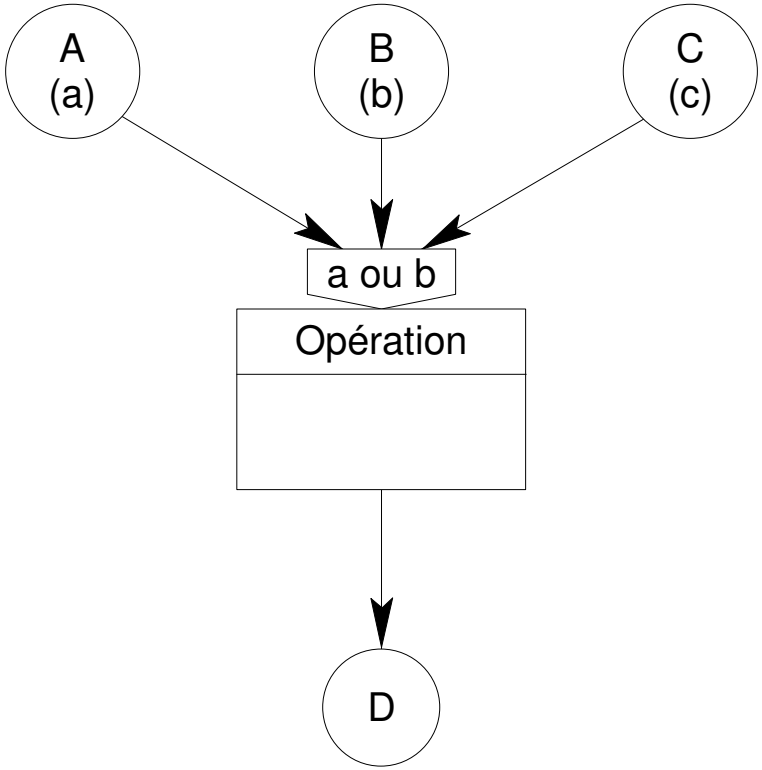
- Règles d'émission : elles doivent
  - Etre mutellement exclusives : deux règles de la même opération ne peuvent pas être vraies en même temps
  - Couvrir tous les cas possibles
- Ne pas répéter les actions et les événements résultants
- Problèmes de synchronisation
  - Il faut simplifier les synchonisations
- Problèmes structurel
  - Il faut éviter les chaînes d'opérations et les événements internes

Simplification des synchronisations



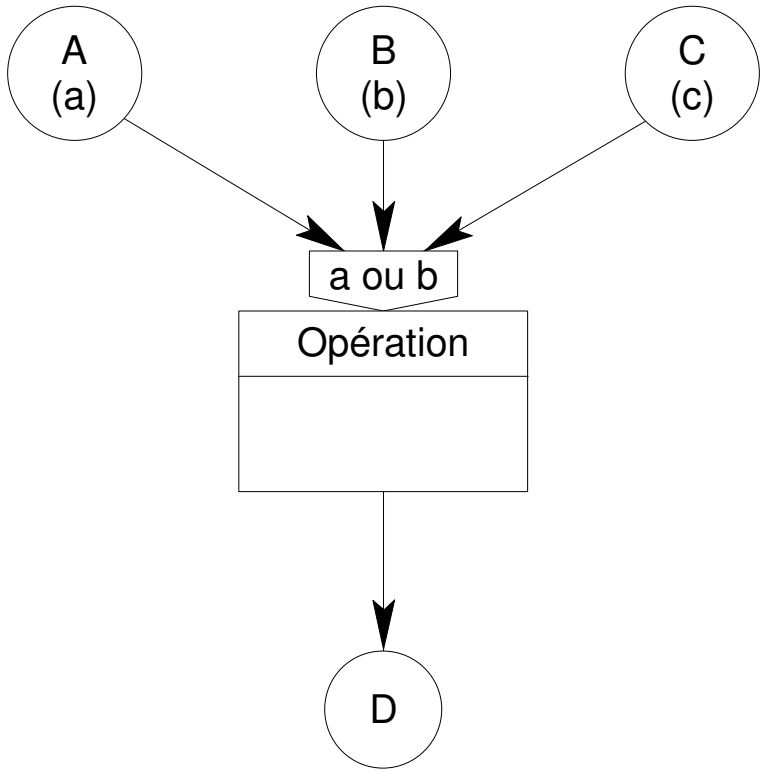
Simplification des synchronisations





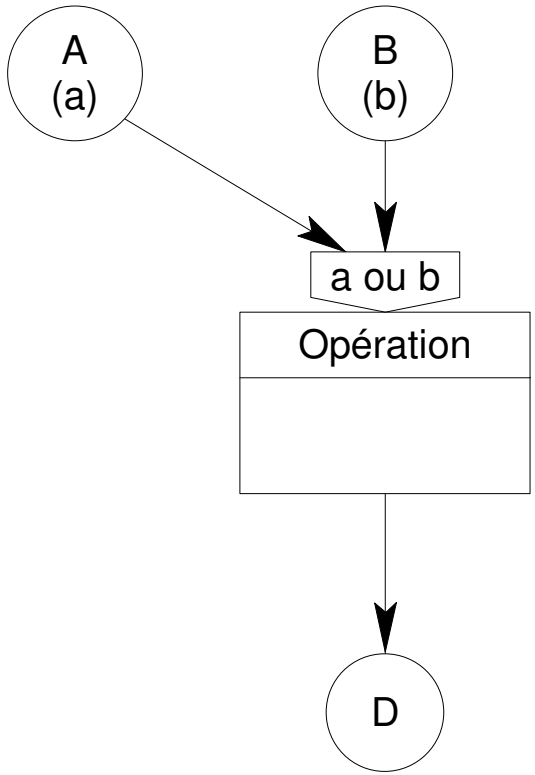
Simplification des synchronisations

La simplification a mis en évidence que C n'était pas nécessaire

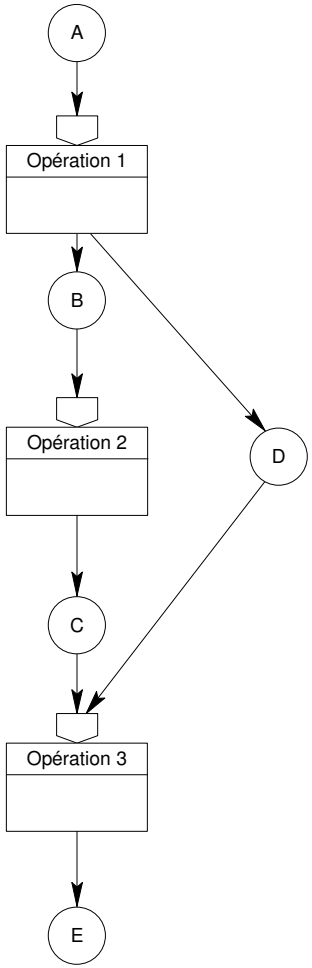


Simplification des synchronisations

La simplification a mis en évidence que C n'était pas nécessaire

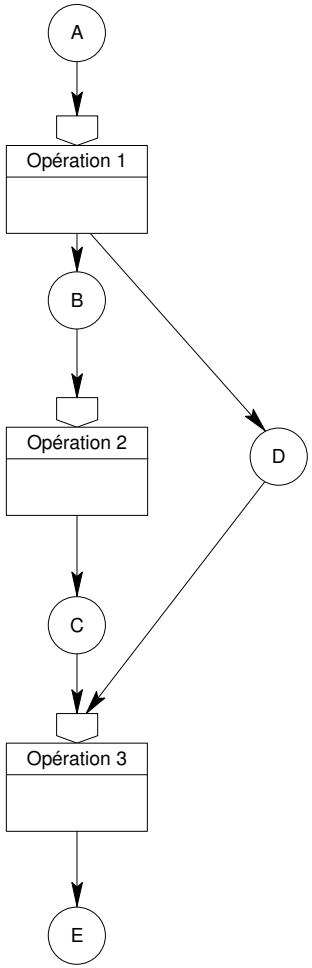


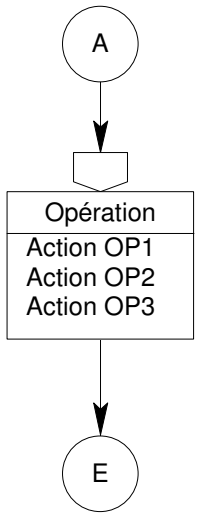
Réduction des chaînes d'opérations



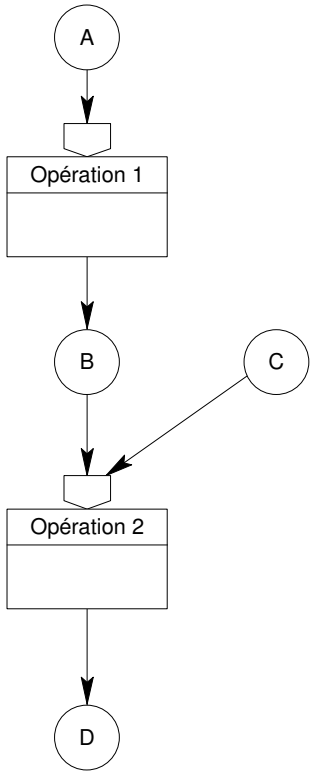
- De A à E, les opérations s'enchaînent de manière systématique
- On supprime les événements internes B, C et D

Réduction des chaînes d'opérations

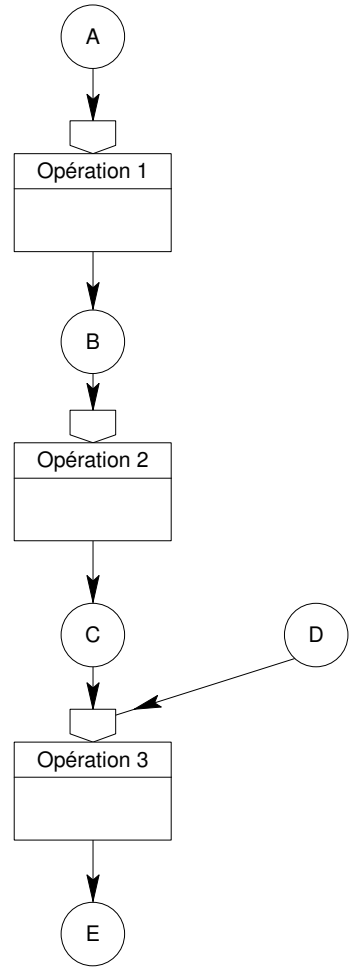




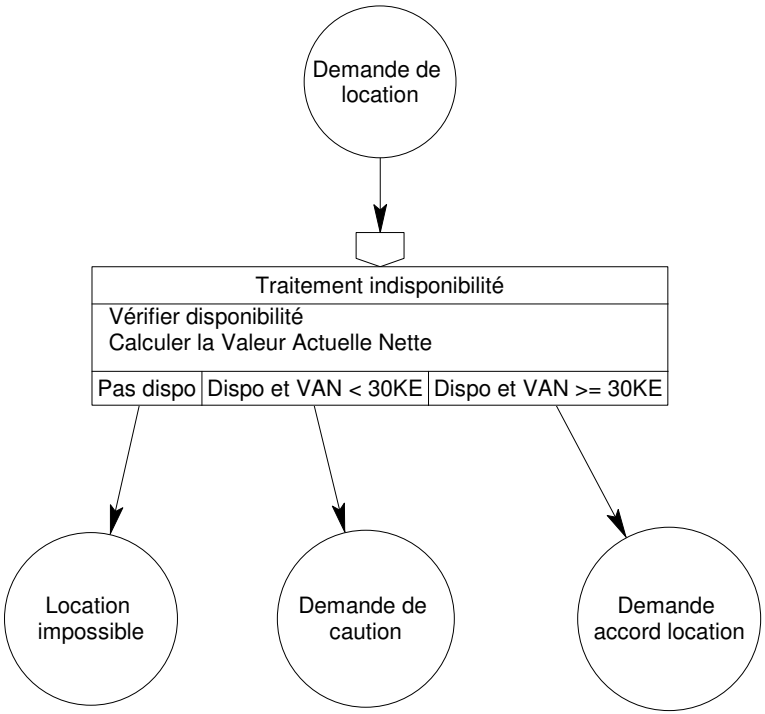
Autres exemples



Chaînes à réduire à une seule opération

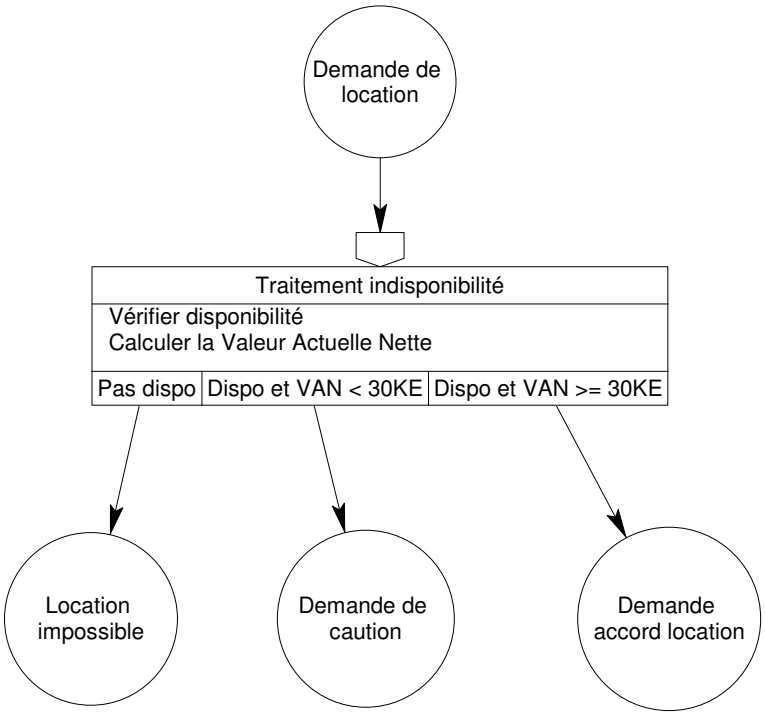


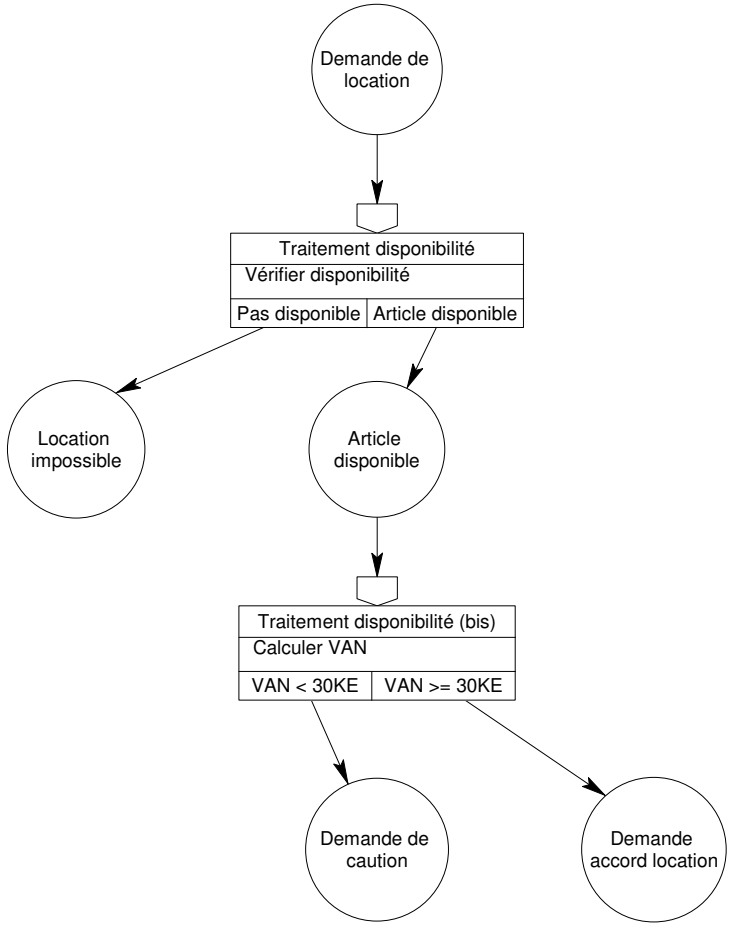
Cas d'introduction d'événements internes



Calculer la VAN ne se fait pas en cas d'indisponibilité

Cas d'introduction d'événements internes





Calculer la VAN ne se fait pas en cas d'indisponibilité

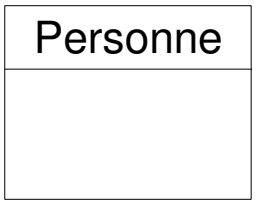
Modèle Conceptuel des Données

- Modèle Entité / Association
  - Souvent nommé Entité-Relation
- Repose sur les concepts de
  - Entités
  - Associations
  - Propriétés
- Permet de décrire un ensemble de données relatives à un domaine défini afin de les intégrer ensuite dans une Base de Données

Entité et entité type

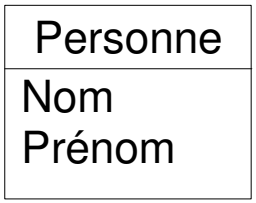
- Entité : Une entité est un objet, une chose concrète ou abstraite qui peut être reconnue distinctement
  - Ex : Jean-Claude, Momo, Ma Voiture, Son 4x4, l'Île de France, la Bretagne
- Entité type : Une entité type est la représentation commune que l'on adopte pour des entités qui possèdent les mêmes caractéristiques
  - Ex : Personne, Voiture, Région

Une entité est une *occurence* d'une entité type (ou instance)



Propriété (ou attribut)

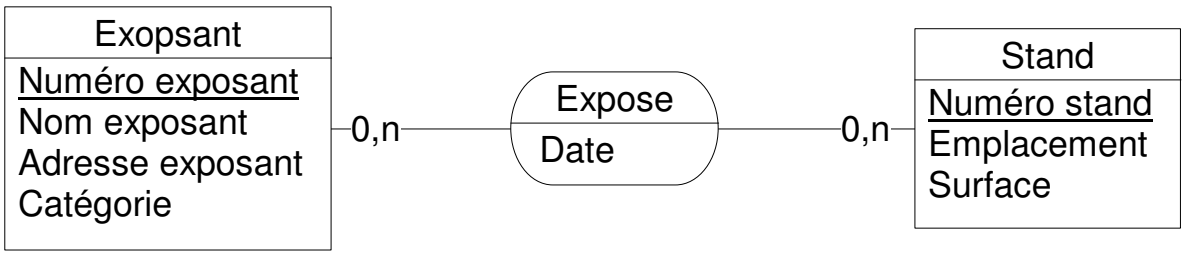
- Propriété : caractéristique associée à une entité type
  - Ex : L'âge d'une personne, la puissance d'une voiture, le numéro d'un produit...
  - On associe un *domaine* à chaque propriété, qui définit l'ensemble des valeurs possibles que peut prendre la propriété
- Valeur : Valeur que prend une propriété (à l'intérieur du domaine) pour une entité particulière
  - Ex : 28 ans pour l'âge de Jean-Claude, 150cv pour la puissance de son 4x4



Association et association type

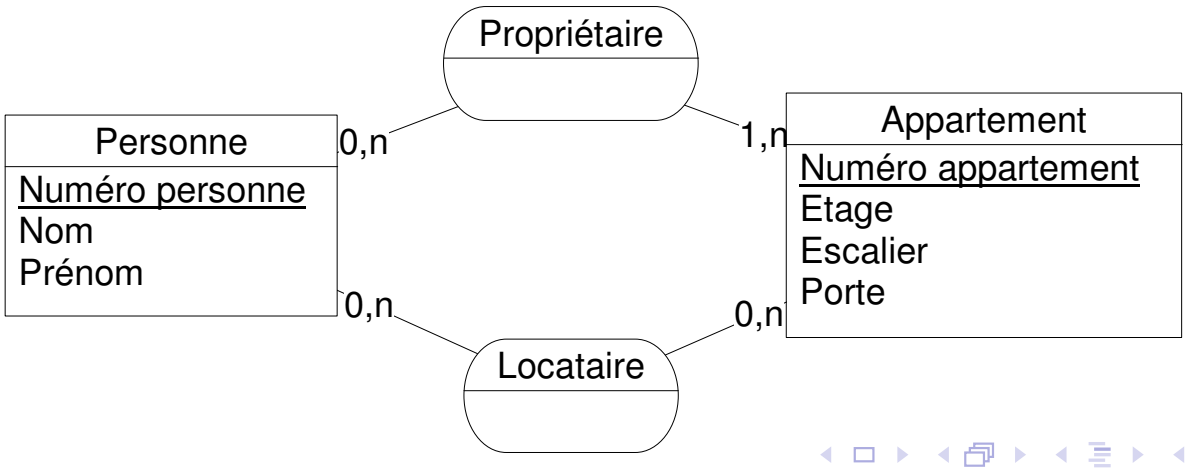
- Association : lien entre plusieurs entités
  - Ex : Le mariage de Momo et de Jeanne, celui de Jean-Claude et d'Eglantine
- Association type : représentation d'un ensemble de relations qui possèdent les mêmes caractéristiques, lien entre plusieurs entités type
  - Ex : Le mariage de deux personnes

- Une association type peut avoir des propriétés



Association et association type

- Association : lien entre plusieurs entités
  - Ex : Le mariage de Momo et de Jeanne, celui de Jean-Claude et d'Eglantine
- Association type : représentation d'un ensemble de relations qui possèdent les mêmes caractéristiques, lien entre plusieurs entités type
  - Ex : Le mariage de deux personnes
- Il peut y avoir plusieurs associations type liant les mêmes entités si la sémantique est différente



Abus de langage

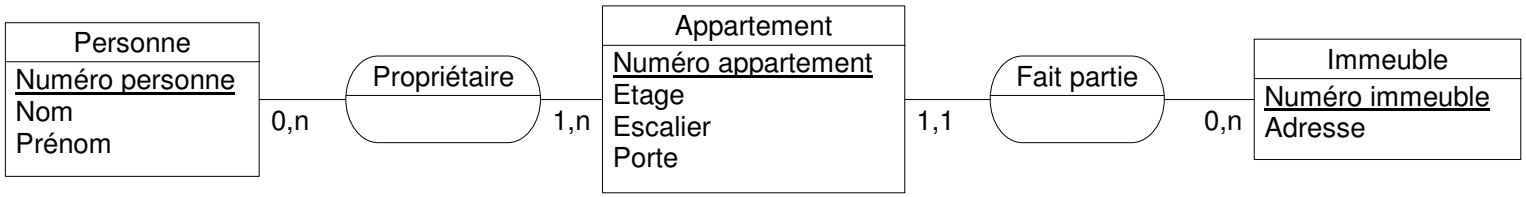
- Souvent, on parle d'« Entité » à la place d'« Entité Type ». Dans la suite, comme c'est d'usage, nous utiliserons les termes :
  - *Entité* pour *entité type*
  - *Occurence d'entité* pour *entité*
- De même, on utilise souvent « Association » plutôt que « Association Type ». Dans la suite, comme c'est d'usage, nous utiliserons les termes :
  - *Association* pour *Association type*
  - *Occurence d'association* pour *Association*

Identifiants

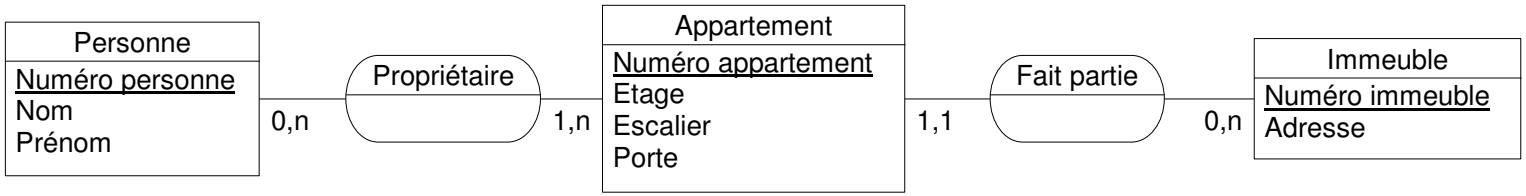
- Identifiant : une ou plusieurs propriétés d'une entité ou d'une association qui ont une valeur unique pour chaque occurence de l'entité ou de l'association
  - Ex : Le numéro de SECU d'une personne, le numéro d'immatriculation d'une voiture...
  - On souligne les identifiants d'une entité
  - L'identifiant d'une association est un sous-ensemble des identifiants des entités liés

Cardinalités

- Cardinalité d'une association : le nombre de fois minimal et maximal qu'une occurence d'une des entités associée peut intervenir dans l'association
  - Ex : un client peut commander entre 1 et n produits



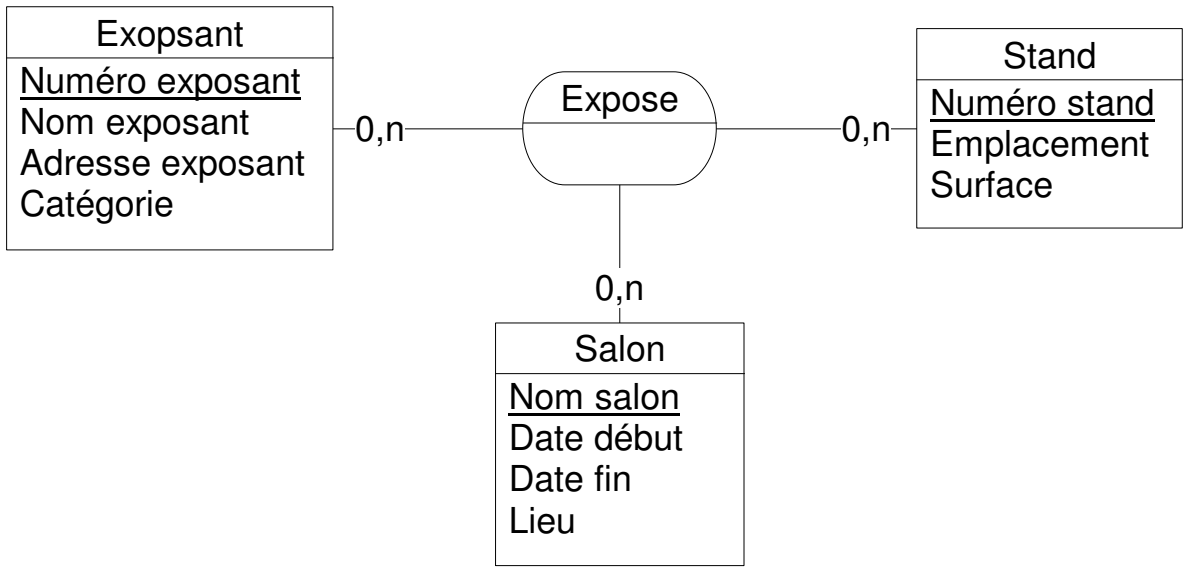
Cardinalités



- Cardinalité minimale
  - 0 si une occurence de l'entité peut exister tout en n'intervenant dans aucune occurence de l'association
  - 1 si une occurence de l'entité ne peut exister que si elle intervient dans au moins une occurence de l'association
  - n : cas rare à éviter
- Cardinalité maximale
  - 1 si une occurence de l'entité ne peut pas être impliquée dans plus d'une occurence de l'association
  - n si une occurence de l'entité ne peut être impliquée dans plus d'une occurence de l'association

Dimension d'une association

- Dimension : Nombre de « pattes » de l'association
  - Binaire, *ternaire* ou n-aire

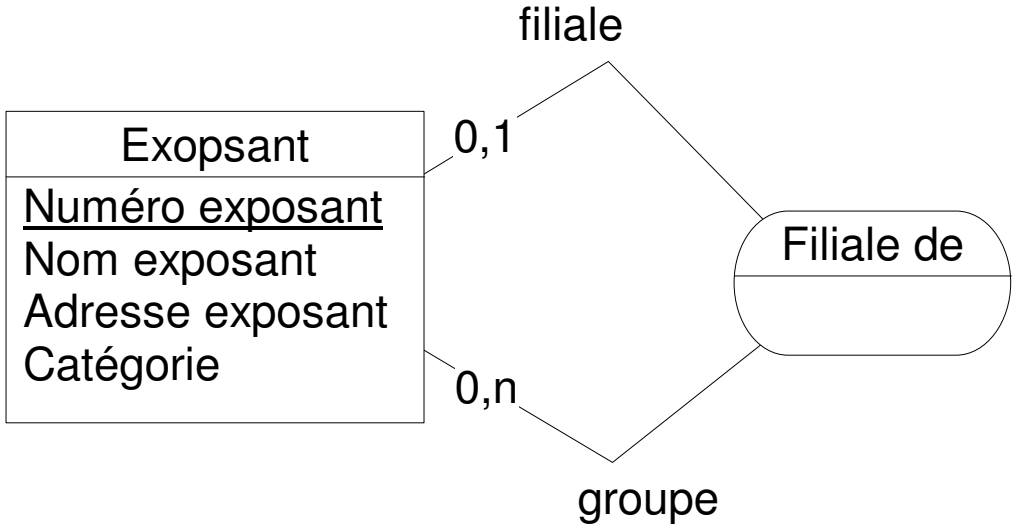


Types d'asociations

- En fonction des cardinalités
  - 1:1 si toutes la cardinalités maximales valent 1
  - 1:n s'il existe au moins une cardinalité maximale à n et une à 1
  - n:m si toutes la cardinalités maximales valent n

Associations réflexives

- Association réflexive : Une association dont plusieurs « pattes » lient la même entité. Dans ce cas, plusieurs occurences de la même entité seront associées



- On peut libeller chaque « pattes » par son rôle dans l'association

Remarques

- Il est parfois difficile de faire un choix entre entité et association
  - Ex : Un mariage est-il une association entre deux personnes ou une entité pour lequel on veut conserver un numéro, une date, un lieu, etc. et que l'on souhaite manipuler en tant que tel ?
  - Souvent, le contexte aide à décider
    - Lorsqu'on ne parvient pas à trouver d'identifiant pour une entité, il faut se demander s'il ne s'agit pas en fait d'une association. Si ce n'est pas le cas, un identifiant arbitraire numérique entier peut faire l'affaire
    - Lorsque toutes les pattes d'une association portent la cardinalité 11, il faut se demander si ce type-association et les types-entités liés ne décrivent pas en fait un seul type-entité

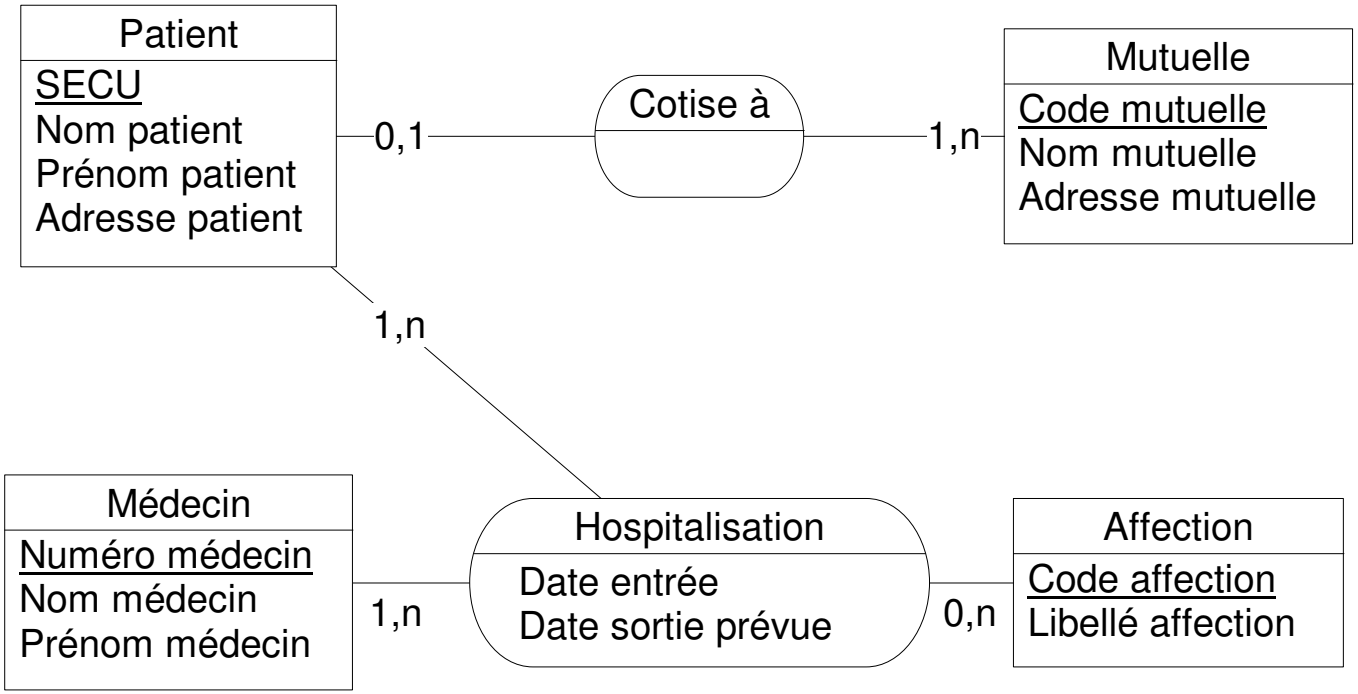
Cohérence entre données et traitements

- A chaque opération, on associe un MCD partiel : une Vue Externe des Données
  - On s'assure ainsi que toutes les données nécessaires sont représentées
- Le MCD global est l'union de toutes les VED
- Pour chaque élément du MCD global, on vérifie que celui-ci est utilisé dans au moins une opération
  - On s'assure ainsi que seules les données nécessaires sont représentées
- On s'appuie souvent sur des documents existants pour réaliser les VED

Dépendances fonctionnelles

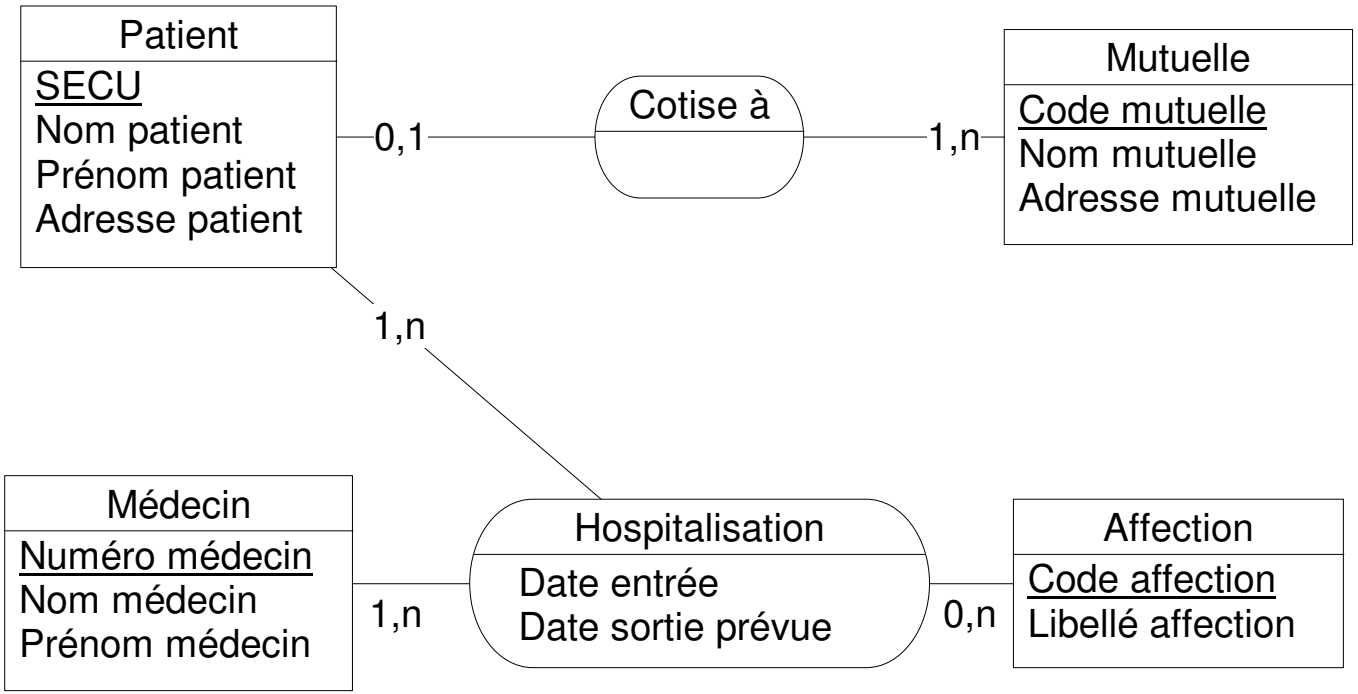
- Une propriété (ou un groupe de propriétés) Y dépend fonctionellement d'une autre propriété (ou groupe de propriétés) X si
  - Etant donné une valeur de X, il lui correspond une valeur unique de Y. On note
  - $X \rightarrow Y$ (X détermine Y)
- Cette relation est transitive : si $X \rightarrow Y$ et $Y \rightarrow Z$ alors $X \rightarrow Z$
  - Cependant, on ne représente que les DF élémentaires

Dépendances fonctionnelles



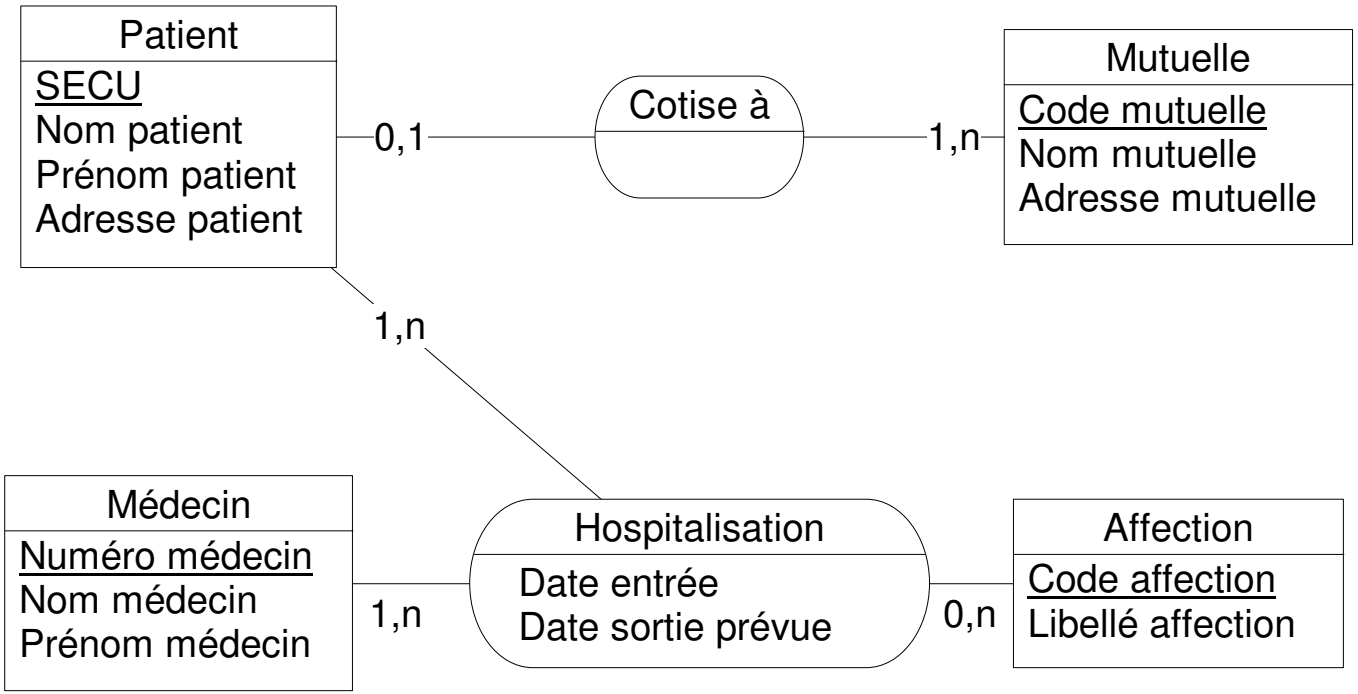
- Les propriétés non identifiantes d'une entité dépendent fonctionnellement de l'ensemble des identifiants
  - Ex : SECU $\rightarrow$ NomPatient, PrénomPatient, AdressePatient

Dépendances fonctionnelles



- L'identifiant d'une association de type nm dépend fonctionnellement des identifiants des entités liées
  - Ex : SECU, NuméroMédecin, CodeAffection → DateEntrée, DateSortie

Dépendances fonctionnelles



- Une cardinalité 11 ou 01 est la source d'une dépendance fonctionnelle de l'identifiant du côté 11 vers l'autre côté de l'association
  - Ex : SECU $\rightarrow$ CodeMutuelle

1ère Forme Normale (1FN)

- Toutes les entités et les association possèdent un identifiant
- Aucune propriété n'est à valeurs multiples (popriétés atomiques)

1ère Forme Normale (1FN)

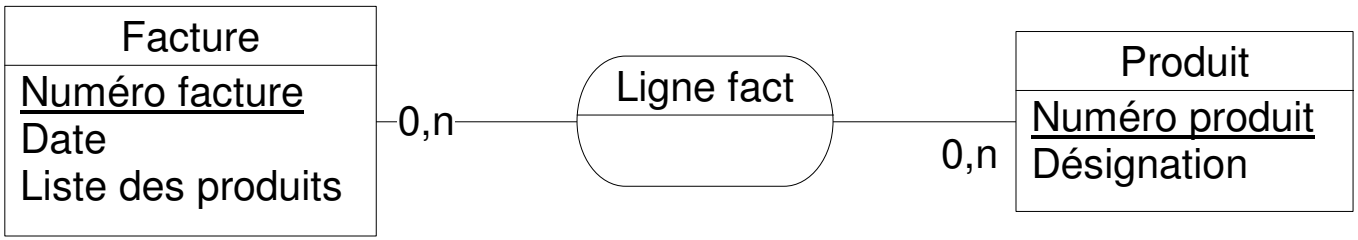
- Toutes les entités et les association possèdent un identifiant
- Aucune propriété n'est à valeurs multiples (popriétés atomiques)

Facture
Numéro
Date
Liste des produits

- Ici, « liste des produits » n'est pas atomique, c'est une liste

1ère Forme Normale (1FN)

- Toutes les entités et les association possèdent un identifiant
- Aucune propriété n'est à valeurs multiples (popriétés atomiques)

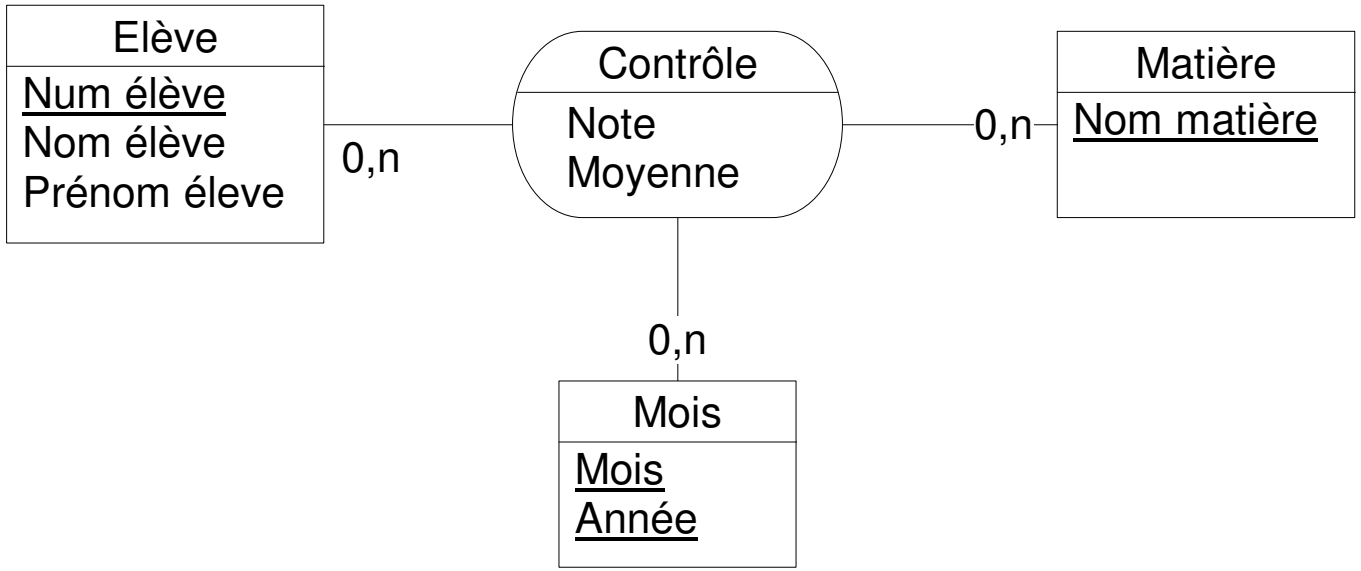


2e Forme Normale (2FN)

- Le modèle est en 1FN
- Toutes les DF entre les propriétés sont élémentaires
  - Toute propriété n'appartenant pas à une clé ne dépend pas seulement d'une partie de son identifiant identifiant
  - Les propriétés d'une entité ne doivent dépendre que de l'identifiant de l'entité et non d'une partie de cet identifiant

2e Forme Normale (2FN)

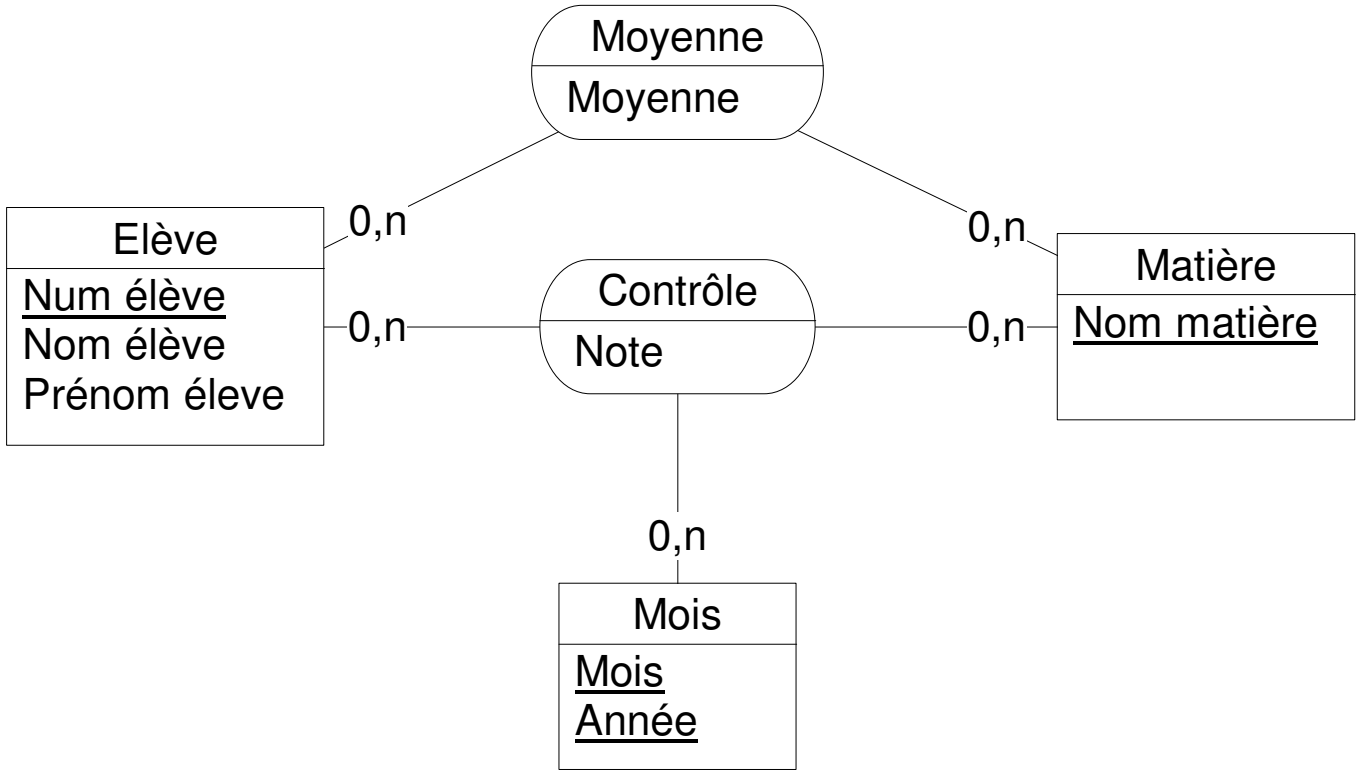
- Le modèle est en 1FN
- Toutes les DF entre les propriétés sont élémentaires



- Ici, d'après le schéma,
  NumEleve, NomMatière, Mois, Année → *Moyenne*
- Or, NumEleve et NomMatière suffisent

2e Forme Normale (2FN)

- Le modèle est en 1FN
- Toutes les DF entre les propriétés sont élémentaires



3e Forme Normale (3FN)

- Le modèle est en 2FN
- Toutes les DF entre les propriétés sont directes
  - Les propriétés d'une entité doivent dépendre de l'identifiant de l'entité de manière directe
  - Toute propriété n'appartenant pas à un identifiant ne dépend pas d'un attribut non identifiant

3e Forme Normale (3FN)

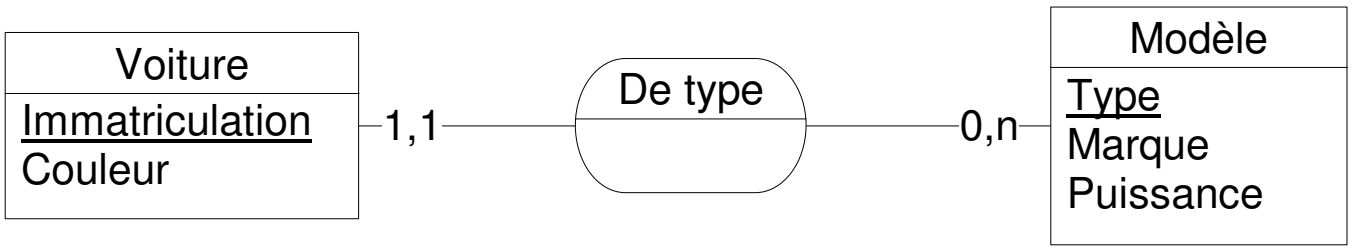
- Le modèle est en 2FN
- Toutes les DF entre les propriétés sont directes

Voiture
Immatriculation
Couleur
Type
Puissance
Marque

- Or, *Type* $\rightarrow$ *Marque*, *Puissance* alors que Type n'est pas un identifiant

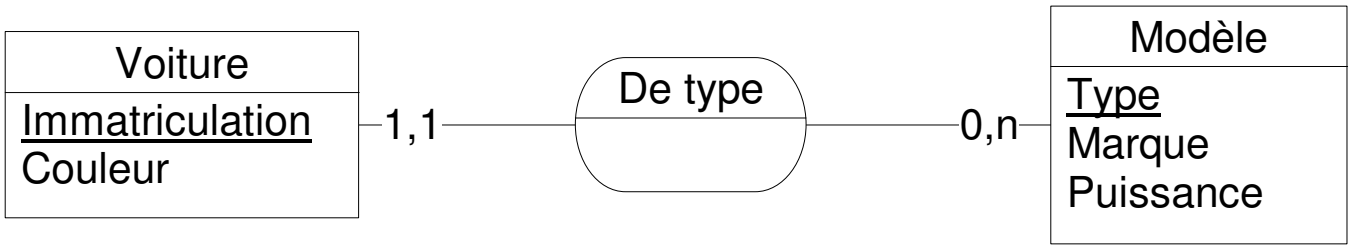
3e Forme Normale (3FN)

- Le modèle est en 2FN
- Toutes les DF entre les propriétés sont directes



3e Forme Normale (3FN)

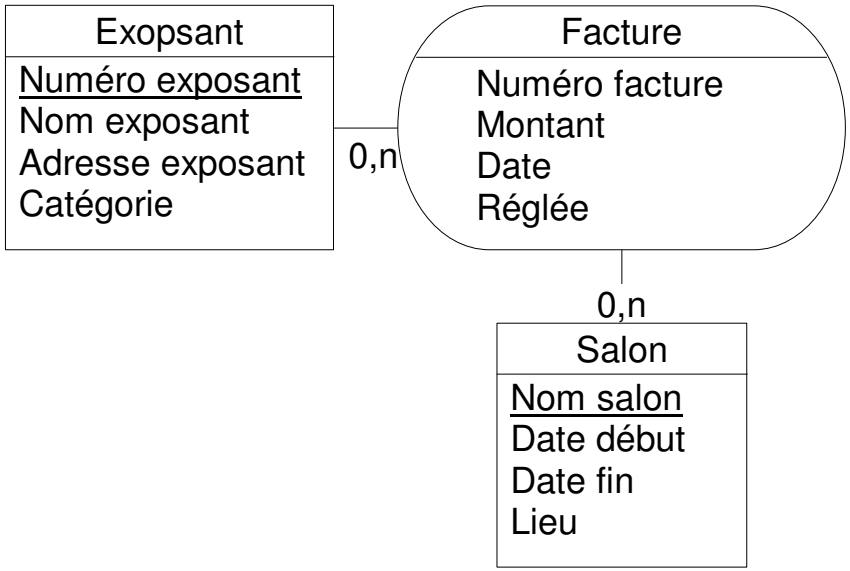
- Le modèle est en 2FN
- Toutes les DF entre les propriétés sont directes



- Très bien mais si on voulait rajouter un numéro de facture...

3e Forme Normale (3FN)

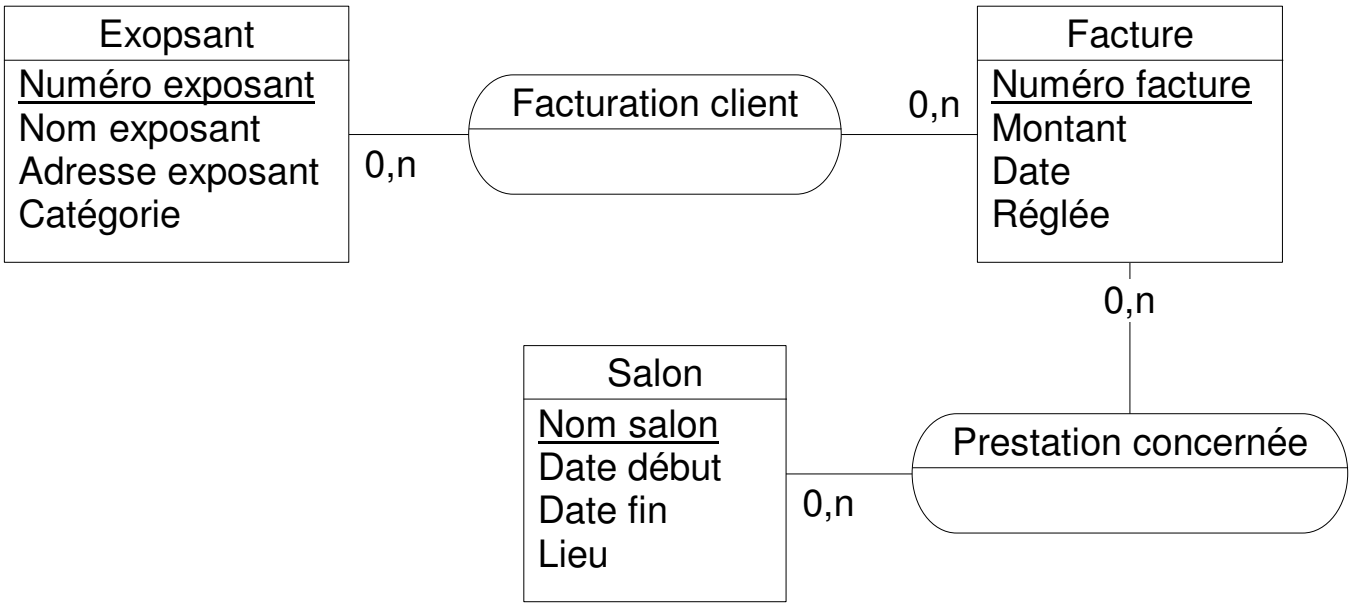
- Le modèle est en 2FN
- Toutes les DF entre les propriétés sont directes



- Or, *NumFact* → *Montant, Date, Réglée* alors que NumFact n'est pas un identifiant

3e Forme Normale (3FN)

- Le modèle est en 2FN
- Toutes les DF entre les propriétés sont directes



Forme normale de Boyce-Codd (BCNF)

- Le modèle est en 3FN
- Les seules dépendances fonctionnelles élémentaires sont celles dans lesquelles un identifiant détermine une propriété
  - Pour les identifiants composés de plusieurs propriétés, ces dernières ne doivent pas être dépendantes d'une autre propriété de l'entité (pour éviter les cycles de DF)

Forme normale de Boyce-Codd (BCNF)

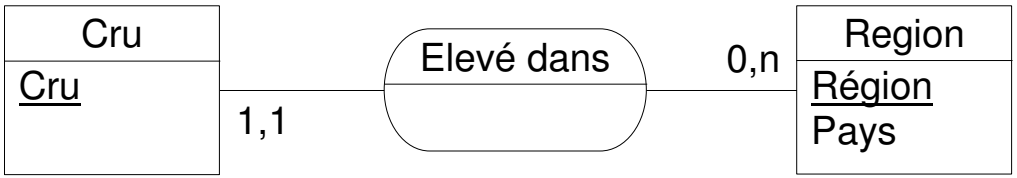
- Le modèle est en 3FN
- Les seules dépendances fonctionnelles élémentaires sont celles dans lesquelles un identifiant détermine une propriété

Vin
Cru Pays Région

- Or, *Région* $\longrightarrow$ *Pays*

Forme normale de Boyce-Codd (BCNF)

- Le modèle est en 3FN
- Les seules dépendances fonctionnelles élémentaires sont celles dans lesquelles un identifiant détermine une propriété



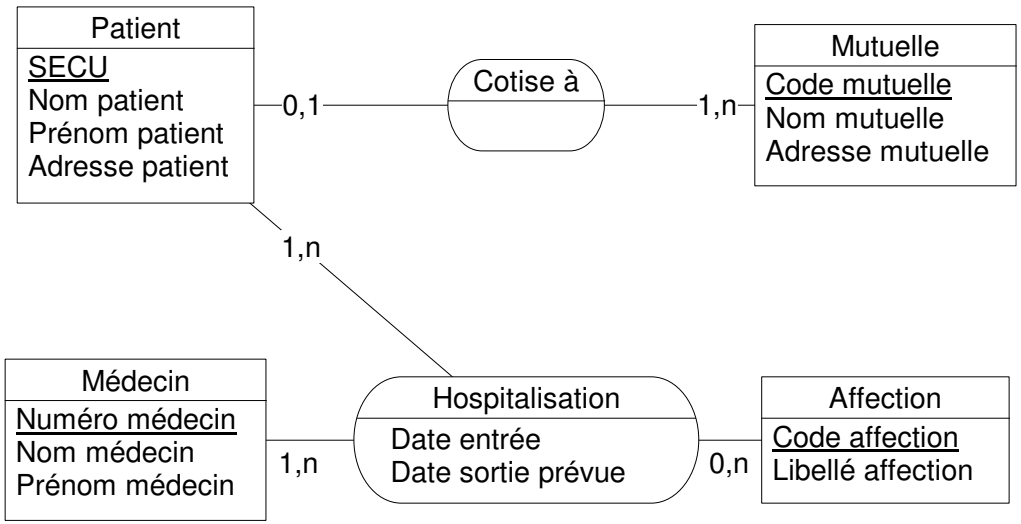
- On a alors, *Cru* $\longrightarrow$ *Region* et *Region* $\longrightarrow$ *Pays*
- Attention : Même si elle peut être retrouvée par jointure, on a perdu la dépendance *Cru, Pays* $\longrightarrow$ *Région*

Un MCD ne doit pas nécéssairement être en BCNF, il faut peser le pour et le contre avant de perdre des dépendances fonctionnelles

Modèle relationnel

- Modèle relationnel : Ensemble de schémas relationnels de la forme *Relation(clé1, ... clén, att1, ... attm)*

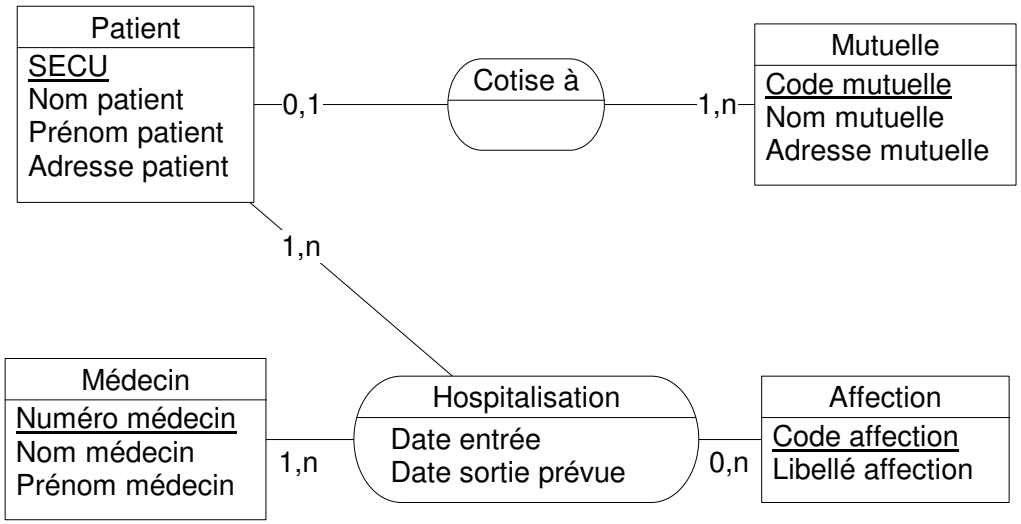
Passage du MCD au MLD relationnel



- Règle 1 : Chaque entité avec au moins une propriété non identifiante donne lieu à un schéma relationnel, les identifiants deviennent les clés

Patient *(SECU, NomPatient, PrenomPatient, AdressePatient)*
Médecin *(NuméroMédecin, NomMédecin, PrénomMédecin)*
Mutuelle *(CodeMutelle, NomMutuelle, AdresseMutuelle)*
Affection *(CodeAffection, LibelléAffection)*

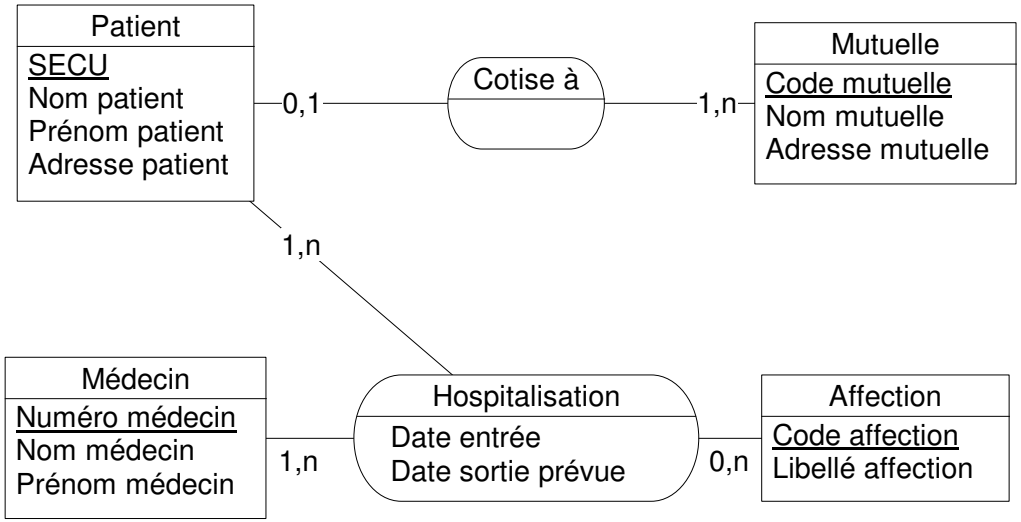
Passage du MCD au MLD relationnel



- Règle 2 : Les associations de type 1:n donnent lieu à l'ajout de l'identifiant côté 1 vers le côté n, en tant qu'attribut non-clé)

Patient *(SECU, NomPatient, ... , CodeMutuelle)*

Passage du MCD au MLD relationnel



- Règle 3 : Les associations de type n:m donnent lieu à la création de nouveaux schémas relationnels
  - Les identifiants des entités liées deviennent des clés
  - Les propriétés de l'association deviennent des attributs simples

Hospitalisation (*NuméroMedecin*, *SECU*, *CodeAffection*, *DateEntrée*, *DateSortie*)

Modèle Organisationnel des Traitements

$$\text{MOT} = \text{MCT} + \text{lieu} + \text{moment} + \text{nature}$$

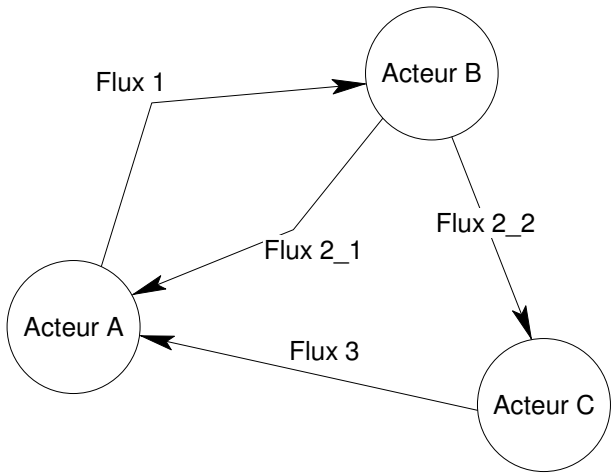
- Lieu
  - Qui exécute ? Acteurs (MCC)
- Moment
  - Quand exécute-t-on l'opération ?
  - Agencement temporel
- Nature
  - Manuelle
  - Automatique
  - Interactive

Du MCT au MOT

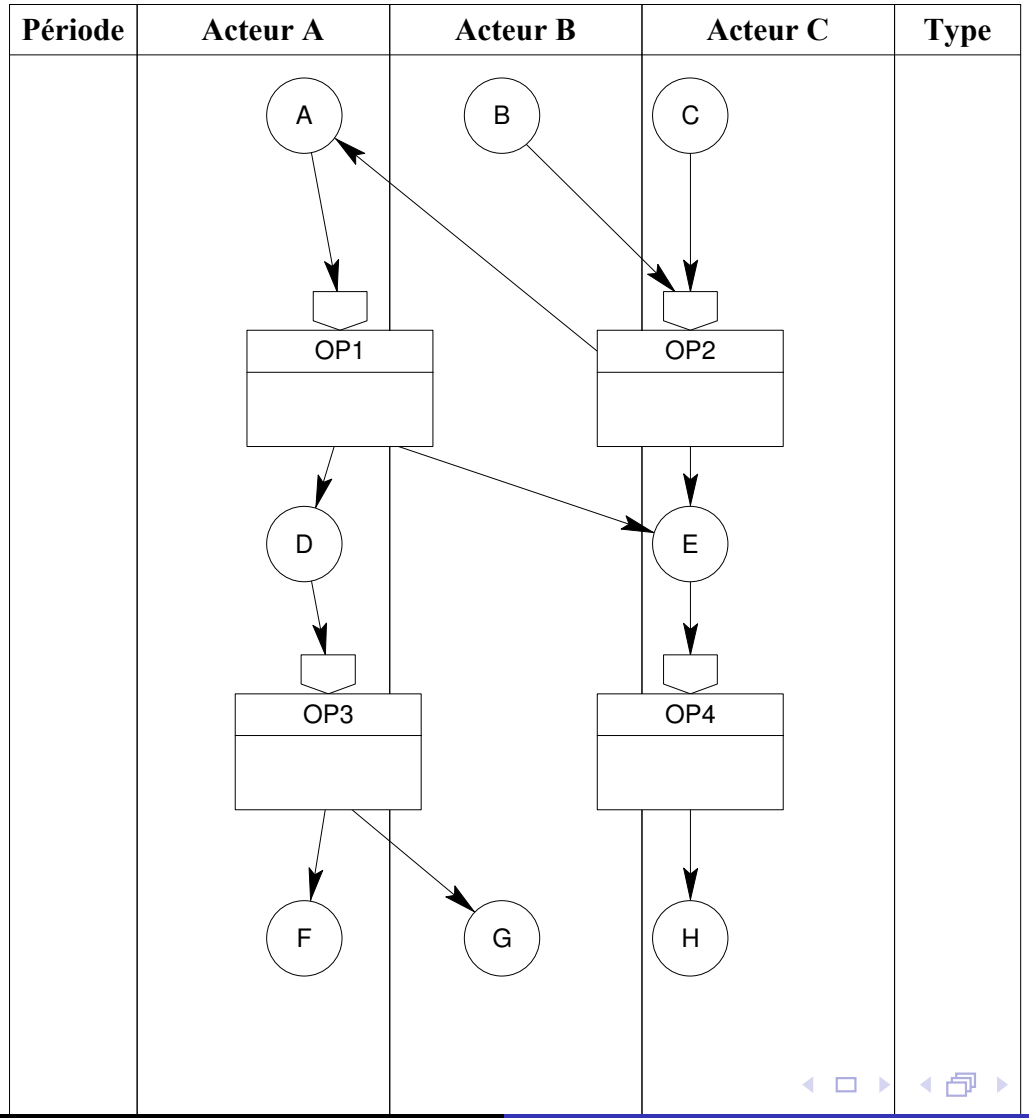
1. Importer la liste des acteurs du MCC
2. Importer le MCT

Import de la liste des acteurs

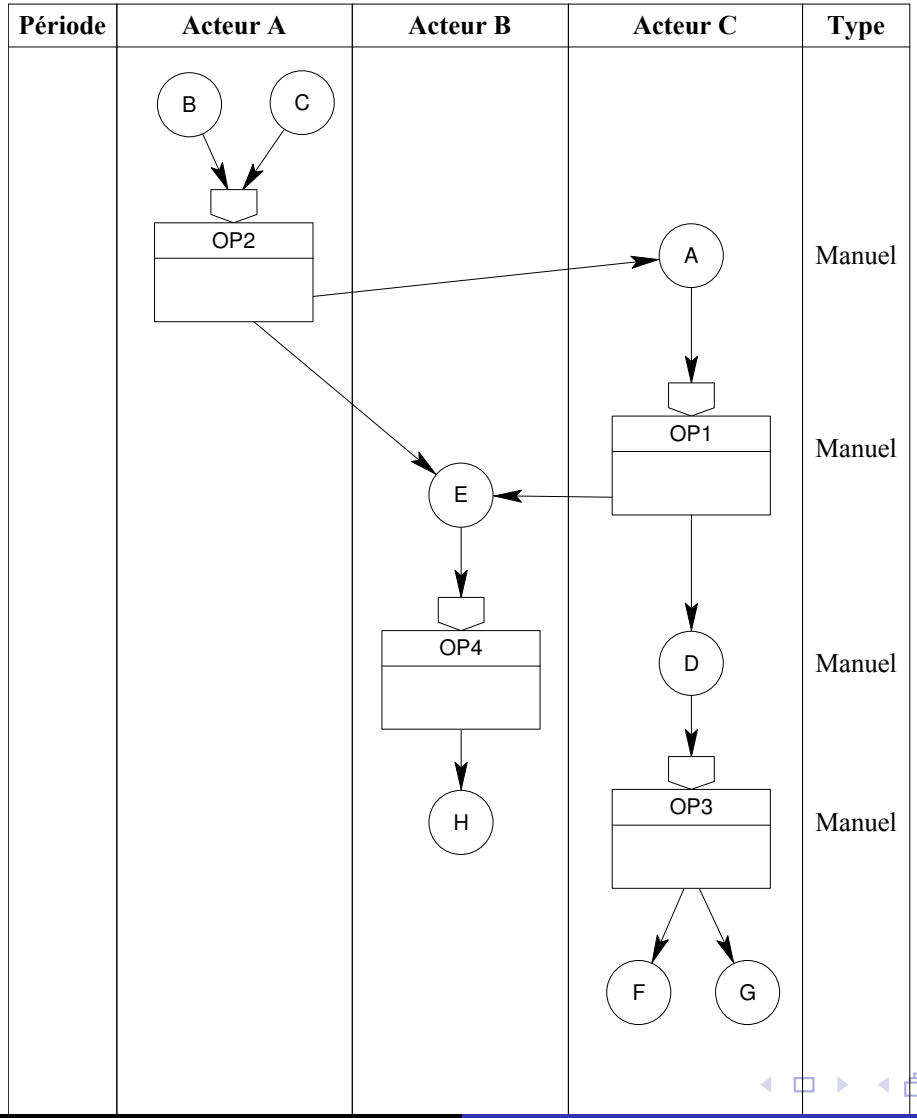
Période	Acteur A	Acteur B	Acteur C	Type



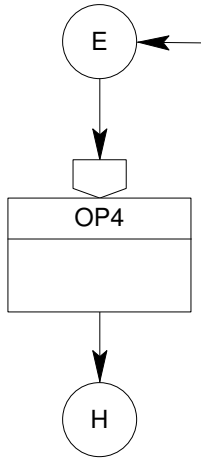
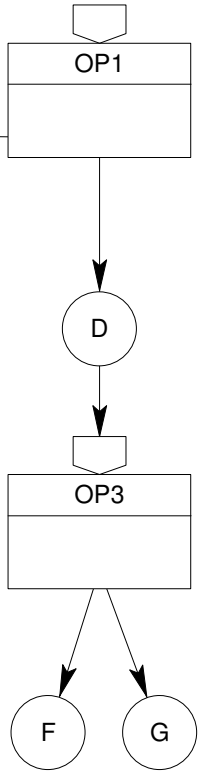
Import du MCT



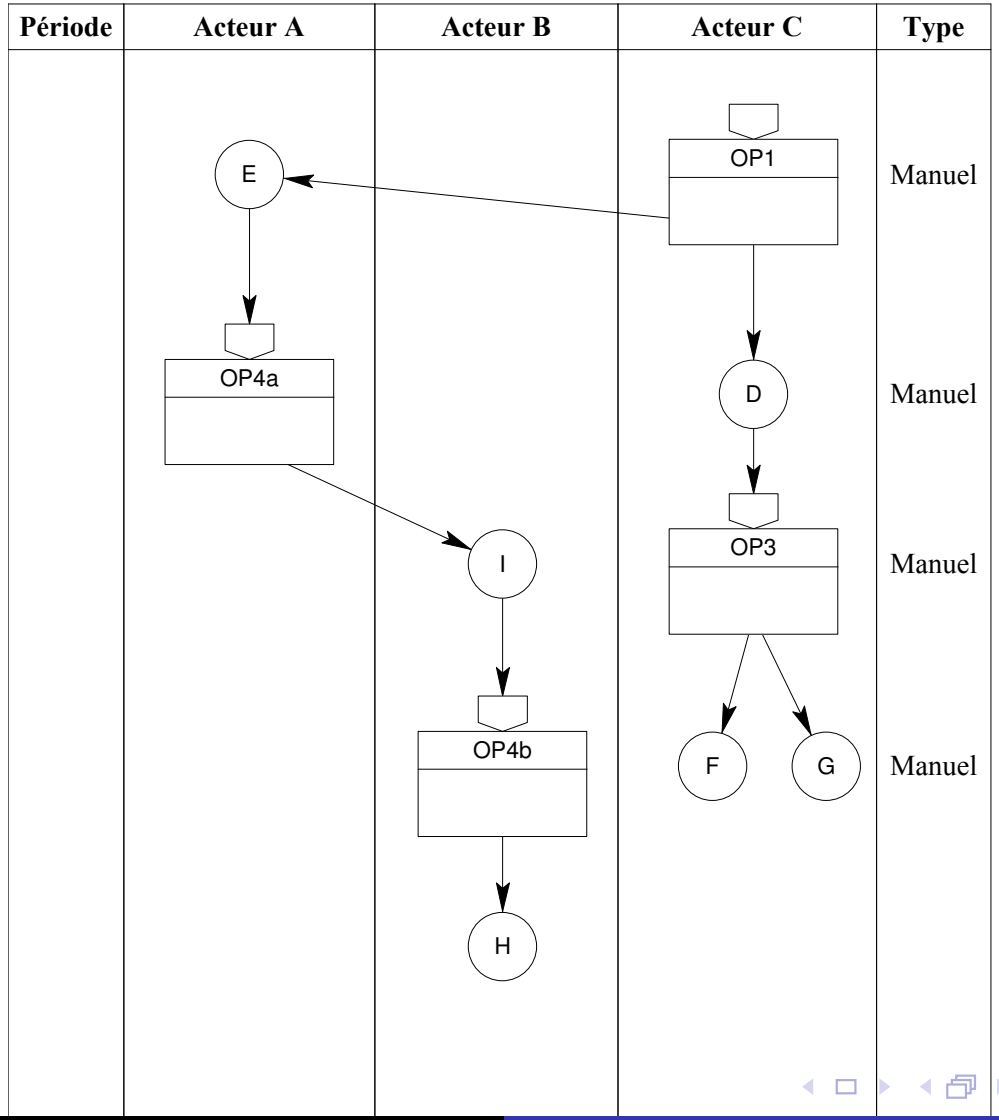
Répartition des opérations en les acteurs



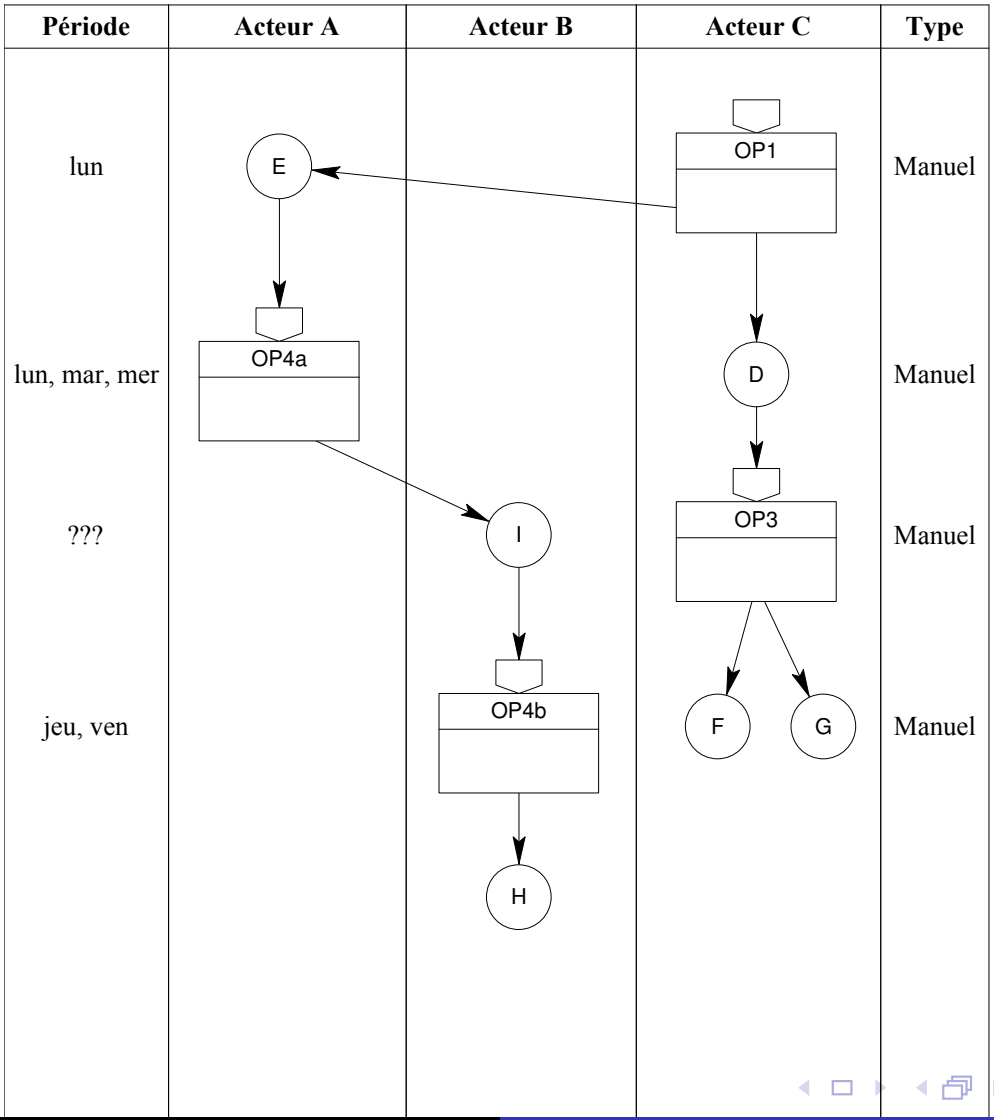
Répartition des opérations en les acteurs

Période	Acteur A	Acteur B	Acteur C	Type
				Manuel Manuel Manuel

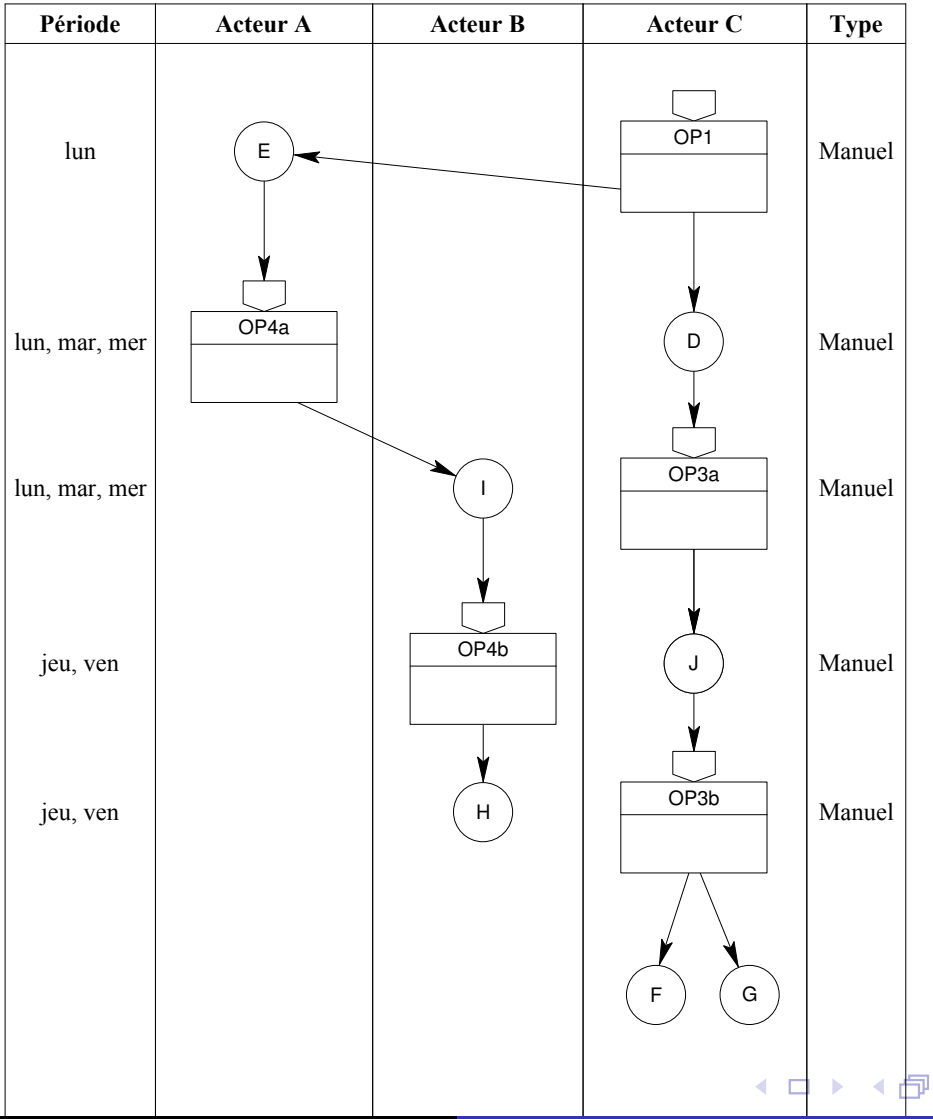
Division des traitements répartis entre plusieurs acteurs



Définition des périodes de traitement



Division des traitements répartis sur plusieurs périodes



Division des traitements répartis sur plusieurs périodes

Période	Acteur B	Acteur C	Type
lun, mar, mer	I	OP3a	Manuel
jeu, ven	OP4b	J	Manuel
jeu, ven	H	OP3b	Manuel
		F, G	

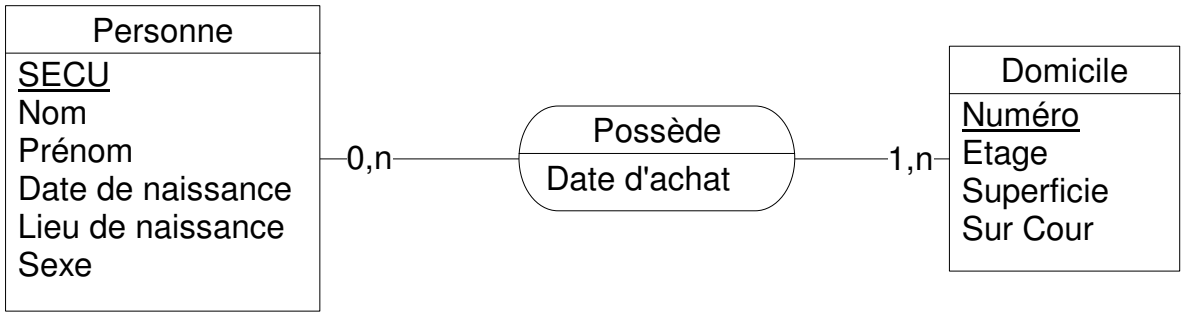
Définition des types de traitements

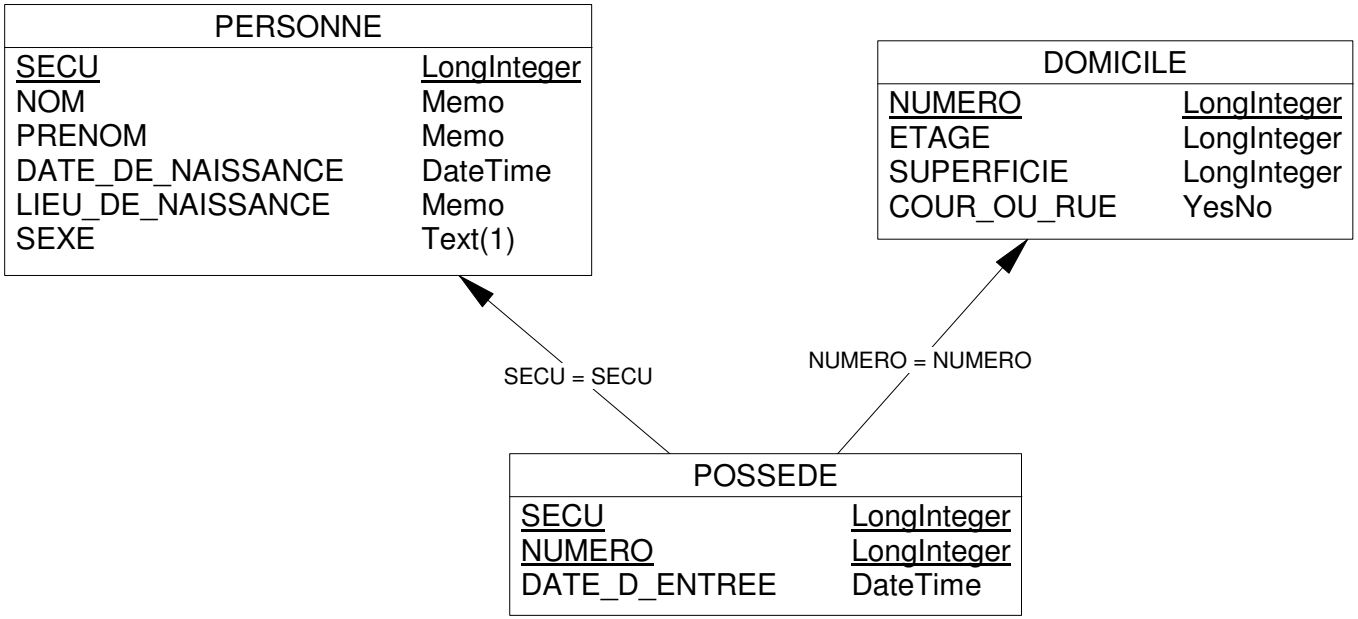
Période	Acteur B	Acteur C	Type
lun, mar, mer	I	OP3a	Manuel
jeu, ven	OP4b	J	Manuel
jeu, ven	H	OP3b1	Manuel
		K	
jeu, ven		OP3b2	Automatique
		F G	

Modèle Physique des Données (MPD)

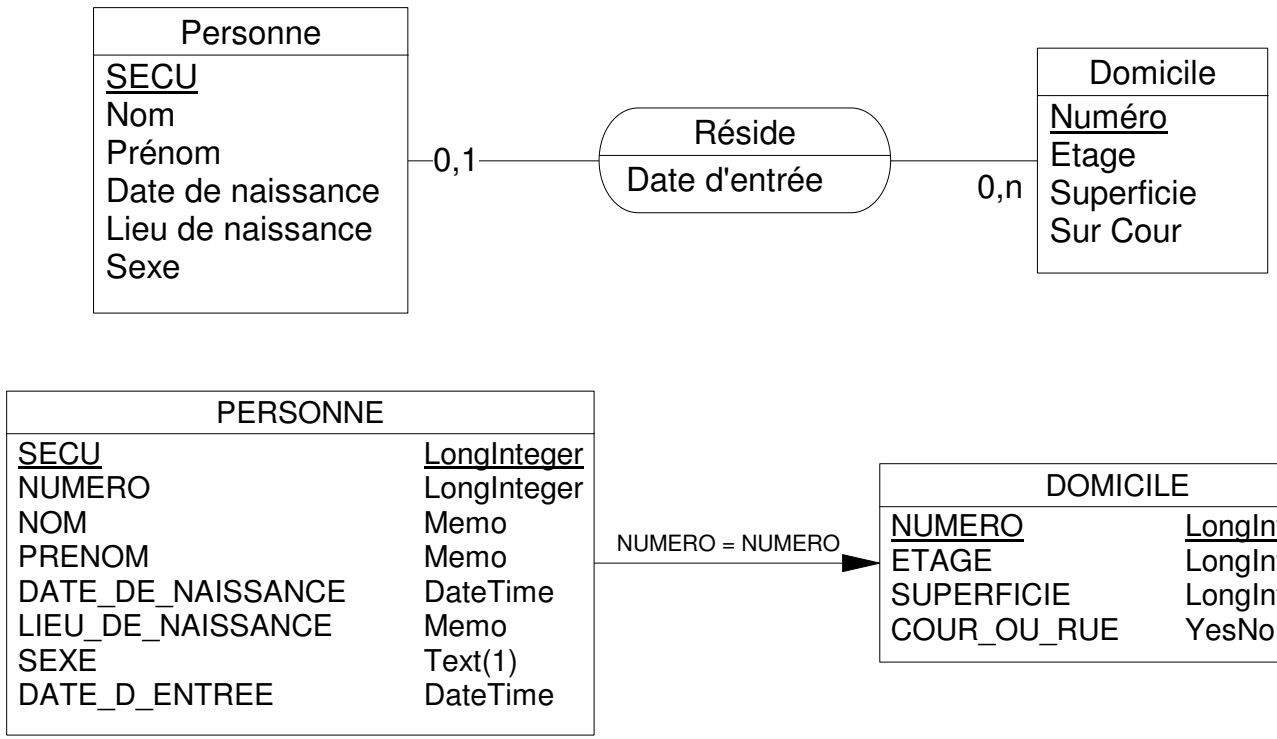
- Transformation en tables des
  - Entités et associations si on produit le MPD directement à partir du MCD
  - Schémas relationnels si on produit un MLD
- Dépend de la base de données cible
  - Types de données
  - Domaines des propriétés
- Les attributs qui permettent d'indexer les tables sont des clés primaires
- Les attributs (non clés primaires) qui font référence aux clés primaires d'autres tables sont des attributs secondaires

Associations n:m





Associations 1:n



Associations 1:1