

Modélisation de systèmes par une approche Objet avec UML et l'outil Objecteering de Softeam



Franck DARRAS
Hervé PINGAUD
Centre DR/GI

Objectifs

- Apprendre à utiliser *Objecteering*
- Posséder une formation initiale à UML

Achat d 'Objecteering en Mai 2001

Phase d 'apprentissage à l 'EMAC

Utilisation DE - DR - SG ?

Quelques références

- **Modélisation Objet avec UML, Pierre Alain MULLER, Eds EYROLLES**
- **UML pour l 'analyse d 'un système d 'information, le cahier des charges du maître d 'ouvrage, C. MORLEY, J.HUGUES, B.LEBLANC, DUNOD**
- **UML par la pratique, Etudes de cas et exercices corrigés, P. ROQUES, Eds EYROLLES**

<http://www.softteam.fr/>, <http://www.rational.com/>

<http://www.omg.org/>, <http://www.uml.org/>

Contenu de la formation

- **9h -Introduction à l 'approche objet et premiers éléments de notation UML**
- **9h 45 Premiers pas avec Objecteering (TP)**
- **11h 15 Etude de cas n°1 - Point de vue fonctionnel (TP)**
- **14h 30 Etude de cas n°2 - Point de vue statique (TP)**
- **15h 45 Etude de cas n°3 - Point de vue dynamique (TD)**
- **17h Debriefing**
- **12h 30 – 14h Déjeuner**

UML ?

- UML est un **langage standard** pour visualiser, spécifier, construire et documenter les systèmes d 'information
- UML signifie « Unified Modeling Language »
- UML essaie de réunir plusieurs « écoles »
 - Concepts de modélisation de données (entité-relation)
 - Concepts de modélisation métier (workflow)
 - Concepts de modélisation objet
- Il permet de transcender la notion de contraintes d 'implantation liées aux langages et aux systèmes

L 'approche objet

- Approches fonctionnelle et objet
- Les objets
 - définition
 - caractéristiques fondamentales
 - communication entre objets
- Les classes
 - définition
 - description des classes
- Les relations entre les classes
 - association, multiplicité, agrégation
 - correspondance entre classes et objets
- Les hiérarchies de classes
 - généralisation
 - héritage (principe, délégation)
 - polymorphisme

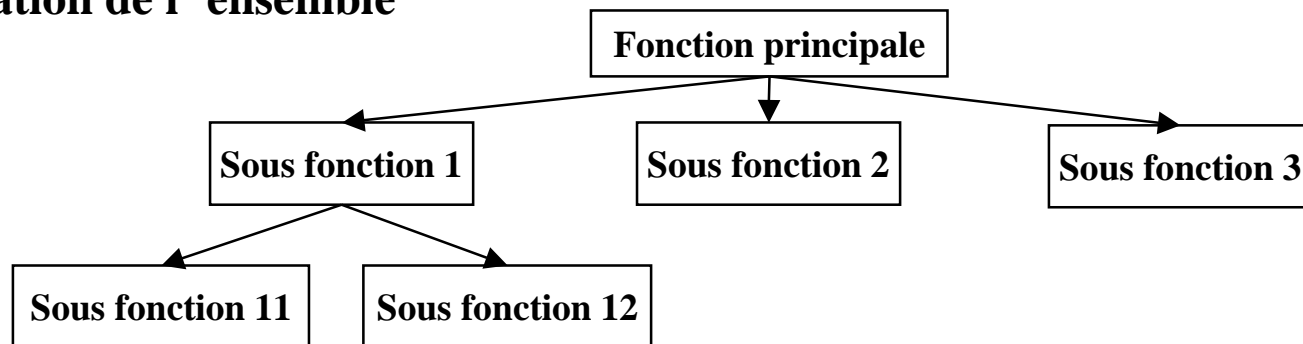
Approches fonctionnelle et objet

Analyse et conception de *systemes* complexes :

- diviser, décomposer pour comprendre,
- composer, réunir pour construire

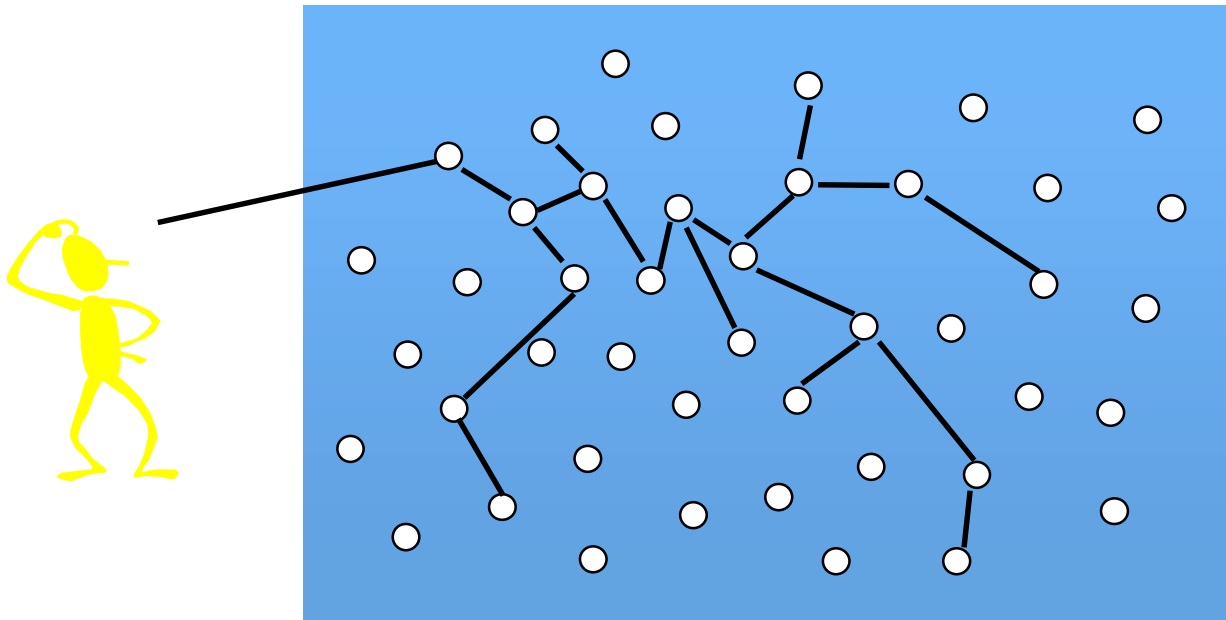
La représentation du système utilise *un modèle* qui permet de formaliser la démarche ou *méthode* d'analyse et de conception

Dans l'approche fonctionnelle, on décrit le système par décomposition en sous système correspondant à des fonctions plus ou moins élémentaires qui participent à la représentation de l'ensemble



La hiérarchie doit être stable au sens où une évolution fonctionnelle ne doit pas provoquer des modifications structurelles lourdes. Souvent, le caractère distribué des données au sein des fonctions appelle de telles modifications.

Un système est une société d'objets qui coopèrent



Les objets : définition

L 'objet est une unité atomique formée de l'union d 'un état et d 'un comportement, il est désigné par son identifiant. C 'est une notion qui permet d 'appréhender des objets matériels très simplement. Il est parfois plus difficile de capter les entités abstraites en termes d 'objet.

- **Construction itérative facilitée par un couplage faible entre composants du modèle**
- **Possibilité de réutiliser des éléments d'un développement à un autre**

L 'identité permet de distinguer tout objet de façon non ambiguë, et cela, indépendamment de son état. Elle est souvent construite avec un identifiant naturel du domaine du problème.

Ex: ma voiture

Un attribut est un trait caractérisant l 'objet qui lui est propre et qui prend des valeurs dans un domaine de définition donné. L 'état regroupe les valeurs instantanées de tous les attributs d 'un objet

Ex : la couleur est bleue, le poids à vide est de 979 kg, la puissance fiscale est de 12 CV, sa localisation courante est le garage

Les objets : caractéristiques et communication

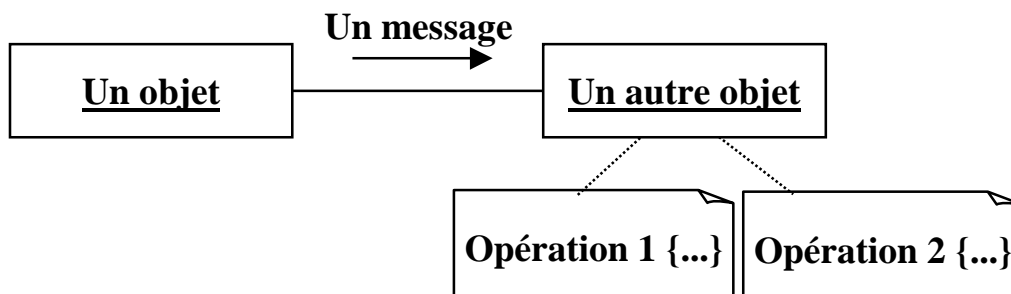
En règle général, l'état d'un objet est variable car certains attributs changent de valeur en fonction du comportement du système

Ex : Sa localisation courante est la route de Teillet

Le comportement regroupe toutes les compétences d'un objet et décrit les actions et les réactions de cet objet. Chaque atome de comportement est appelé opération (ou méthode). Les opérations sont déclenchées par un stimulation interne ou externe sous forme d'un message envoyé par l'objet lui même ou par un autre objet. C'est un stimuli.

Ex: la localisation courante est modifiée par une opération « déplace »

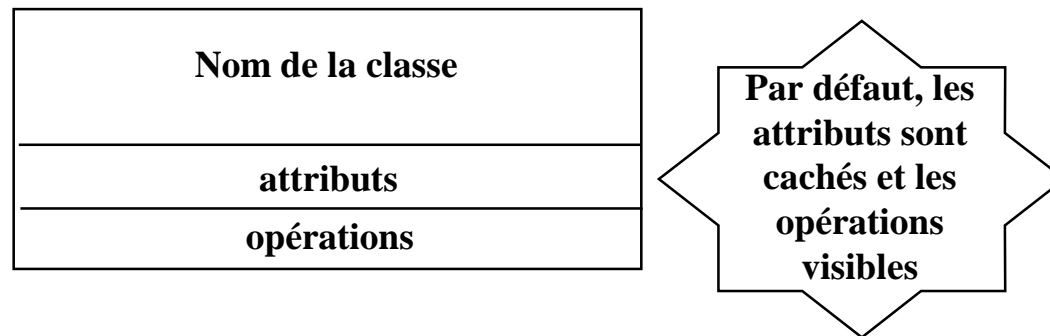
Il existe donc des liens entre les objets qui stipulent qu'ils peuvent être en interaction grâce à ce moyen qu'est le message.



Les classes

1- Définition

- La classe est le domaine de définition d'un ensemble d'objets, c'est à dire qu'on peut leur reconnaître des similitudes sur la façon des les identifier, sur les types d'état accessibles et sur le rôle qu'ils jouent
- Chaque objet appartient à une classe et il est généré par un processus d'instanciation de la classe



Les classes : 2- La description des classes

- La spécification décrit le domaine de définition et les propriétés des instances (notion de type dans les langages classiques)
- La réalisation décrit comment la spécification est réalisée, contient le corps des opérations et les données nécessaires à leur fonctionnement
- Une classe passe un contrat avec les autres classes :
 - elle s'engage à fournir les services publiés dans sa spécification
 - les autres classes s'engagent à ne pas faire usage des connaissances autres que celles décrites dans la spécification

mécanisme d'encapsulation

- Les règles de visibilité viennent compléter ou préciser l'encapsulation

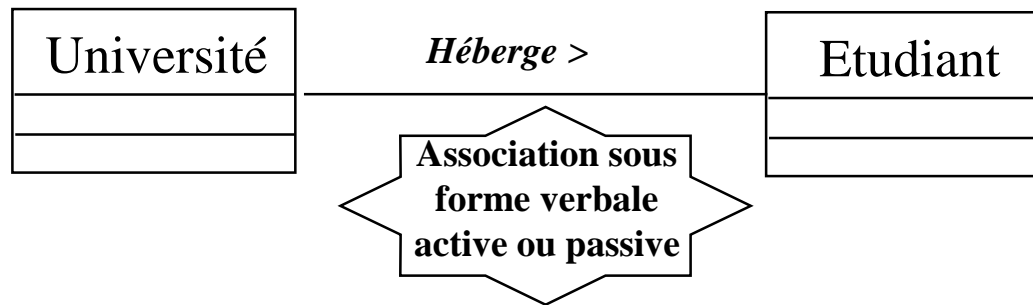
- niveau privé (-)
- niveau protégé (#)
- niveau public (+)

X
(-,#,+) Attribut
(-,#,+) Opération

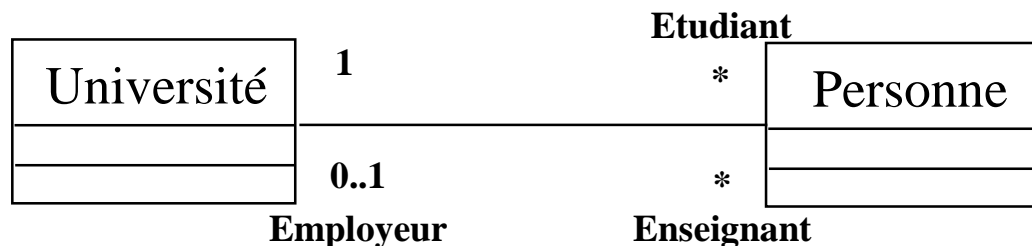
Les relations entre les classes

1- Association

- l'association exprime une connexion sémantique bidirectionnelle entre les classes. Elle décrit la structure, l'organisation du système



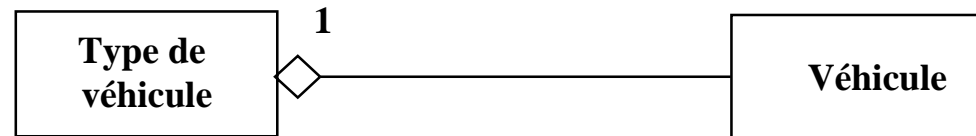
- Les noms de rôles prennent tout leur intérêt lorsque plusieurs associations relient deux mêmes classes
- Une information de multiplicité précise le nombre d'instances qui participent à la relation



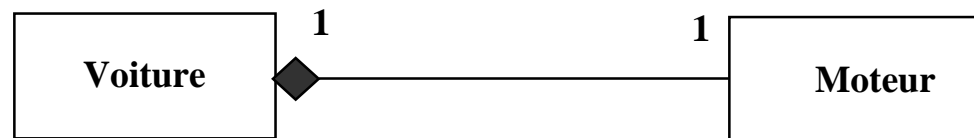
Les relations entre les classes

2- Agrégation et composition

- l'agrégation est une forme particulière d'association qui exprime un couplage plus fort entre classes
- elle favorise la propagation de valeurs d'attributs et d'opérations de l'agrégat vers les composants



- la composition est un cas particulier d'agrégation dans laquelle la vie des composants est liée à celle de l'agrégat (contenance physique)



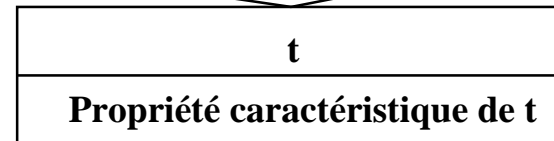
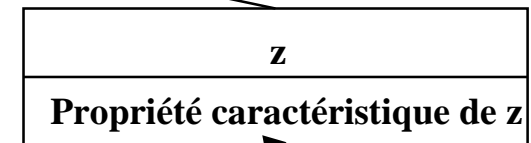
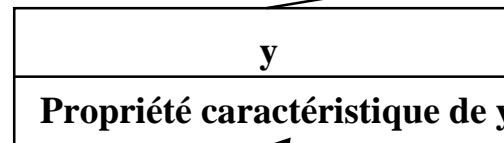
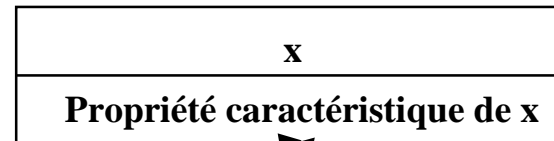
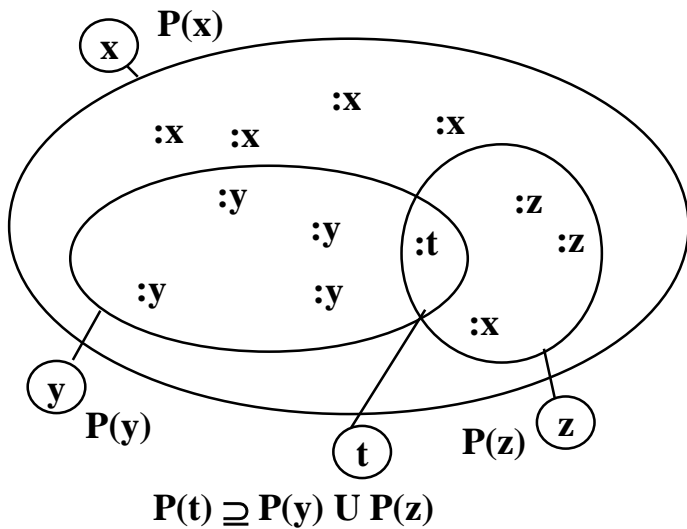
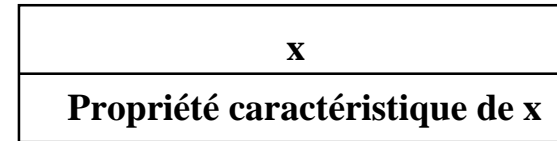
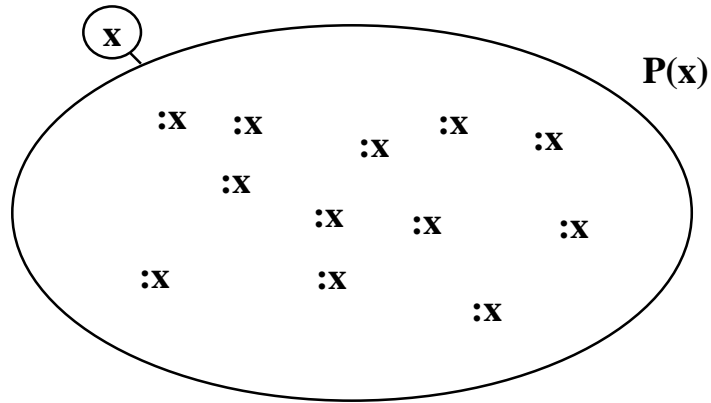
Les relations entre les classes

3-Correspondance entre classes et objets

- Chaque objet est une instance de classe et ne peut pas changer
- Certaines classes, abstraites, ne peuvent pas être instanciées
- Chaque lien est instance d'une relation
- Les liens relient les objets, les relations relient les classes
- Un lien indique que deux objets se connaissent et peuvent échanger des messages
- Un lien entre deux objets implique une relation entre les classes des deux objets

Les hiérarchies de classes

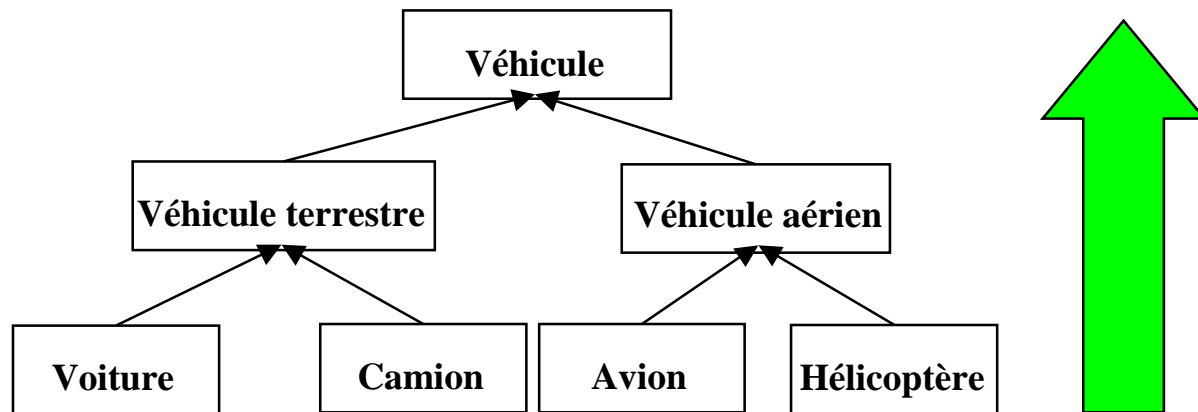
1- Des ensembles aux classes



Les hiérarchies de classes

2- généralisation/spécialisation

- La généralisation consiste à factoriser les éléments communs d'un ensemble de classes dans une classe plus générale (super classe)



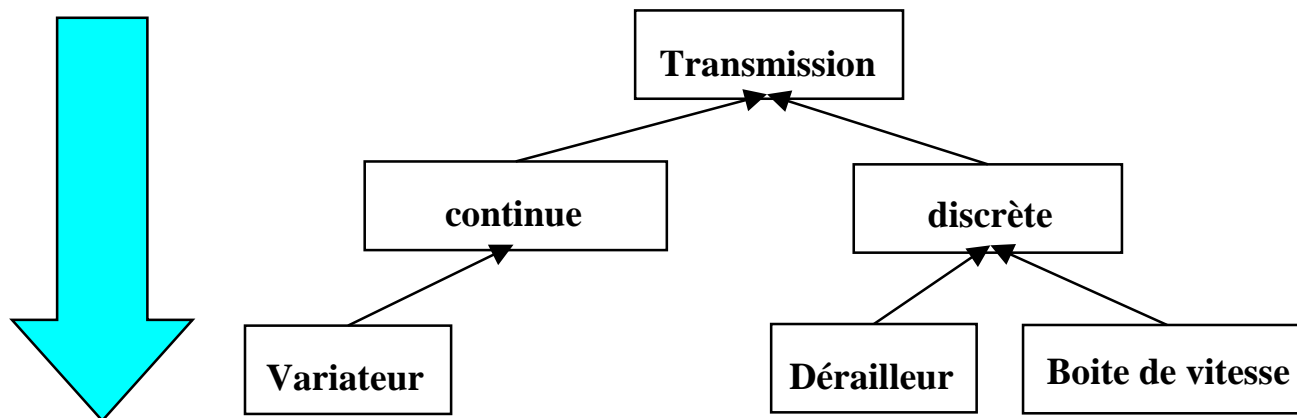
Exemple de hiérarchie de classes

La généralisation ne concerne que les classes, elle n'est pas instanciable en liens, ne porte aucune indication de multiplicité, est non réflexive, non symétrique, mais transitive

Les hiérarchies de classes

3- généralisation / spécialisation

- La spécialisation permet de capturer les particularités d'un ensemble d'objets non discriminés par les classes identifiées



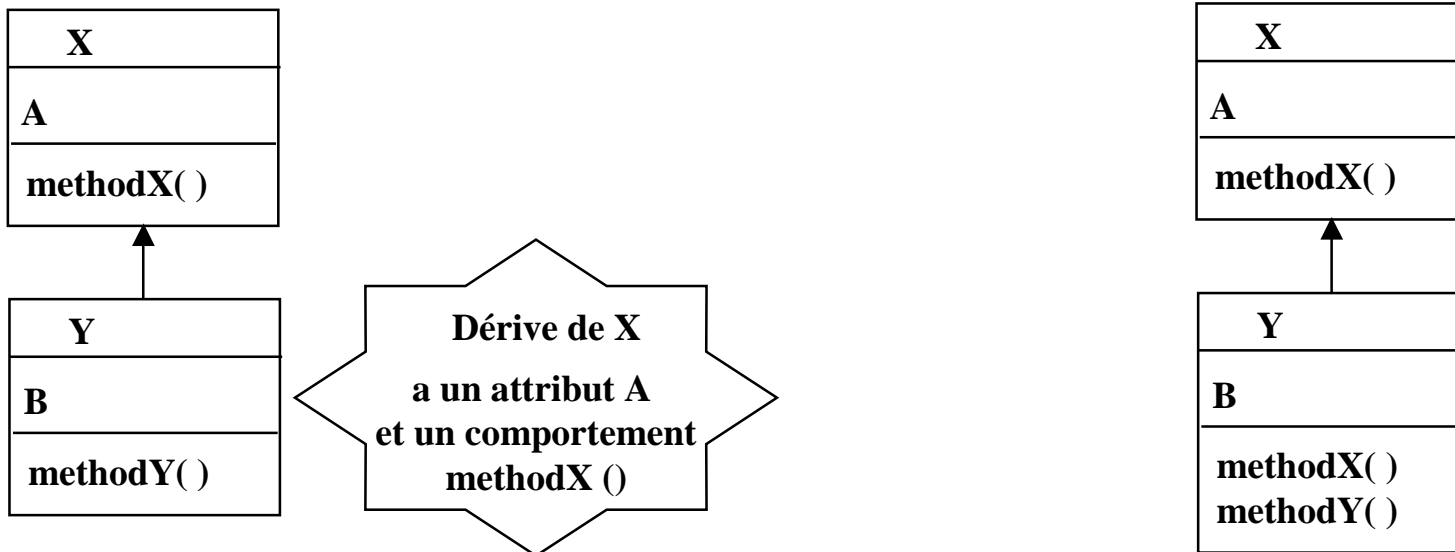
Exemple de hiérarchie de classes

C'est un atout pour faciliter la démarche d'extension et de réutilisation

Les hiérarchies de classes

4- héritage

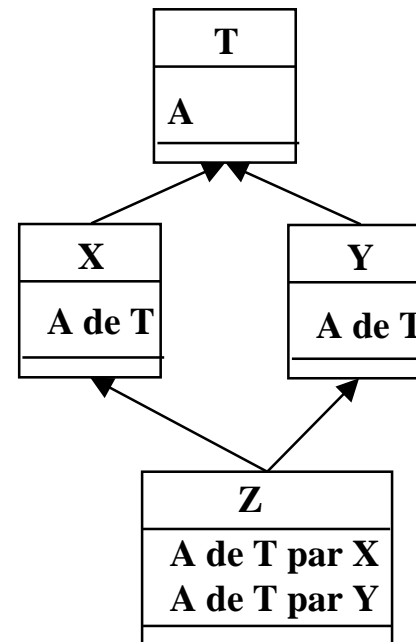
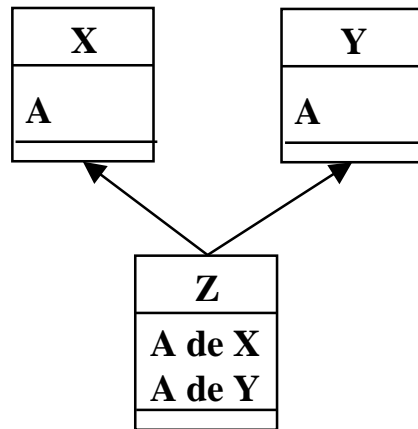
- La réalisation de la classification se fait par héritage
- C'est une technique de construction de classe à partir de super classe(s) en partageant des attributs, des opérations et parfois des contraintes, au sein d'une hiérarchie



Les hiérarchies de classes

5- héritage multiple

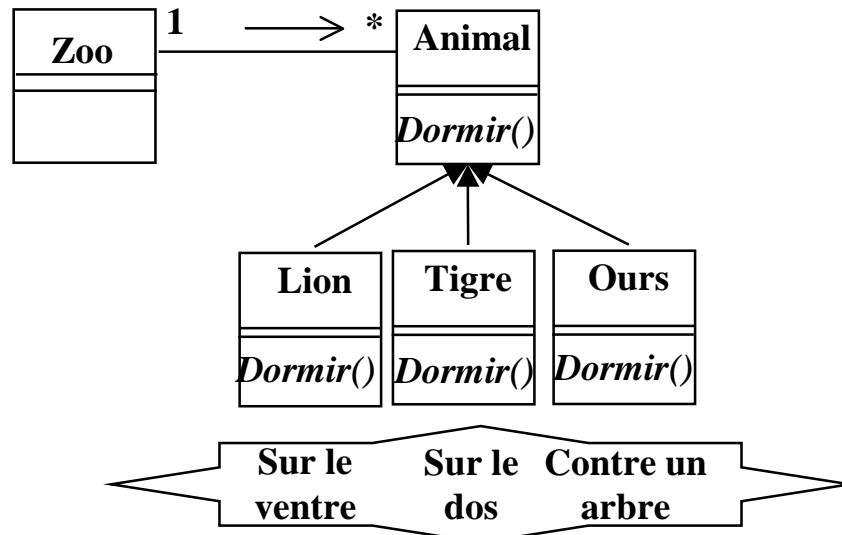
- L'héritage n'effectue pas une union des propriétés, mais une somme
- La réalisation peut induire des conflits, des problèmes de collision de noms lors de la propagation des attributs et des opérations des classes parents vers les sous classes



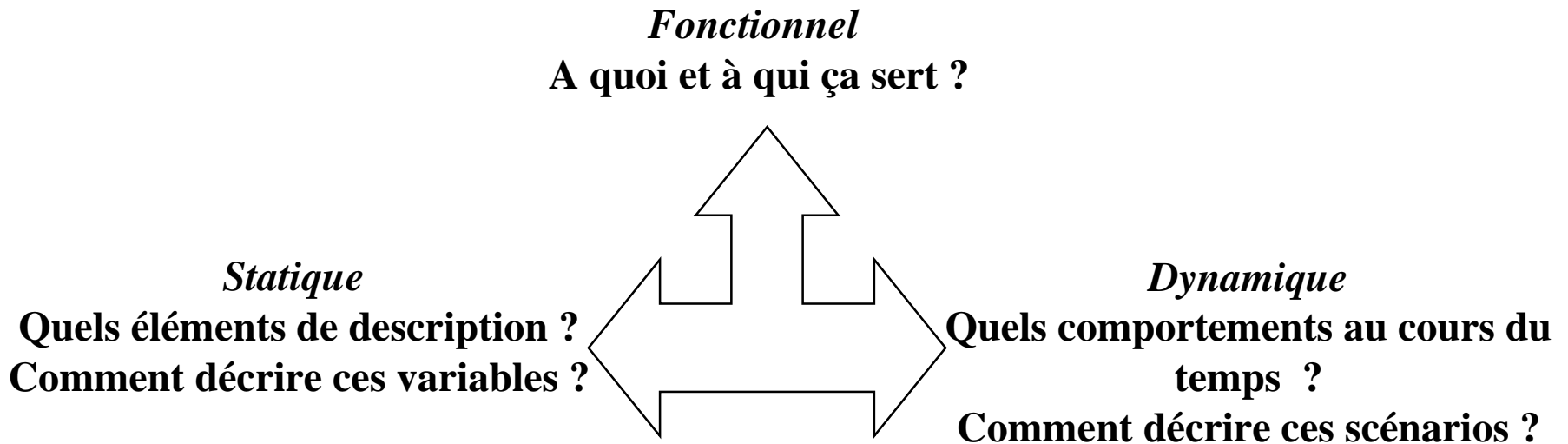
Les hiérarchies de classes

6- polymorphisme

- Le polymorphisme permet de déclencher des opérations différentes en réponse à un même message venant d'un parent
- Chaque sous classe hérite de la spécification des opérations de la super classe, mais a la possibilité de modifier localement le comportement



3 points de vue dans l'analyse de systèmes



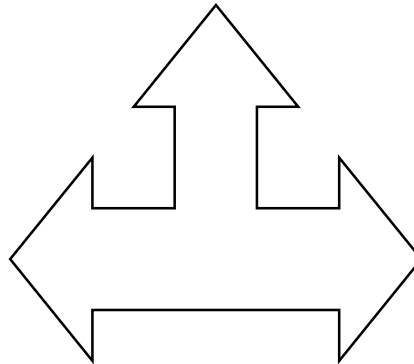
3 cas d'études

Fonctionnel

Cas n°1

Diagramme des cas d'utilisation

Diagramme de séquence



Statique

Cas n°2

Diagramme de classes

Dynamique

Cas n°3

Diagramme d'état transition

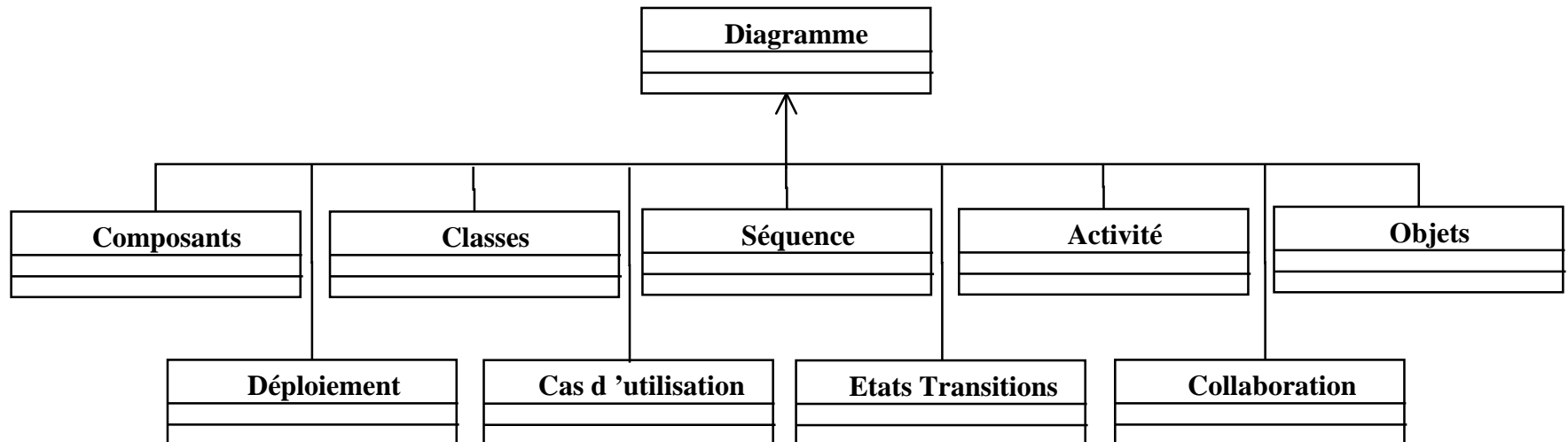
Fin de la partie 1

Notation UML

- Les concepts de base
- Les diagrammes de classes
- Les cas d'utilisation
- Les diagrammes d'objets
- Les diagrammes de collaboration
- Les diagrammes de séquence
- Les diagrammes d'états transitions
- Les diagrammes d'activités
- Les diagrammes de composants
- Les diagrammes de déploiement

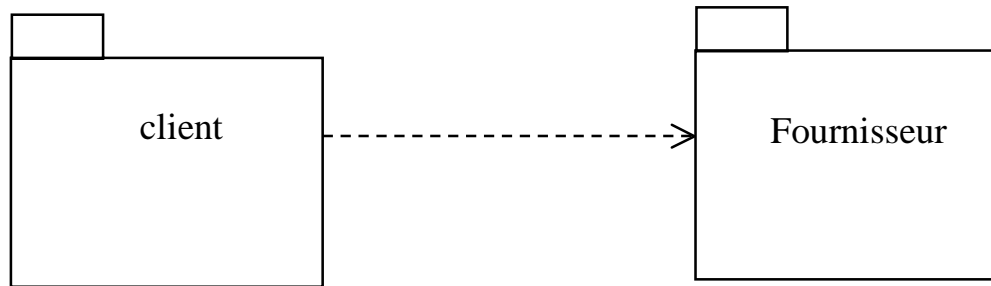
Concepts de base

- UML définit 9 sortes de diagrammes qui représentent les différents points de vue de la modélisation
- La plupart des diagrammes se présentent sous la forme de graphes composés de sommets et d'arcs



Concepts de base : le paquetage

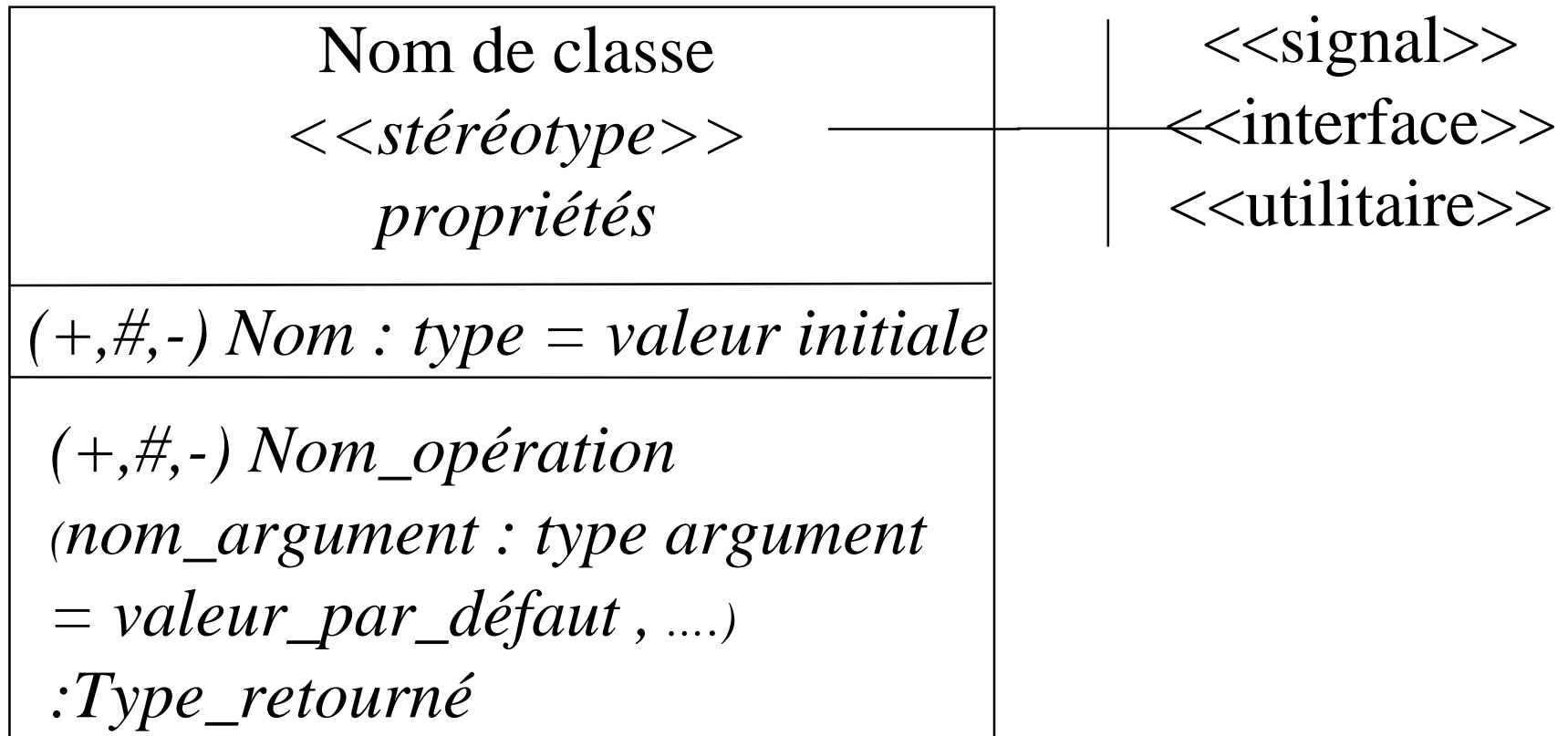
- Les paquetages offrent un mécanisme général pour la partition des modèles et le regroupement des éléments de modélisation
- Chaque paquetage correspond à un sous ensemble du modèle et contient, selon le modèle, des classes des objets, des relations, des composants, ainsi que les diagrammes associés



- Une relation de dépendance spécifie qu'au moins une classe du client utilise au moins une classe du fournisseur

Les diagrammes de classes

1-Notation de la classe

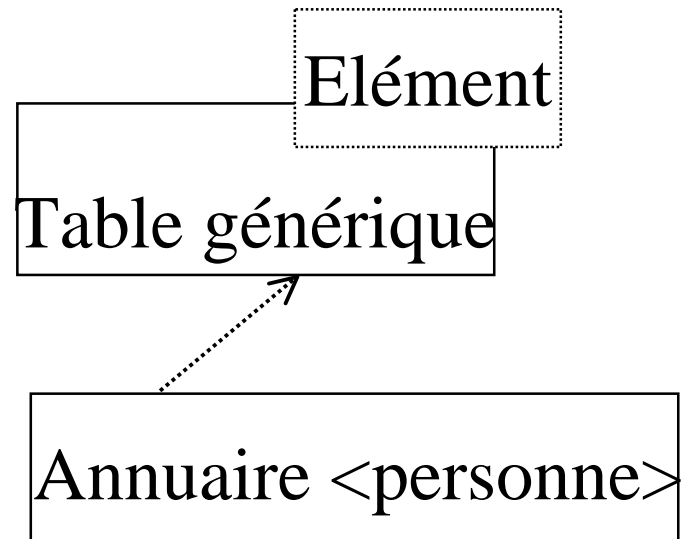


Les diagrammes de classes

1-Des classes spéciales



Représentation d'une interface par un petit cercle relié à la classe qui fournit les services

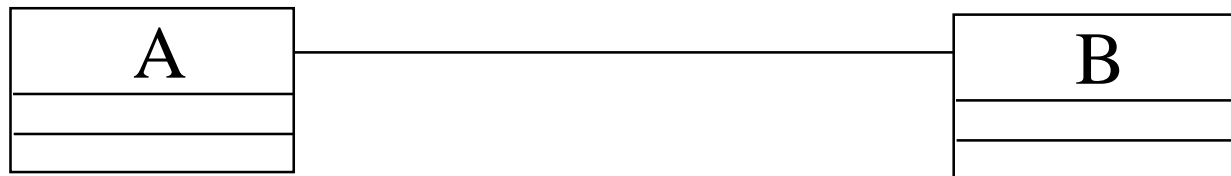


Instance d'une classe paramétrable

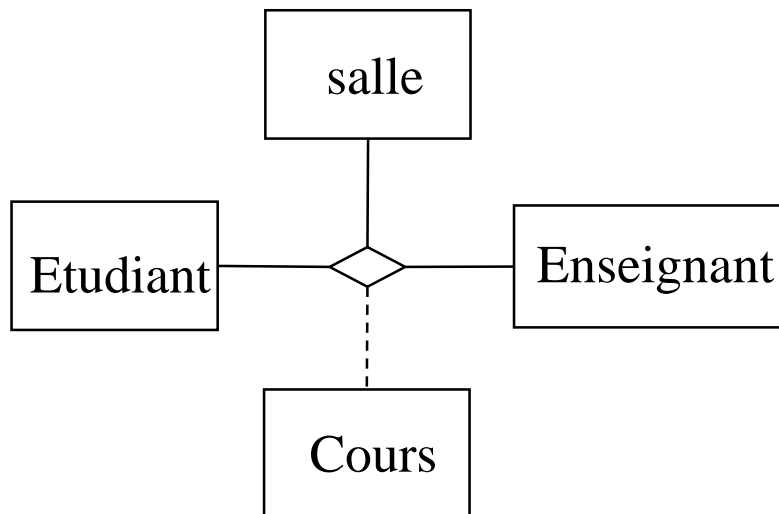
Le diagramme des classes

2-Les associations

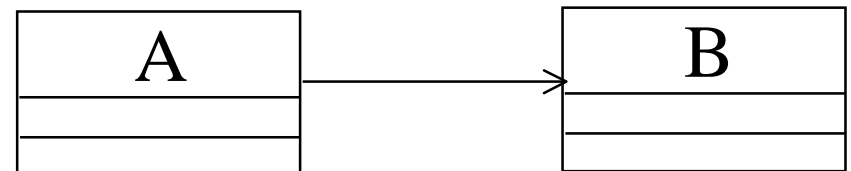
Généralement binaire et navigable dans les deux directions



Arité supérieure à 2

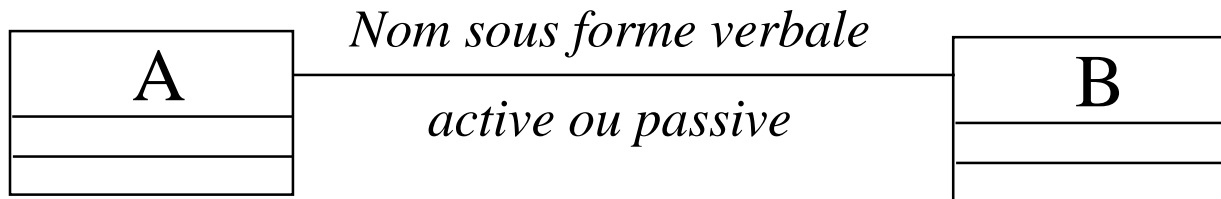


Chemin de navigation



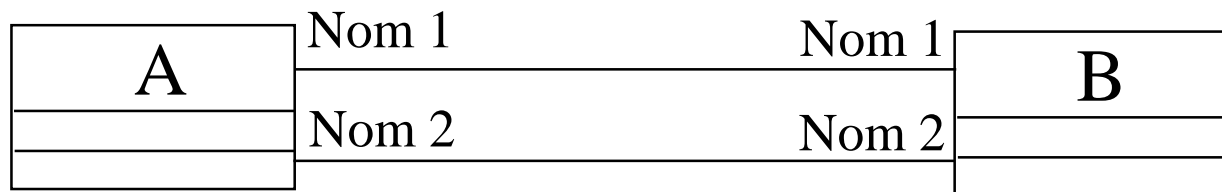
Le diagramme des classes

2-Nommage des associations



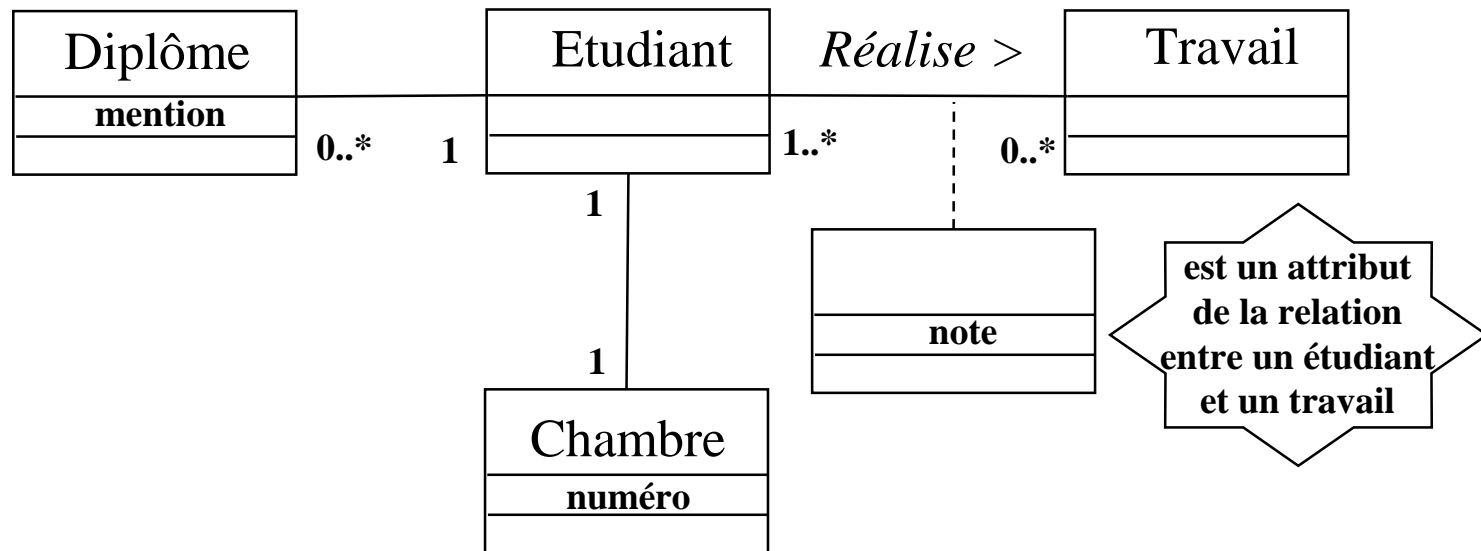
Ne pas confondre nom d'association et message

Nommage des rôles



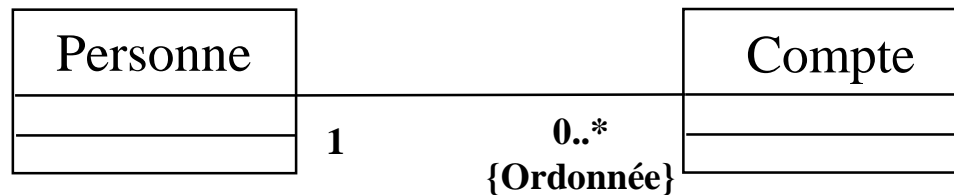
Le diagramme des classes

2-Multiplicité des relations (contraintes sur le nombre de liens)
et placement des attributs pour les types N vers N

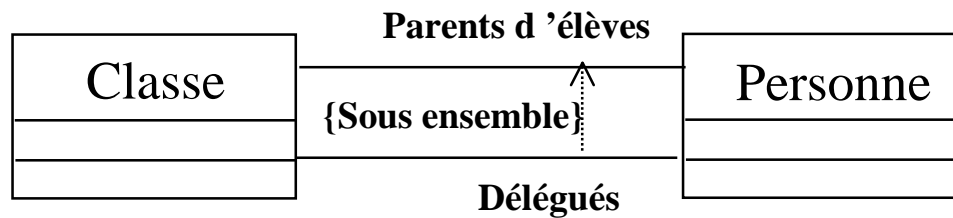


Le diagramme des classes

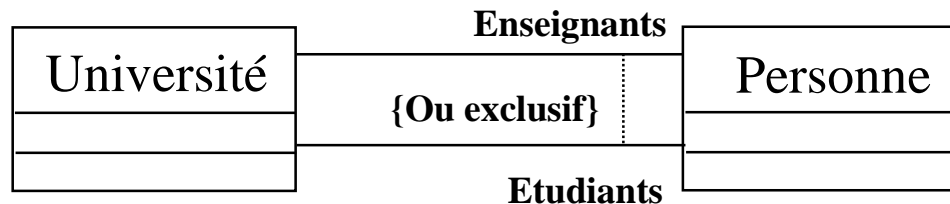
2-Contraintes sur les associations



La collection
des comptes est
ordonnée



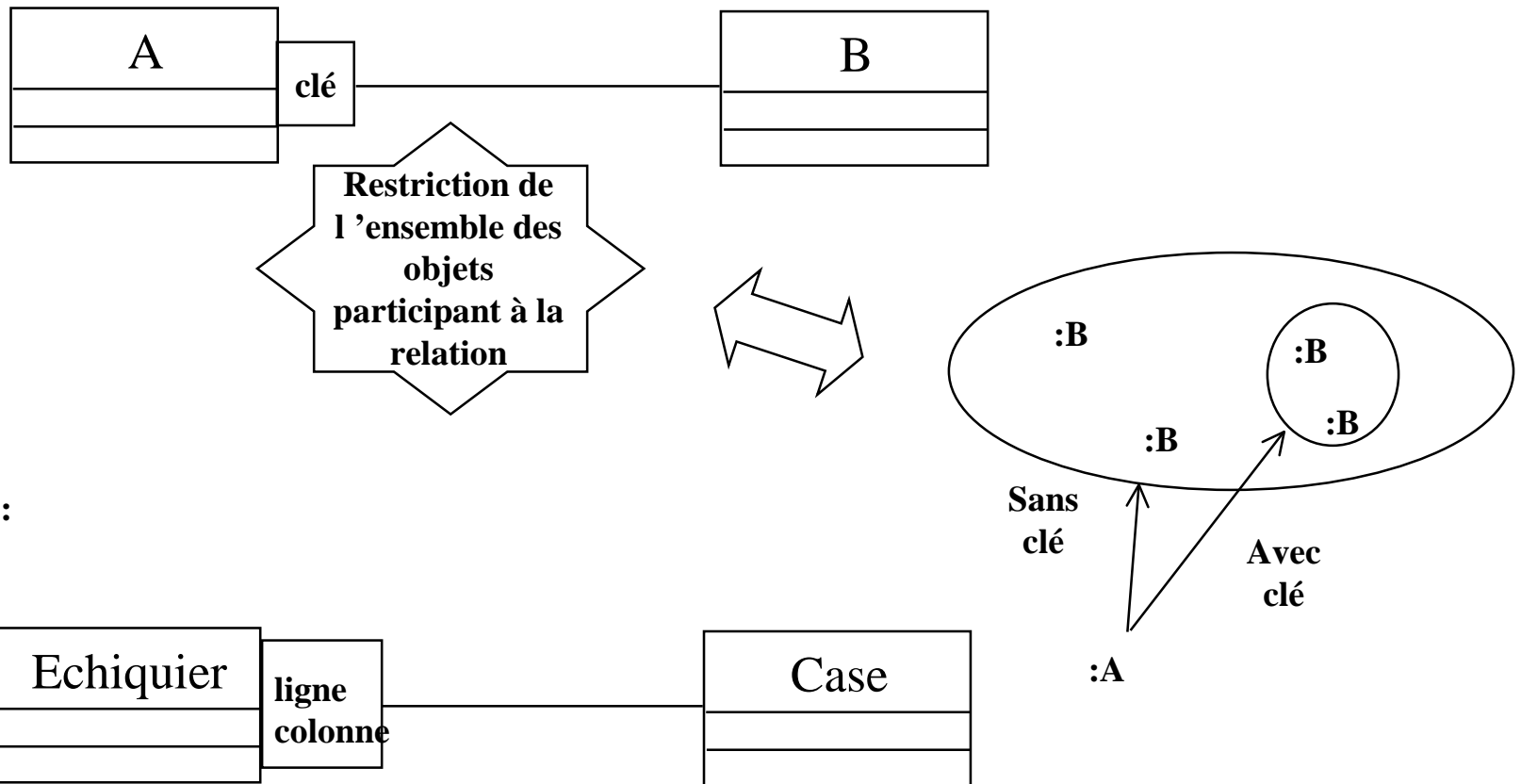
La Délégués
sont
aussi des
parents
d 'élèves



Les associations
sont
mutuellement
exclusives

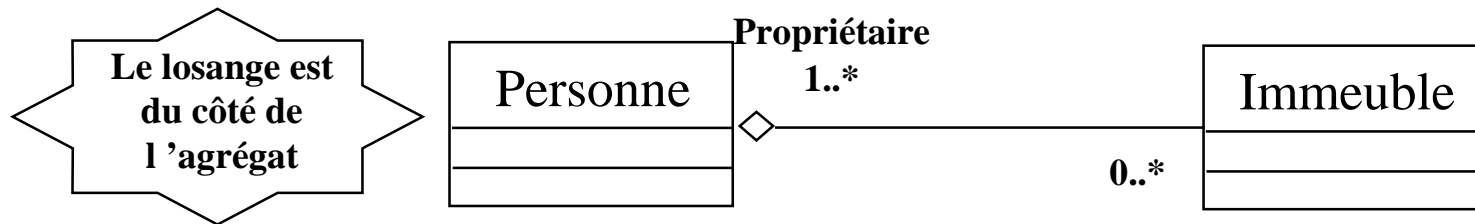
Le diagramme des classes

2-Restriktion des associations

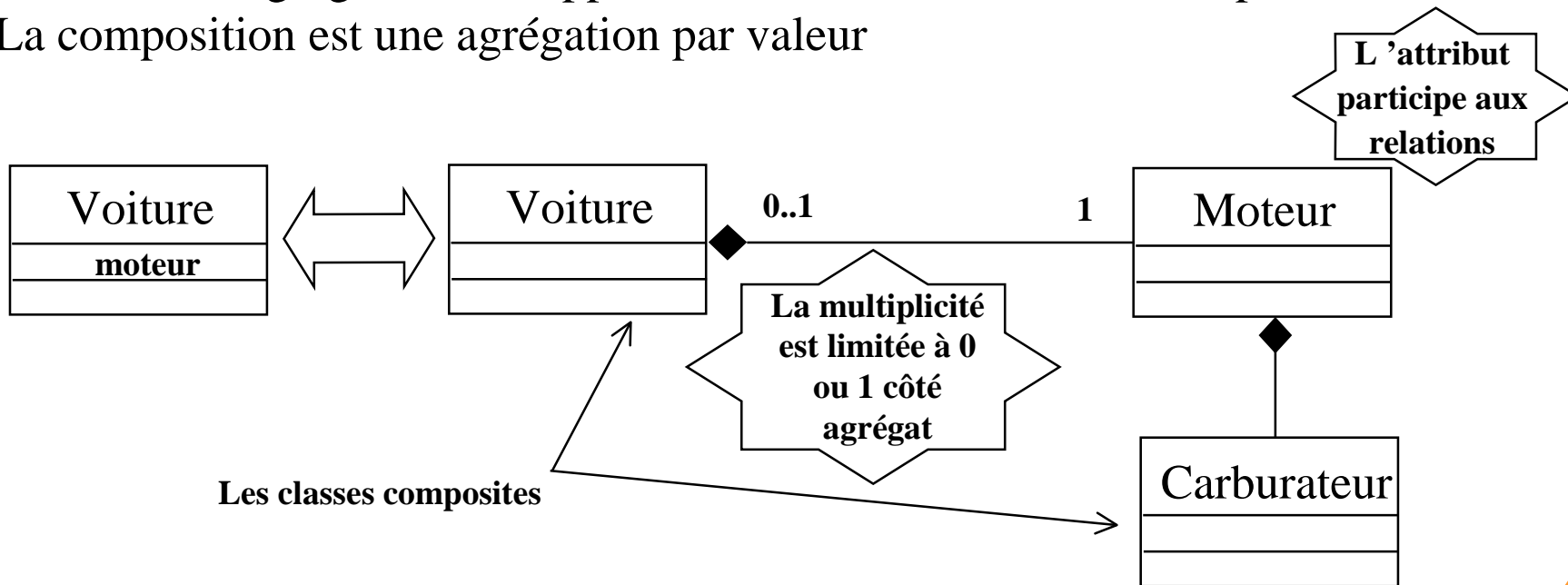


Le diagramme des classes

3- Les agrégations



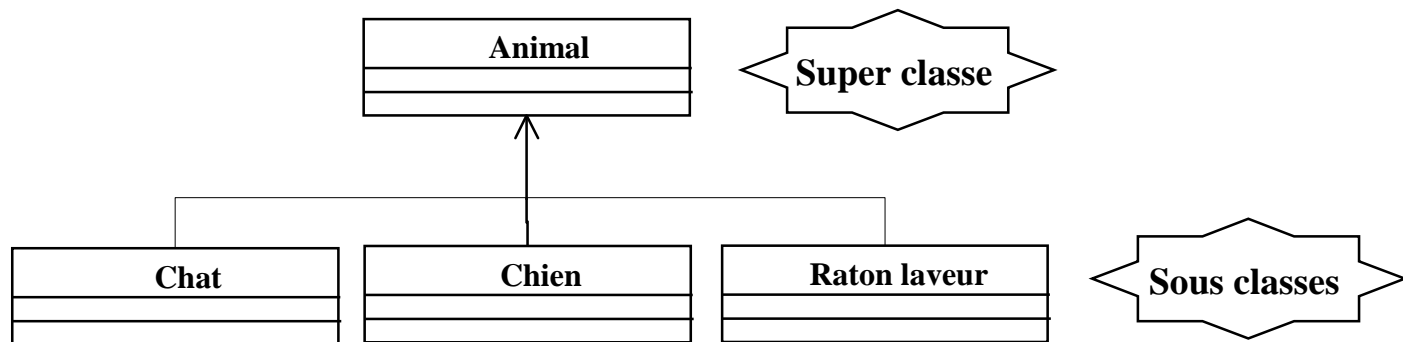
- La notion d'agrégation ne suppose aucune forme de réalisation particulière
- La composition est une agrégation par valeur



Le diagramme des classes

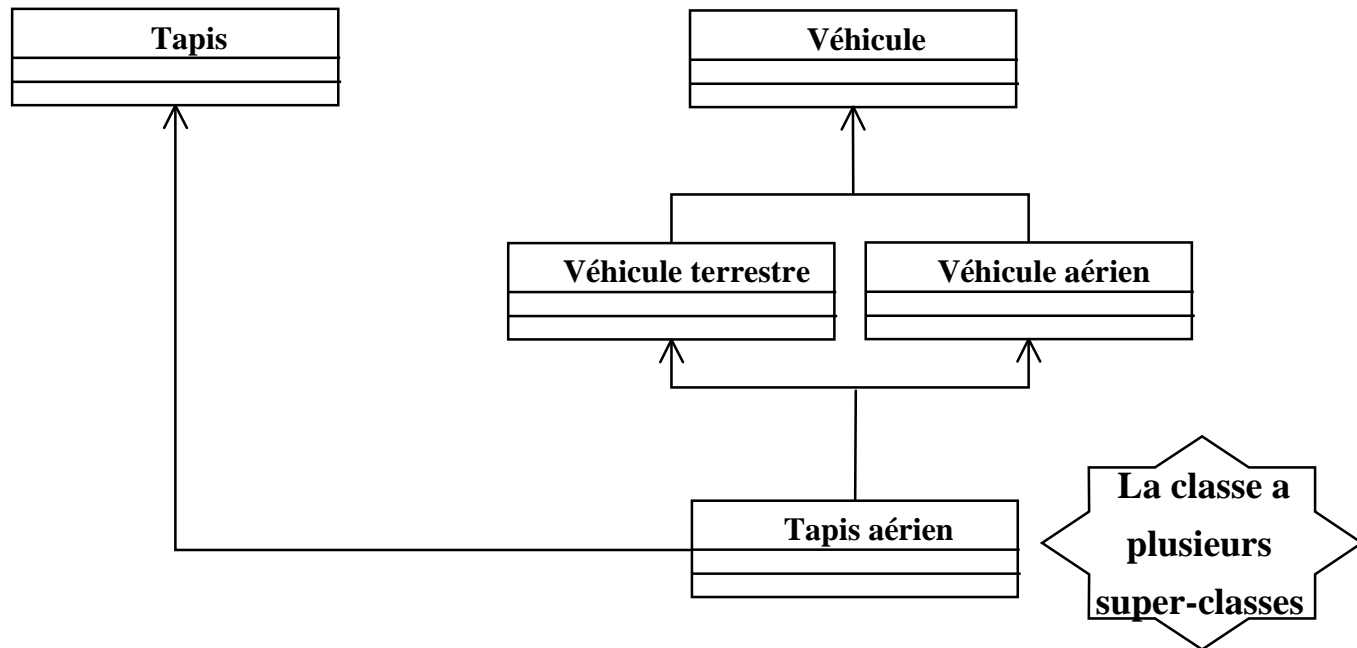
4- La généralisation

- Point de vue porté sur un arbre de classification
- Signifie *est un* ou *est une sorte de*
- Ex : Un chat *est un* animal est une généralisation
Un chat *a* deux oreilles est une composition
- L'élément plus spécifique peut être raffiné dans le respect de son ascendance



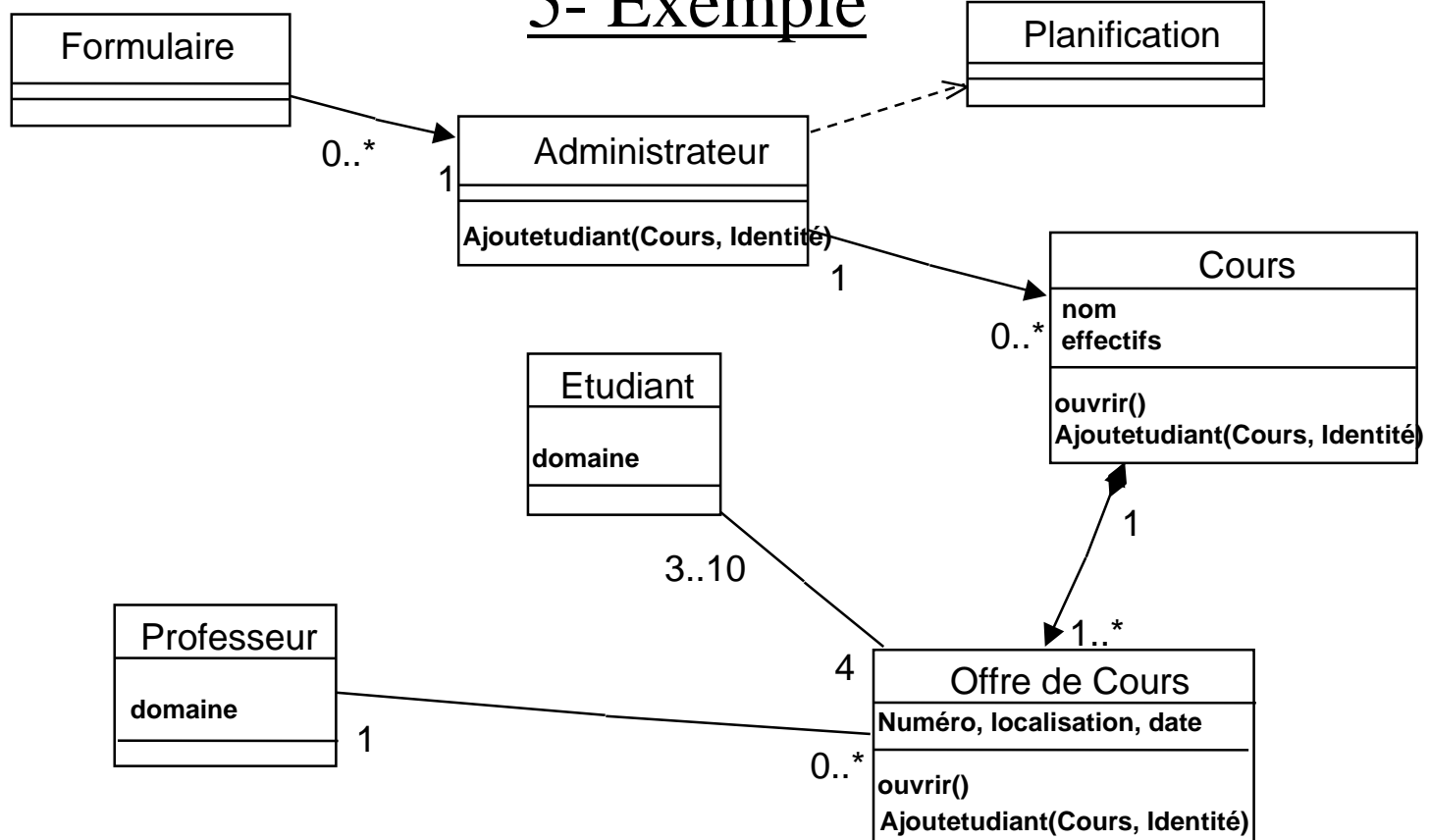
Le diagramme des classes

4- La généralisation multiple



Le diagramme des classes

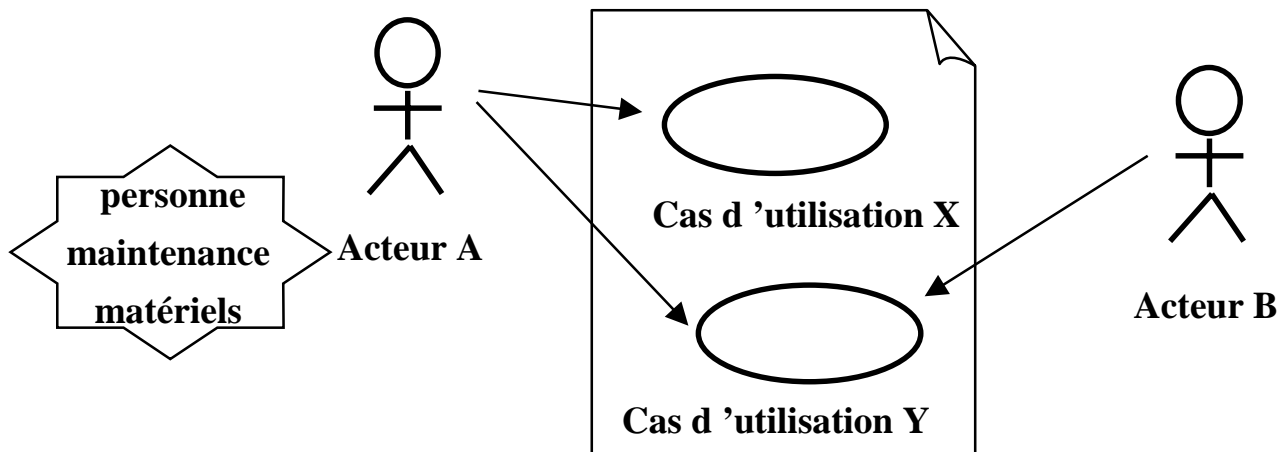
5- Exemple



Les cas d'utilisation

1- Définition

- Les « use cases » décrivent, sous la forme d'actions et de réactions, le comportement d'un système du point de vue utilisateur
- Etude des interactions pour une seule catégorie d'utilisateurs à la fois
- Formalisme simple pour faciliter l'expression du besoin



Les cas d'utilisation

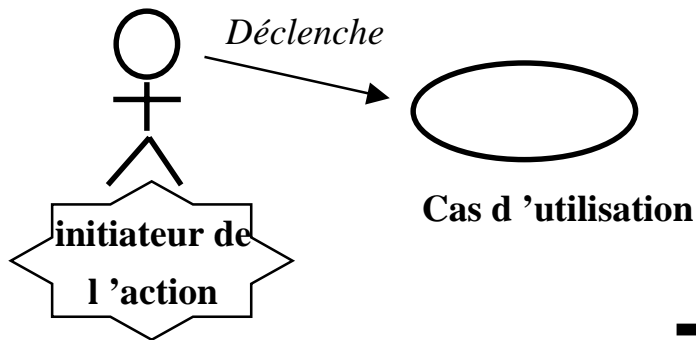
1- Définition

- La description d'un cas d'utilisation comprend :
 - le début du cas, l'événement initiateur
 - la fin du cas d'utilisation, l'événement terminal
 - l'interaction entre les cas d'utilisation et les acteurs,
 - les échanges d'information : paramètres des interactions système/acteurs
 - la chronologie et l'origine des informations (internes/externes)
 - les répétitions de comportement
 - pendant que
 - autre chose
 - fin pendant
 - les situations optionnelles
 - l'acteur choisit l'option
 - X
 - Y
 -
 - Puis continue en

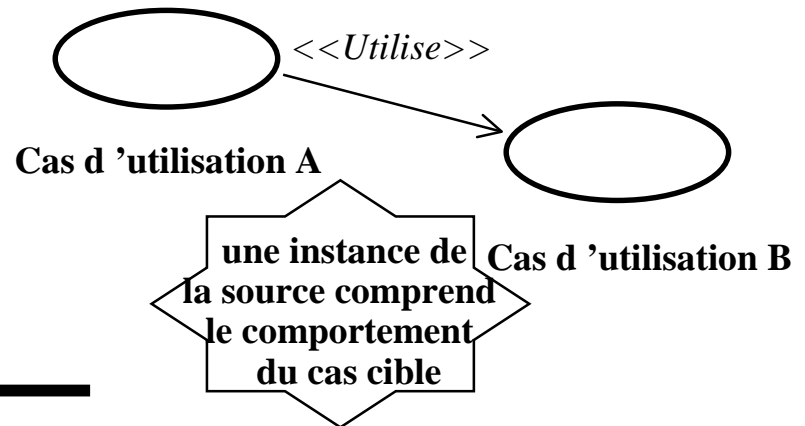
Les cas d'utilisation

2- Les relations entre cas d'utilisation

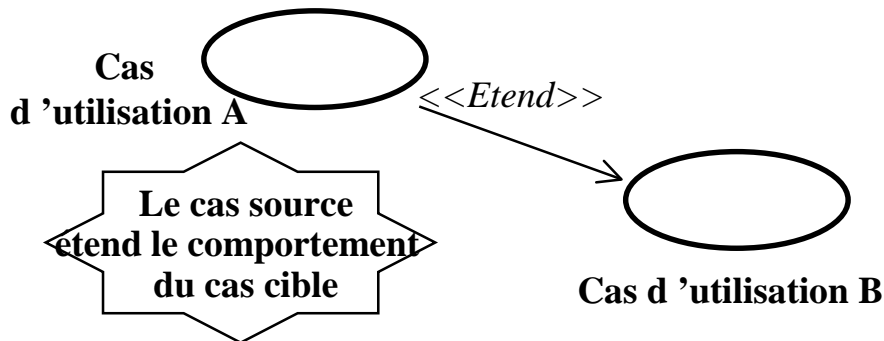
- La relation de communication



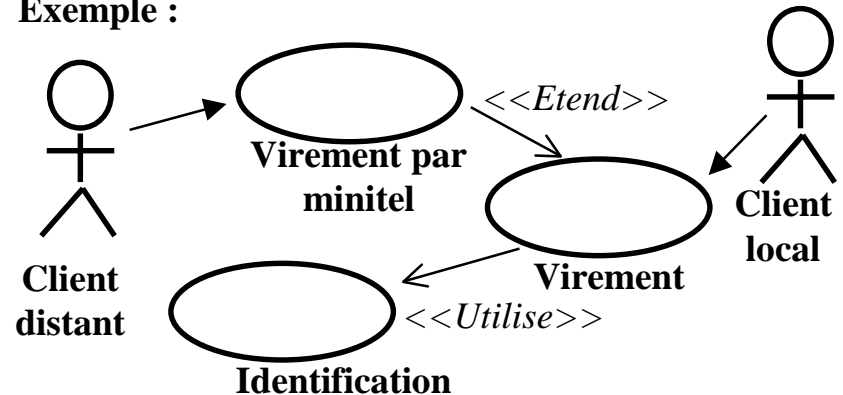
- La relation d'utilisation



- La relation d'extension

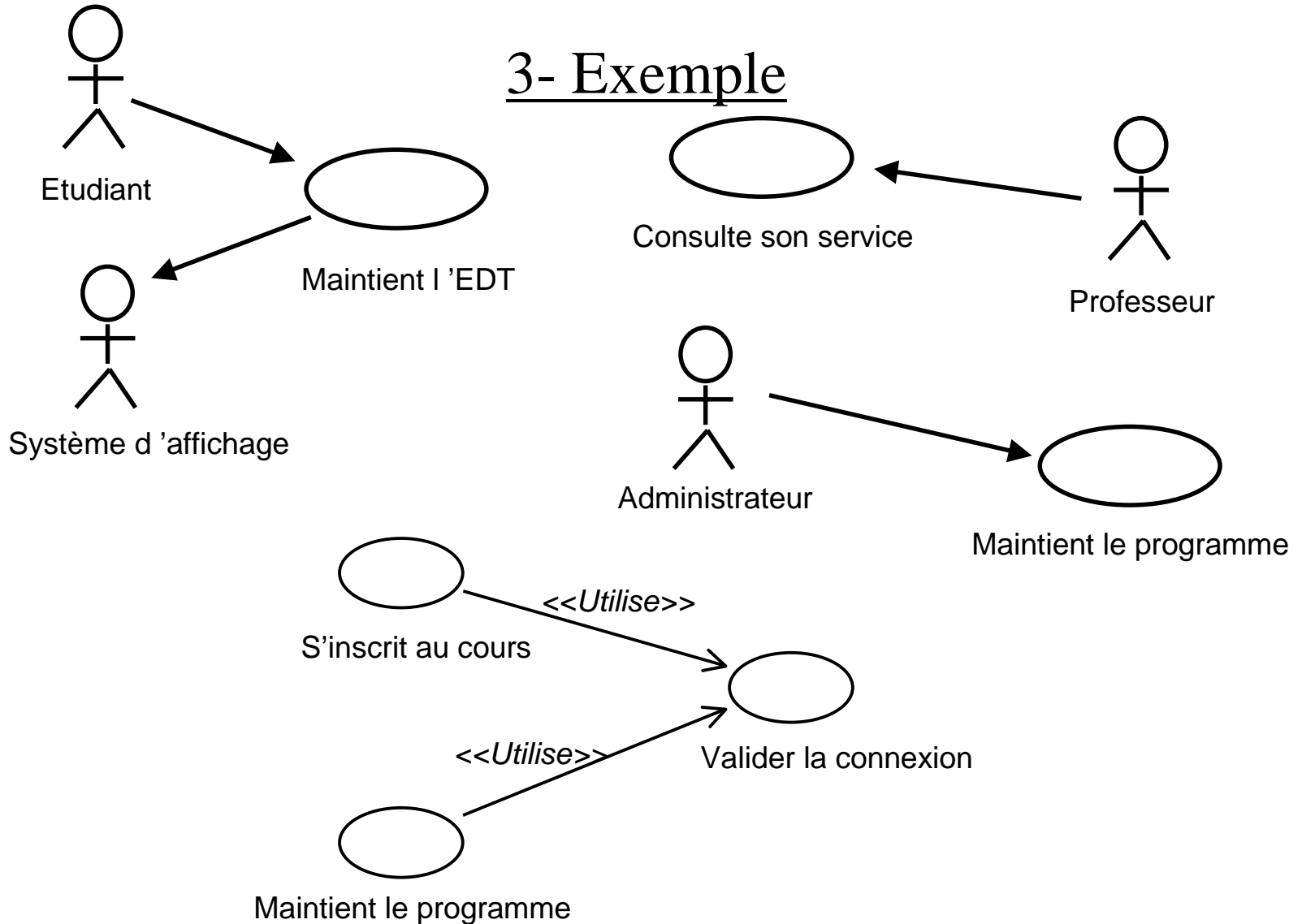


Exemple :



Les cas d'utilisation

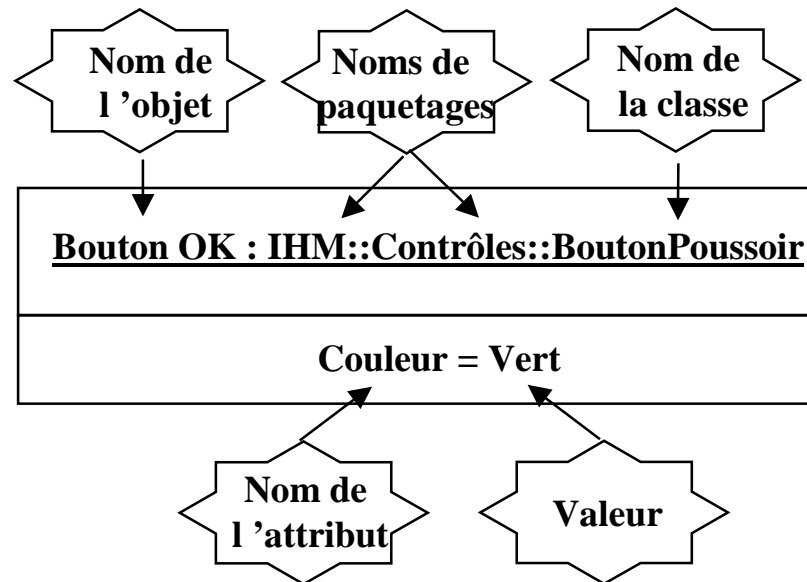
3- Exemple



Les diagrammes d'objets

- Ils montrent les liens entre objets, c'est la nature statique du système
- Ils s'utilisent pour définir un contexte (avant/après interaction, par exemple)

1-Notation d'objet



Les diagrammes d'objets

2- Représentation des liens

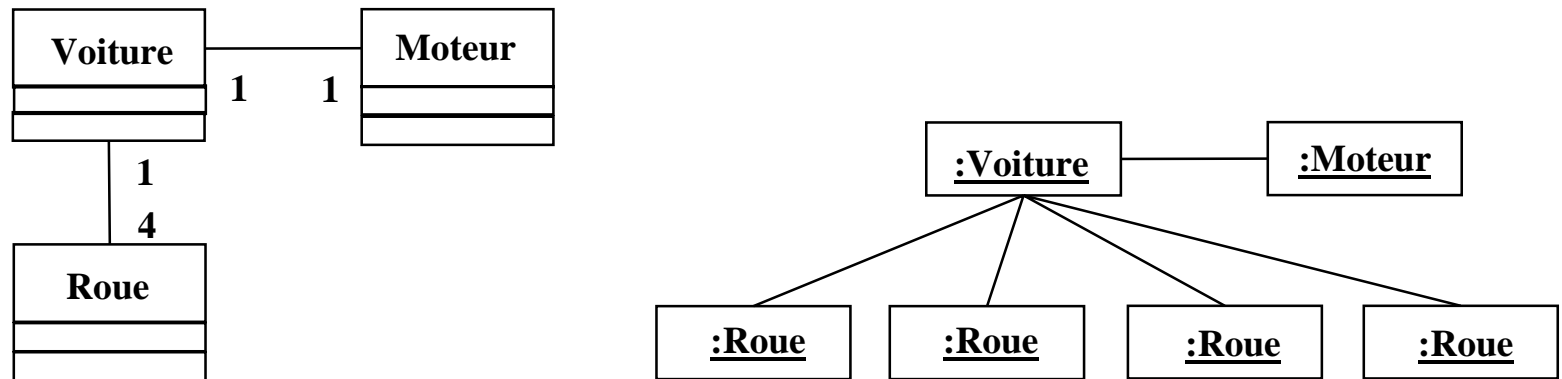


Diagramme de classes

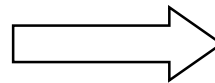
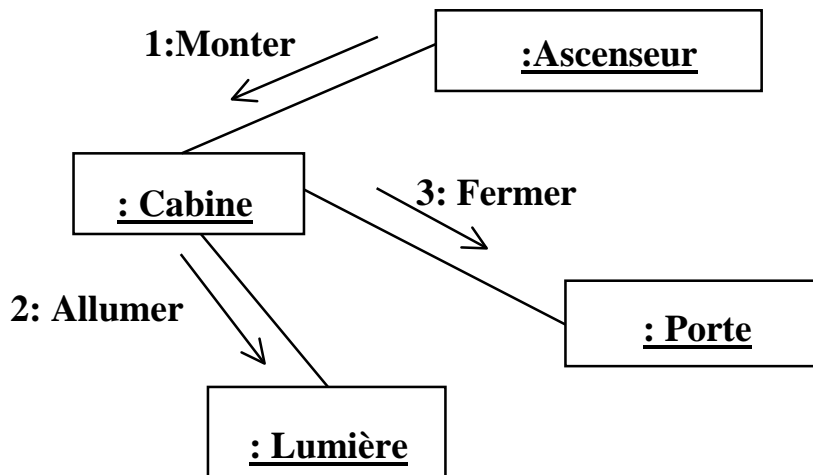


Diagramme d'objets

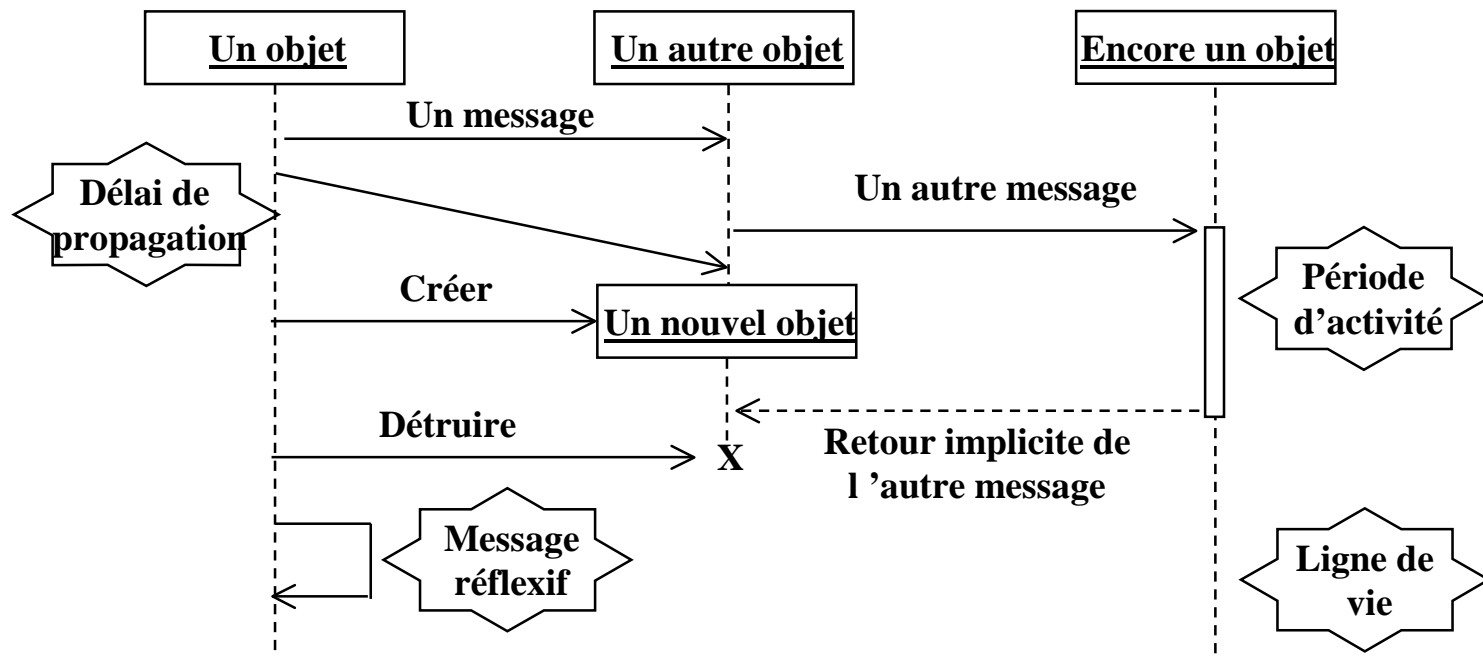
Les diagrammes de collaboration

- Ils montrent les interactions entre objets (par l'envoi de messages)
- Ils expriment le contexte d'un groupe d'objets (objets et liens)



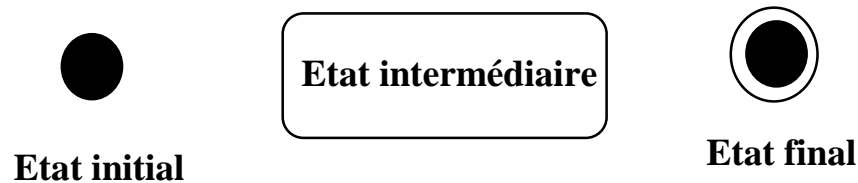
Les diagrammes de séquence

- Ils montrent les interactions entre objets selon un point de vue temporel
- Ils s'utilisent à la documentation des cas d'utilisation (# messages/événements)
- Puis, ils portent sur toutes les formes de messages entre objets : appel de procédure, événement discret, signal entre flots d'exécution, interruption matérielle

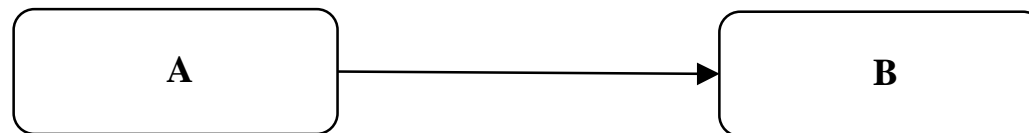


Les diagrammes d'états-transitions

- Ils décrivent le comportement dynamique de groupes d'objets
- Ils visualisent des automates d'états finis (états/transitions)
- Un état est caractérisé par un jeu de valeurs des attributs de l'objet
- Ils utilisent le formalisme *Statecharts* (D.Harel, Science of computer prog., Vol 8, 1987)



- Les états sont reliés par des connexions unidirectionnelles, appelées transitions
- La transition d'état survient lors d'événement dans le domaine du problème
- La transition d'état est instantanée car le système doit toujours être déterminé

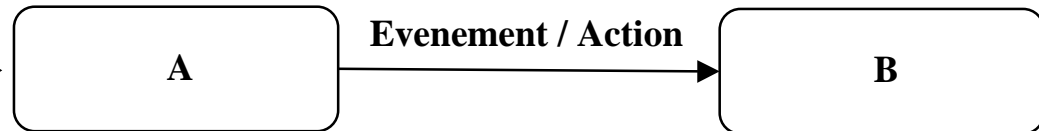


Les diagrammes d'états-transitions

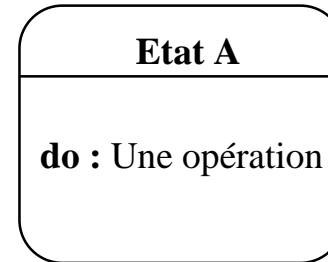
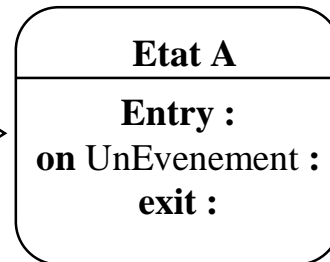
une garde est une condition booléenne qui valide ou non le déclenchement d'une transition



L'action pointe une opération déclarée dans la classe de l'objet destinataire

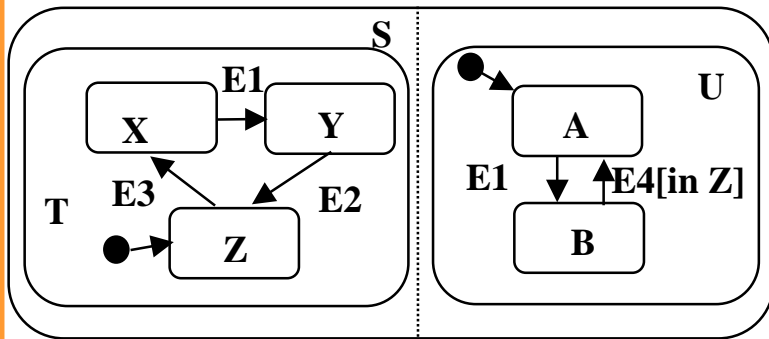
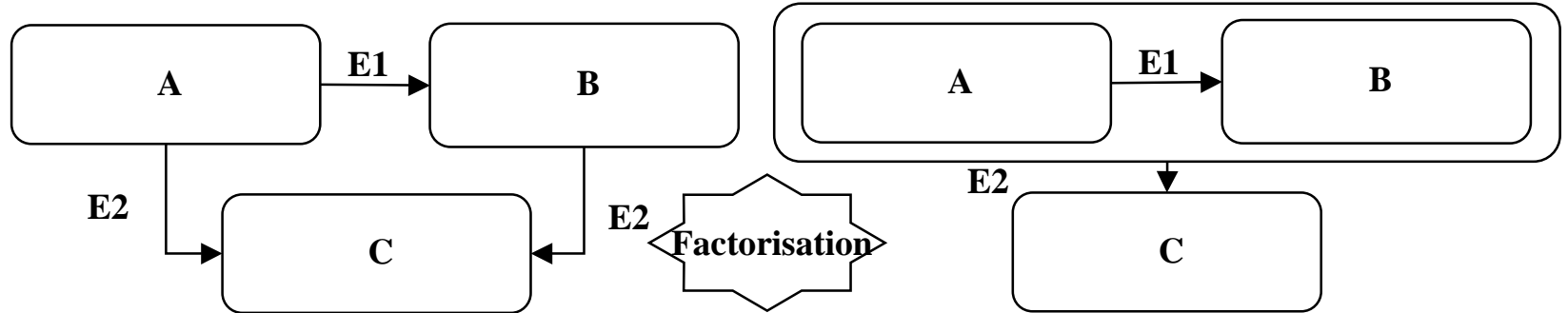


Les états peuvent aussi contenir des actions déclenchées au début, pendant ou à la fin de l'état

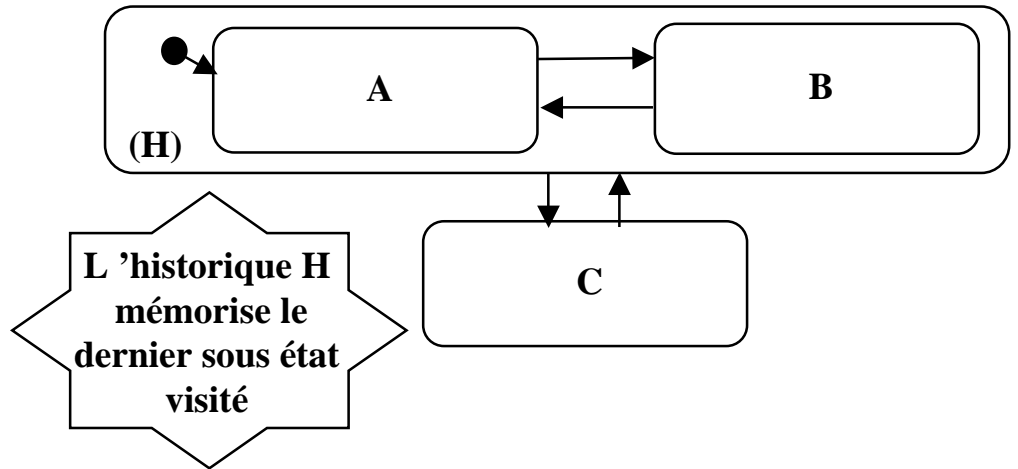


L'opération est exécutée pendant que l'objet est dans un état donné

Les diagrammes d'états-transitions



S est produit cartésien de T et U



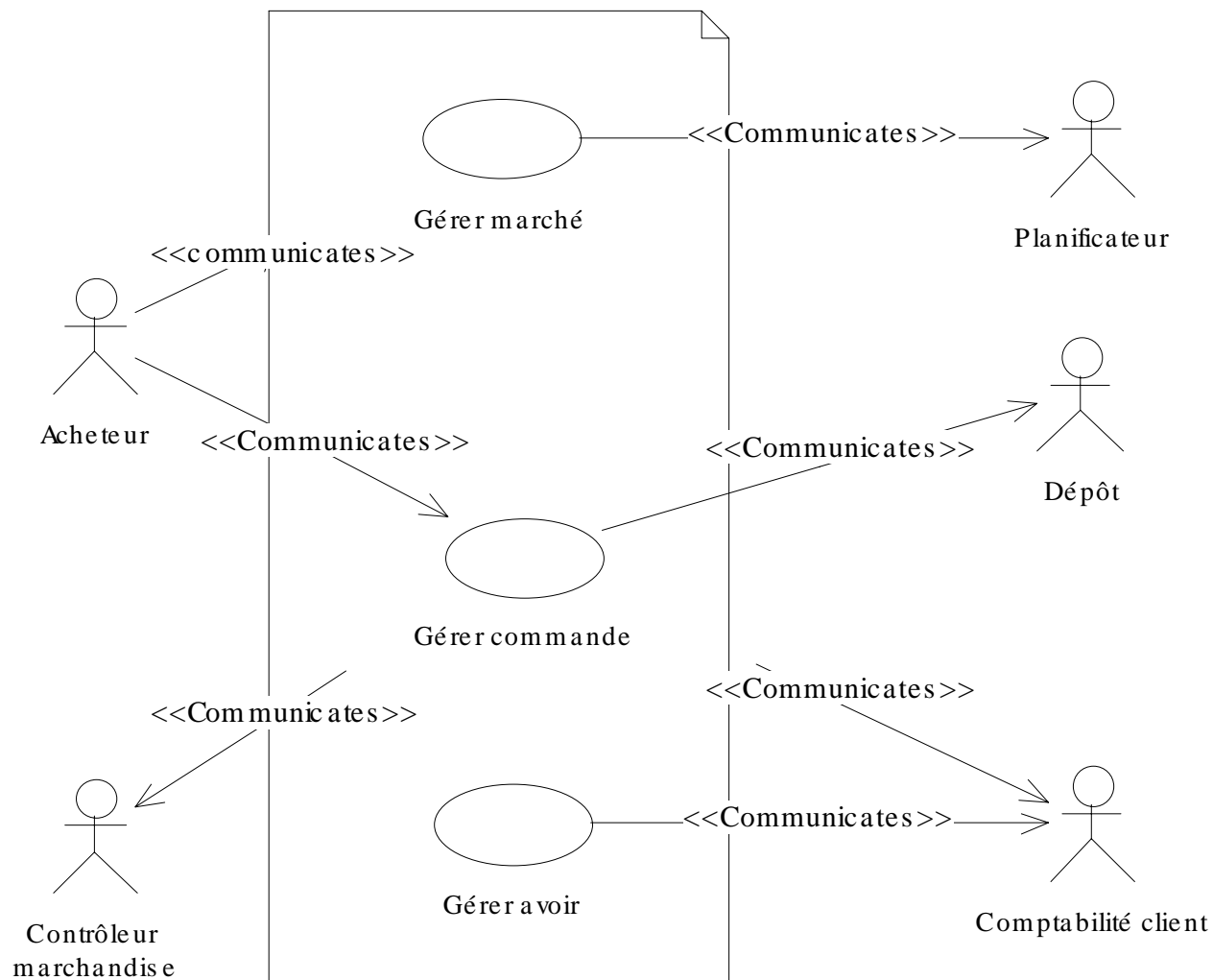
Modélisation du métier :

- **étude du périmètre et des intervenants extérieurs à l'entreprise**
- **étude des processus de l'entreprise**
- **étude des travailleurs et des entités de l'entreprise**
- **étude des workflows des processus**
- **étude des structures organisationnelles**

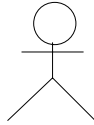
Les modèles métier doivent prendre en compte aussi bien les aspects dynamiques, c'est à dire les flux d'événements à l'intérieur du métier, que les aspects statiques du métier, sa structure, son architecture.

Un processus métier est l'ensemble des activités internes d'un métier dont l'objectif est de fournir un résultat observable et mesurable pour un utilisateur individuel du métier

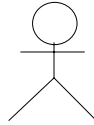
Un diagramme de cas d'utilisation est un graphe d'acteurs, un ensemble de cas englobés par la limite du système, des associations de communication entre les acteurs et les cas d'utilisation



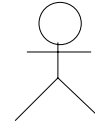
Un acteur est un type stéréotypé représentant une abstraction qui réside juste en dehors du système à modéliser



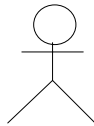
Acheteur



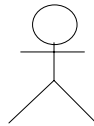
Service juridique



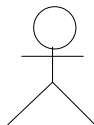
Dépôt



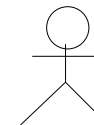
Comptabilité client



Planificateur

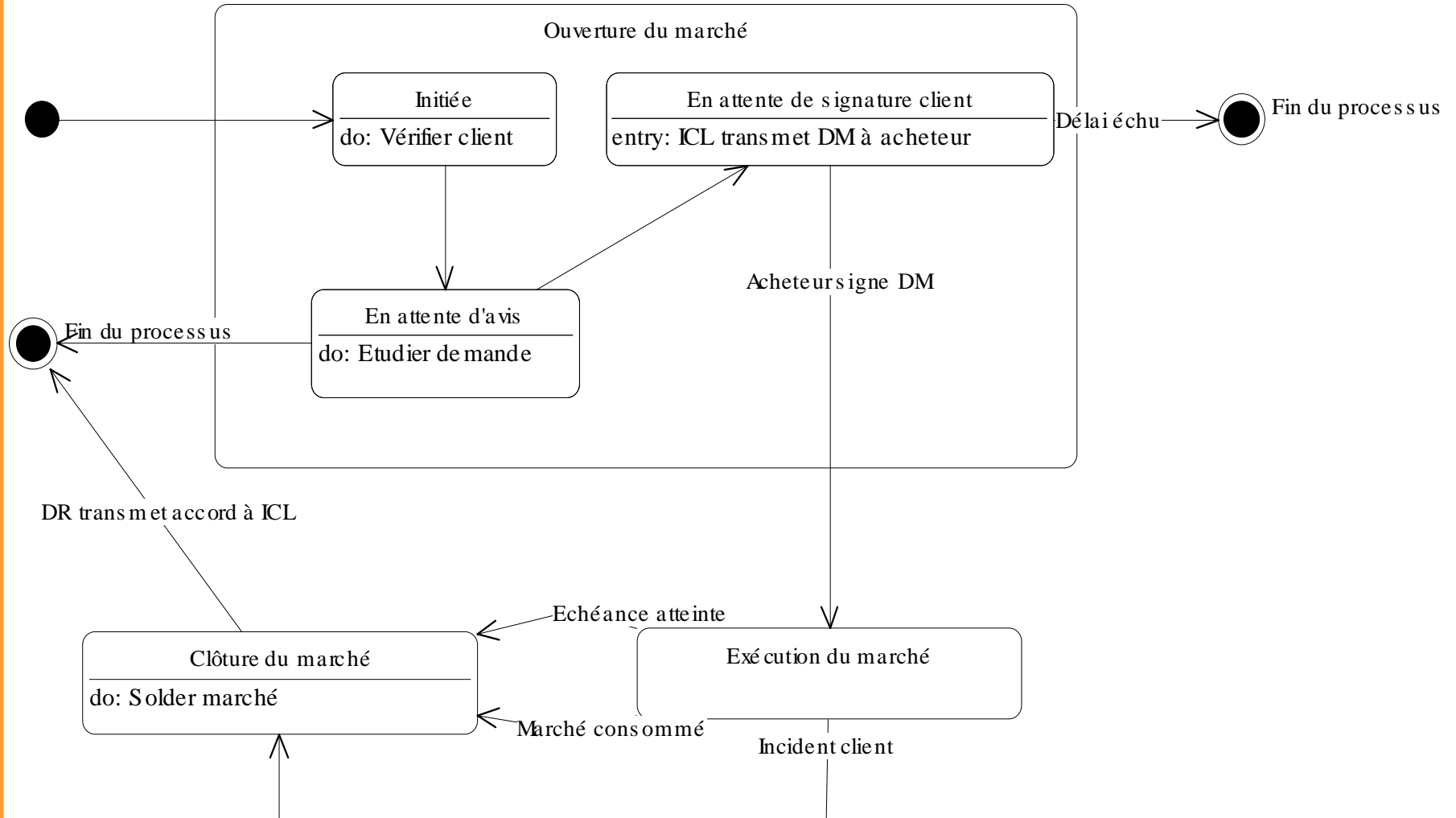


Contrôleur
marchandise



Expert qualité

Description interne d'un processus métier : Gérer marché



Description interne d'un processus métier : Gérer commande

