

INGENIERIE DES SYSTEMES D'INFORMATION

UML

Frédérique LAFoux
Université Paris-Dauphine
L3 Continue – MIDO

Année 2012-2013

OBJECTIFS

- INTRODUCTION AU SI
- MODELISATION UML

DEROULEMENT

- 10 COURS + 10 TDS + 1 EXAMEN

PLAN DU COURS

- **PARTIE I** Présentation des **S**ystèmes d'**I**nformation
- **PARTIE II** UML

BIBLIOGRAPHIE SI

AUSTIN R. – NOLAN R. – O'DONNELL S. – The adventures of an IT leader – Harvard Business Press - 2009

DESFRAY P. – RAYMOND G. – TOGAF en pratique (*The Open Group Architecture Framework*) – modèles d'architecture d'entreprise – Dunod - 2012

GABAY J. - GEBRE B. - La conduite des projets d'évolution des systèmes d'information – InterEditions - 1999

COULEAU-DUPONT A. – TOMBAREL R. - Management des systèmes d'information - DSCG 5 – Manuel, applications et corrigés – Editions Nathan – 2011

KUEVIAKOE D. – Guide du DSI, Profil et responsabilités du Directeur des Systèmes d'Information – Editions ESKA Interactive - 2007

Multi auteurs – Management et gouvernance des SI – Hermès Science - 2009

PEAUCELLE J.L. – Systèmes d'Information. Le point de vue des gestionnaires – Economica – 1999

BIBLIOGRAPHIE UML

BOOCH G. - RUMBAUGH J. - JACOBSON I. - Le guide de l'utilisateur UML – Eyrolles – 2000

DEBRAUWER L. – UML2, Maitrisez la modélisation – 2 volumes – Eni - 2010

GABAY J. - GABAY D - UML 2 Analyse et conception – Dunod – 2008

MULLER P.A. - GAERTNER N. - Modélisation objet avec UML – Eyrolles - 2000

ROQUES P. - VALLEE F. - UML en action – Eyrolles - 2000

ROQUES P. – UML 2 par la pratique - Etude de cas et exercices corrigés – Eyrolles - 2011

RUMBAUGH J., BLAHA M. - Modélisation et conception orientées objet avec UML - Pearson Education, Paris – 2005

FOWLER M. – UML Distilled Third Edition- Addison-Wesley – 2005

Site de l'OMG : <http://www.omg.org/spec/UML/2.3/>

SOURCES

SI :

<http://www.volle.com>

<http://michelvolle.blogspot.com>

<http://www.laurent-bloch.org/>

UML :

<http://www.uml-diagrams.org>

PRÉSENTATION DES SYSTEMES D'INFORMATION

- HISTORIQUE
- DEFINITION
- MOA-MOE
- CYCLE DE VIE PROJET
- ARCHITECTURES

HISTORIQUE



Pourquoi, Comment, Quand ?

La situation au départ

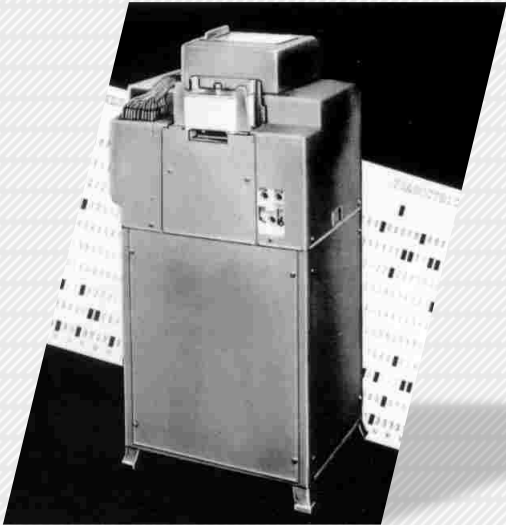
Quels sont les problèmes rencontrés ?

La naissance du SI

Les dates clefs

HISTORIQUE

LA SITUATION AU DEPART : LA MECANOGRAPHIE

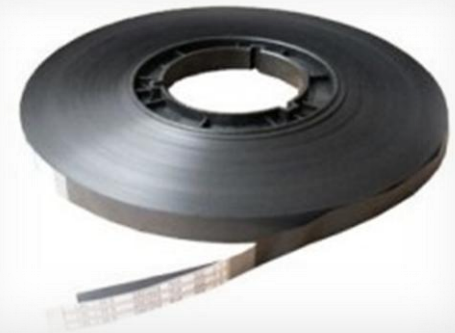


- Créée fin du XIXe siècle pour statistiques (Hollerith)
- Exemple : gestion de stock, paye, banque
- Traitement par lots (*batch processing*)
- Problèmes :
 - Cartes perforées fragiles
 - Bourrage possible
 - Très volumineux
 - Pas en temps réel

HISTORIQUE

LA SITUATION AU DEPART : **EVOLUTIONS** (*hardware*)

- Passage des Cartes perforée aux **Bandes Magnétiques** (1953) avec l'IBM 701
- Puis viennent les premiers **Disques Magnétiques** (1956) avec l'IBM RAMAC
- Langages : Assembleur, **Cobol**, **Fortran**, Pascal



HISTORIQUE

LES EVOLUTIONS **CONTINUENT** (*software*):

- Pas de **persistance** des données, RAM trop petite :

 *Naissance du **Fichier***

- A chaque traitement, la RAM est vidée dans un Fichier :

 ***Incohérence** entre les fichiers*

- Difficultés pour les utilisateur de différencier RAM et Fichier

 *Naissance du **SGBD***

HISTORIQUE

NAISSANCE DES SI

- 1966 : Création d'**IMS** (Information Management System) par IBM



Gestion de stock pour les missions spatiales



Séparation du **Code** et des **Données**



Gestion des accès aux **Données**

HISTORIQUE

VERS LA MODELISATION :

- Outil équivalent au dessin industriel pour l'industrie mécanique
 - ➔ Organigramme (XIXème siècle)
 - ➔ Merise (1980s)
 - ➔ SADT (1981)
 - ➔ UML (1997)

DEFINITION

Kezako ?, périmètre...

Définition générale

Ingénierie Sémantique

Ingénierie des Processus

Ingénierie Contrôle & Stratégie

DEFINITION

DEFINITION GENERALE :

L'expression « **SI** » : fin des années **60**, Pourquoi ?

➔ *Apparition de l'automate programmable : **ordinateur***

Un **Système d'Information (SI)** est un ensemble **organisé** d'éléments qui permet de **regrouper**, de **classifier**, de **traiter** et de **diffuser** de l'information sur un phénomène donné.

DEFINITION

DEFINITION GENERALE : QUALITES

- **Pertinence**

Qualité de ce qui se rapporte à la question posée, de ce qui est approprié

- **Fiabilité**

Aptitude de donner les mêmes résultats dans les mêmes conditions

- **Robustesse / Pérennité**

Aptitude à fonctionner même dans des conditions anormales

- **Validité / Conformité**

Aptitude à réaliser exactement les tâches définies par sa spécification

- **Extensibilité / Evolutivité**

Faculté d'adaptation aux changements de spécification

- **Réutilisabilité**

Aptitude à être réutilisé pour de nouvelles applications

- **Compatibilité**

Aptitude des logiciels à pouvoir être combinés les uns avec les autres

DEFINITION

DEFINITION GENERALE : QUALITES

Pour veiller à la qualité, on va mettre en place des méthodes :

- **CMMI** : Capability Maturity Model Integration
- **ITIL** : Information Technology Infrastructure Library
- **AGILE**

DEFINITION

DEFINITION GENERALE :

Le SI doit répondre à des **exigences** :

- Langage → Ingénierie **Sémantique**
- Action → Ingénierie **Processus**
- Contrôle → Ingénierie **Contrôle**
- Stratégie → Ingénierie **Affaire**

DEFINITION

DEFINITION GENERALE :

Systeme d'Information != Systeme Informatique

- Le système d'information :
 - Réel constituée d'informations organisées,
 - D'événements
 - D'acteurs
 - Des processus
 - Finalité de gestion.
- Le système Informatique :
 - Ensemble d'objets techniques nécessaires au fonctionnement du SI.

 **Navigation & Construction Navale**

DEFINITION

INGENIERIE SEMANTIQUE:



Vision Simpliste du SI :

- Alimente le SI avec des **données**
- Puis on **saisit** et on les **traite**
- On **produit** alors des **résultats**
- Que l'on **conserve**, ainsi que les données saisies
- Afin d'être **consultées** et **transmises**

DEFINITION

INGENIERIE SEMANTIQUE:

Donnée : Définition + Mesure

Information : donnée que l'on transmet

Qualité des données :

- Définition pertinente
- Mesure exacte



Rôle de l'administrateur de données

DEFINITION

INGENIERIE SEMANTIQUE:

Référentiel : Ensemble de

- règles
- documents
- base de données



Identifiants, Nomenclatures, Définitions

2 Formes :

- Documentaire (word, print...)
- Physique (base de données)

DEFINITION

INGENIERIE PROCESSUS:

Processus:

- Succession de tâches qui concourent à l'élaboration d'un produit.
- Emergence début années 90

Modélisation de processus:



Gain de 20-30 % en coût production

- UML
- BPMN



DEFINITION

INGENIERIE **CONTROLE**:

- Il faut que le SI soit bien **utilisé**
➔ *Formation Utilisateurs*

INGENIERIE **STRATEGIE**:

- Problème du positionnement, de ciblage, segment de marché
➔ *Importance de l'**Urbanisation***
- Conjuguer les **ressources** du SI et l'**Urbanisation**
➔ *Assurer la satisfaction des ambitions du SI*

MOA-MOE



Qui fait quoi ?

MOA

MOE

INTERACTIONS

MOA-MOE

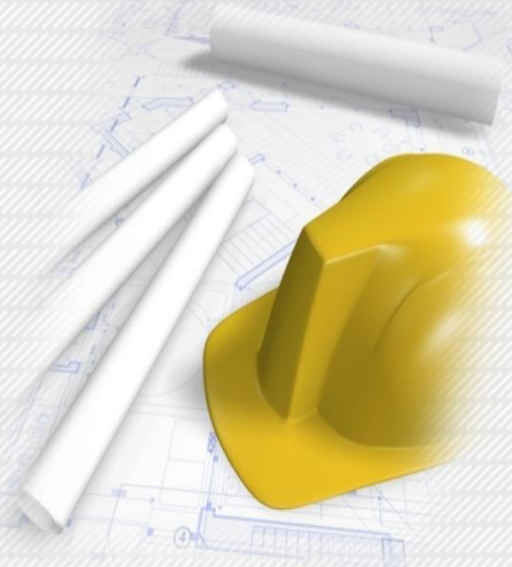
DEFINITION : GENERALITES

Provient du **BTP** :

- **MOA** : Maitrise d'**ouvrage**
➔ *Celui qui fait construire une maison*
- **MOE** : Maitrise d'**œuvre**
➔ *Celui qui organise la construction*

Sens **étymologique** :

- **Ouvrage** : fait de produire
- **Œuvre** : produit



MOA-MOE

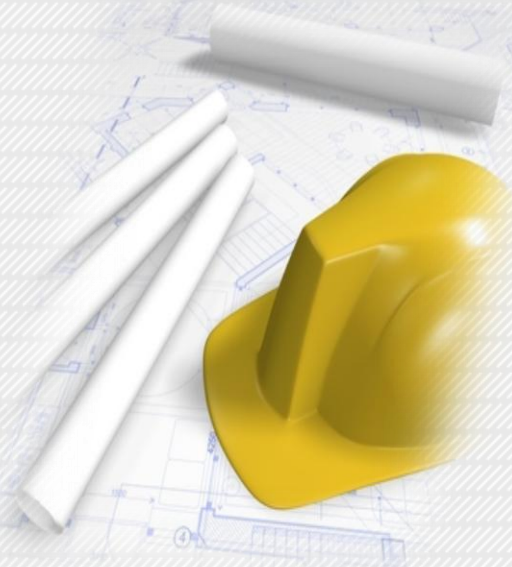
DEFINITION : MOA (Maitrise d'ouvrage)

- Entité responsable de l'organisation et des méthode de travail autour du SI, responsable de la bonne compréhension entre les métiers et la DSI. Donneur d'ordre de la DSI.

➔ Située entre les utilisateurs et la MOE.

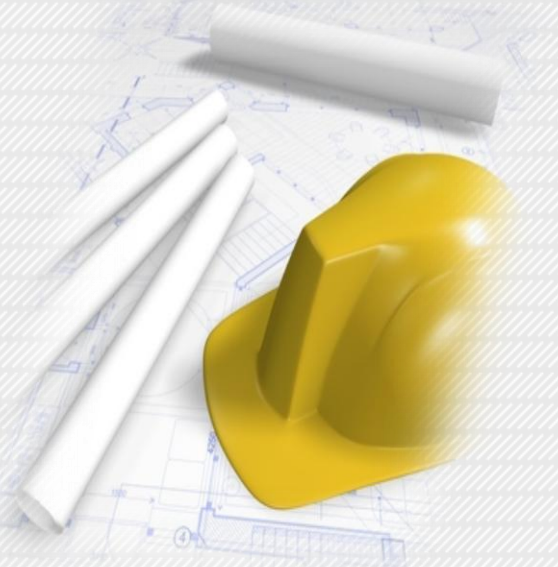
- Rôles MOA :

- décrire les **besoins**,
- le **cahier des charges**,
- établit le **financement** et le **planning** général des projets,
- fournit au MOE les **spécifications fonctionnelles générales** et **valide** la recette fonctionnelle des produits,
- **coordonne** les instances projets entre les utilisateurs métiers et la MOE,
- assura la responsabilité de **pilotage du projet** dans ses grandes lignes, adapte le périmètre fonctionne en cas de retard dans les travaux , pour respecter la date de la livraison finale.



MOA-MOE

DEFINITION : MOE (Maitrise d'oeuvre)



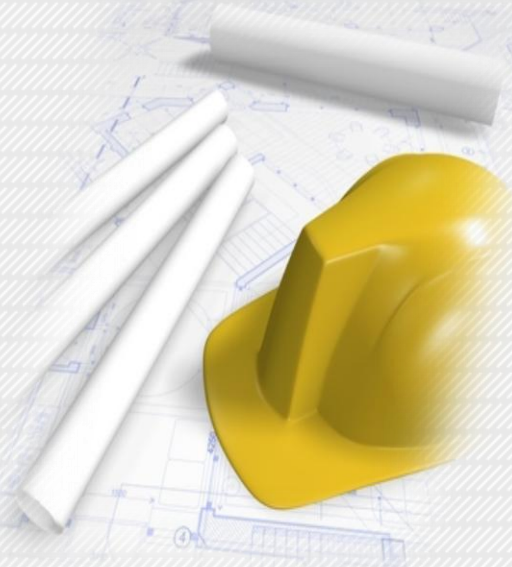
- **Réalisateur technique** du projet, elle en conçoit la solution informatique.
- La MOA est son **client**
- Composée de la DSI de l'entreprise et du réalisateur (prestataires, éditeurs...)

MOA-MOE

DEFINITION : **DES CONFLITS**

Communication => Conflits

- Commercial / Production
- R&D / Marketing
- ➔ Arbitrage revient au DG ou MOAS



CYCLE DE VIE PROJET



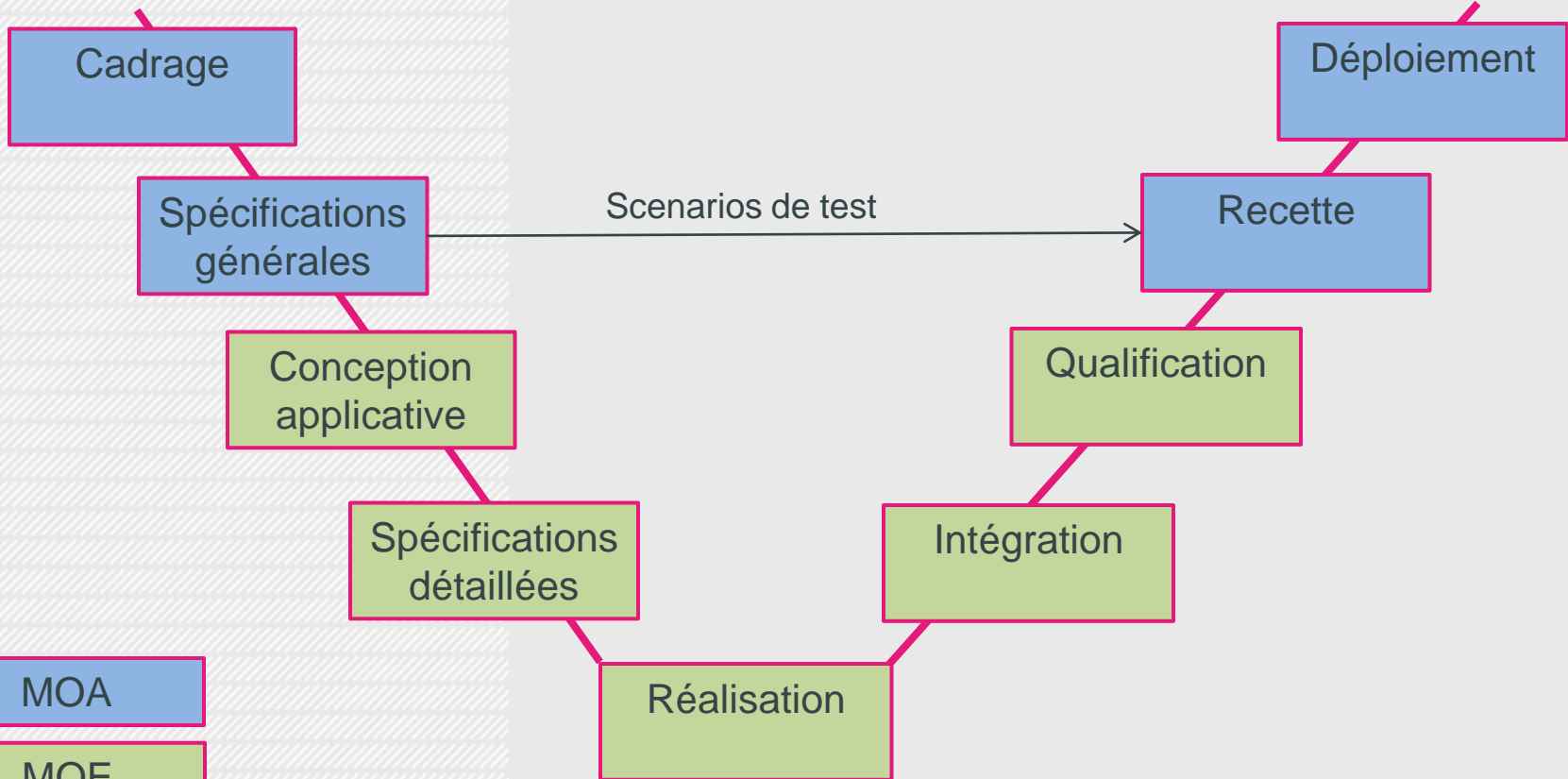
On fait quoi ? Quand ? Avec qui ?...

Le cycle en V

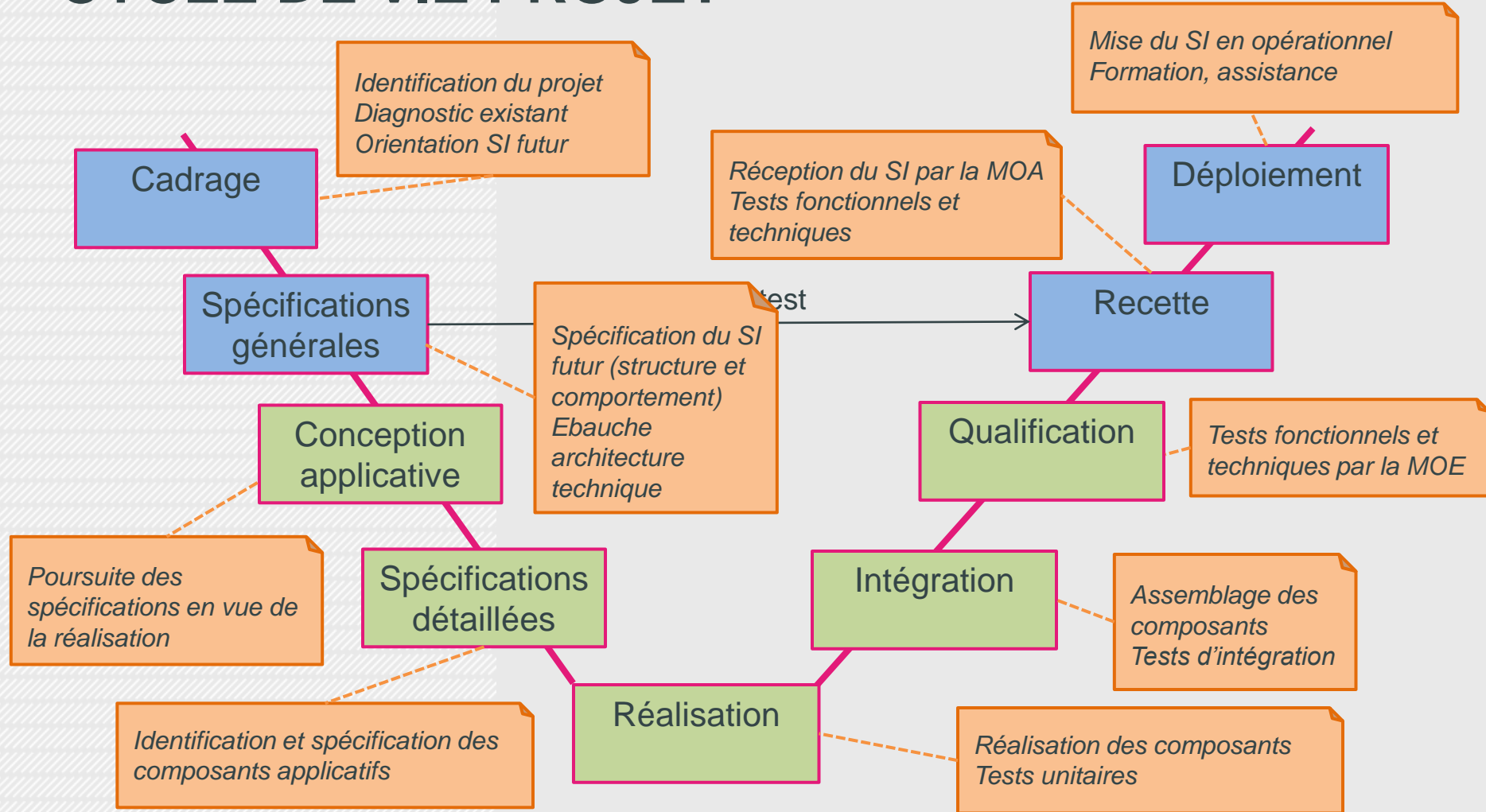
Détails

CYCLE DE VIE PROJET

LE CYCLE EN V :



CYCLE DE VIE PROJET



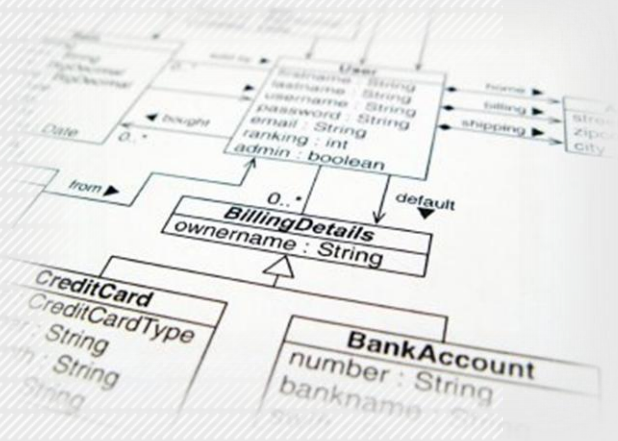
SENSIBILISATION A LA MODELISATION

Pourquoi ? Dans quels buts ?

Définition

Historique

Exemples



SENSIBILISATION A LA MODELISATION

DEFINITION:

Modèle :

- **Abstraction** de la réalité
- Description de tout ou partie d'un système dans un langage bien défini.
- Ensemble de **concepts**, **règles**, un **formalisme**
- Vue subjective mais pertinente de la réalité

Pourquoi Modéliser ?

- Fournir des **spécifications** claires : produire, exploiter
- **Clarifier** les objets, les concepts, les référentiels, les processus.

- Pour quel processus je travaille ?
- Quel rôle j'ai dans ce processus ?
- Quel est l'ensemble des processus de mon entreprise ?

SENSIBILISATION A LA MODELISATION

HISTORIQUE :

- 1980s : Apparition des langages programmation OO (c++)
- Réflexion standardisation de l'OMG (Object Management Group)
 - ➔ Echech, trop de protestations
- Quelques années plus tard....
 - ➔ 1997 : UML 1.0
- Pendant ce temps la... en France...
 - ➔ 1980s : Merise

SENSIBILISATION A LA MODELISATION

EXEMPLES:

- Développement classique
- Echanges de données
- Développement : MDA (Model Driven Architecture)

INTRODUCTION UML



Kezako ? Avec quoi ?

Définition

Historique

Objectifs

Remarques

Les Diagrammes

Aperçu

Metamodel

INTRODUCTION UML



DEFINITION:

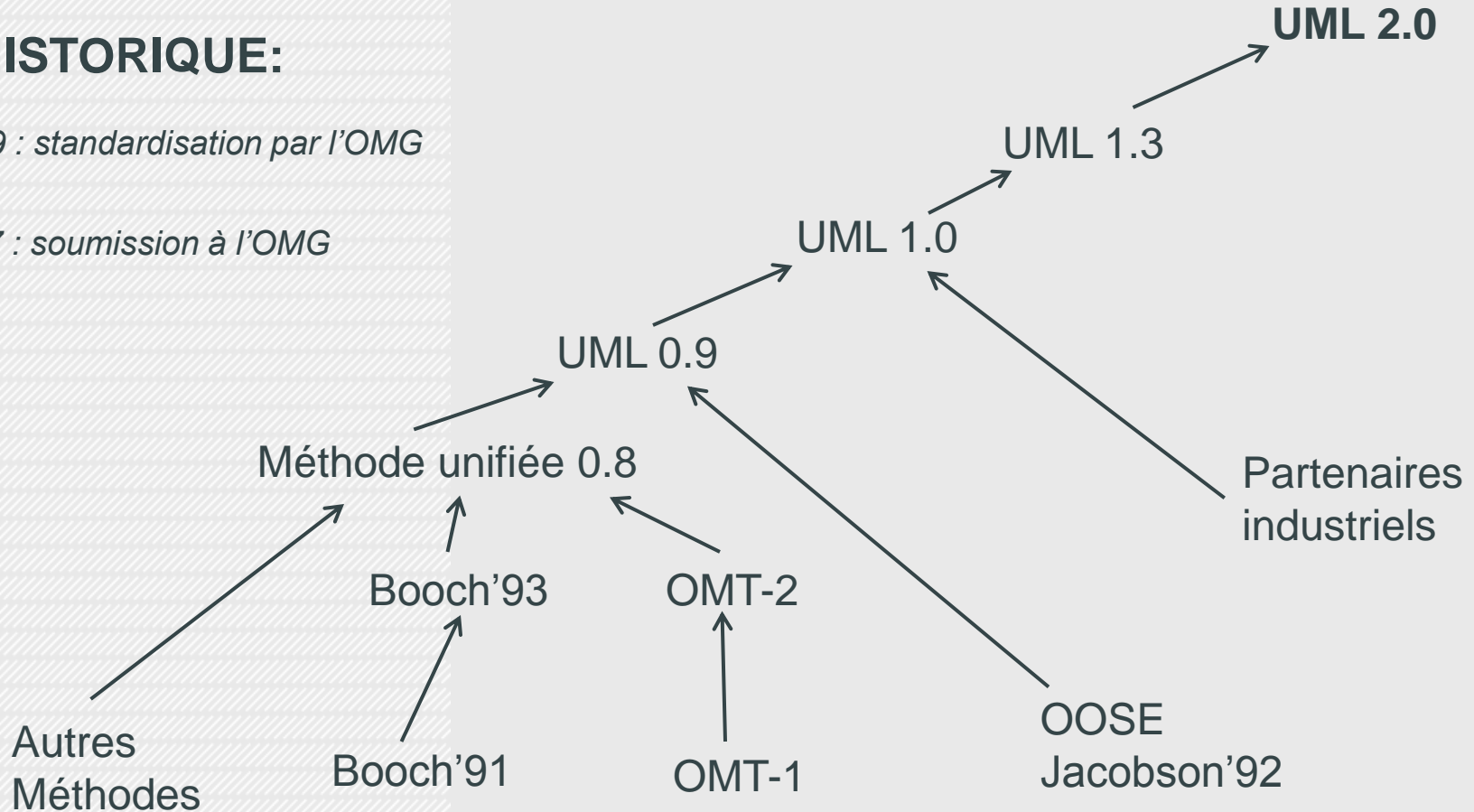
- **Langage** destiné aux phases amont de la réalisation d'un logiciel.
- Technique de modélisation unifiée issue de méthodes orientées objets plus anciennes :
 - OMT (James Rumbaugh),
 - OOSE (Ivar Jacobson),
 - OOD (Grady Booch)
- Pourquoi des **standards** ?
 - ➔ *Faciliter la **communication***

SENSIBILISATION A LA MODELISATION

HISTORIQUE:

1999 : standardisation par l'OMG

1997 : soumission à l'OMG



INTRODUCTION UML



OBJECTIFS:

- Représenter des **systemes** entiers.
- Prendre en compte les facteurs d'**échelle**
- Créer un langage de **modélisation**
 - Utilisables par les hommes & machines
 - Compatible avec toutes les techniques de réalisation
 - Adapté à toutes les phases du développement
- **Indépendant** des langages de programmation

INTRODUCTION UML



OBJECTIFS:

- **Visualiser**
 - *Chaque symbole graphique a une sémantique*
- **Spécifier**
 - *de manière précis et complète, sans ambiguïté*
- **Construire**
 - les classes, les relations,
- **Documenter**
 - les diagrammes, notes, contraintes, exigences

INTRODUCTION UML

REMARQUES :

- UML = Langage
 - ➔ *Pas une méthode*
- Des problèmes de **communication** persistes :
 - diagrammes complexes : difficiles à comprendre
 - diagrammes simples : sujets à interprétation
 - « mal s'exprimer » : apparition d'erreurs, d'imprécisions...

INTRODUCTION UML

LES DIAGRAMMES:

- UML utilise des **diagrammes** :
 - Possibilité de visualiser et manipuler des éléments de **modélisation**
 - Représentation **graphique** d'une séquence d'opérations ou de la structure d'un système
 - Simplicité de compréhension

INTRODUCTION UML

LES DIAGRAMMES: 9 Diagrammes au total (UML 1.4)

4 **Statiques** (structures – données) :

- Diagramme d'objets
- Diagramme de classes
- Diagramme de composants
- Diagramme de déploiement

5 **Dynamiques** (comportement) :

- Diagramme de cas d'utilisation
- Diagramme de collaboration
- Diagramme de séquence
- Diagramme d'états-transition
- Diagramme d'activités

INTRODUCTION UML

RELATIONS AVEC CYCLE :

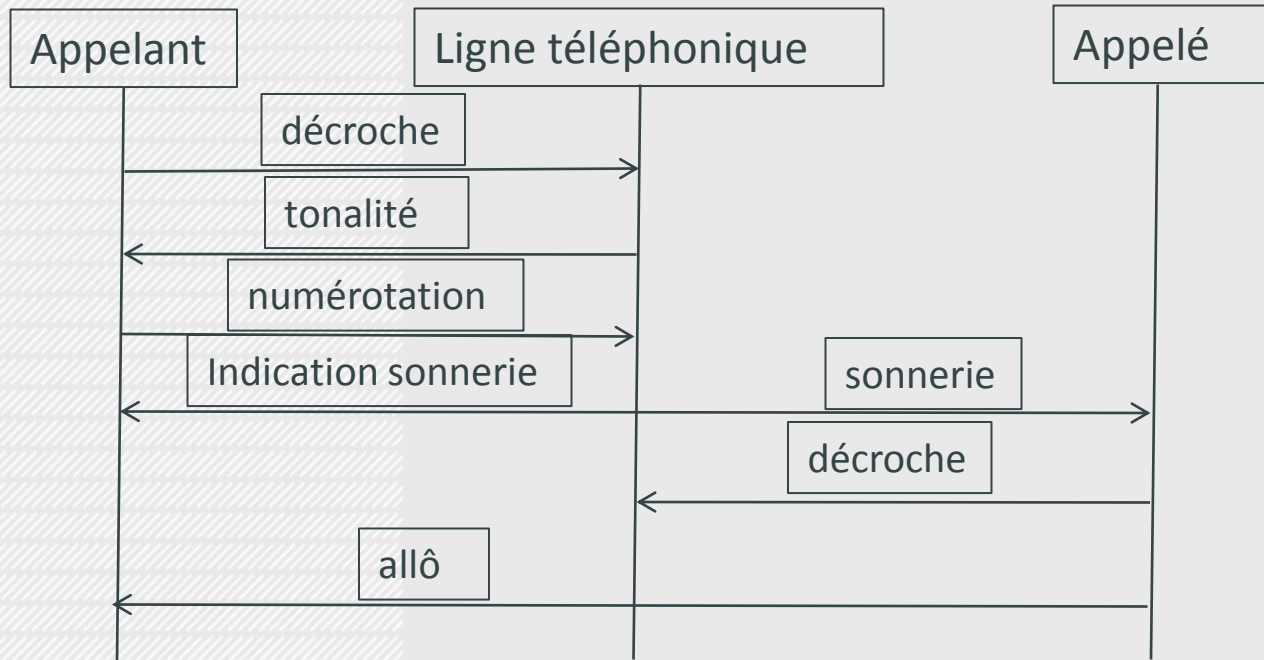
➔ Découverte et analyse des besoins :

- **Diagramme de cas d'utilisation** : décrit les **fonctions** du système selon le point de vue de ses futurs **utilisateurs**
- **Diagramme de séquence** : représentation **temporelle** des objets de leur interactions
- **Diagramme d'activités** : pour des représentations **grossières**, ils représentent le **comportement** d'un méthode ou d'un cas d'utilisation.

INTRODUCTION UML

RELATIONS AVEC CYCLE :

Exemple de diagramme de séquence :



INTRODUCTION UML

RELATIONS AVEC CYCLE :

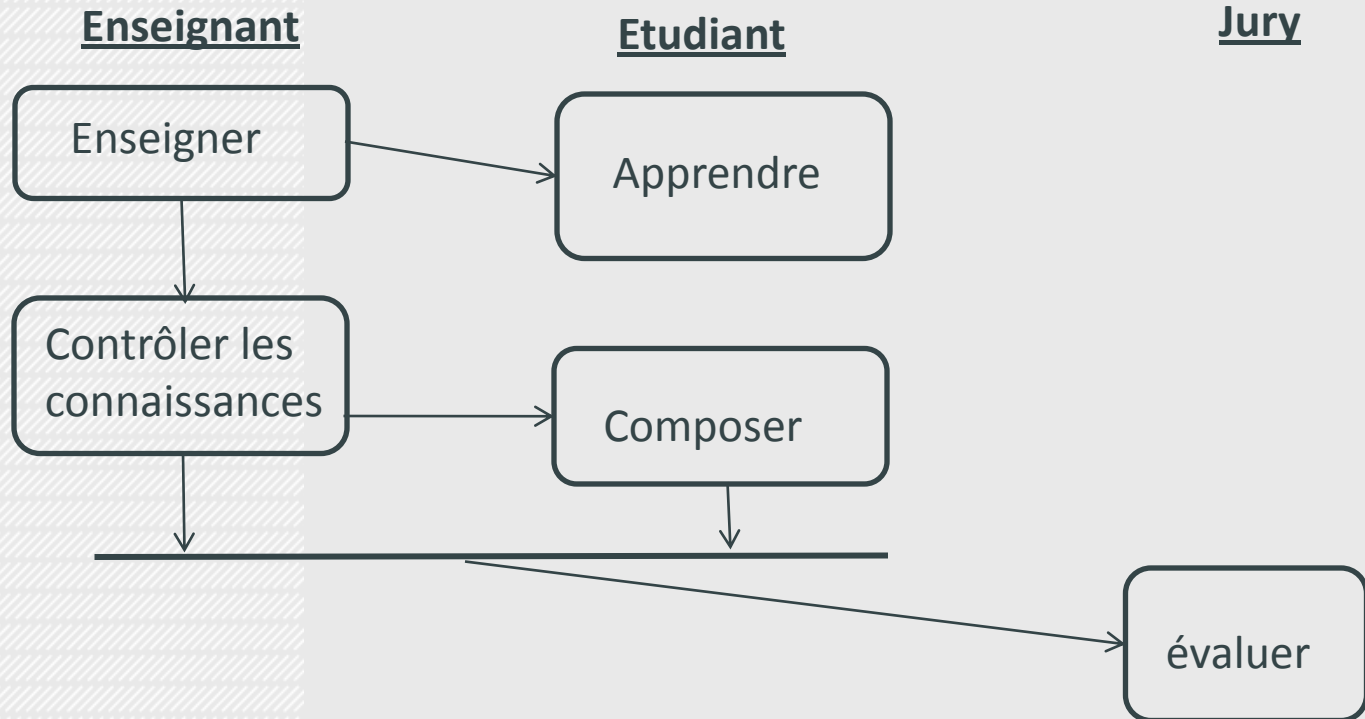
→ Spécification :

- **Diagramme de classes:** structure des données du système définies comme un ensemble de relations et de classes.
- **Diagramme d'objets:** illustration des objets de leurs relations. Ce sont des diagrammes de collaboration simplifiés sans représentation des envois de messages.
- **Diagramme de collaboration:** représentation des interactions entre objets.
- **Diagramme d'états-transitions :** représentation du comportement des objets d'une classe en terme d'états et de transition d'états
- **Diagramme d'activité :** pour des représentations grossières, ils représentent le comportement d'un méthode ou d'un cas d'utilisation.

INTRODUCTION UML

RELATIONS AVEC CYCLE :

Exemple de diagramme d'activités:



INTRODUCTION UML

RELATIONS AVEC CYCLE :

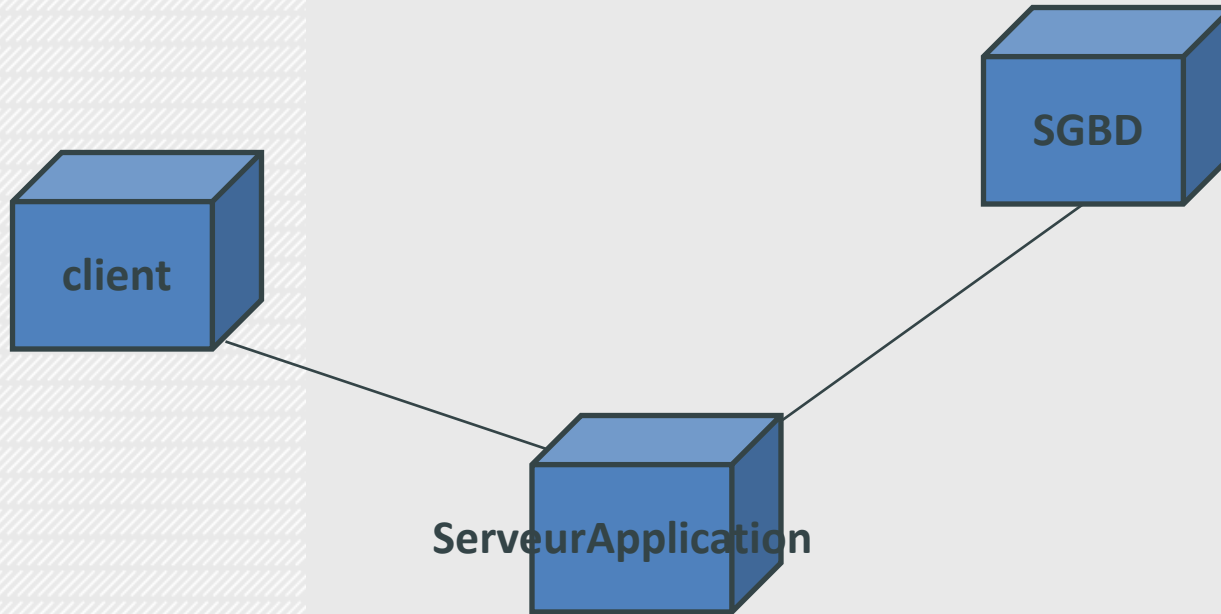
→ Conception:

- **Diagramme de séquence:** représentation des interactions temporelles entre objets et dans la réalisation d'une opération.
- **Diagramme de composants:** architecture des composants physique d'une application.
- **Diagramme de déploiement:** description du déploiement des composants sur les dispositifs matériels

INTRODUCTION UML

RELATIONS AVEC CYCLE :

Exemple de diagramme de déploiement:



INTRODUCTION UML

RELATIONS AVEC CYCLE :

 Attention:

- **Un même type de diagramme peut :**
 - Modéliser des concepts différents
 - Être utilisé à des moments différents du processus de développement
 - Être à différents niveaux d'abstraction

- Ne pas être utilisé

INTRODUCTION UML

ET MAINTENANT...

... ALLONS UN PEU PLUS LOIN

(hors programme)

INTRODUCTION UML

METAMODEL

Méta-méta-modèle

Définit un langage pour
spécifier un méta-modèle

MOF

Méta-modèle

Instance du méta-méta-
modèle, définissant un langage

UML

Modèle

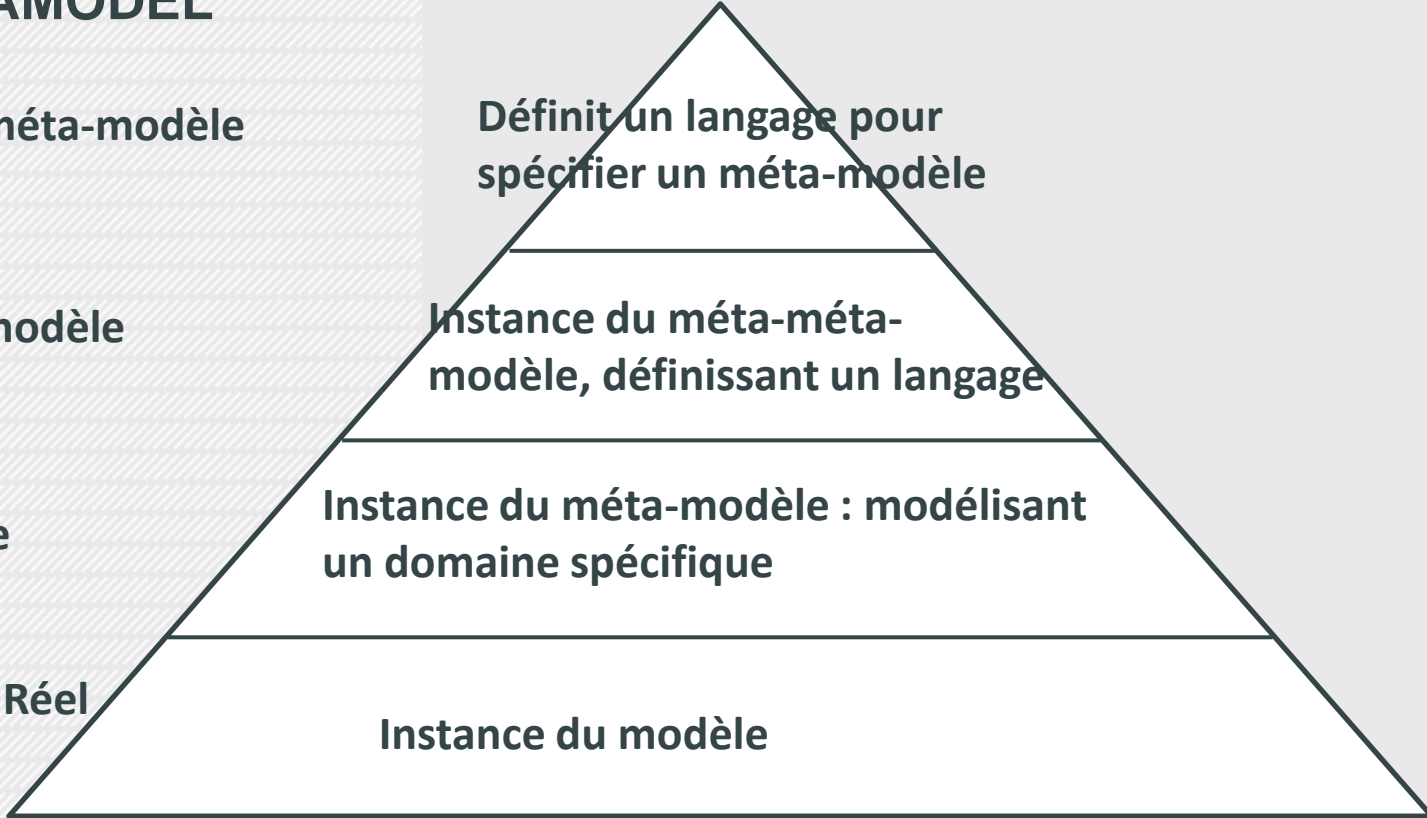
Instance du méta-modèle : modélisant
un domaine spécifique

Diagramme
de classe

Monde Réel

Instance du modèle

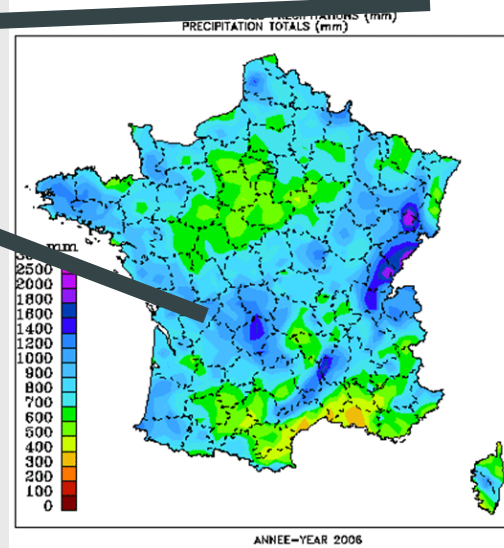
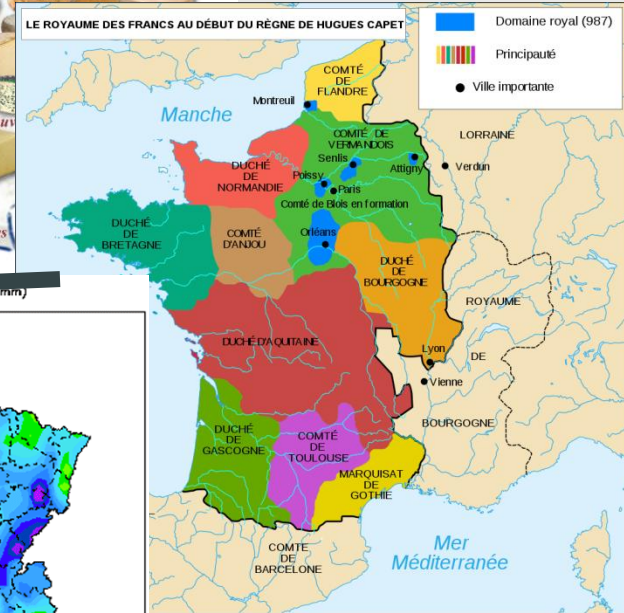
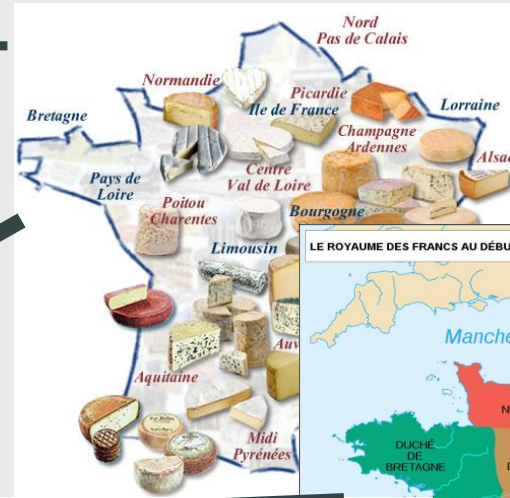
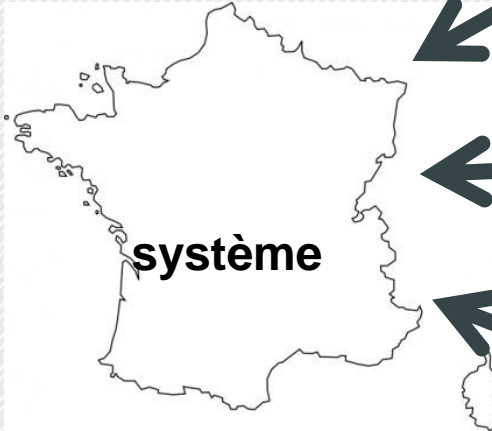
Application



INTRODUCTION UML

MODELE ?

représente



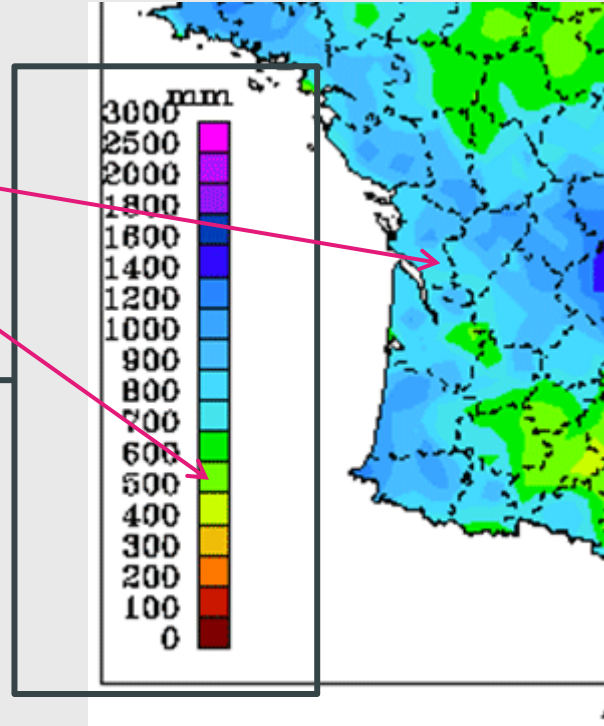
modèles

INTRODUCTION UML

META MODELE

Même notation
graphique, contextes
différents, significations
différentes

La légende représente
le meta-modèle



INTRODUCTION UML

META-META MODELE

Une carte a une légende :

- Dans un coin
- Avec un cadre
- Avec des correspondance icône-texte

UML : GENERALITES



Diagrammes, Stéréotypes, Extensions

Rappel Diagrammes

Niveaux de visions

Formalisme Diagrammes

Stéréotypes

Extensions

UML : GENERALITES

RAPPEL DIAGRAMMES

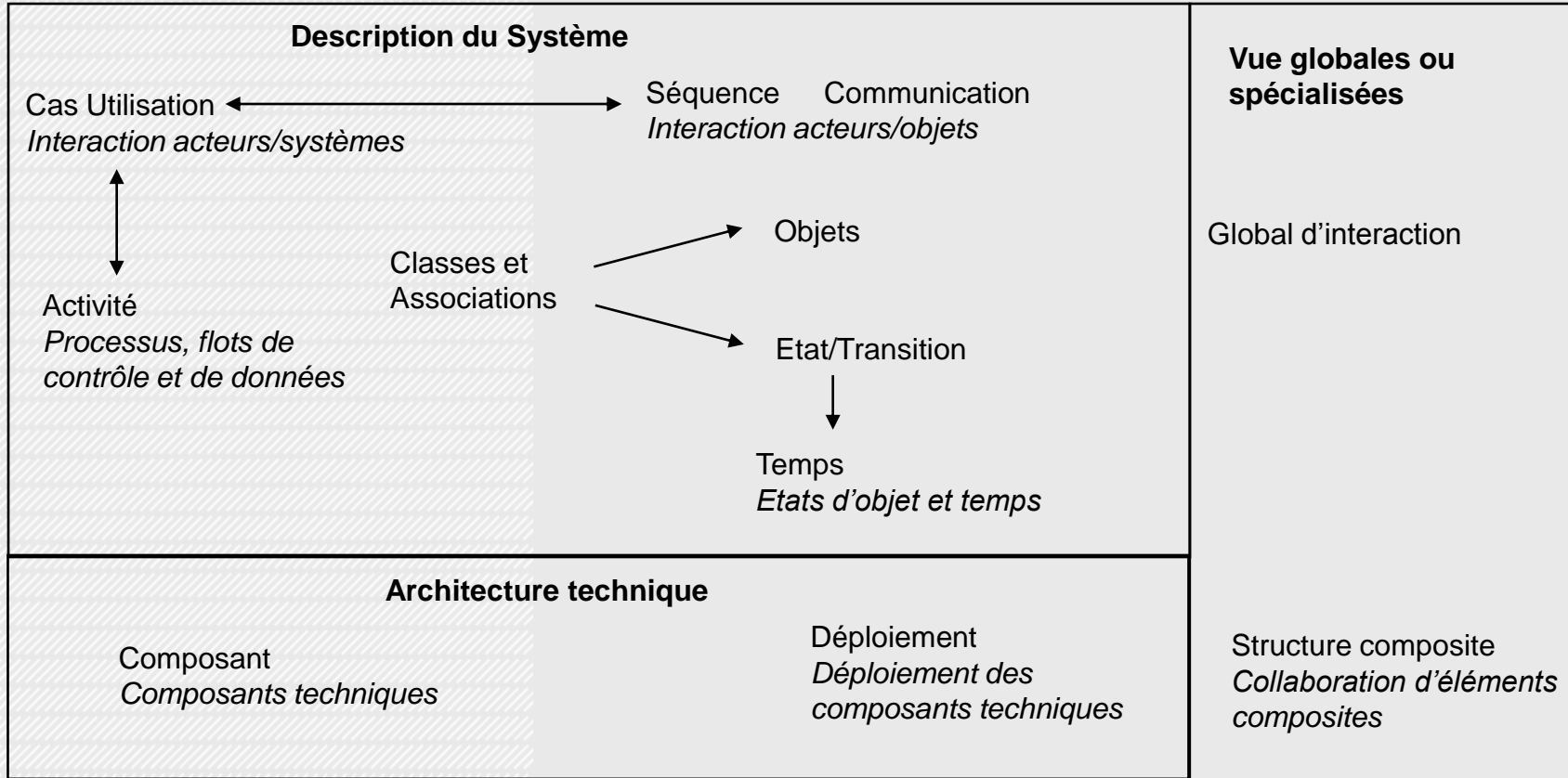
Statiques (structures – données) :

- Diagramme d'objets
Object diagram
- Diagramme de classes
Class diagram
- Diagramme de composants
Component diagram
- Diagramme de déploiement
Deployment diagram
- Diagramme de paquetages
Package diagram
- Diagramme de structure composite
Composite Structure diagram

Dynamiques (comportement) :

- Diagramme de cas d'utilisation
Use case diagram
- Diagramme de collaboration
(**communication**)
Communication diagram
- Diagramme de séquence
Sequence diagram
- Diagramme d'états-transition
State Machine diagram
- Diagramme d'activités
Activity diagram
- Diagramme global d'interaction
Interaction Overview diagram
- Diagramme de temps
Timing diagram

UML : GENERALITES

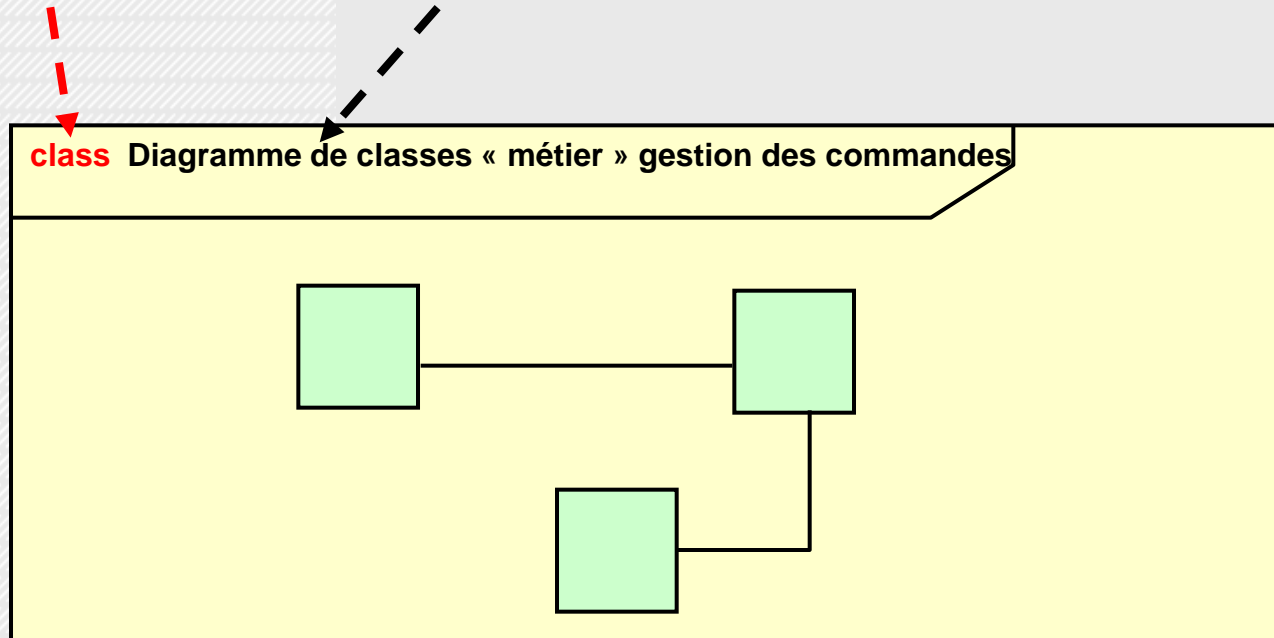


UML : GENERALITES

FORMALISME DIAGRAMME

Type de diagramme

Nom du modèle



UML : GENERALITES

EXTENSION : STEREOTYPES

- **Image** préconçue d'un sujet, habituellement admise et véhiculée.
➔ Représentation **connue** de tout le monde



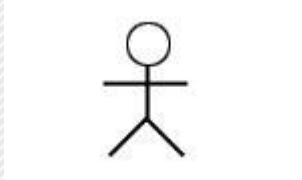
EN UML :

- Mécanisme d'**extensibilité**
- Permet d'**étendre** le vocabulaire
- Associé à : classe, opérations, attributs, paquetages.
- **Notation** : << stereotype >> + élément graphique

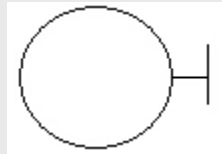
UML : GENERALITES

EXTENSION : STEREOTYPES

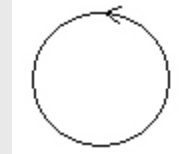
- Inclus dans UML :



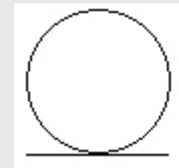
Acteur



Interface
boundary



Controleur
Controller



Entité persistante
entity

- Mais on peut aussi les créer et les définir (extension du méta-model)

RM

<< règle métier >>

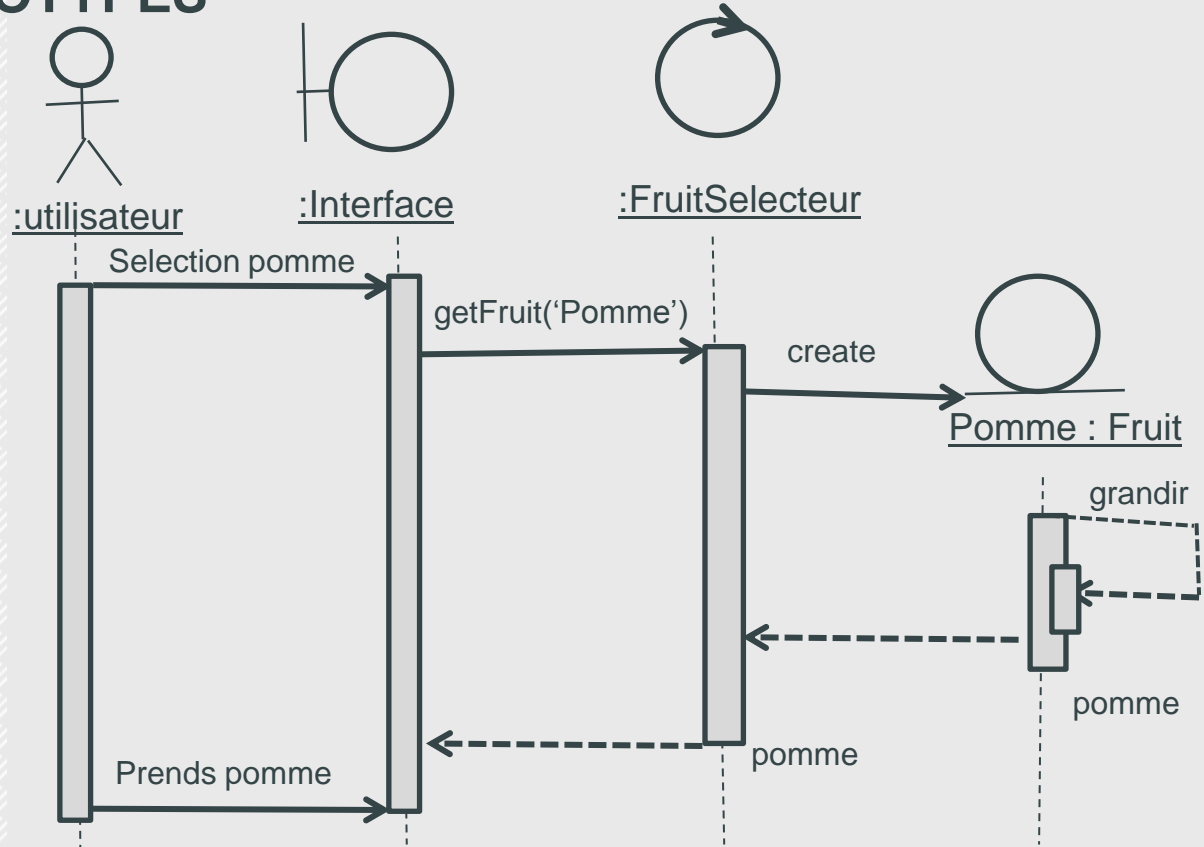
UML : GENERALITES

EXTENSION : STEREOTYPES

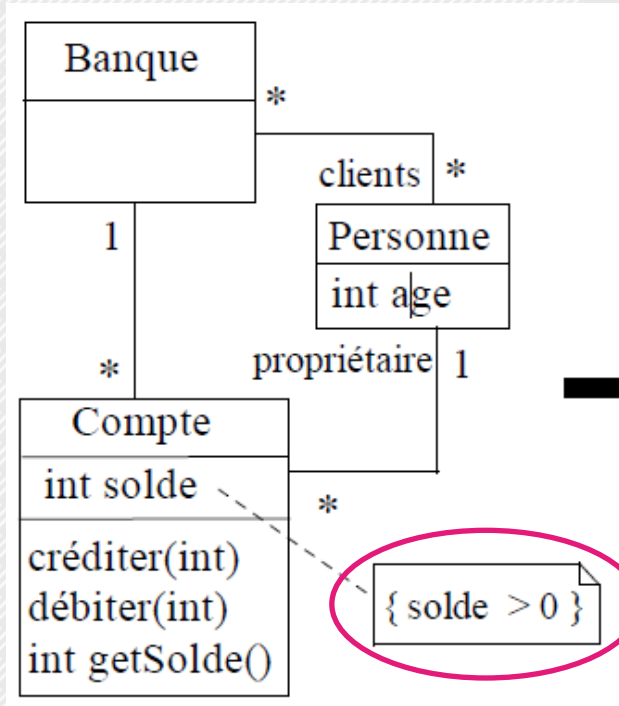


Extrait d'un diagramme de classe

Diagramme de séquence



UML : GENERALITES

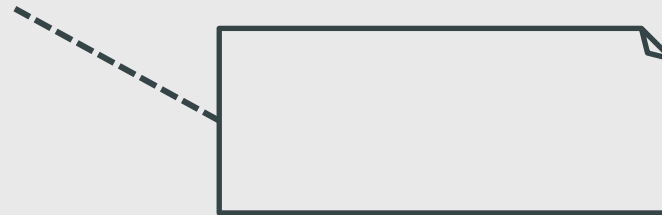


*Diagramme de classe commenté
par de l'OCL*

Autre Extension : Les commentaires

Pour compléter les descriptions, utilisation du symbole avec autant de texte que nécessaire à l'intérieur.

Utilisables dans tous les diagrammes



UML : GENERALITES

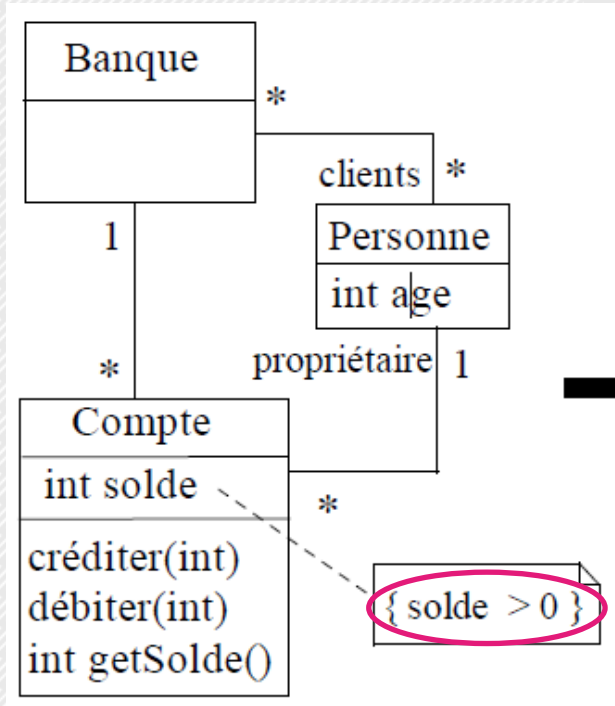


Diagramme de classe utilisant OCL

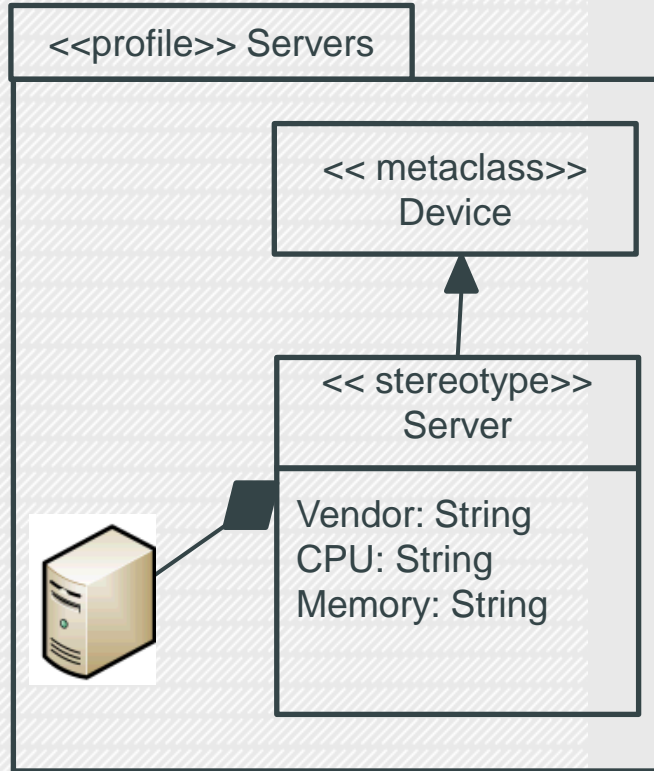
Autre Extension : Les contraintes

Une information sémantique associée à un élément du modèle et que le modèle doit satisfaire pour être correct.

Utilisation des symboles { et } autour du texte décrivant les limites imposées
UML utilise un langage de contrainte : « OCL »
(Object Constraint Language)

Utilisables dans pratiquement tous les diagrammes

UML : GENERALITES



Définition d'un profil

Autre Extension : Les profils

Extension légères des règles de UML pour permettre d'adapter de spécialiser UML à des cas précis et sans contradiction avec les règles de base

Nb : on définit un metaclass : couche meta modèle

Un profile peut définir :

- des classes,
- des datatypes,
- des primitive types,
- des enumerations

/!\ Parfois appelé : Diagramme de profils

UML : GENERALITES

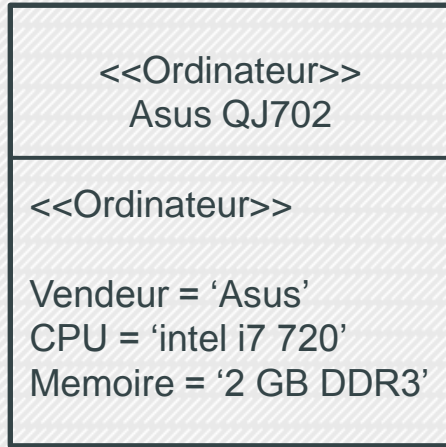


Figure a

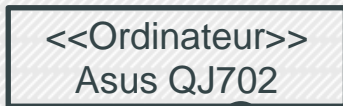
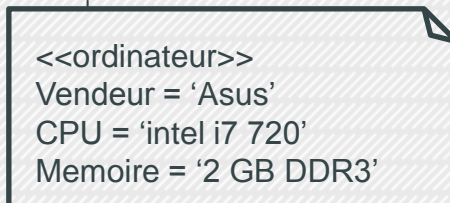


Figure b



AUTRES EXTENSIONS : Tagged Values

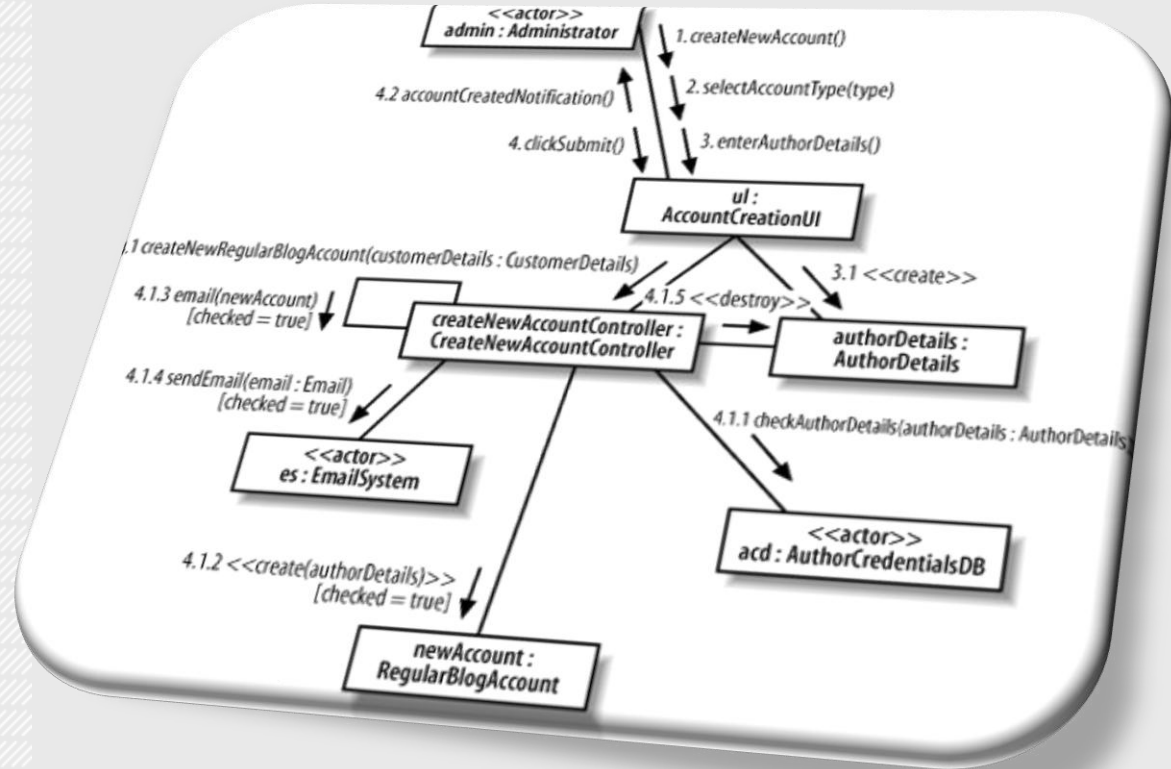
Propriétés définies par l'utilisateur

L'utilisateur peut donner des valeurs par défaut aux attributs. La valeur par défaut est précédée du symbole « égal » (=)

Depuis UML 2.X : seulement sur un attribut défini par un stereotype.

- Une paire 'attribut-valeur' par ligne
- Soit dans un compartiment (figure a)
- Soit dans un commentaire (figure b)

UML : DIAGRAMME DE COMMUNICATION



UML : DIAGRAMME DE COMMUNICATION

GENERALITES :

2 noms :

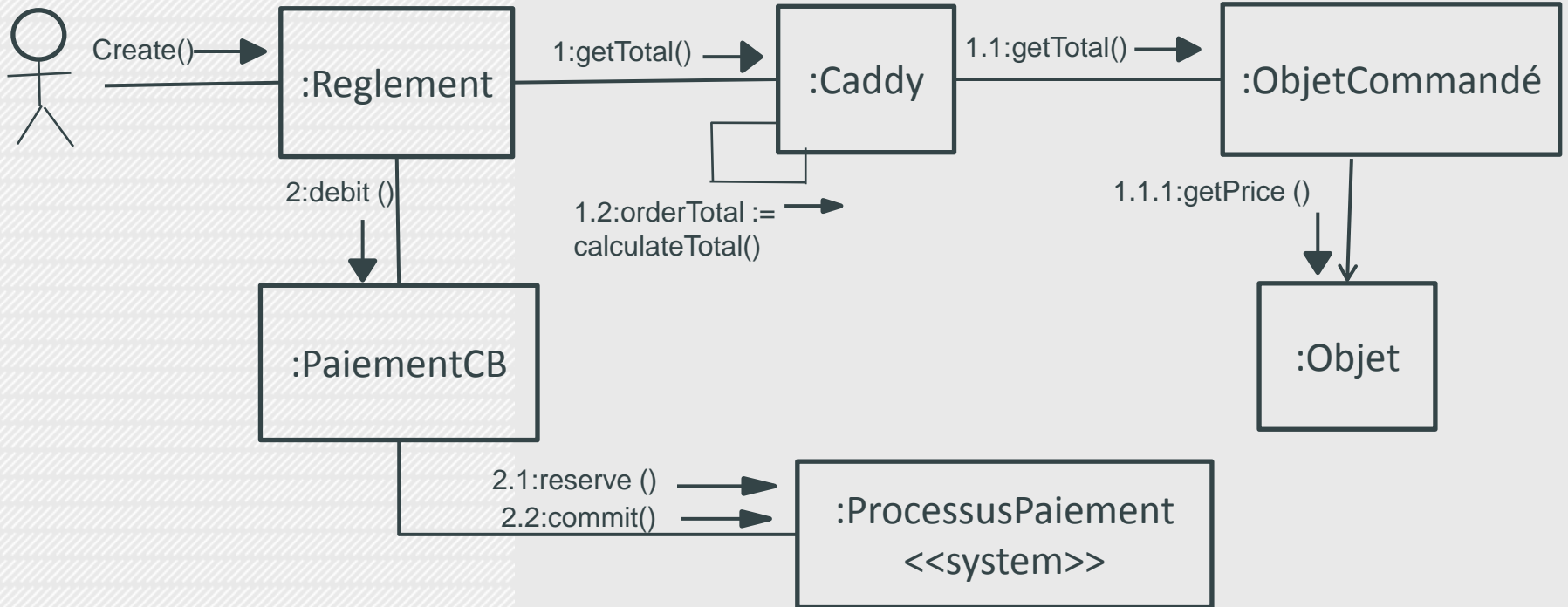
- Diagramme de **collaboration** (UML 1.X)
- Diagramme de **communication** (UML 2.X)

Version **simplifié** du diagramme de **séquence**

2 vues différentes mais logiquement identiques : **isomorphe**

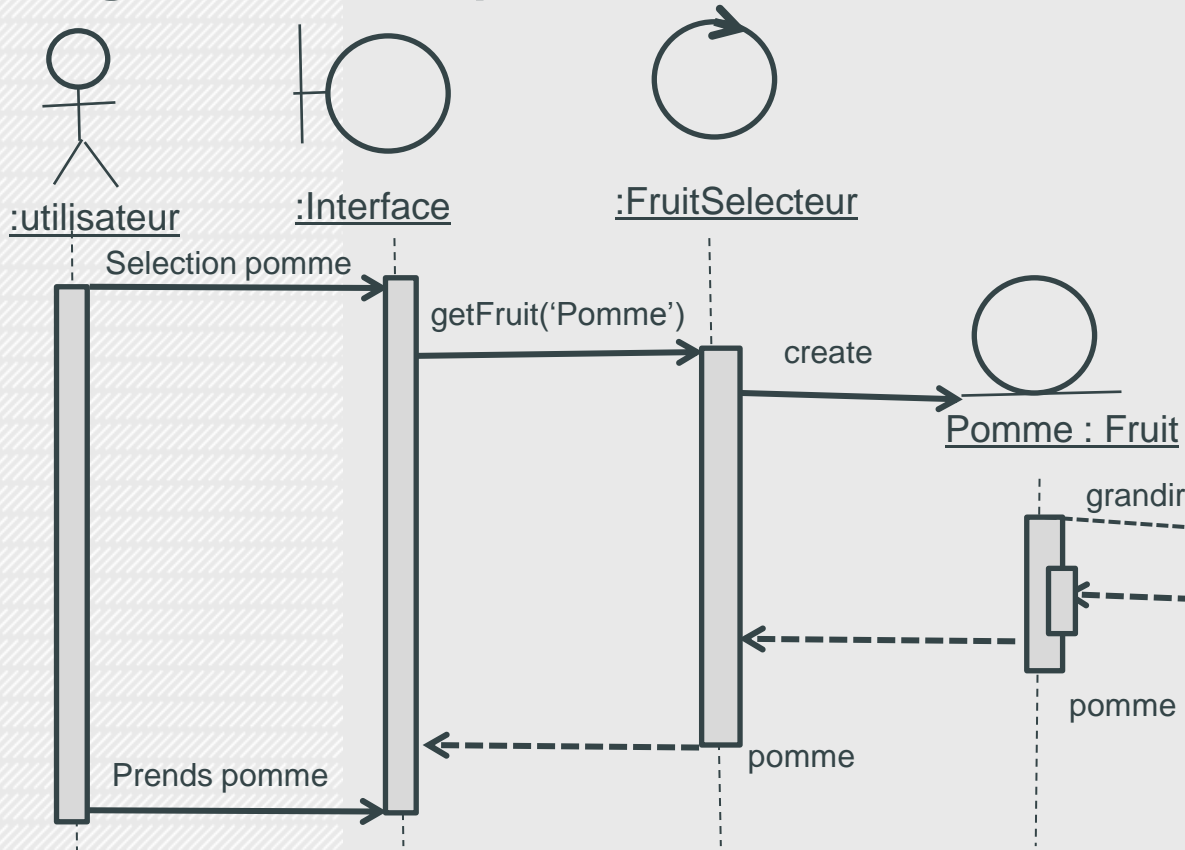
UML : DIAGRAMME DE COMMUNICATION

GENERALITES :



UML : DIAGRAMME DE COMMUNICATION

Rappel : diagramme de séquence



UML : DIAGRAMME DE COMMUNICATION

PRINCIPES :

- **Libre** de placer les participants (objets).
- On fait des **liens** entre eux, et on les numérote (**interactions**)
- Pas de **ligne de vie**

BUTS :

- Comportement **collectif** d'objets
- En vue de réaliser une **opération**

UML : DIAGRAMME DE COMMUNICATION

DEFINITIONS :

- Une collaboration est une collection d'objets et d'acteurs liés entre eux
- Une collaboration définit un ensemble de participants et de relations qui sont sensés pour un périmètre donné
- Une collaboration entre 2 objets travaillants ensemble produit une fonctionnalité.
- Les objets collaborent entre eux par communications (s'échangeant des messages)

UML : DIAGRAMME DE COMMUNICATION

QUAND L'UTILISER ? :

- Phase de **cadrage**
- Début de projet pour **clarifier** le domaine d'étude.

- Représenter collaboration entre le domaine d'étude et les partenaires
Permet de déterminer :
 - Les flux entrants
 - Les **flux** sortants
 - Les **acteurs** externes
 - Les **domaines** connexes

UML : DIAGRAMME DE COMMUNICATION

QUOI UTILISER ? :

Diagramme de **séquence** ou Diagramme de **communication** ?

Majorité des personnes : Diagramme de séquence

Diagramme de communication :

- Adapté pour montrer les liens

Diagramme de séquence:

- Importance des messages

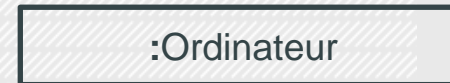
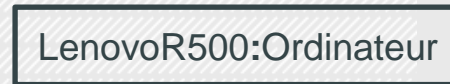
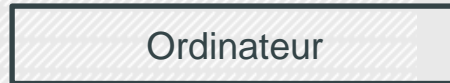
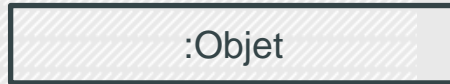
UML : DIAGRAMME DE COMMUNICATION

3 CONCEPTS IMPORTANTS :

- Les Objets
- Les liens d'interactions
- Les Messages

UML : DIAGRAMME DE COMMUNICATION

OBJETS :



- Représenté par un rectangle
- Nommage :
 - Nom de l'objet instancié
 - Nom de l'objet et nom de la classe
 - Nom de la classe
- UML 2.X : plus de soulignement

UML : DIAGRAMME DE COMMUNICATION

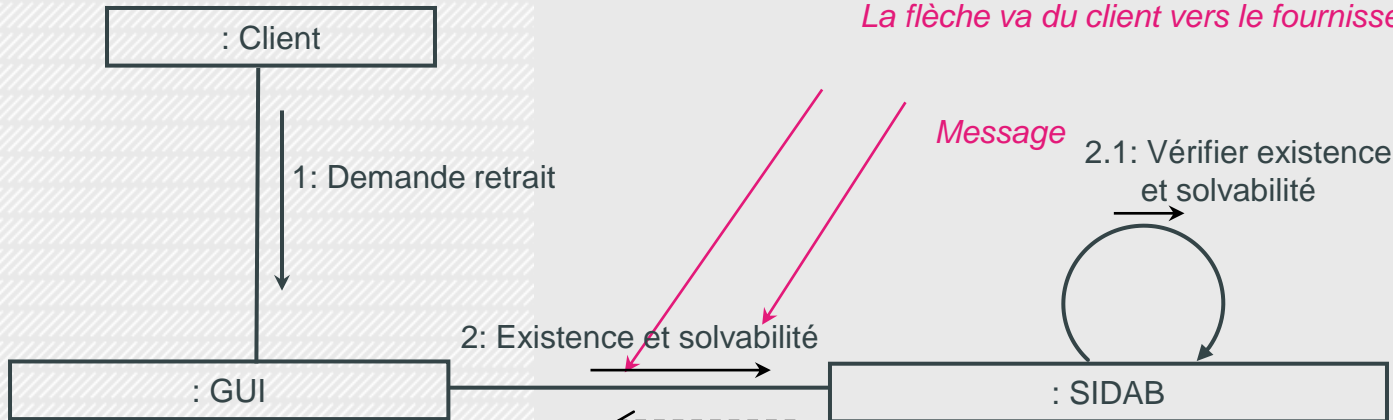
LIENS :

- Indique un chemin de communication entre 2 objets, sur lequel passent les messages



lien d'interaction

- Exemples :



La flèche va du client vers le fournisseur

Message 2.1: Vérifier existence et solvabilité

Représentation des Actions internes à l'objet au moyen de message que l'objet s'envoie à lui même

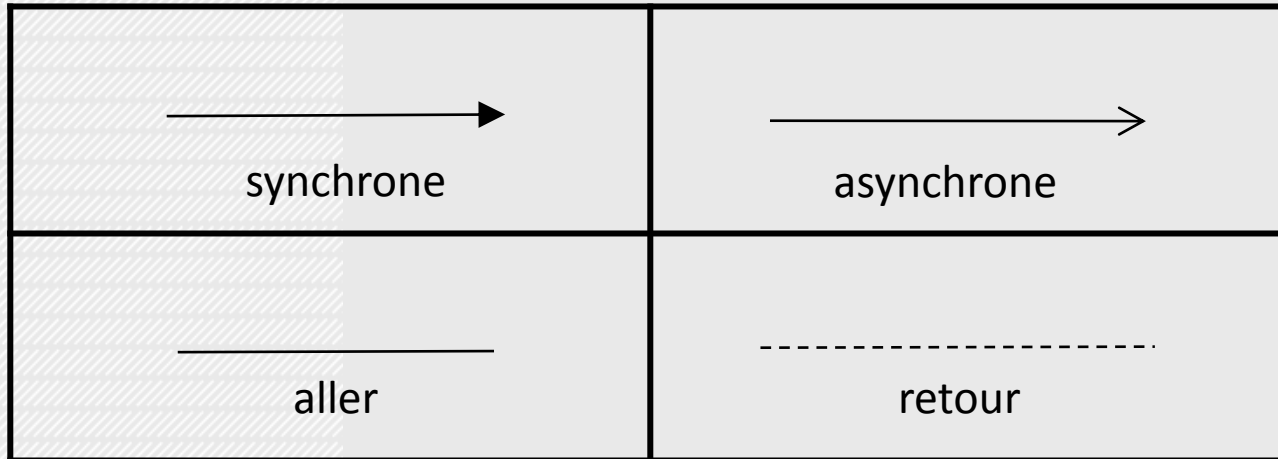
Objet client/appelant

Donnée de retour

Objet fournisseur/appelé

UML : DIAGRAMME DE COMMUNICATION

MESSAGES :



UML : DIAGRAMME DE COMMUNICATION

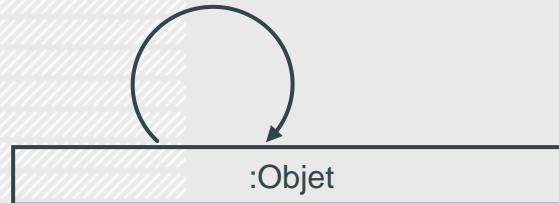
MESSAGES :

- **Synchrone** : Un message est envoyé par à un objet à un autre, et le premier objet **attends** jusqu'à ce que l'action ai finie.
- **Asynchrone** : Un message est envoyé par à un objet à un autre, mais le premier objet **n'attends pas** la fin de l'action.
- **Aller** (plat) : Chaque flèche représente une **progression** d'une étape à une autre dans la séquence. La plupart asynchrone.
- **Retour** : Le **retour** explicite d'un objet à qui le message était envoyé.

UML : DIAGRAMME DE COMMUNICATION

MESSAGES :

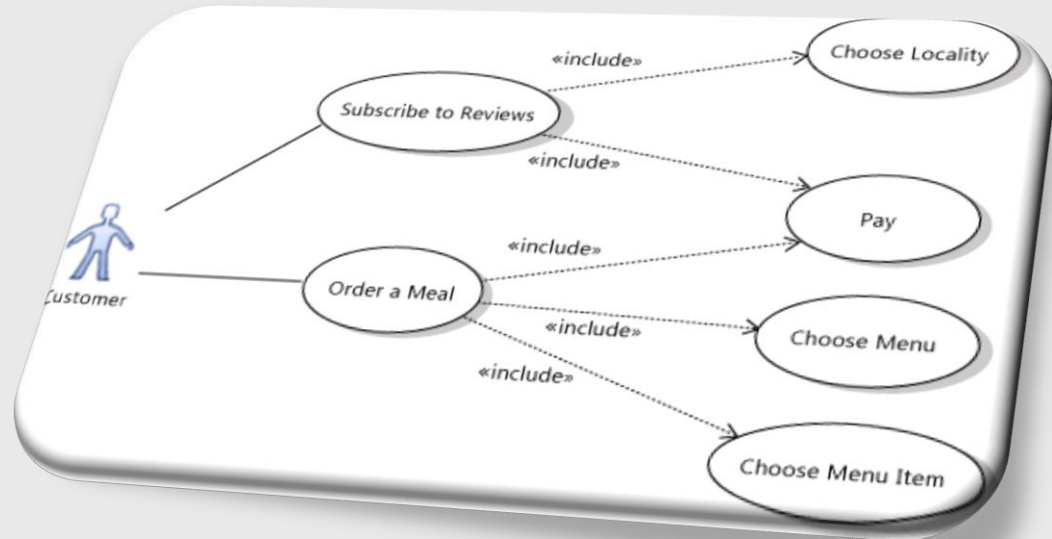
- Messages envoyés **en même temps** :
Utilisation de la notation Chiffre-Lettre
- Messages invoqués **plusieurs fois** :
*Contrainte de boucle * [i=0..9]*
- Messages envoyés à une **condition** :
Ajout d'une expression logique [condition = true]
- Messages envoyés à lui même :



UML : DIAGRAMME DE COMMUNICATION

EXERCICES

UML : DIAGRAMME DE CAS D'UTILISATION



UML : DIAGRAMME DE CAS D'UTILISATION

GENERALITES :


- Le **systeme** existe pour servir ses **utilisateurs**
- Cas d'utilisation = Use cases
- Idée : description du comportement du système du point de vue de son utilisateur (facilite l'expression des besoins)

- Comportement = {Actions} + {Réactions}

UML : DIAGRAMME DE CAS D'UTILISATION

GENERALITES :

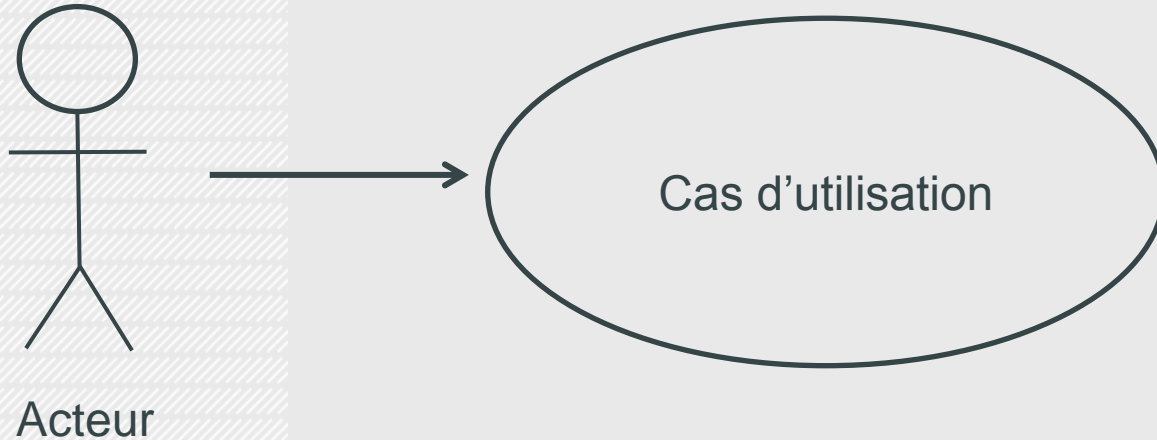
- On part d'un **scénario** (*ex : un client achète un objet et paie sur internet*)
- Mais il peut y avoir des scénarios liés *ex*
 - *échec lors du paiement*
 - *Il s'agit d'un client régulier*
- Mais ces scénarios ont le même **but** : acheter un objet

 Un cas d'utilisation est un ensemble de scénarios liés ensemble par un but commun d'un utilisateur.

- Acteur = entité externe qui agit sur le système

UML : DIAGRAMME DE CAS D'UTILISATION

REPRESENTATION :



UML : DIAGRAMME DE CAS D'UTILISATION

ACTEURS vs UTILISATEURS :

Ne pas confondre **acteur** et **personne** utilisant le système :

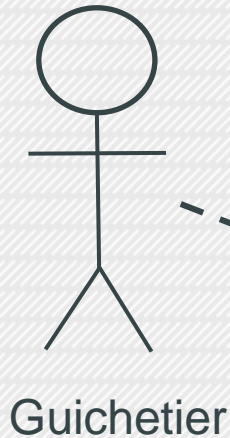
- Une même personne peut jouer plusieurs rôles
- Plusieurs personnes peuvent jouer un même rôle
- Un acteur n'est pas forcément une personne physique.

Types d'acteurs :

- Utilisateurs principaux
- Utilisateurs secondaires
- Périphériques externes
- Systèmes externes

UML : DIAGRAMME DE CAS D'UTILISATION

DEFINITION DES ACTEURS :



Un guichetier est un employé de la banque jouant un rôle d'interface entre le système informatique et les clients qu'il reçoit au comptoir.

Pour chaque acteur :

- choix d'un **identificateur**
- brève **description** (facultatif)

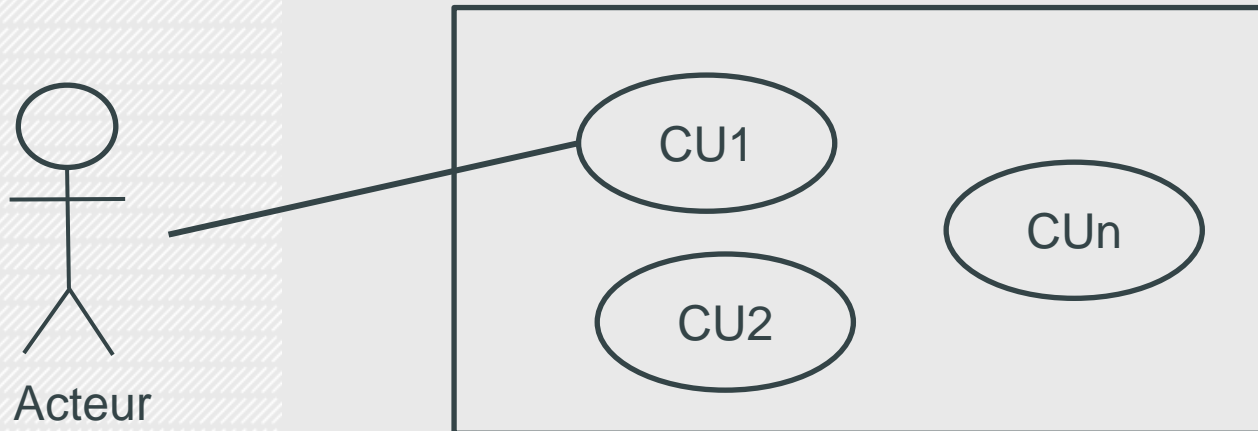
- Acteur **principaux** : utilisent le système
- Acteur **secondaires** : administrent le système

UML : DIAGRAMME DE CAS D'UTILISATION

CAS D'UTILISATION : DEFINITIONS

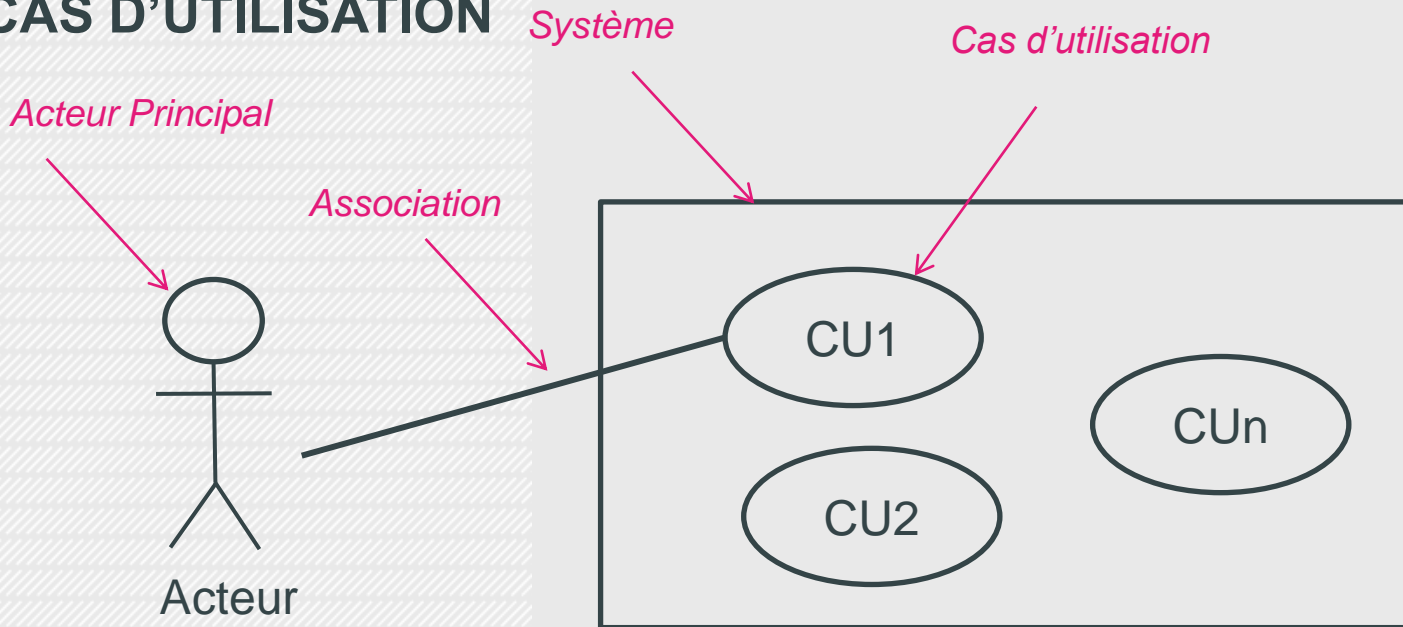
Ensemble des **actions** réalisées par le **système** en réponse à une action d'un **acteur**

Les cas d'utilisation ne doivent pas se chevaucher



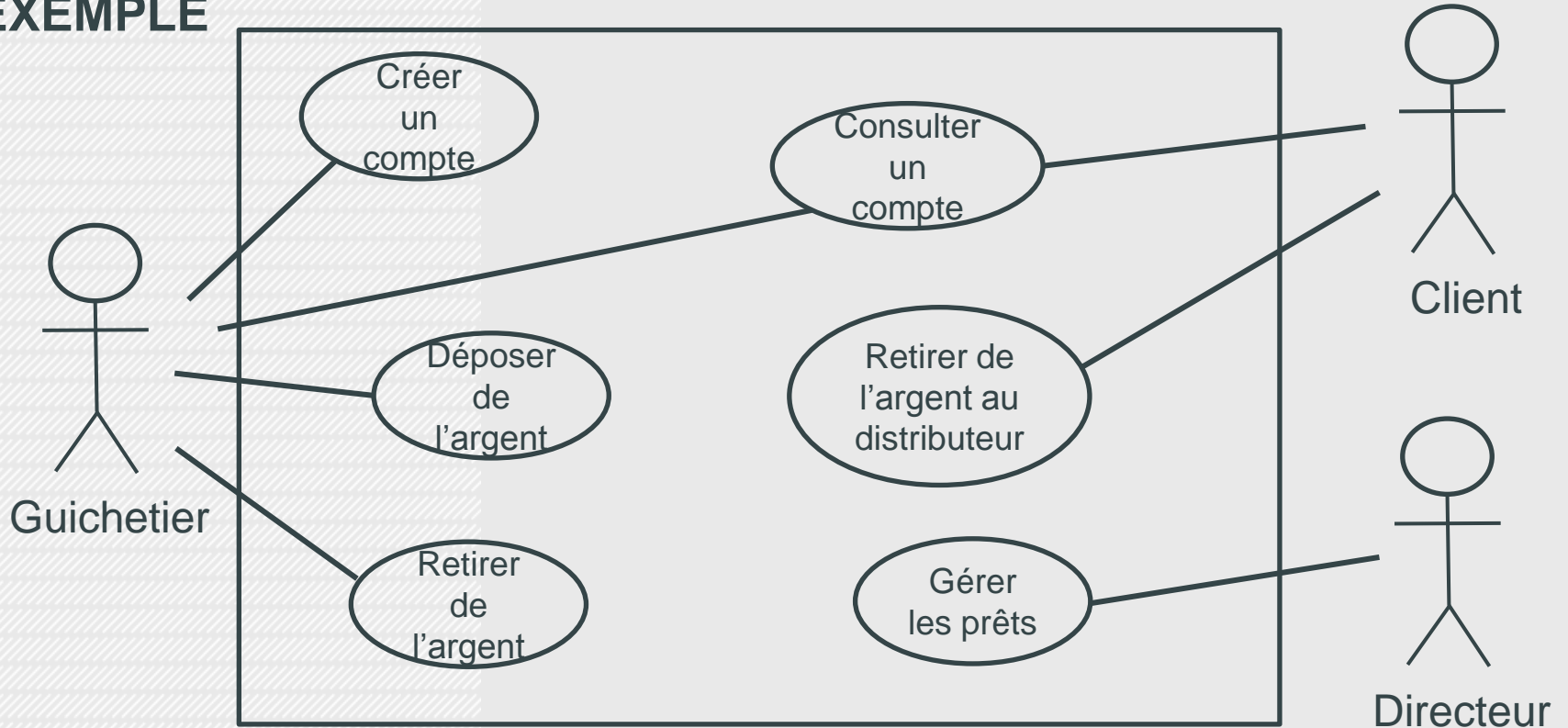
UML : DIAGRAMME DE CAS D'UTILISATION

CAS D'UTILISATION



UML : DIAGRAMME DE CAS D'UTILISATION

EXEMPLE



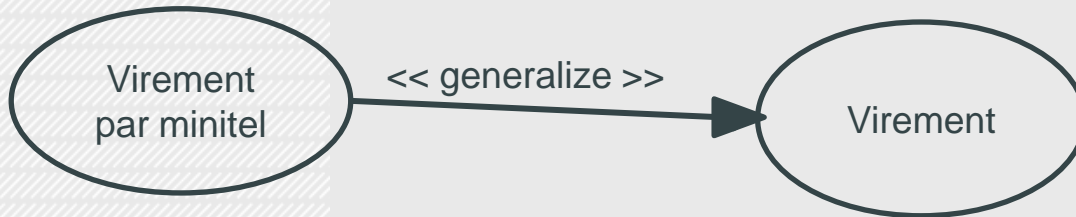
UML : DIAGRAMME DE CAS D'UTILISATION

RELATIONS ENTRE CAS D'UTILISATIONS :

- Généralisation (*generalize*)
- Inclusion (*include*)
- Extension (*extend*)

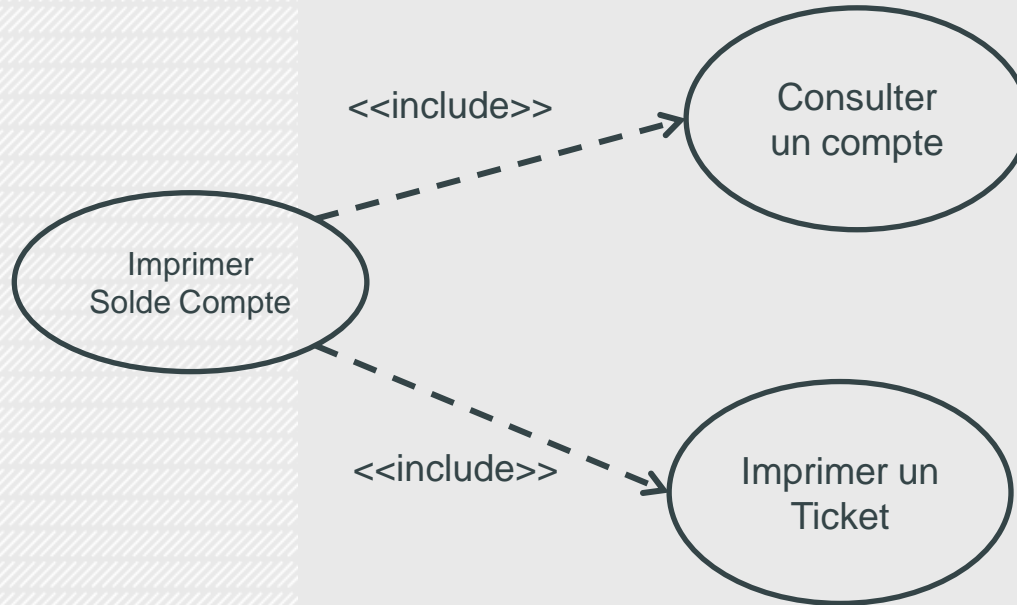
UML : DIAGRAMME DE CAS D'UTILISATION

RELATION GENERALIZE



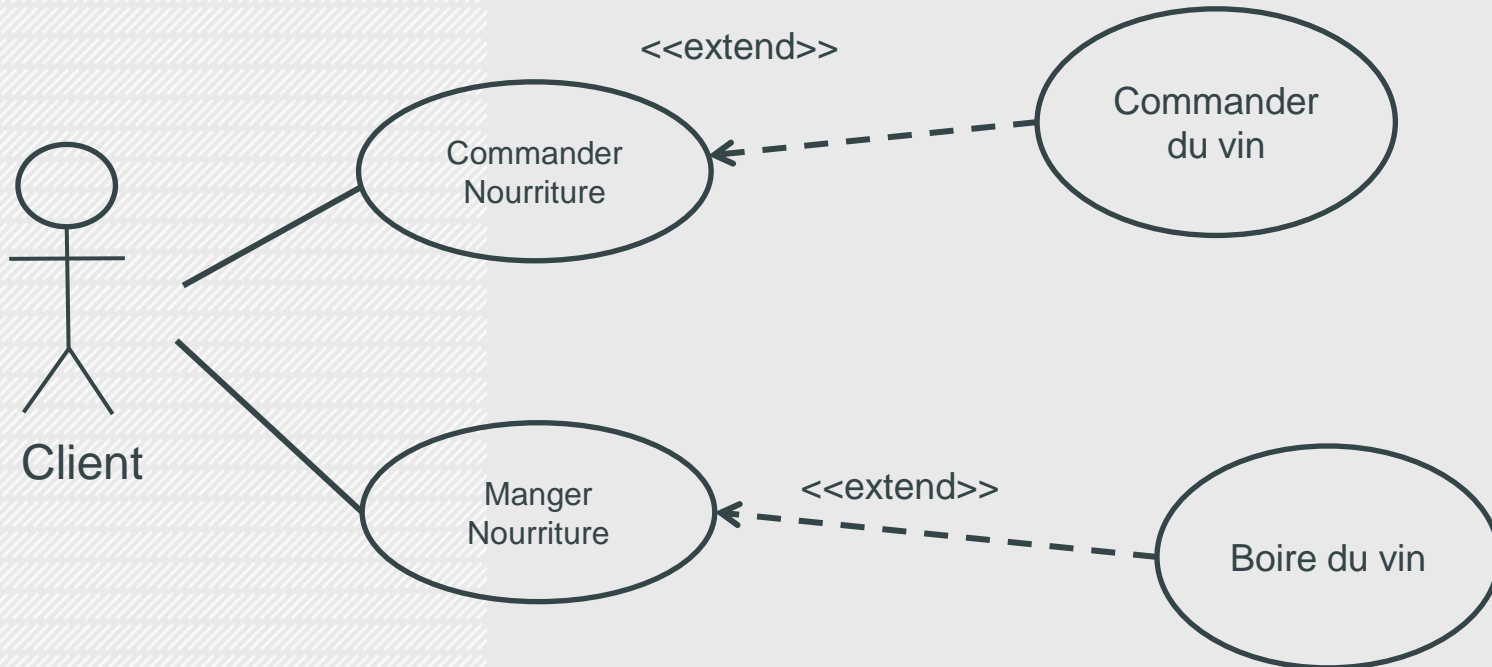
UML : DIAGRAMME DE CAS D'UTILISATION

RELATION INCLUDE :



UML : DIAGRAMME DE CAS D'UTILISATION

RELATION EXTEND :



UML : DIAGRAMME DE CAS D'UTILISATION

EN RESUME :

- Système = ensemble de cas d'utilisation
- Le système possède les cas d'utilisation mais pas les acteurs
- Un cas d'utilisation = ensemble de « chemins d'exécution » possibles
- Un scénario = un chemin particulier d'exécution
- Un scénario = Instance de cas d'utilisation
- Une instance d'acteur crée un scénario

UML : DIAGRAMME DE CAS D'UTILISATION

QUAND L'UTILISER ?

- Outil appréciable pour aider à comprendre les requis fonctionnels d'un système.
- Utile dans les premières phases d'un projet
- Précède les spécifications détaillées

ASTUCES :

- S'aider des flux & des acteurs identifiés dans le diagramme de communication
- Regrouper ces flux identifiés
- Ne pas descendre trop bas dans la description

UML : DIAGRAMME DE CAS D'UTILISATION

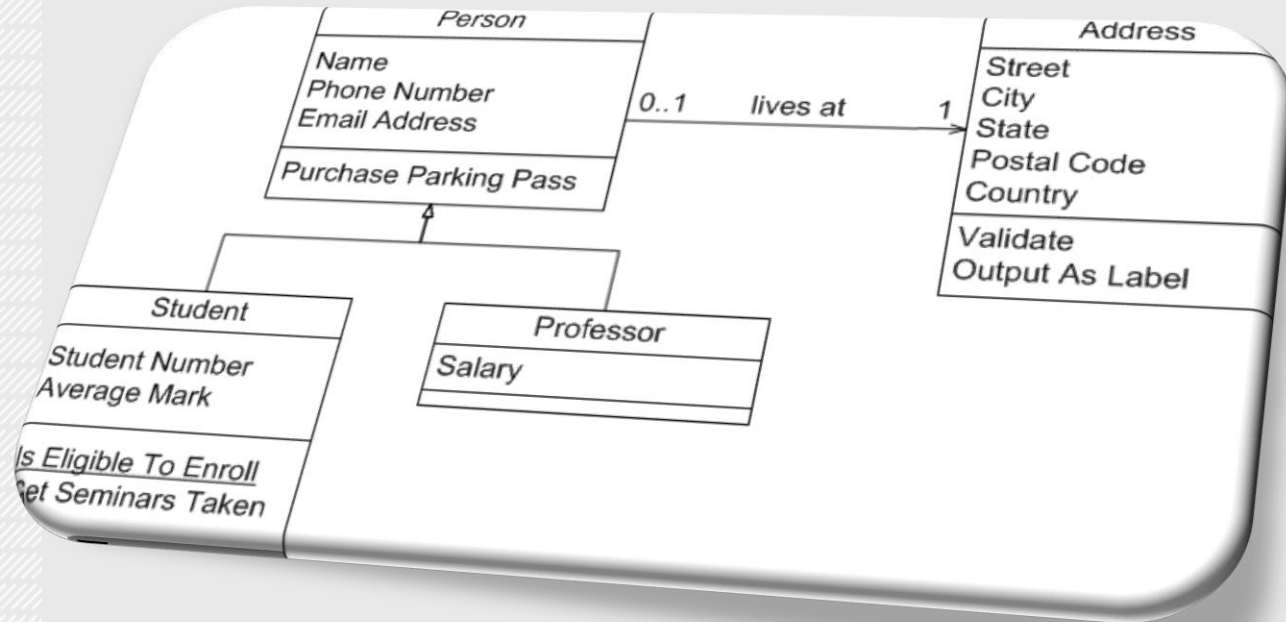
ASTUCES :

- Impossible de décrire tous les scénarios
 - Sélection des scénarios optimaux : interaction la plus fréquente
 - Sélection des scénarios dérivés : certaines alternatives intéressantes
- Commencer par les diagrammes CU qui présentent :
 - Le plus d'enjeux / risque
 - Les plus importants

UML : DIAGRAMME DE CAS D'UTILISATION

EXERCICES

UML : DIAGRAMME DE CLASSES



UML : DIAGRAMME DE CLASSES

APPROCHE OBJET:

- Représentation d'un **concept** ou d'une **chose**, ayant des **limites** et un **sens** dans un **contexte** donné.
- Physique ou Abstraite
- Caractéristiques :
 - Informations
 - Identifiant
 - Comportement
 - Messages
 - Encapsulation

UML : DIAGRAMME DE CLASSES

CARACTERISTIQUES :

- **Informations** : attributs décrivant l'objet
- **Identifiant** : permet d'individualiser les objets entre eux
- **Comportement** : implémenter sous forme de « méthodes »
- **Messages** : moyen de communication entre objets
- **Encapsulation** : gestion des accès

UML : DIAGRAMME DE CLASSES

DEFINITION :

- **Classe :**

modélisation de l'ensemble des objets ayant :

- attributs similaires
- comportement en commun
- relations communes avec d'autres objets

'Titre d'objet': 'Titre de classe'

« Objet »

**Les fleurs
du mal**

<< objet >>

**Œuvre
littéraire**

<< classe >>

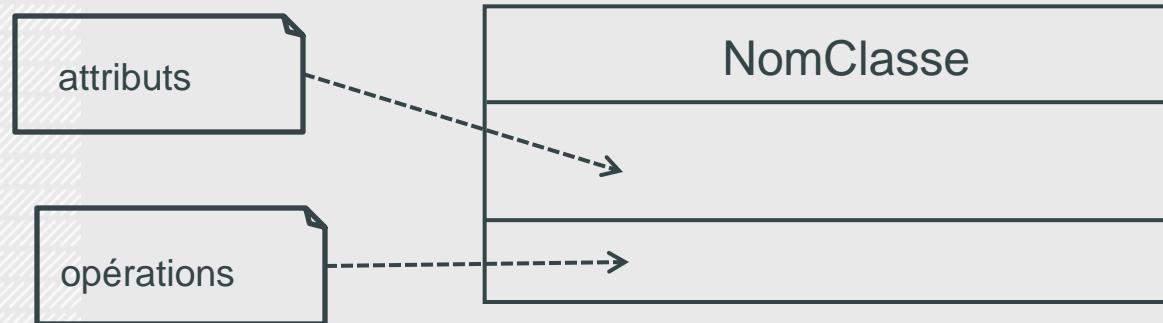
UML : DIAGRAMME DE CLASSES

NOTATION :

- Tout nom de classe commence par une majuscule

NomAttribut : type = ValeurInitiale

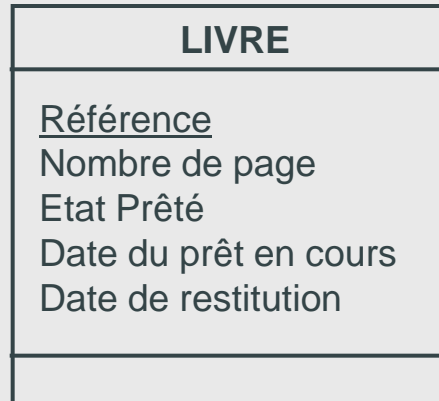
NomOpération(ListeArguments) : typeRetour



UML : DIAGRAMME DE CLASSES

ATTRIBUT :

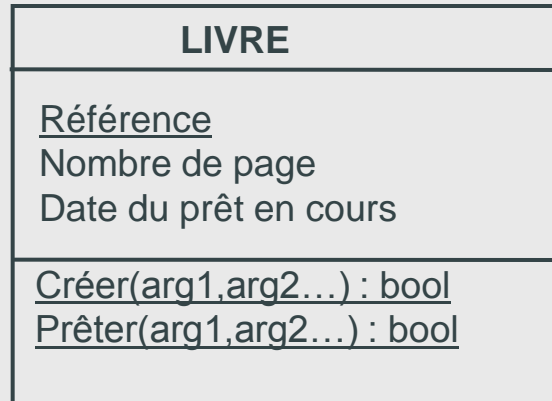
- Donnée gérée par la classe (donc pour toutes les instances)
- Valeur unique par instance (mais la même peut être dans plusieurs occurrences)
- Identifiant est un attribut :
 - particulier, autant de valeurs que d'instances
 - valeurs déterminent les instances de la classe



UML : DIAGRAMME DE CLASSES

OPERATIONS - METHODES:

- **Opération** : effectuée par l'objet
- **Méthode** : effectuée par la classe
- **Signature** (notation) : *NomOperation(arg1,arg2) : TypeOutput*
- **Comportement** : effets observables des opérations



UML : DIAGRAMME DE CLASSES

OPERATIONS :

- **Peut :**



- Accéder à ses propres attributs
- Invoquer une autre opération de son propre objet
- Invoquer une autre opération d'un autre objet (même classe ou autre classe)

- **Ne peut pas:**

- Accéder directement aux attributs d'un autre objet

UML : DIAGRAMME DE CLASSES

LIENS & ASSOCIATIONS:

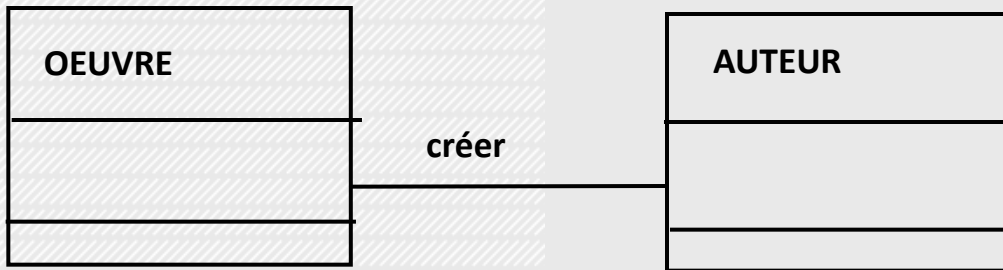
- **Lien :**
 - connexion physique ou conceptuelle entre instances de classes
- **Association:**
 - Instance du lien (ex : rédiger, créer)
- **Rôle :**
 - Spécifie la contribution (ex : être rédigée par)
- Notation :
 - Nom : verbe 
 - Navigation : via flèche 

Par défaut : 2 sens

UML : DIAGRAMME DE CLASSES

TYPES ASSOCIATIONS:

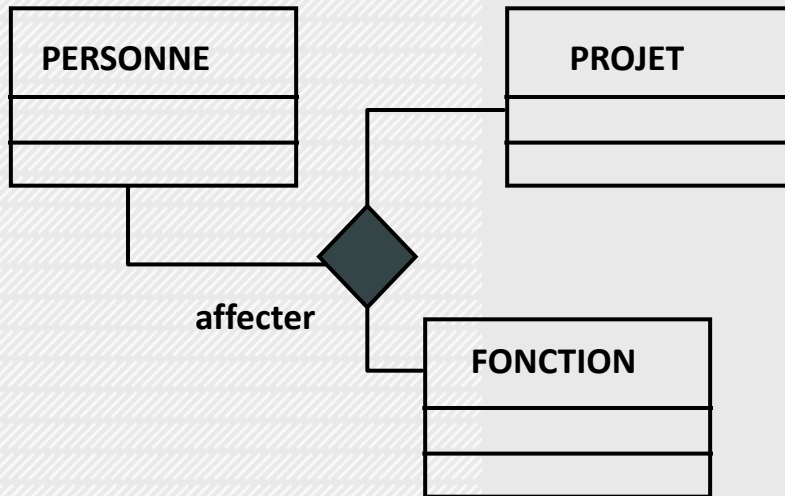
- **Association binaire** : associe 2 classes



UML : DIAGRAMME DE CLASSES

TYPES ASSOCIATIONS:

- **Association n-aire** : associe plus de 2 classes

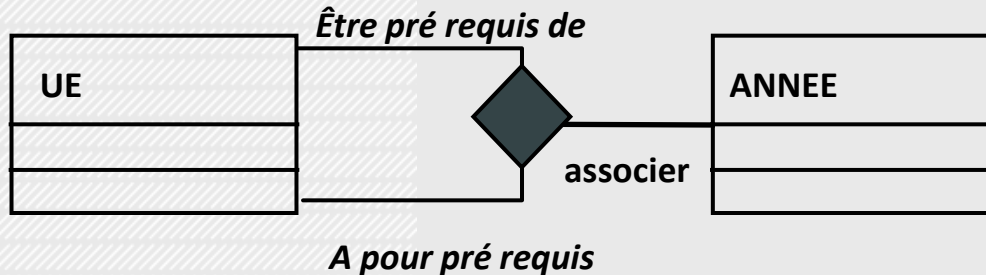


UML : DIAGRAMME DE CLASSES

TYPES ASSOCIATIONS:

- **Association réflexive** : associe une classe à elle même

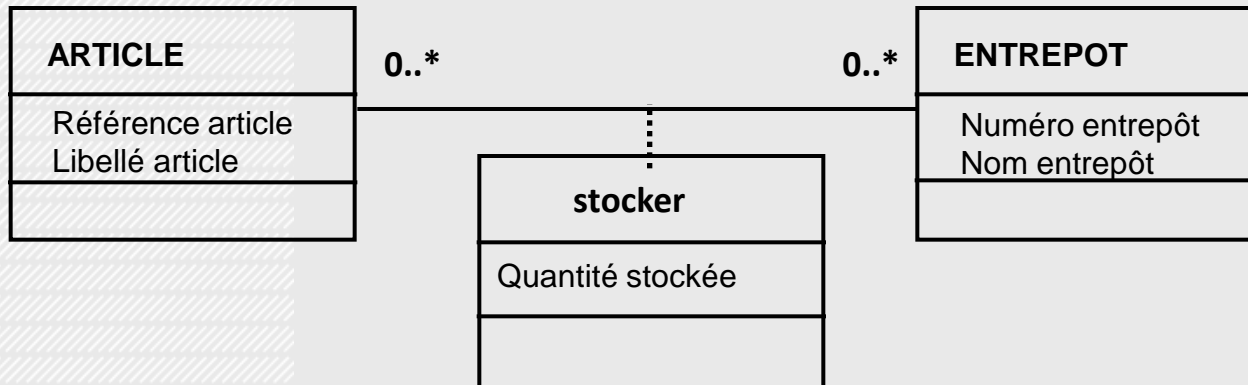
Dans ce cas, indispensable de nommer les associations (rôles)



UML : DIAGRAMME DE CLASSES

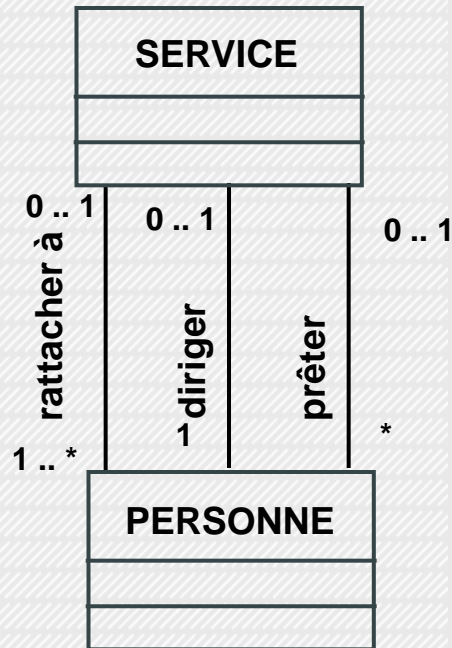
CLASSE D'ASSOCIATION:

- Définition : classe liée à une association
- Classe qui a les caractéristiques d'une association ou Association qui a les caractéristiques d'une classe.
- Nécessaire quand attributs ou opérations liées à l'ensemble des classes de l'association
- Utile pour simplifier



UML : DIAGRAMME DE CLASSES

MULTIPLICITES (cardinalités):



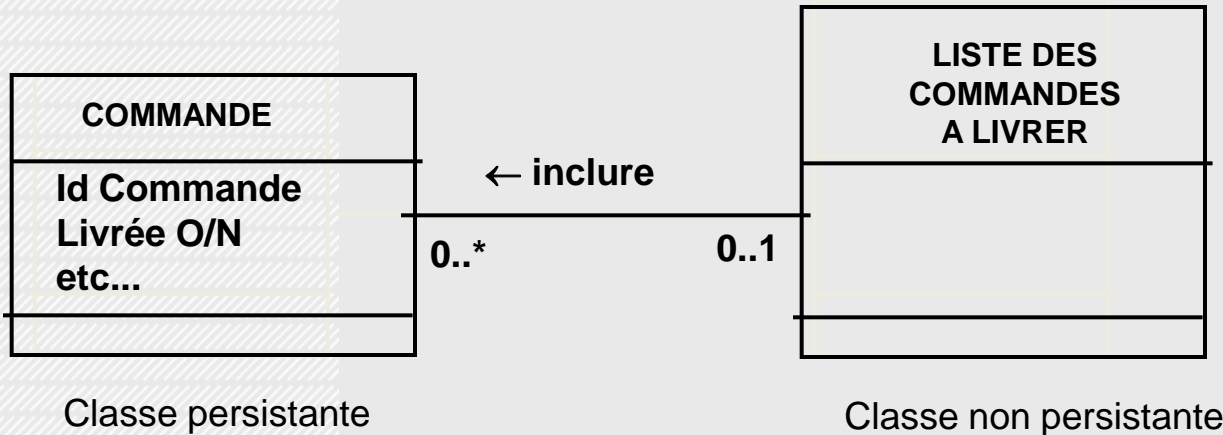
1	Un et un seul
0..1	Zéro ou Un
M..N	De M à N (entiers naturels)
0..*	De Zéro à plusieurs
*	
1..*	De Un à plusieurs
N	Exactement N (entier naturel)

UML : DIAGRAMME DE CLASSES

CLASSES PERSISTANTES :

- **DEFINITION :**

Une classe est persistante quand elle perdure dans le temps



UML : DIAGRAMME DE CLASSES

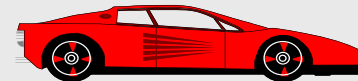
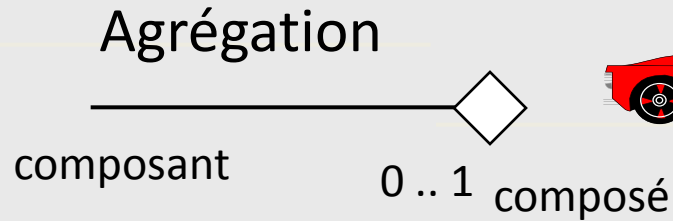
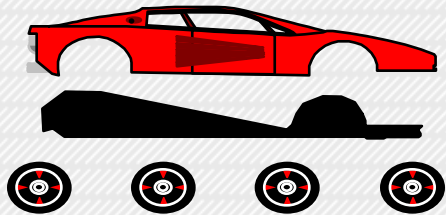
CONCEPTS AVANCES...

UML : DIAGRAMME DE CLASSES

AGREGATION

- **DEFINITION :**

Forme d'association qui spécifie la relation entre un **ensemble** ou « **composé** » et sa **partie** ou « **composant** ».

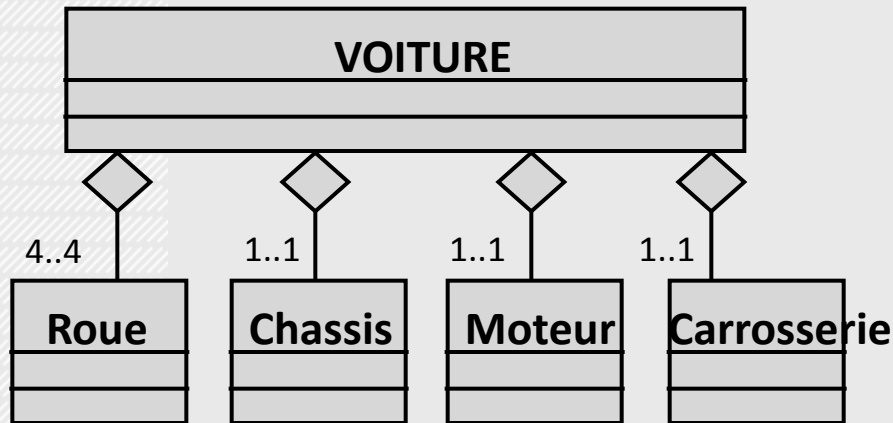


Contrainte D'intégrité Fonctionnelle **FAIBLE**

UML : DIAGRAMME DE CLASSES

AGREGATION

- La partie (composant) peut exister sans l'agrégat (composé)
- Si le composé (agrégat) disparaît, le composant (partie) peut continuer d'exister

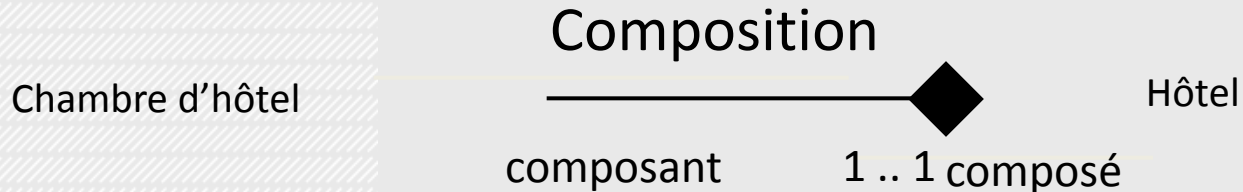


UML : DIAGRAMME DE CLASSES

COMPOSITION

- **DEFINITION :**

Forme d'agrégation quand l'**ensemble** ou « **composé** » est responsable de la création et de la destruction de ses parties. Le Composant n'existe que dans l'association au composé.

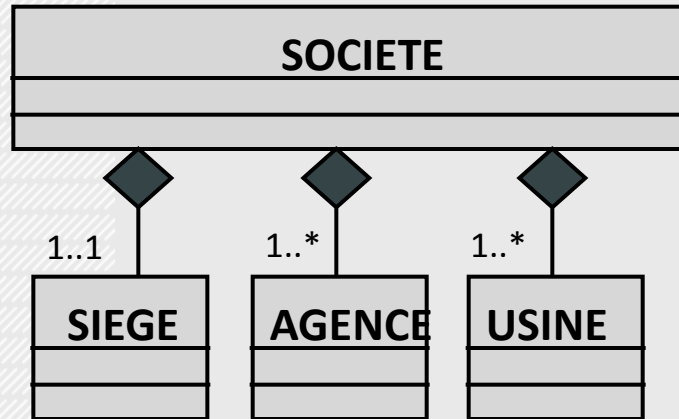


Contrainte D'intégrité Fonctionnelle **FORTE**

UML : DIAGRAMME DE CLASSES

COMPOSITION

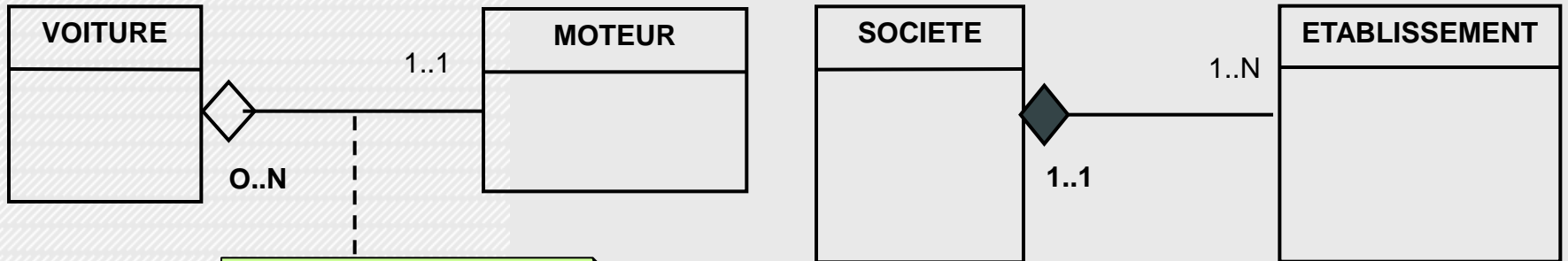
- La partie (composant) n'existe pas sans l'agrégat (composé)
- Si le composé (agrégat) disparaît le composant (partie) disparaît aussi



UML : DIAGRAMME DE CLASSES

MULTIPLICITES DES AGREGATIONS ET COMPOSITIONS:

- Une agrégation peut avoir N maxi du côté de la classe agrégat
 - car l'élément agrégeant peut servir (en même temps ou successivement) à plusieurs agrégats
- Une composition ne peut avoir que 1..1 du côté de la classe composée
 - car le composant, ne pouvant exister indépendamment du composé, est obligatoirement lié à un seul

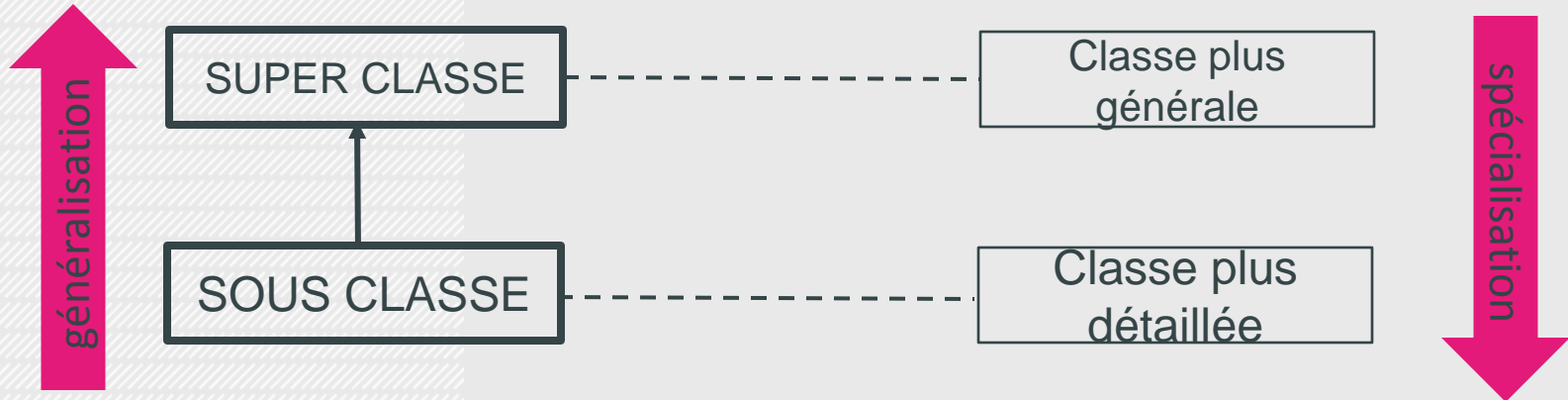


Un moteur peut appartenir à plusieurs voitures (mais à une seule voiture en même temps)

UML : DIAGRAMME DE CLASSES

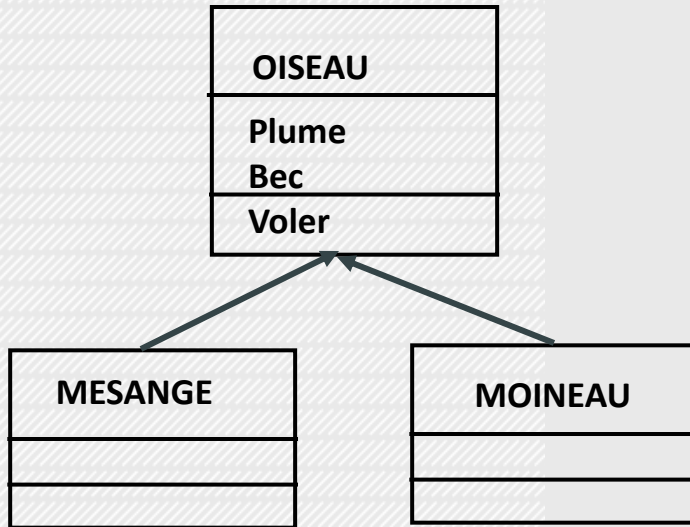
LA CLASSIFICATION :

- 2 Approches :
 - La généralisation
 - La spécialisation



UML : DIAGRAMME DE CLASSES

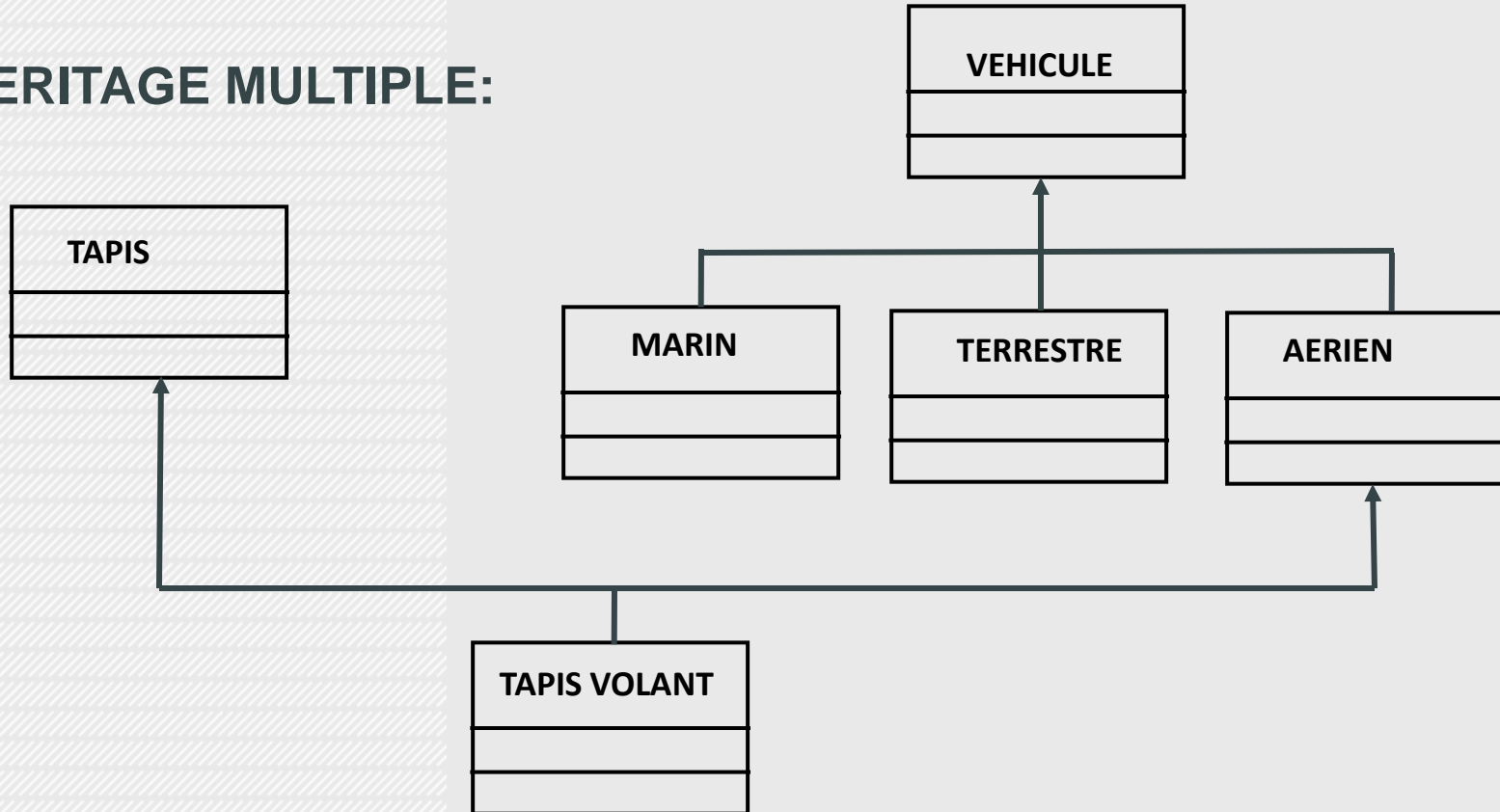
GENERALISATION:



- Factoriser les éléments communs (attributs, opérations, contraintes) d'un ensemble de classes dans une classe plus générale appelée Super-Classe (ou classe mère).
- Tous les attributs qui se trouvent dans la Super-Classe sont vrais dans toutes les sous-classes.

UML : DIAGRAMME DE CLASSES

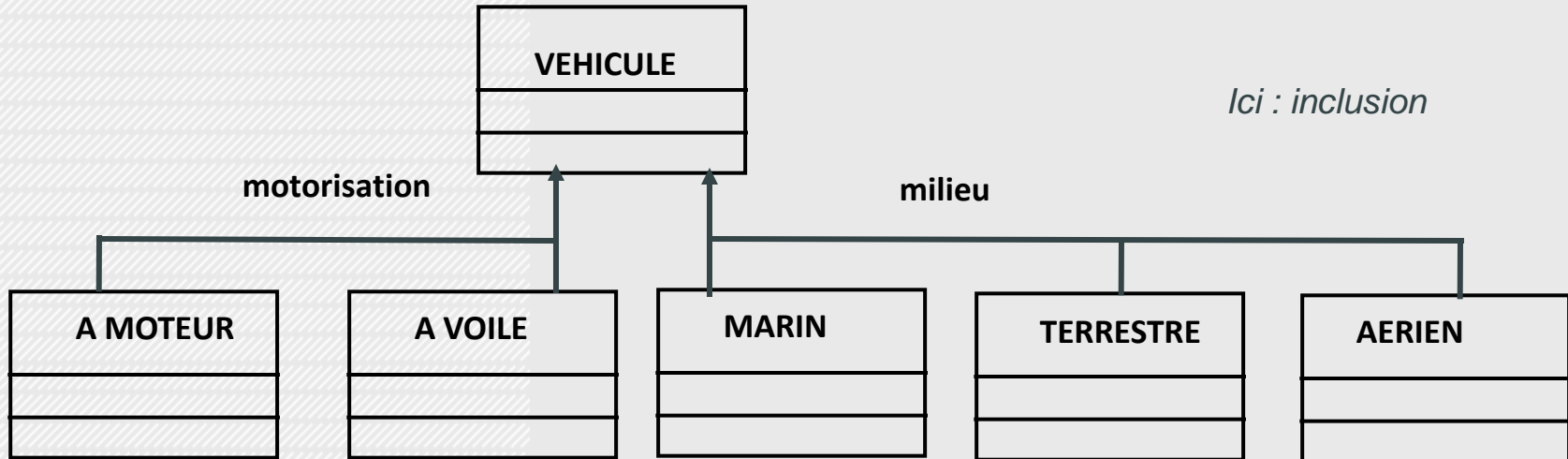
HERITAGE MULTIPLE:



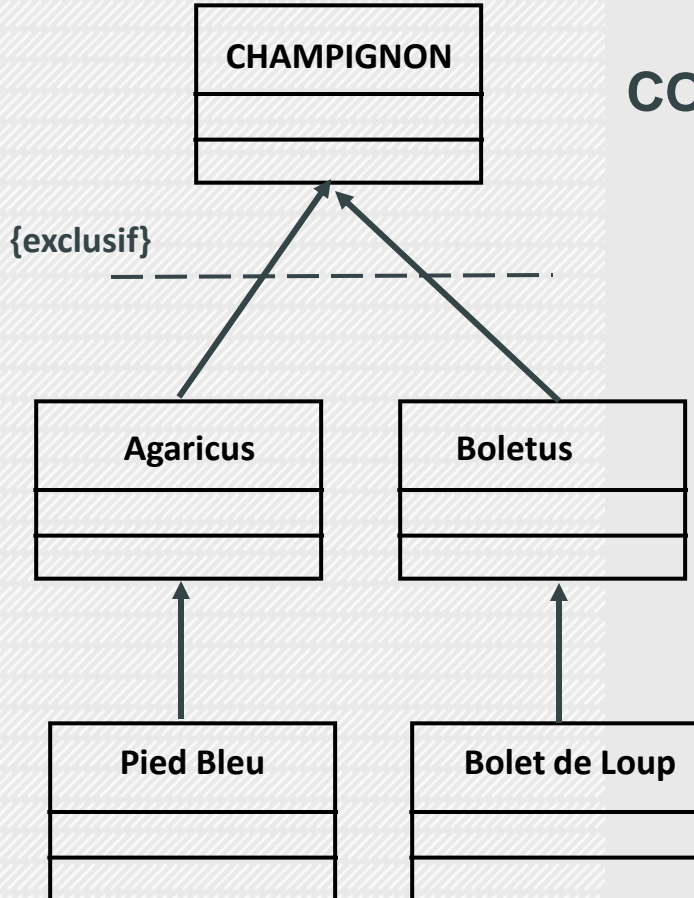
UML : DIAGRAMME DE CLASSES

CONTRAINTES DE GENERALISATION:

- Une classe peut être spécialisée selon plusieurs critères.
- Certaines contraintes peuvent être posées sur les relations de généralisations.
- Par défaut, la généralisation symbolise une **exclusion**.



UML : DIAGRAMME DE CLASSES

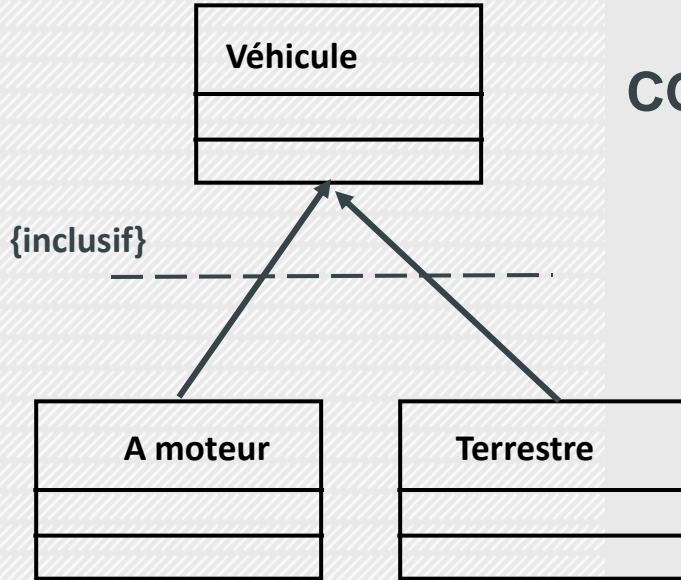


CONTRAINTES DE GENERALISATION :

• {DISJOINT} = {EXCLUSIF}

- Un champignon peut être :
Soit un agaricus
Soit un Boletus
Ou Aucun des deux.

UML : DIAGRAMME DE CLASSES



CONTRAINTES DE GENERALISATION :

- {CHEVAUCHEMENT} = {INCLUSIF}
- Un Véhicule peut être :
 - A moteur
 - Et Terrestre

UML : DIAGRAMME DE CLASSES

CONTRAINTES DE GENERALISATION :

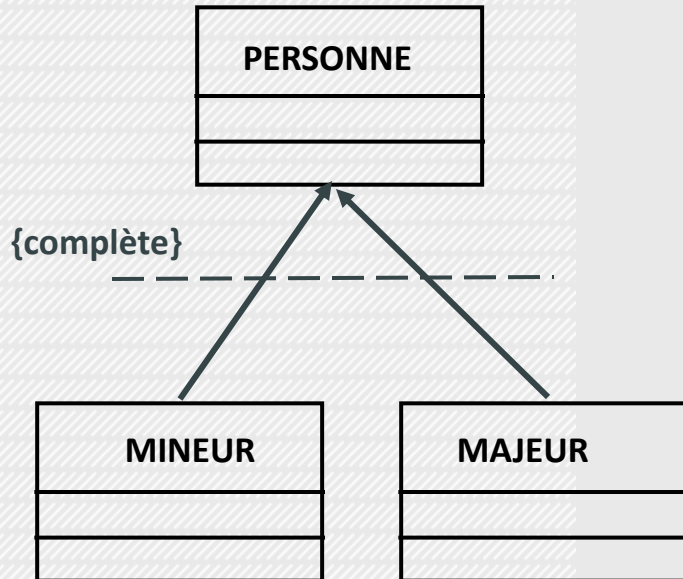
- { COMPLETE }

- Une personne est :

- Soit Mineure

- Soit majeure

- Mais d'un aucun autre statut

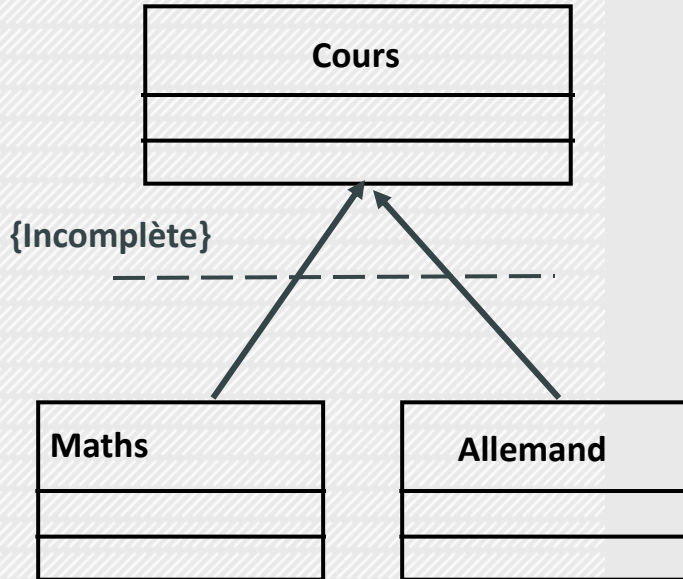


UML : DIAGRAMME DE CLASSES

CONTRAINTES DE GENERALISATION:

• { INCOMPLETE }

- Un Cours peut-être :
Soit de maths
Soit d'allemand
Ou d'un autre non représenté.

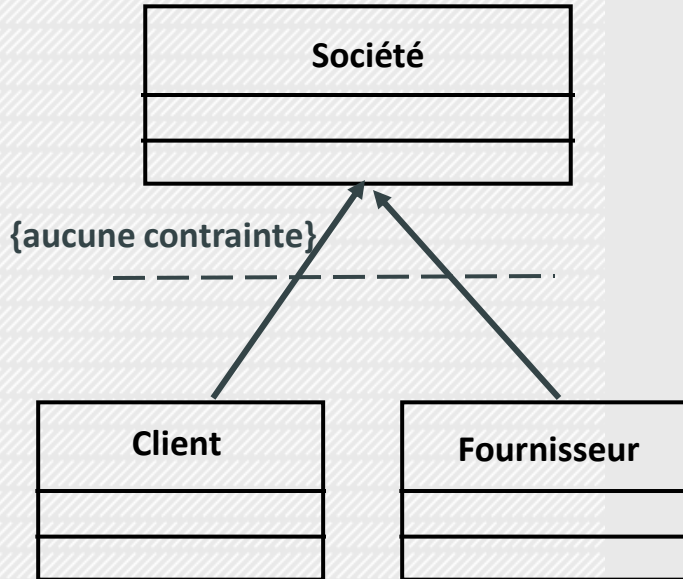


UML : DIAGRAMME DE CLASSES

CONTRAINTES D'ASSOCIATIONS:

• { INCOMPLET RECOUVREMENT } (= { AUCUNE CONTRAINTE })

- Une Société peut être :
Client
Fournisseur
Ou autre chose...



UML : DIAGRAMME DE CLASSES

CONTRAINTES D'ASSOCIATIONS:

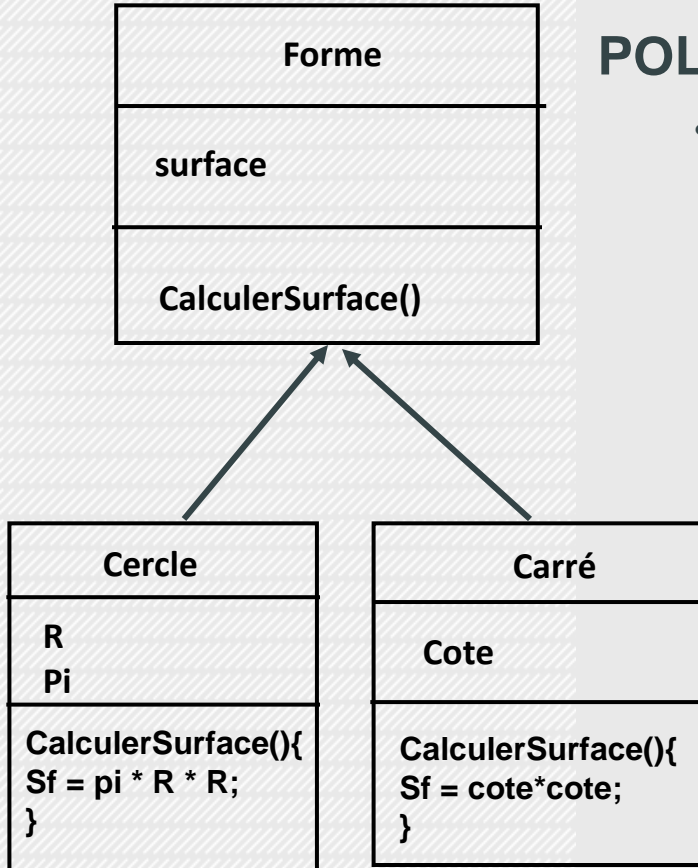
- On peut combiner ces contraintes :
- exclusif + incomplète : Exclusion
- exclusif + complète : Partition
- inclusif + complète : Totalité
- inclusif + incomplète : Aucune contrainte

UML : DIAGRAMME DE CLASSES

POLYMORPHISME

- Une opération est polymorphe lorsque :

- Elle est appelée sous forme générique au niveau de la Super-Classe
- Elle est spécifiée au niveau de chaque Sous-Classe



UML : DIAGRAMME DE CLASSES

CONSEILS Généralisation - spécialisation:

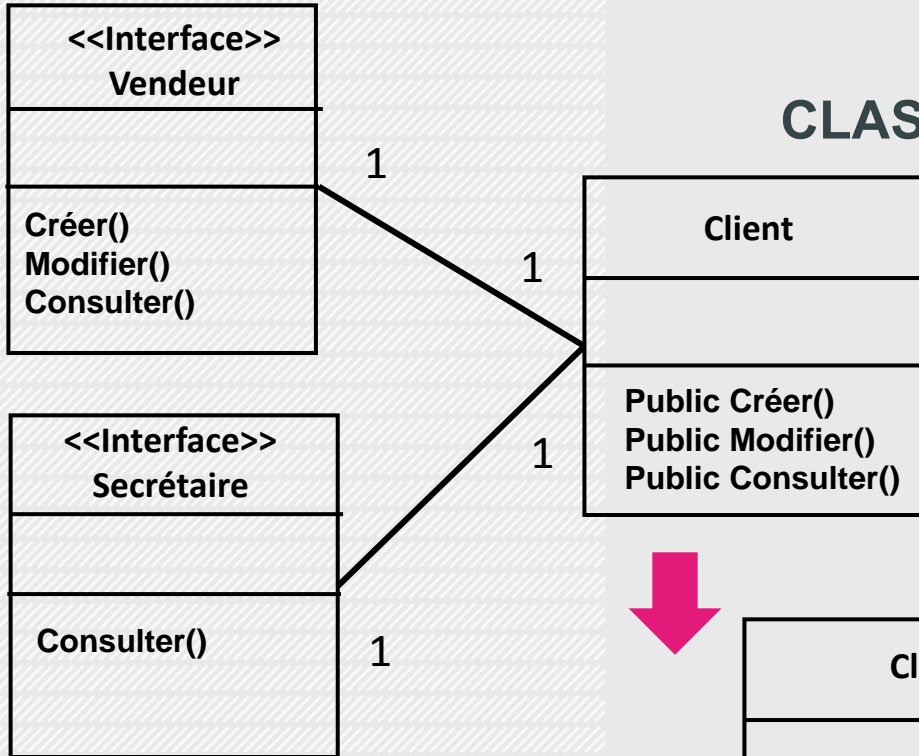
- Ne spécialiser que si cela apporte un gain.
- Ne spécialiser que sur un seul niveau
- Par défaut, la généralisation symbolise une décomposition exclusive.

UML : DIAGRAMME DE CLASSES

CLASSES ABSTRAITES :

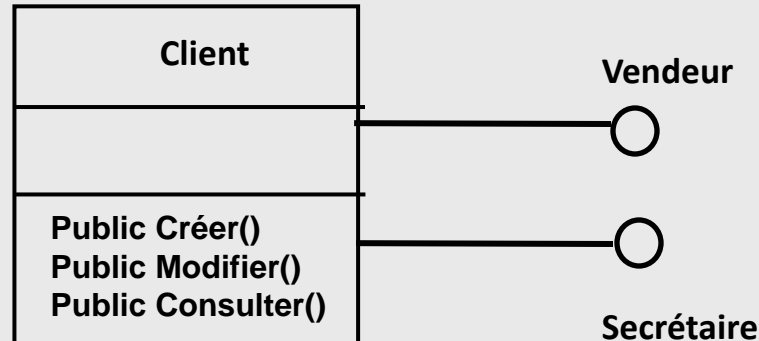
- Ne peut être instanciée
- On instancie une sous-classe.
- Indiquée en *italique* de manière générale.

UML : DIAGRAMME DE CLASSES



CLASSES D'INTERFACE:

- Décrit le comportement externe visible d'une classe, d'un objet ou d'une autre entité.



UML : DIAGRAMME DE CLASSES

EXERCICES

UML : FORMES NORMALES

PREMIERE FORME NORMALE (1FN) :

- Chacun des attribut contient une valeur atomique

ID	Titre	Auteur	Genre
1	CESAR	PAGNOL	Roman
2	Les CHOUANS	BALZAC	Roman

UML : FORMES NORMALES

DEUXIEME FORME NORMALE (2FN):

- Première forme normale
- ET : les attributs n'appartenant pas à la clef primaire ne dépendent pas fonctionnellement d'une partie de la clef.
- Soit : Operation (**N° Compte**, **CodeOpe**, **DateOpe**, Nom, Prenom ,LibelOpe, Somme)
- Devient :
 - Compte (**N°Compte**, Nom, Prénom)
 - Libelle (**CodeOpe**, LibelOpe)
 - Operation (**N°Compte**, **DateOpe**, **CodeOpe**, Somme)

UML : FORMES NORMALES

TROISIEME FORME NORMALE (3FN):

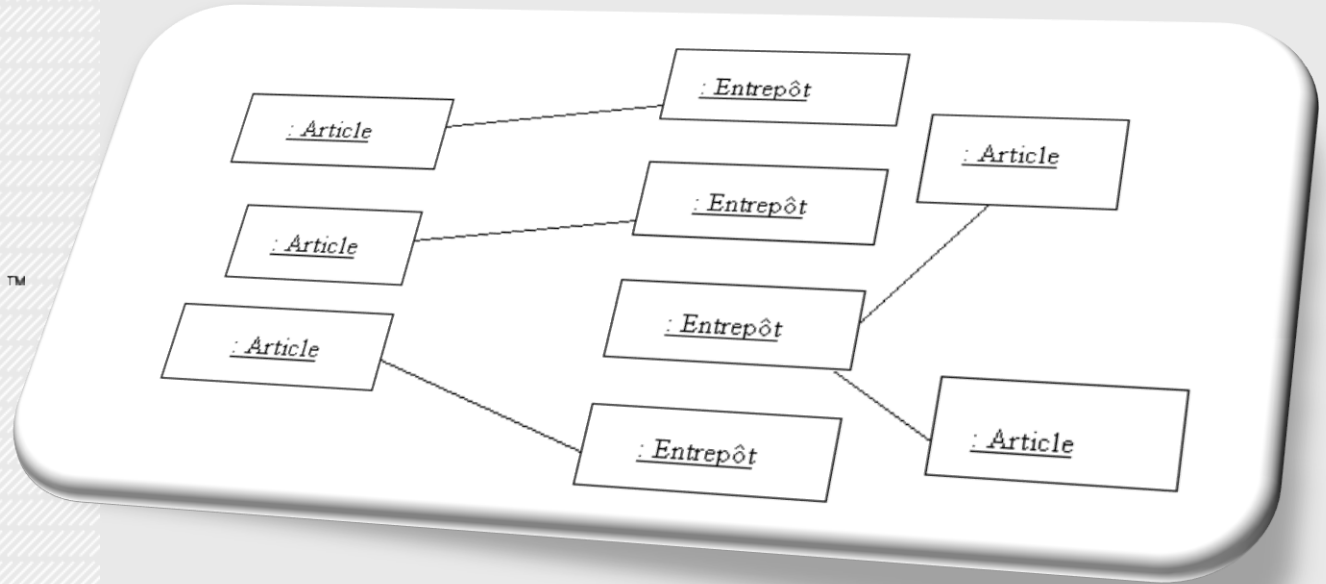
- Deuxième forme normale
- ET : tout attribut non clef ne dépend pas fonctionnellement d'un autre attribut.
- Soit : Adherent(**Codeadh**, NomAdh, AdresseAdh, TypeAdh, CotisTyp, DatPaiCot, NbSaladh)
- Devient :
 - Type(**TypeAdh**, CotisAdh)
 - Adherent(**Codeadh**, NomAdh, AdresseAdh, **TypeAdh**, DatPaiCot, NbSaladh)
- Permet de regrouper les attributs dépendants fonctionnellement pour éviter la redondances de données et de faciliter la maintenance.

UML : FORMES NORMALES

RESUME :

- 1FN :
 - tout attribut contient une valeur atomique
- 2FN :
 - tous les attributs non-clefs sont dépendants fonctionnellement de la clef primaire
- 3FN :
 - tout attribut n'appartenant pas à une clef ne dépend pas d'un attribut non clef

UML : DIAGRAMME D'OBJETS



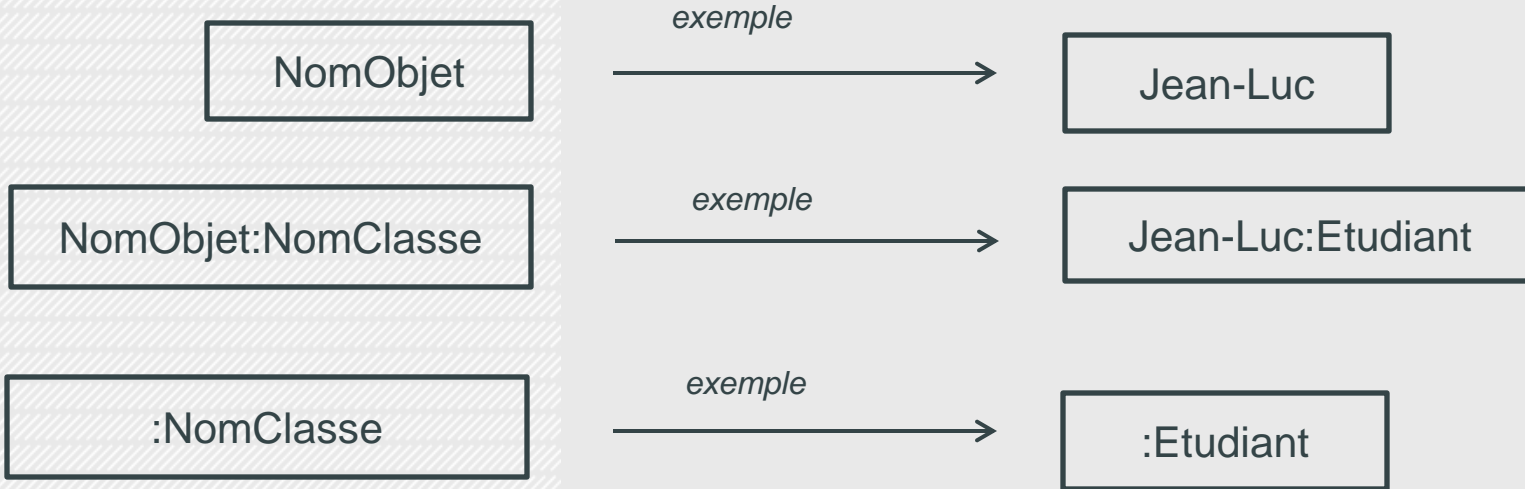
UML : DIAGRAMME D'OBJETS

DEFINITION:

- OBJET : Sens précis dans le contexte du problème étudié.
- Diagramme d'objets montre les instances de classes avec leurs liens.
- Utile pour :
 - Vérifier le diagramme de classes
 - Bien identifier les multiplicités (associations n-aires)
 - Rendre le diagramme de classes plus accessible

UML : DIAGRAMME D'OBJETS

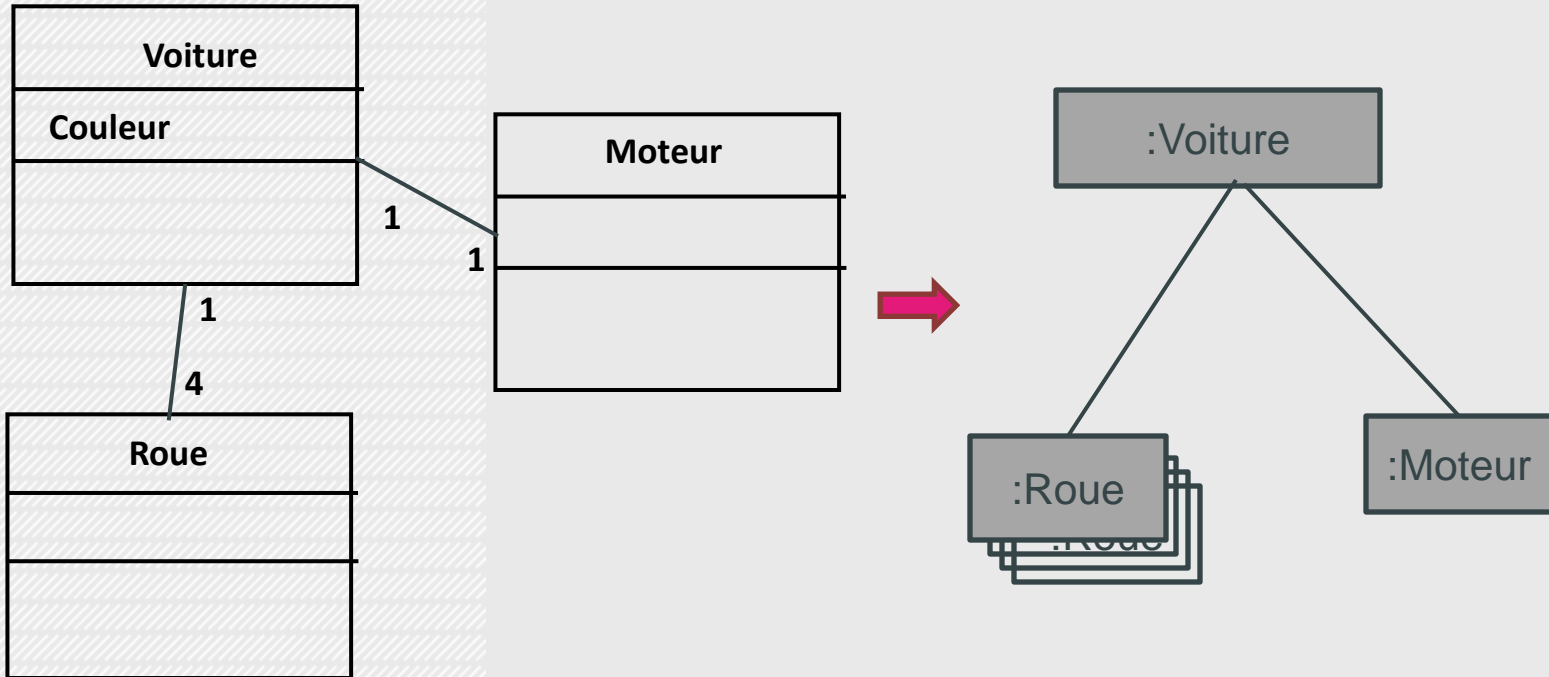
REPRESENTATION:



UML : DIAGRAMME D'OBJETS

REPRESENTATION:

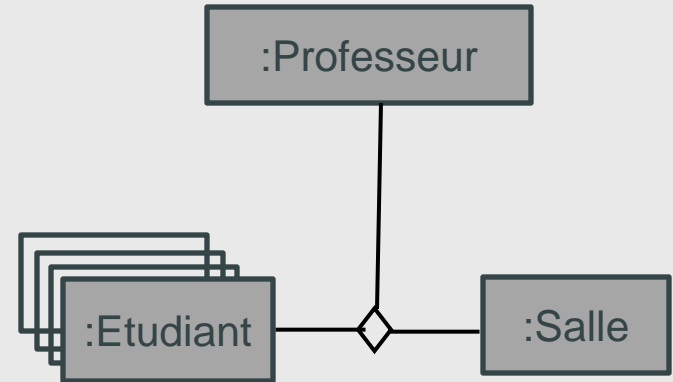
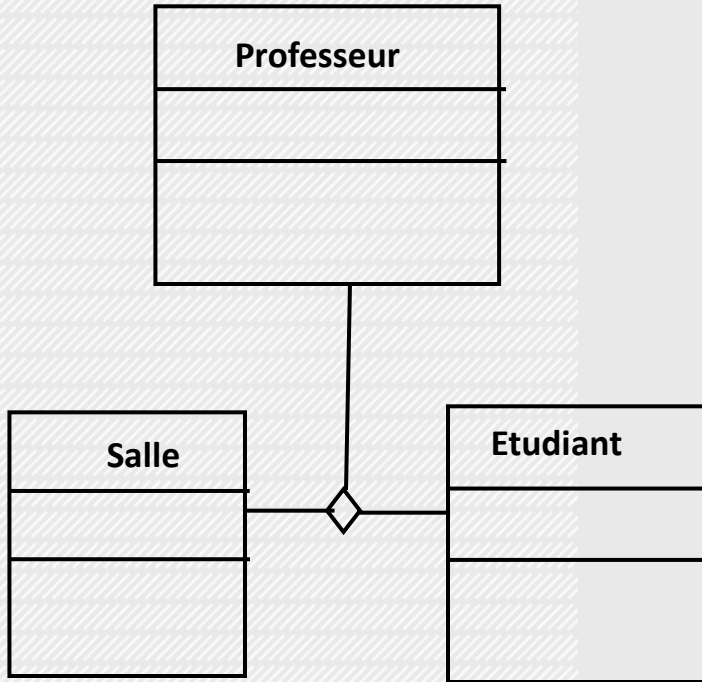
- Attributs et cardinalités optionnels



UML : DIAGRAMME D'OBJETS

REPRESENTATION:

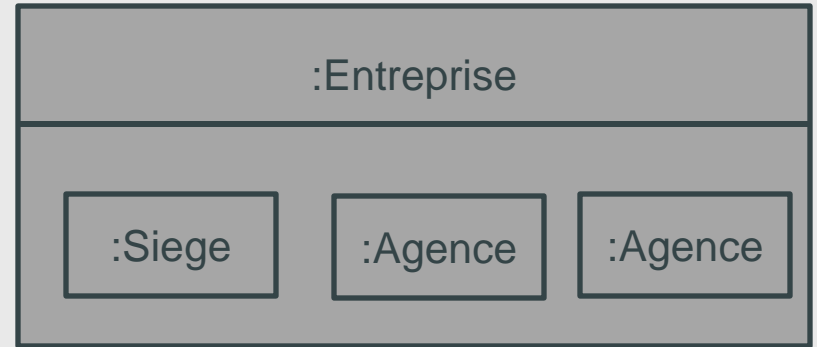
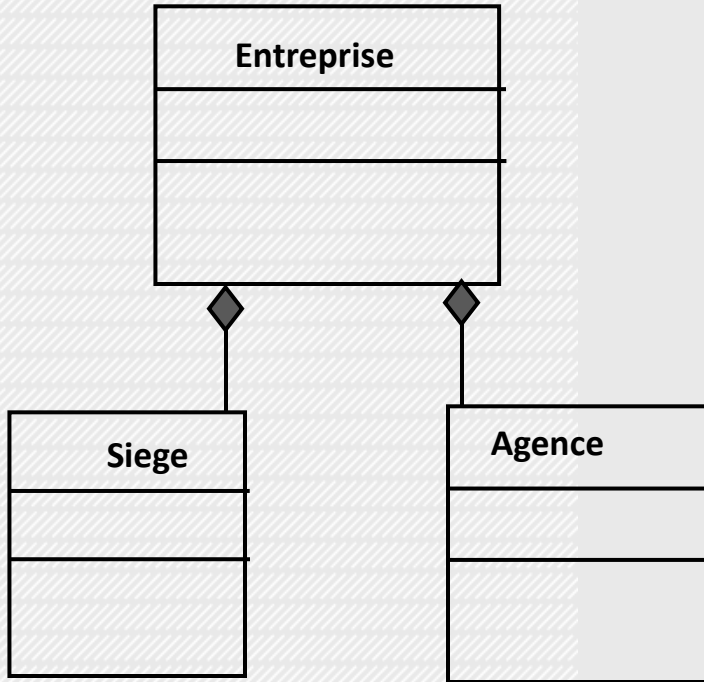
- Lien d'arité supérieure à 2 et multiplicité



UML : DIAGRAMME D'OBJETS

REPRESENTATION:

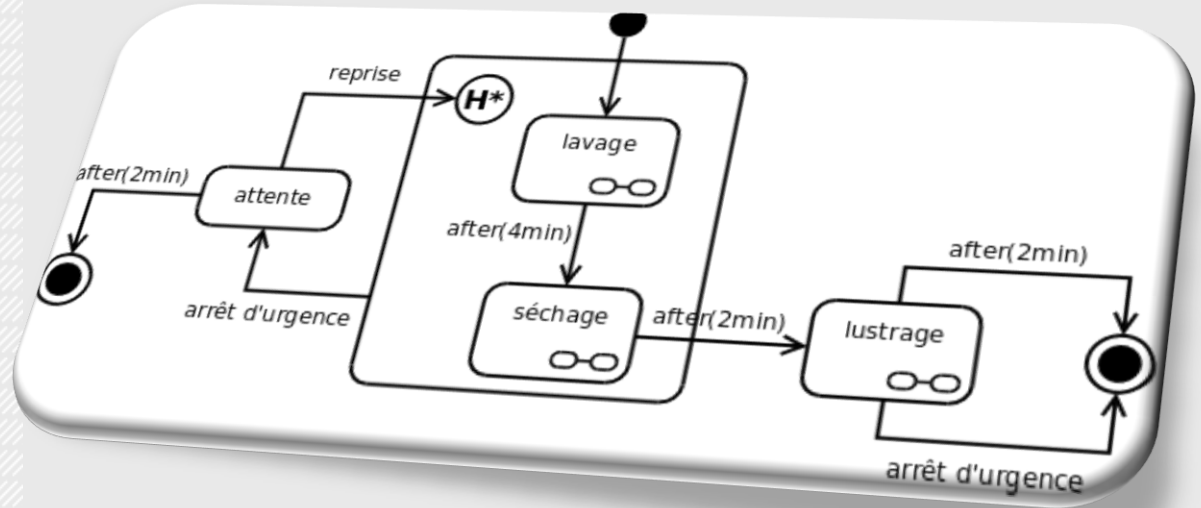
- Composition



UML : DIAGRAMME D'OBJETS

EXERCICES

UML : DIAGRAMME D'ETATS-TRANSITIONS



UML : DIAGRAMME D'ETATS-TRANSITIONS

DEFINITION:

- Décrit le **comportement** des objets d'une classe au moyen d'un automate d'états associés à la classe
- Le comportement est modélisé par un graphe :
 - Nœuds = **états possibles** des objets
 - Arcs = transitions d'état à état
- Une **transition** :
 - exécution d'une action
 - réaction de l'objet sous l'effet d'une occurrence **d'événement**

UML : DIAGRAMME D'ETATS-TRANSITIONS

DEFINITION:

- Description du **cycle de vie** d'un objet d'une classe :
 - Les **états** qui peuvent être pris par les **objets** d'une classe
 - Les **événements** qui provoquent la **transition d'un état à un autre**
 - Les **actions** subies/provoquées qui accompagnent un **changement d'état**
 - Les **activités** qui surviennent tant que l'objet est dans un **état donné**.

UML : DIAGRAMME D'ETATS-TRANSITIONS

OBJECTIFS:

- **Globalement** : étudier les états d'un Système d'Information
- **Plus particulièrement** : Comprendre le système d'information en s'intéressant aux **classes** qui présentent des traitements complexes
 - On se limite aux classes qui sont cruciales pour le champ de l'étude
 - On se limite aux états qui sont d'un intérêt pour le champ de l'étude
- Fournir une **représentation** dynamique du **comportement** des objets d'une classe
- Aider à déterminer les **événements** qui occasionnent les transitions
- Aider à déterminer les **opérations** qui vont permettre ces transitions

UML : DIAGRAMME D'ETATS-TRANSITIONS

UTILITE :

- Décrire un processus (type workflow)
- Aider à concevoir les activités, les IHM
- Aider à construire les scénarios de test
- Aider à déterminer les états pour gérer les risques de dysfonctionnement

UML : DIAGRAMME D'ETATS-TRANSITIONS

NOTION D'ETAT:

- Un état = étape dans le **cycle de vie** d'un objet
- Chaque objet possède à un instant donné un état particulier
- Chaque état est identifié par un nom.
- Un état est **stable** et **durable**
- Chaque diagramme d'états-transitions comprend un état
- Il est possible de n'avoir aucun état final : *un système qui ne s'arrête jamais.*



Etat initial

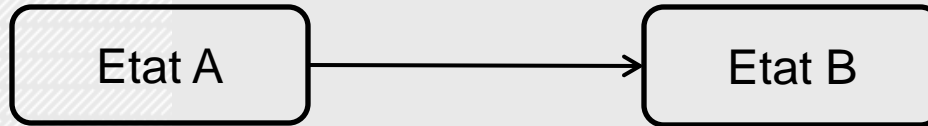


Etat final

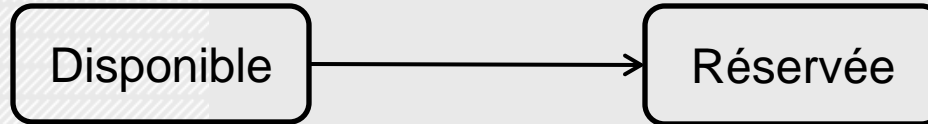
UML : DIAGRAMME D'ETATS-TRANSITIONS

NOTION DE TRANSITION:

- Les états sont reliés par des connexions unidirectionnelles appelées **transitions**



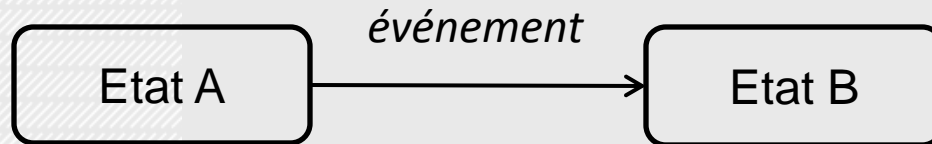
- Ex : place de parking



UML : DIAGRAMME D'ETATS-TRANSITIONS

NOTION D'EVENEMENT:

- Un événement correspond à l'occurrence d'une situation donnée dans le domaine étudié
- Un événement est une information instantanée qui doit être traitée à l'instant où il se produit



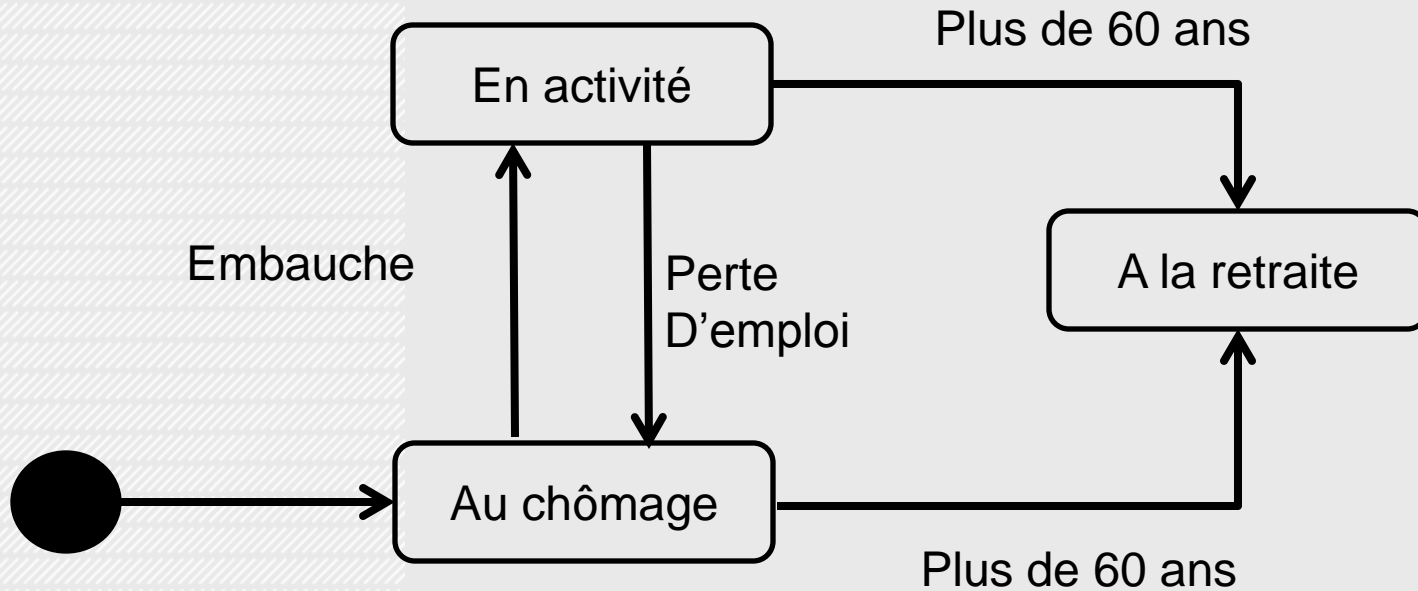
UML : DIAGRAMME D'ETATS-TRANSITIONS

NOTION D'ÉVÉNEMENT:

- Syntaxe d'un événement :
 - Nom de l'événement (Nom de paramètre : Type,.....)
- La description complète d'un événement est donnée par :
 - Nom de l'événement
 - Liste des paramètres
 - Objet expéditeur
 - Objet destinataire
 - Description textuelle

UML : DIAGRAMME D'ETATS-TRANSITIONS

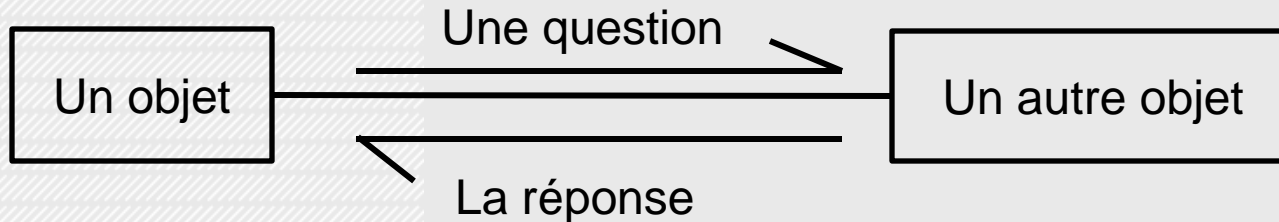
NOTION D'EVENEMENT:



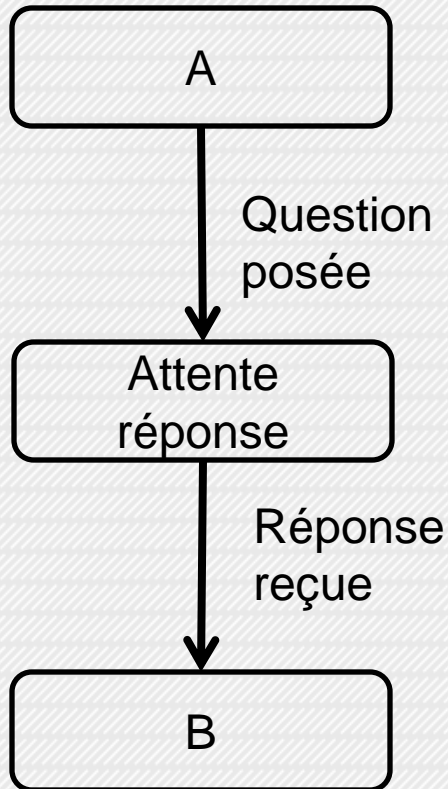
UML : DIAGRAMME D'ETATS-TRANSITIONS

COMMUNICATION ENTRE OBJETS PAR EVENEMENTS

- La communication est de type asynchrone, atomique et unidirectionnelle



UML : DIAGRAMME D'ETATS-TRANSITIONS



COMMUNICATION ENTRE OBJETS PAR EVENEMENTS

- L'objet émetteur de la requête se met en attente de la réponse de l'objet récepteur de la requête

UML : DIAGRAMME D'ETATS-TRANSITIONS

EN RESUME :

- Etat d'un objet :
 - Situation d'un objet que l'on désire connaître et gérer.
- Transition :
 - Passage d'un objet d'un état à un autre. Elle est déclenchée par un événement
- Evénement :
 - Stimulus qui provoque une (ou plusieurs) transition(s). A chaque stimulus peut correspondre une **action** responsable des modifications de l'objet (les valeurs des attributs)

UML : DIAGRAMME D'ETATS-TRANSITIONS

ETATS SPECIAUX:

- 2 états prédéfinis :
 - état de démarrage : obligatoire, unique
 - état de fin : optionnel, peut-être multiple

Création de l'objet



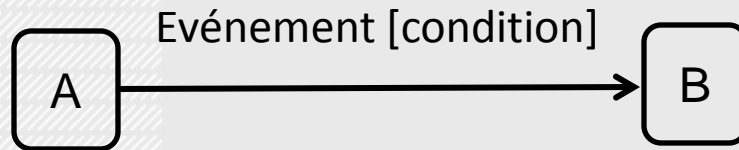
Fin de vie de l'objet



UML : DIAGRAMME D'ETATS-TRANSITIONS

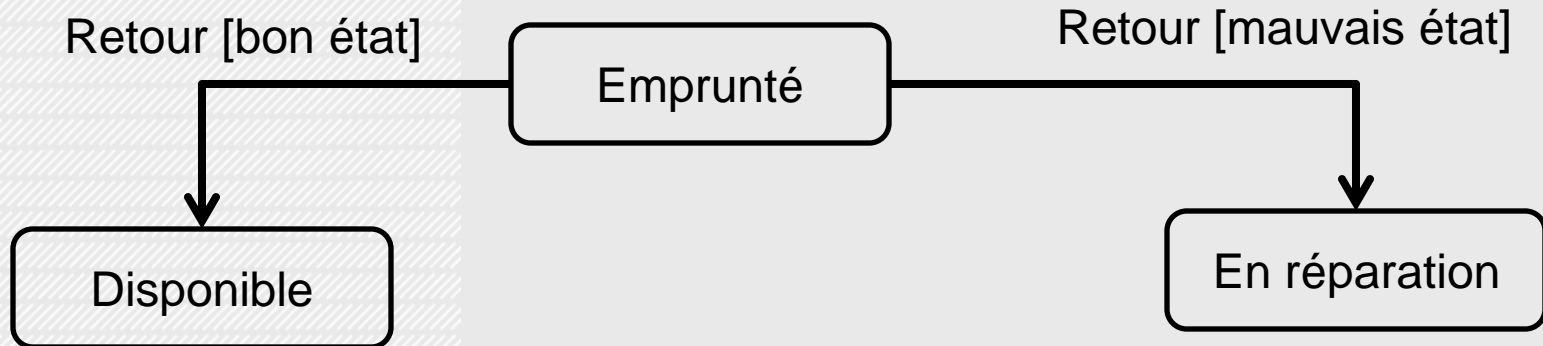
NOTION DE GARDE:

- Une garde est une condition booléenne qui permet ou non le déclenchement d'une transition lors de l'occurrence d'un événement.



UML : DIAGRAMME D'ETATS-TRANSITIONS

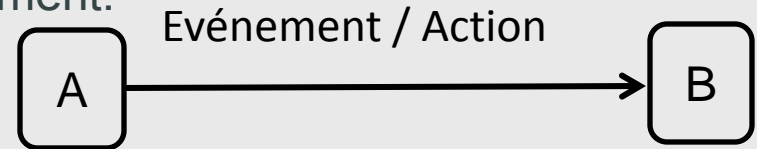
COMMUNICATION ENTRE OBJETS PAR EVENEMENTS :



UML : DIAGRAMME D'ETATS-TRANSITIONS

NOTIONS D'OPERATION ET D'ACTION :

- Action et activités : Le lien entre les **opérations** définies dans la spécification d'une **classe** et les **événements** apparaissant dans la diagramme d'états-transitions
- Chaque transition peut avoir une action à exécuter lorsqu'elle est déclenchée
- L'action est considérée comme instantanée et atomique
- Une action correspond à l'exécution d'une des opérations déclarées dans la classe de l'objet destinataire de l'événement.



- L'action a accès aux paramètres de l'événement ainsi qu'aux attributs de l'objet sur lequel elle s'applique

UML : DIAGRAMME D'ETATS-TRANSITIONS

ACTIONS DANS UN ETAT :

- Les états peuvent également contenir des actions :

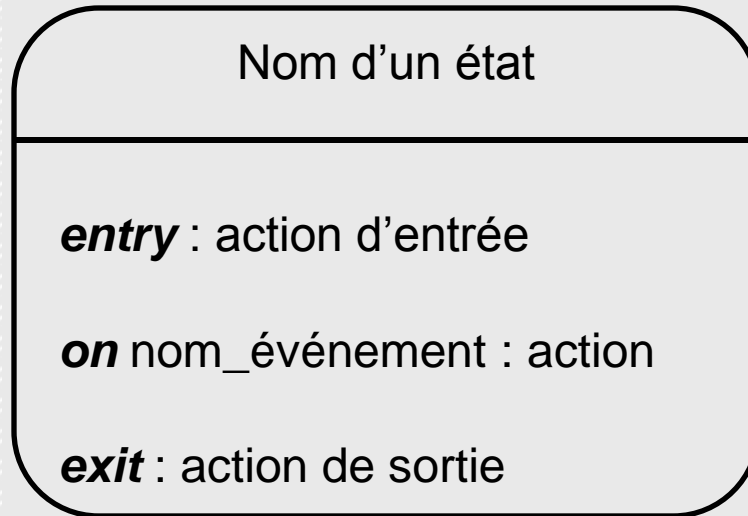
Elles sont exécutées :

- à l'entrée (*entry*) ou à la sortie (*exit*) de l'état
- lorsqu'une occurrence d'événement interne (*on*) survient
(exécutée lors de l'occurrence d'un événement qui ne conduit pas à un autre état)

UML : DIAGRAMME D'ETATS-TRANSITIONS

ACTIONS DANS UN ETAT :

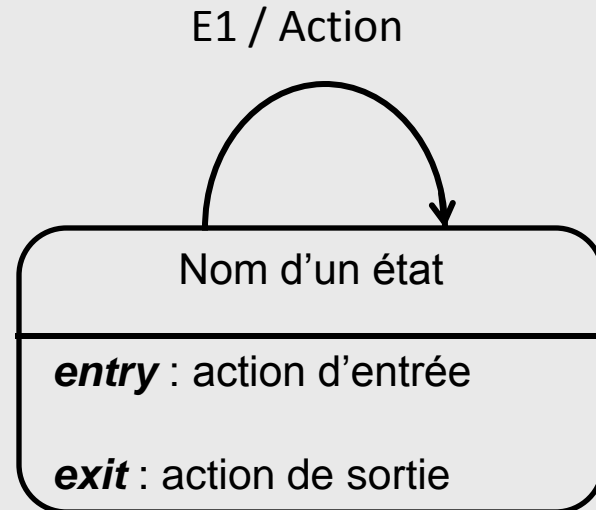
- Représentation :



UML : DIAGRAMME D'ETATS-TRANSITIONS

OPERATIONS, ACTIONS ET ACTIVITES :

- Un événement interne n'entraîne pas l'exécution des actions de sortie et d'entrée, contrairement au déclenchement d'une transition réflexive.



UML : DIAGRAMME D'ETATS-TRANSITIONS

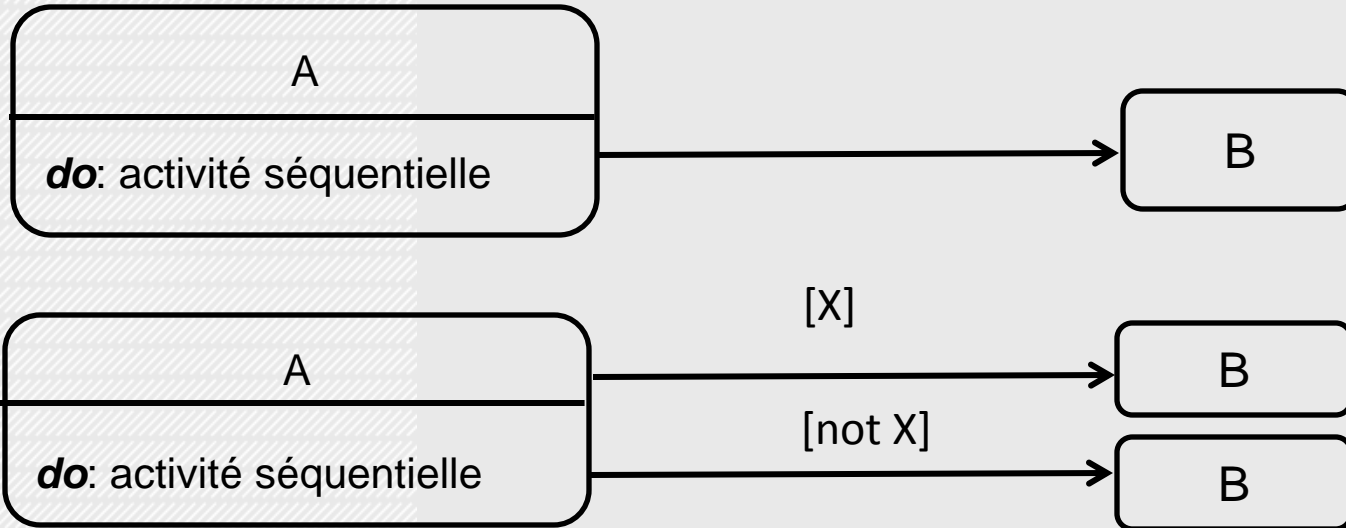
OPERATIONS, ACTIONS ET ACTIVITES :

- Contrairement à une action, une activité est une opération qui dure un certain temps
- Les activités sont associées aux états
 - commencent quand on est entré dans l'état
 - s'exécutent jusqu'à la fin si elles ne sont pas interrompues par une transition sortante (donc tant que l'état ne change pas)
 - peuvent être interrompues car elles ne modifient pas l'état de l'objet
- Les activités sont notées dans la partie inférieure de l'état

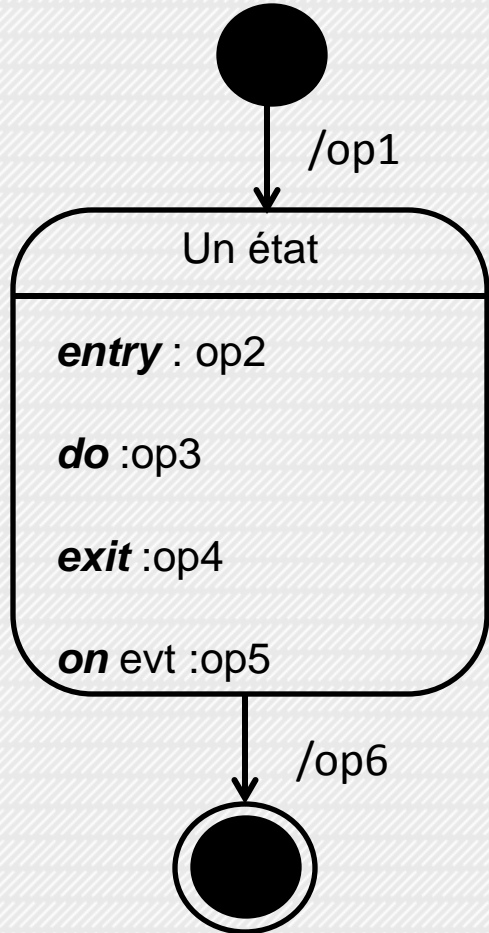
UML : DIAGRAMME D'ETATS-TRANSITIONS

OPERATIONS, ACTIONS ET ACTIVITES :

- Lorsqu'une activité se termine, les transitions automatiques (sans événement), mais éventuellement protégées par des gardes, sont déclenchées



UML : DIAGRAMME D'ETATS-TRANSITIONS

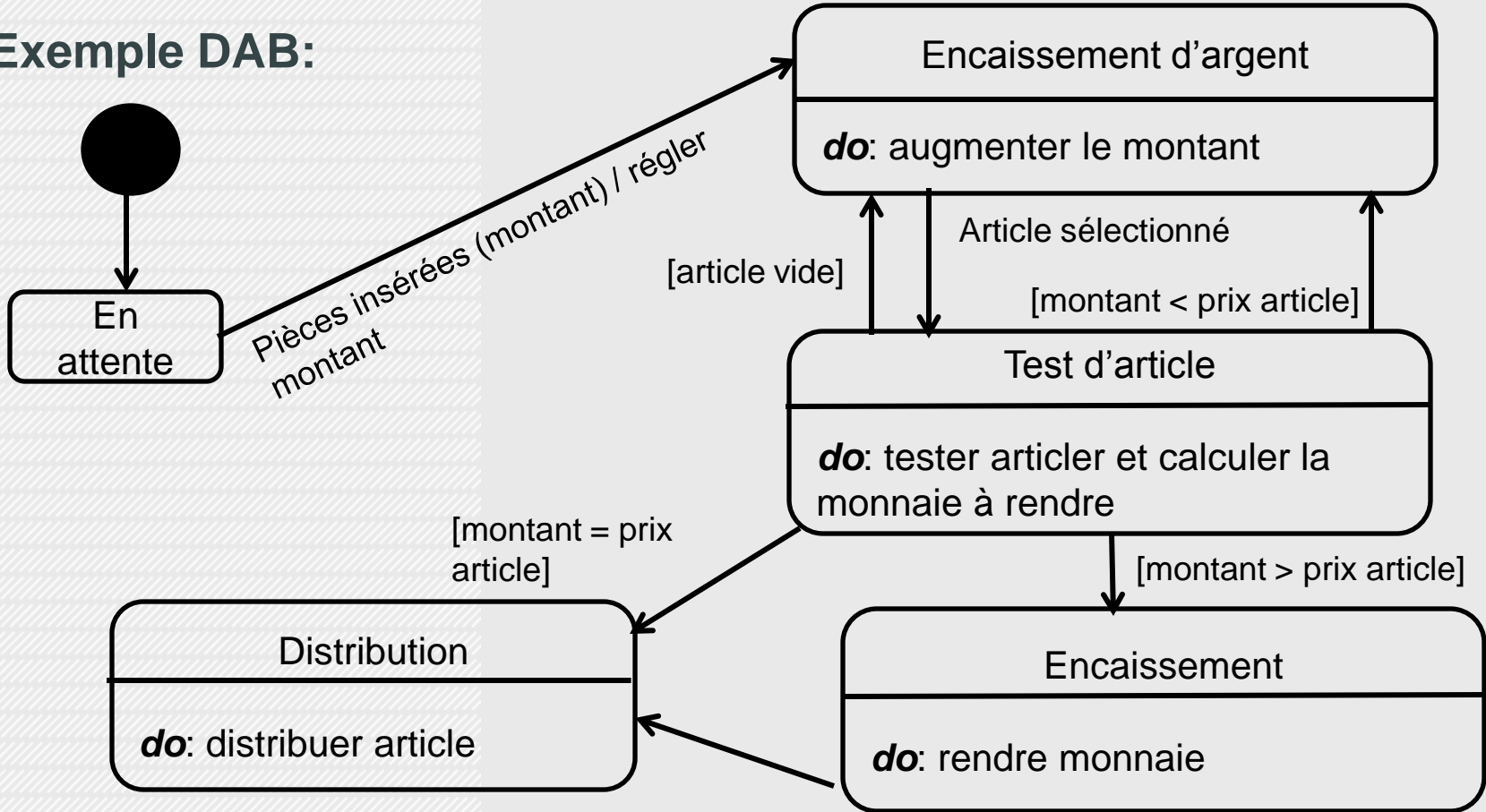


OPERATIONS, ACTIONS ET ACTIVITES :

- 6 manières d'associer une opération à une transition :
 - l'action **associée** à la transition d'entrée (op1)
 - l'action **d'entrée** de l'état (op2)
 - l'**activité** dans l'état (op3)
 - l'action de **sortie** de l'état (op4)
 - l'action **associée** aux **événements internes** (op5)
 - l'action **associée** à la **transition** de la sortie de l'état (op6)

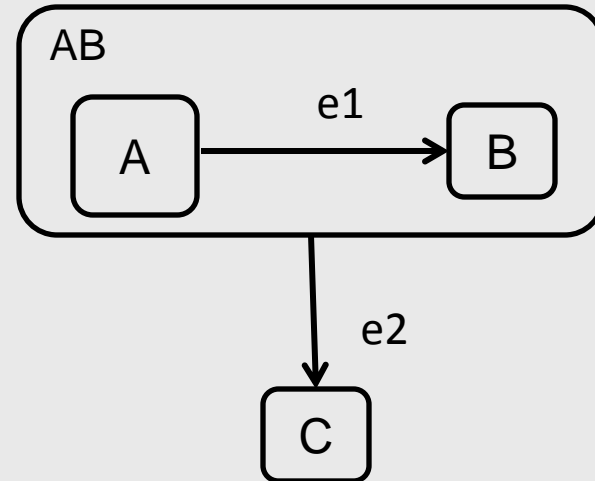
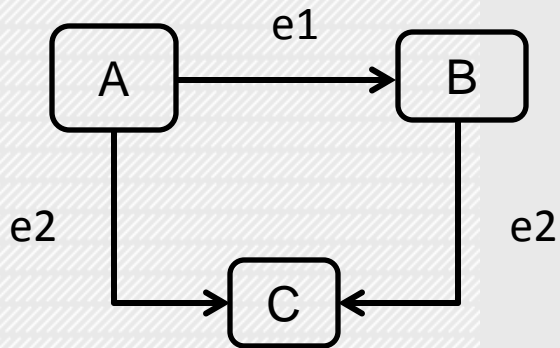
UML : DIAGRAMME D'ETATS-TRANSITIONS

Exemple DAB:



UML : DIAGRAMME D'ETATS-TRANSITIONS

Généralisation d'états:



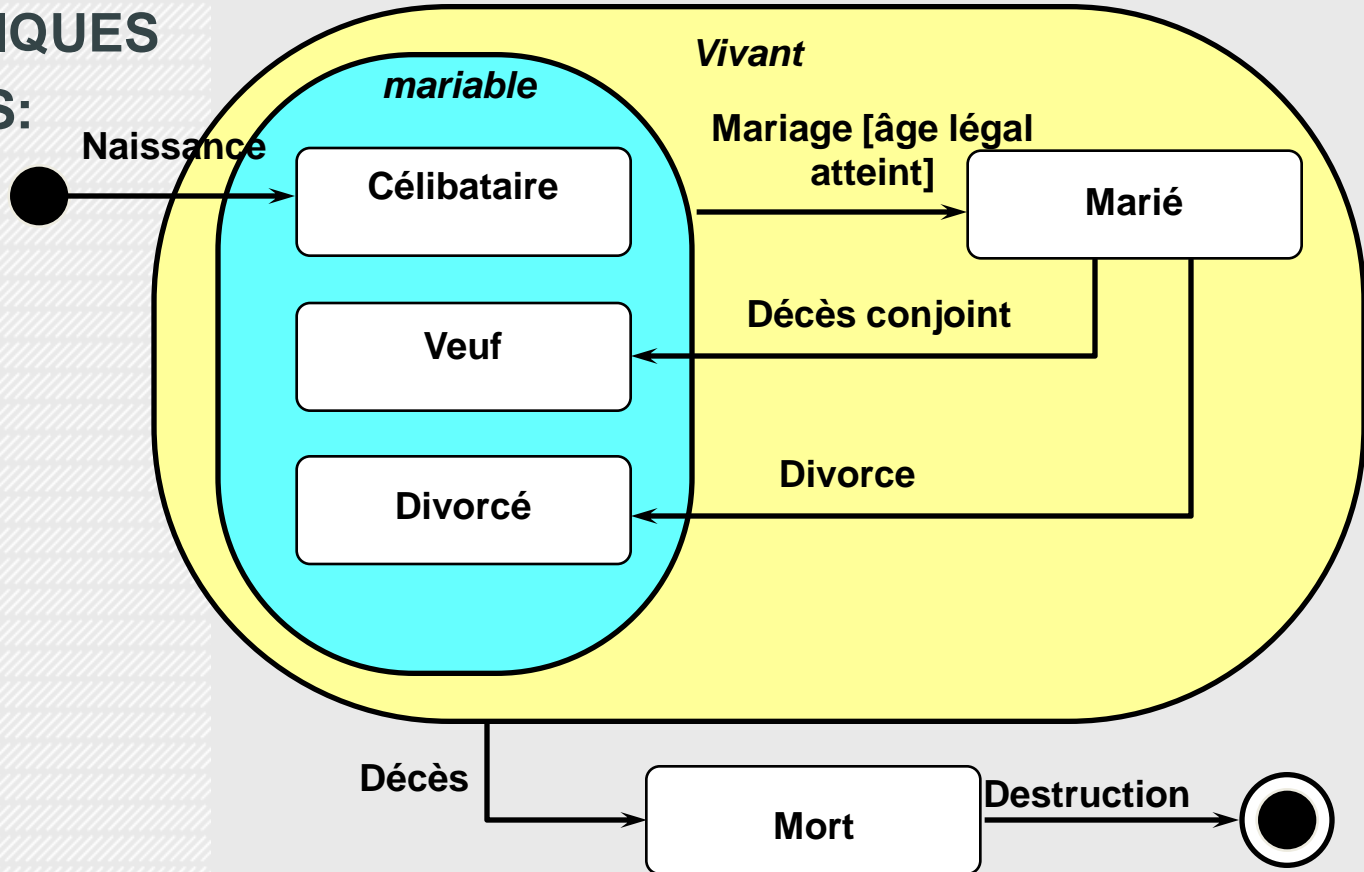
UML : DIAGRAMME D'ETATS-TRANSITIONS

ETATS IMBRIQUES - COMPOSITES:

- Si le diagramme d'état transition devient trop complexe, on peut utiliser des états imbriqués pour le simplifier
- Un super-état ou état composite est un état qui englobe d'autres états appelés sous-états
- Le nombre d'imbrication n'est pas limité (ne pas abusé sinon problème de lisibilité)

UML : DIAGRAMME D'ETATS-TRANSITIONS

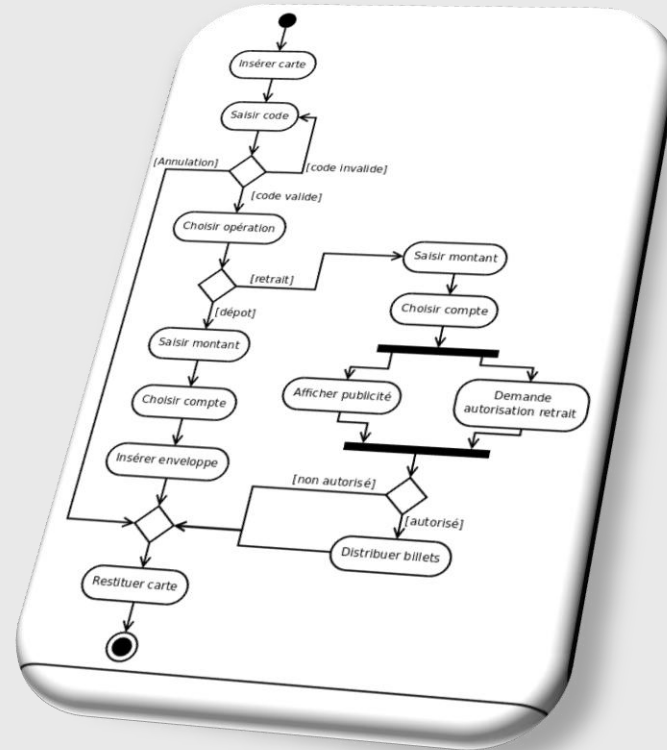
ETATS IMBRIQUES
COMPOSITES:



UML : DIAGRAMME D'ETATS-TRANSITIONS

EXERCICES

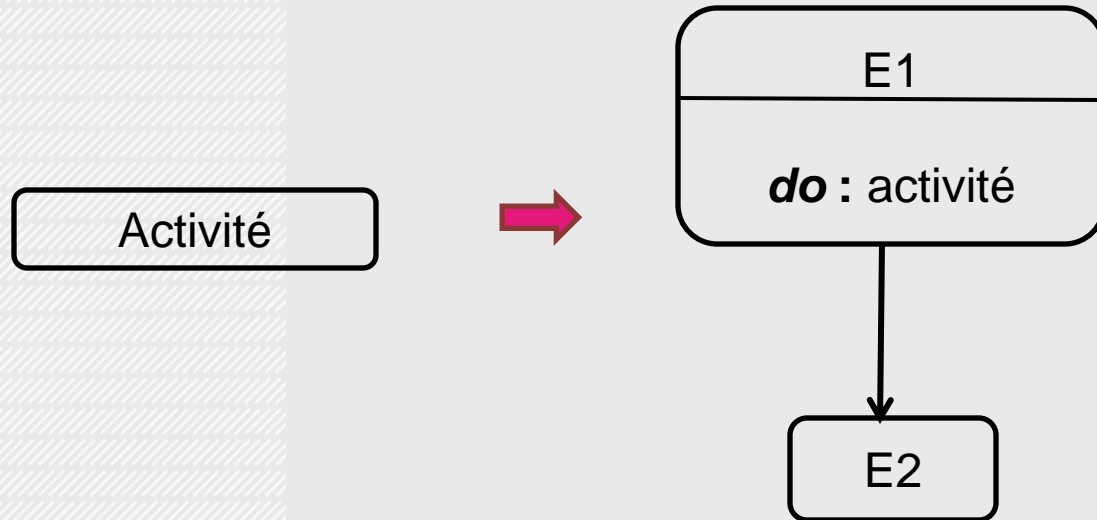
UML : DIAGRAMME D'ACTIVITES



UML : DIAGRAMME D'ACTIVITES

DEFINITION:

- Variante des diagrammes d'états-transitions

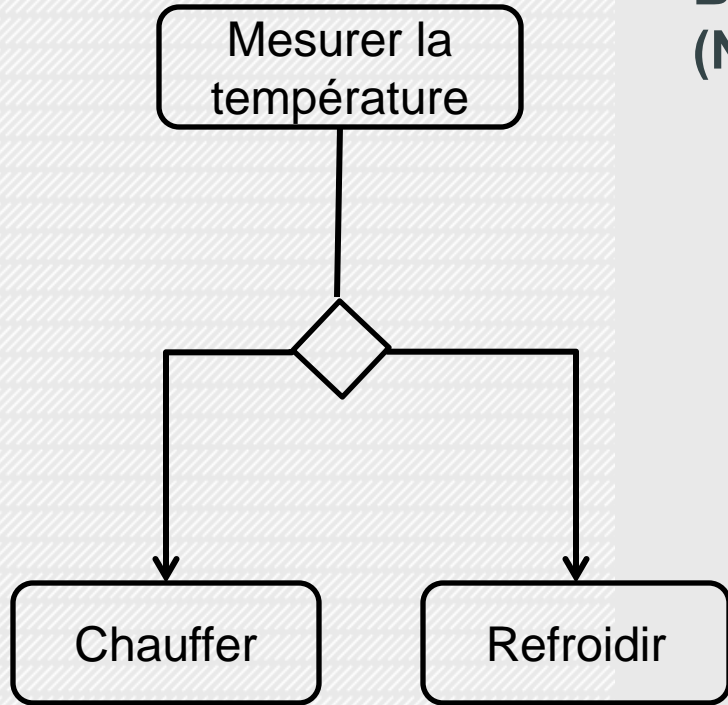


UML : DIAGRAMME D'ACTIVITES

DEFINITION:

- Variante des diagrammes d'états-transitions
- Le diagramme d'activité permet de représenter le comportement interne d'un use case ou processus.
- Représente le déroulement des traitements en les regroupant dans des étapes appelées « Activité »
- La question réside dans comment décomposer les traitements, jusqu'où aller dans la décomposition (quels critères)

UML : DIAGRAMME D'ACTIVITES



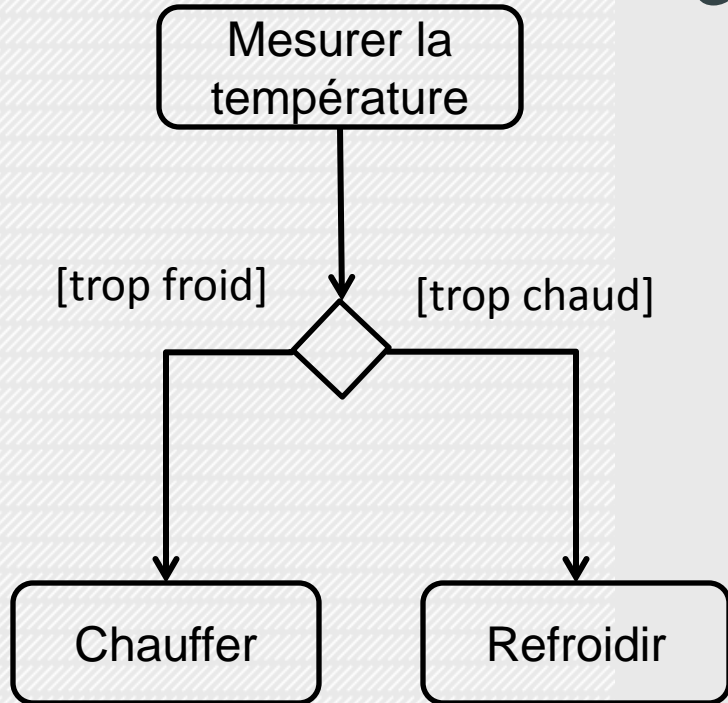
BRANCHEMENT CONDITIONNEL (NŒUD):

- Un losange représente un branchement conditionnel (des décisions) qui offre des possibilités d'alternatives
- Il permet d'introduire des conditions booléennes (vrai / faux) mutuellement exclusives, notées entre crochets (notion de « condition surveillée » ou « expression de garde »).

UML : DIAGRAMME D'ACTIVITES

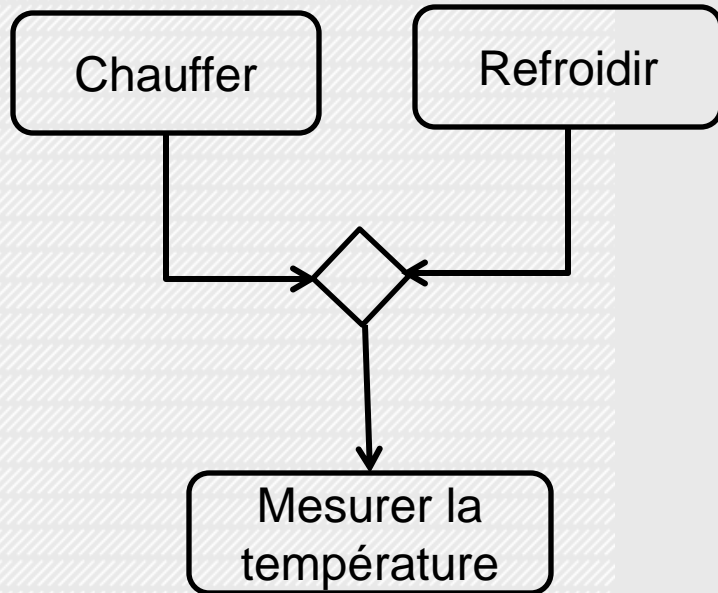
GARDES:

- Les gardes sont les bales des transitions dont elles valident le déclenchement



UML : DIAGRAMME D'ACTIVITES

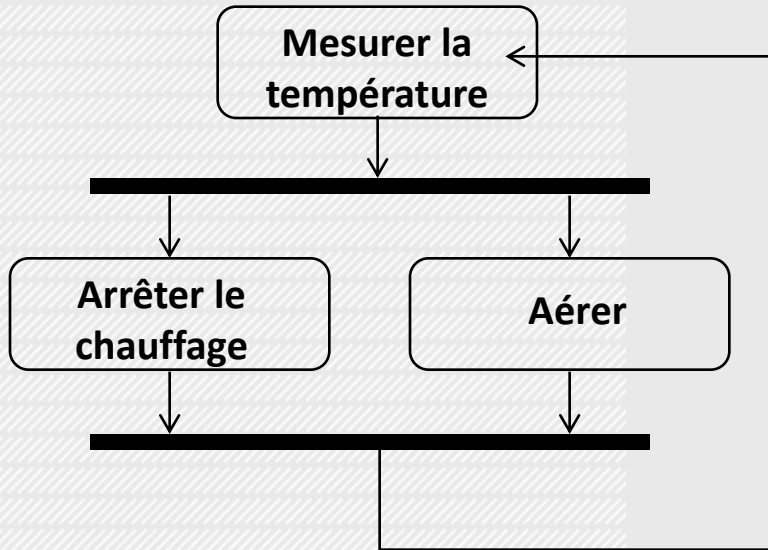
NŒUD DE CONVERGENCE (CONFLUENCE):





- Un losange est utilisé pour modéliser un point de convergence : endroit où deux chemins d'exécution mutuellement exclusifs se rejoignent. A partir de ce point de convergence la dernière partie du trajet est commune.

UML : DIAGRAMME D'ACTIVITES

PARALLELISME ET SYNCHRONISATION :

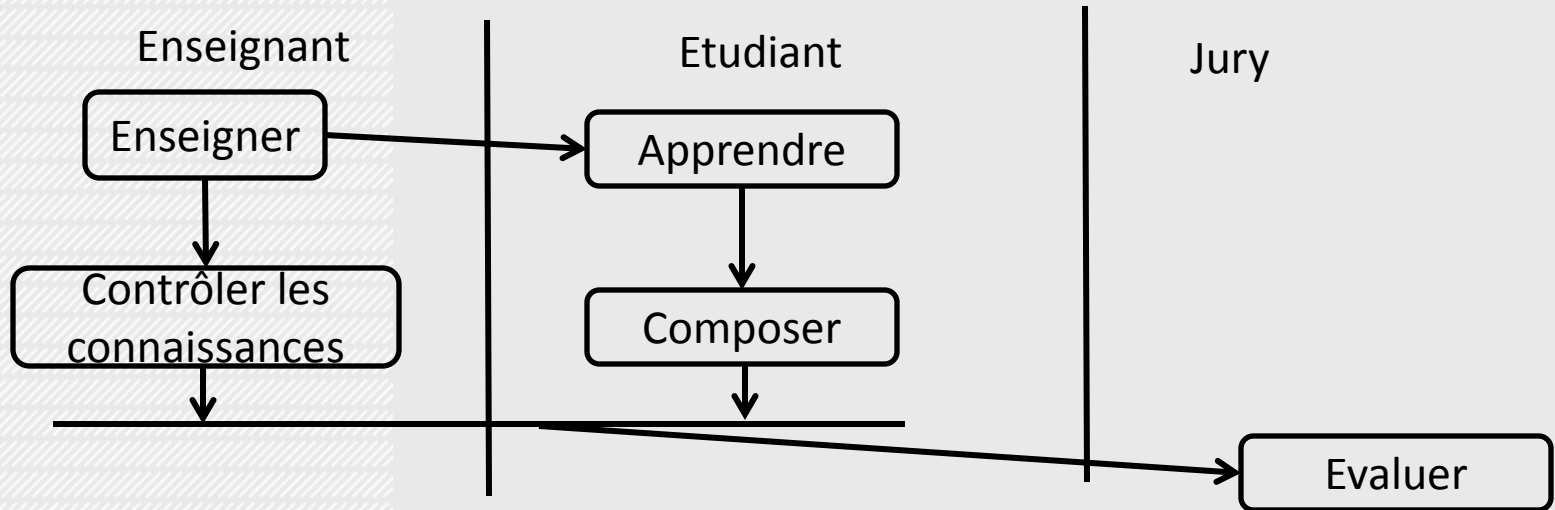


- On représente une synchronisation et le parallélisme par une barre verticale ou horizontale:  
- Parallélisme utilisé pour représenter des déroulements parallèles.
- Synchronisation utilisée pour représenter la fin des traitements parallèles.

UML : DIAGRAMME D'ACTIVITES

DECOUPAGE (COULOIR D'ACTIVITE – SWIM LANE):

- Pour montrer les différentes responsabilités au sein d'un mécanisme ou d'une organisation, on schématise des couloirs d'activités.
- Chaque activité est allouée à un couloir correspondant à la ressource concernée : partenaire, travailleur

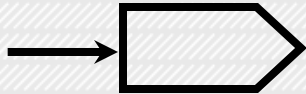


UML : DIAGRAMME D'ACTIVITES

AUTRES NOTATIONS :



- Signal reçu : un événement pour le processus étudié (externe ou interne au SI)
 - On le représente par un pentagone concave



- Signal envoyé : un résultat émis par le processus étudié (externe ou interne au SI)
 - On le représente par un pentagone convexe



- Evénement temporel : une date ou un délai
 - On le représente par deux triangles isocèles inversés (en tête bêche)



- Nœud de départ du diagramme
le diagramme d'activité peut comporter plusieurs nœuds de départ



- Nœud de fin du diagramme (pour une séquence d'activités)
le diagramme d'activité peut comporter plusieurs nœuds de fin

UML : DIAGRAMME D'ACTIVITES

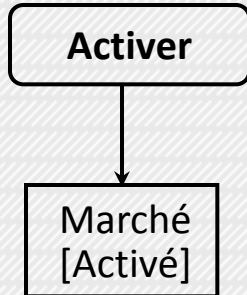
AUTRES NOTATIONS - Nœud d'objet :

- Souvent, différentes activités manipulent un même objet qui change alors d'état selon le degré d'avancement du mécanisme.

- Deux utilisations :

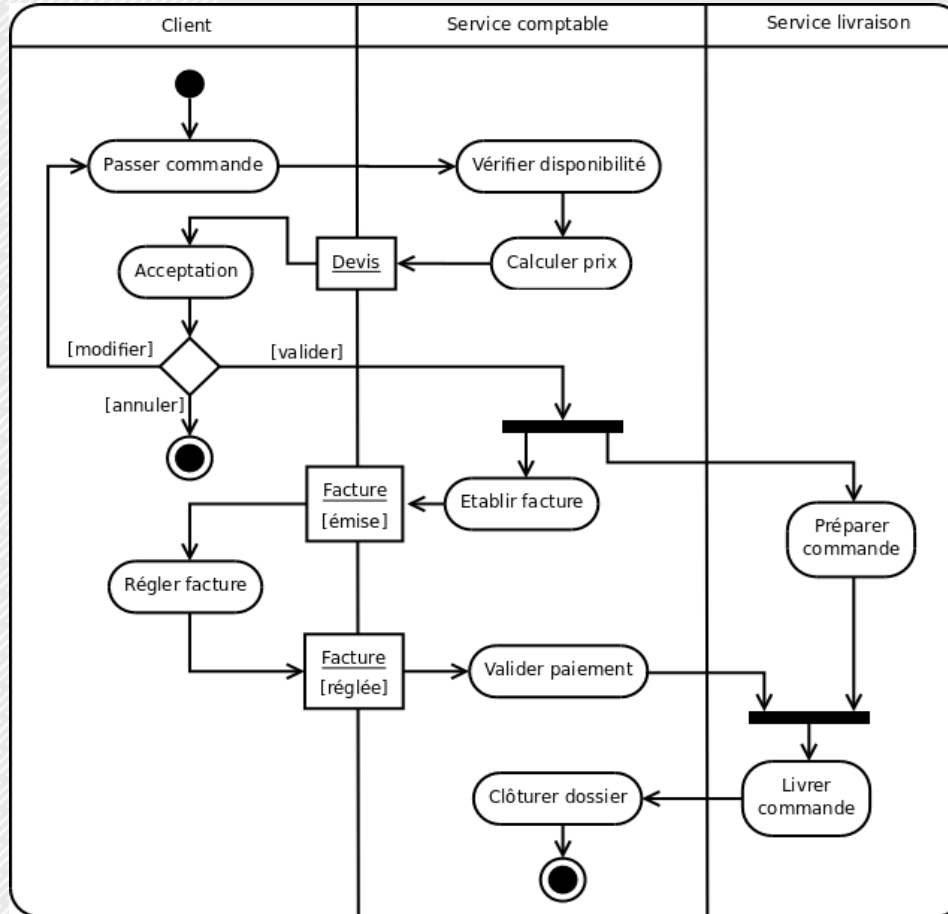
- Une information associée à l'activité (lié par une flèche en pointillés) pour indiquer qu'un message initialise l'objet visé dans l'état indiqué entre crochet. UML V1

- Un résultat de l'activité (lié par une flèche pleine) et repris comme événement pour l'activité suivante. UML V2



UML : DIAGRAMME D'ACTIVITES

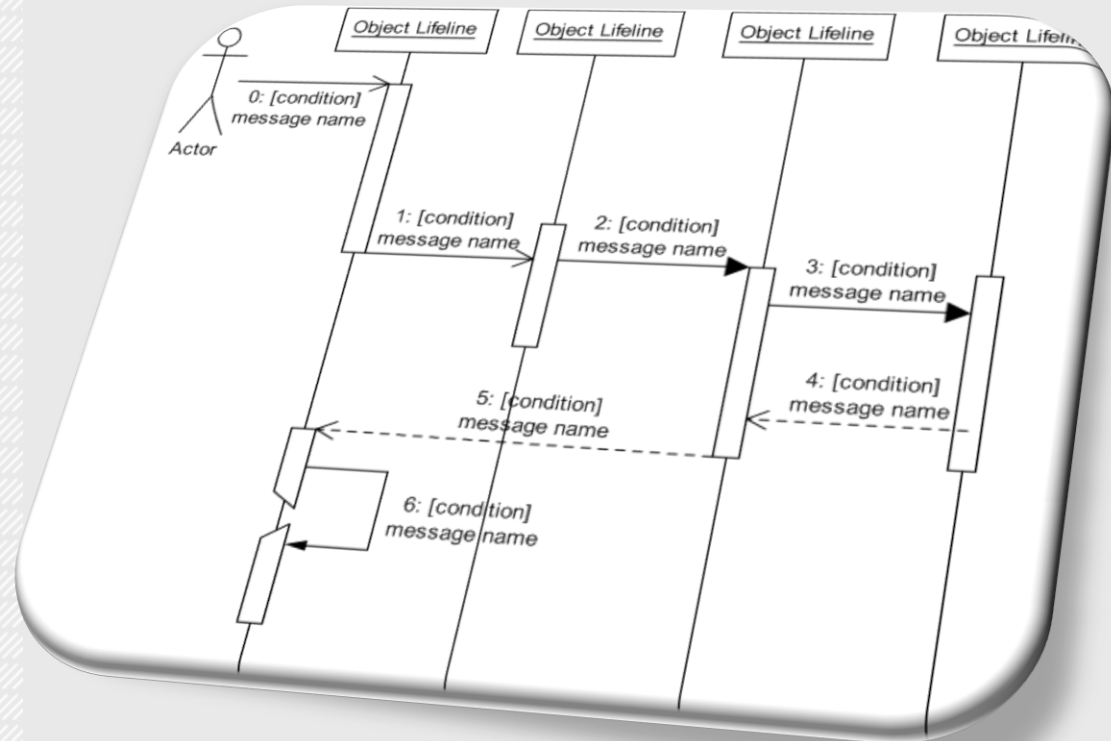
EXEMPLE



UML : DIAGRAMME D'ACTIVITES

EXERCICES

UML : DIAGRAMME DE SEQUENCE



UML : DIAGRAMME DE SEQUENCE

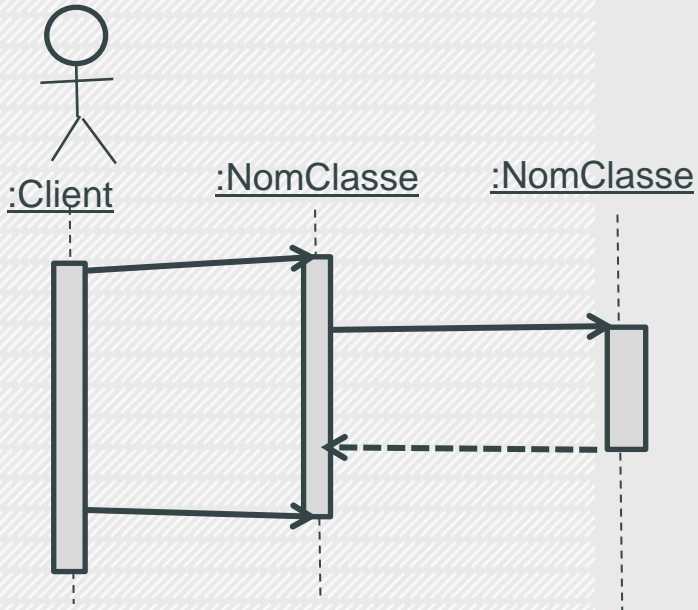
DEFINITION :

- Modélisation des interactions entre objets suite à un événement externe.
- Aspect temporel : messages asynchrones ou synchrones
- Un diagramme de séquence montre
 - des réactions du système aux actions des utilisateurs
 - interactions entre les objets du système selon un point de vue temporel,
 - la chronologie des envois de messages, des échanges d'informations
 - création et la manipulation des objets

UML : DIAGRAMME DE SEQUENCE

DEFINITION :

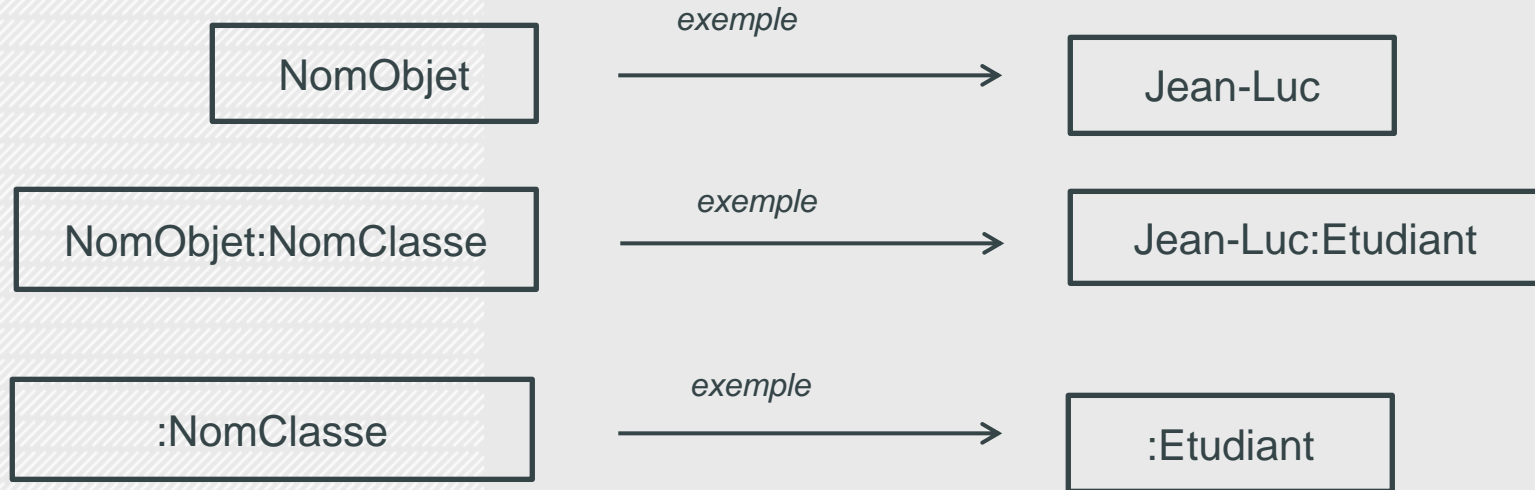
- 3 éléments fondamentaux :
 - Les Objets
 - La ligne de vie, ligne verticale en pointillée partant de chaque objet
 - Les messages/stimulus, signal représenté par des flèches de natures différentes
- 2 dimensions :
 - horizontale : interactions entre objets
 - verticale : le temps



UML : DIAGRAMME DE SEQUENCE

DEFINITION :

- Nommage de l'objet :



UML : DIAGRAMME DE SEQUENCE

CONCEPT – LIGNE DE VIE:

:NomClasse



• Représente l'existence de l'objet à un moment particulier de son cycle de vie ainsi que les actions qui peuvent être déclenchées par les messages sur cet objet :

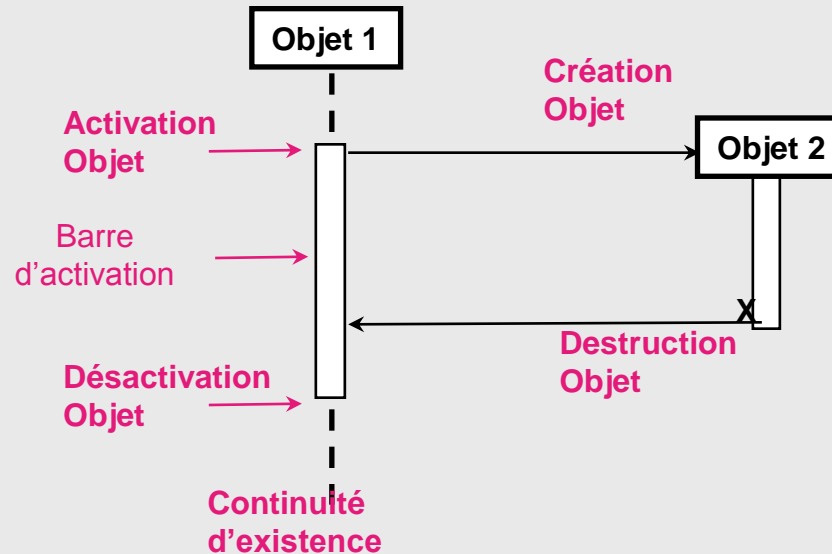
- Création
- Destruction
- Activation
- Désactivation

UML : DIAGRAMME DE SEQUENCE

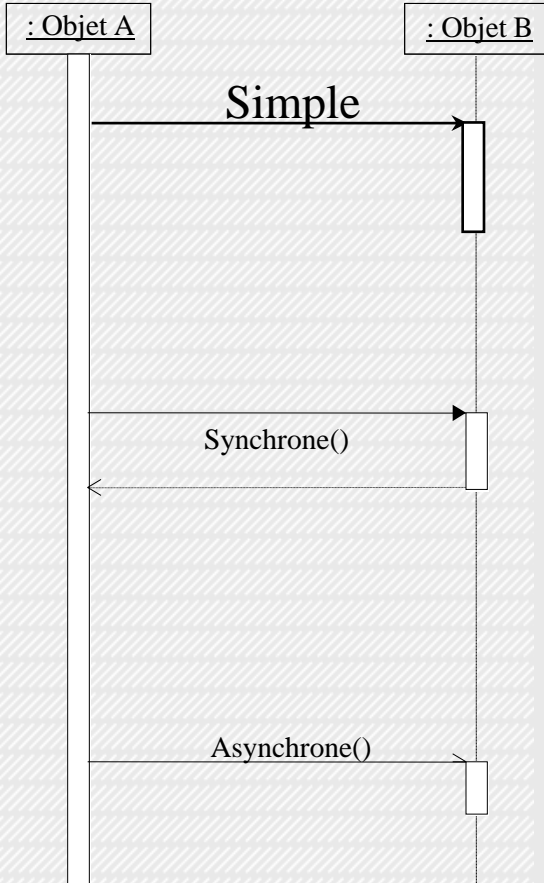
CONCEPT – LA BARRE D’ACTIVATION :

- Représente de manière explicite la durée pendant laquelle un objet accomplit une action au moyen d’un rectangle superposé à la ligne de vie.

:NomClasse



UML : DIAGRAMME DE SEQUENCE



CONCEPT – MESSAGES:

- Message simple : aucune caractéristique
—————>
- Message synchrone : l'expéditeur est bloqué jusqu'à réception de la réponse.
—————>
- Message asynchrone : n'interrompt pas l'exécution de l'expéditeur qui n'attend pas de réponse du récepteur.
—————>
- Message réponse :
 - Facultatif pour synchrone : ----->
 - Obligatoire pour asynchrone : ----->

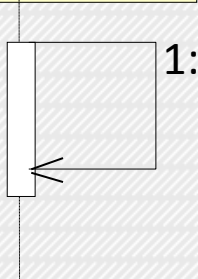
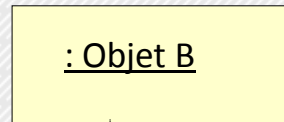
UML : DIAGRAMME DE SEQUENCE

AUTRE CONCEPT – MESSAGES:

- Notion de Garde :

Représentation de condition d'envoi de message

[condition] message



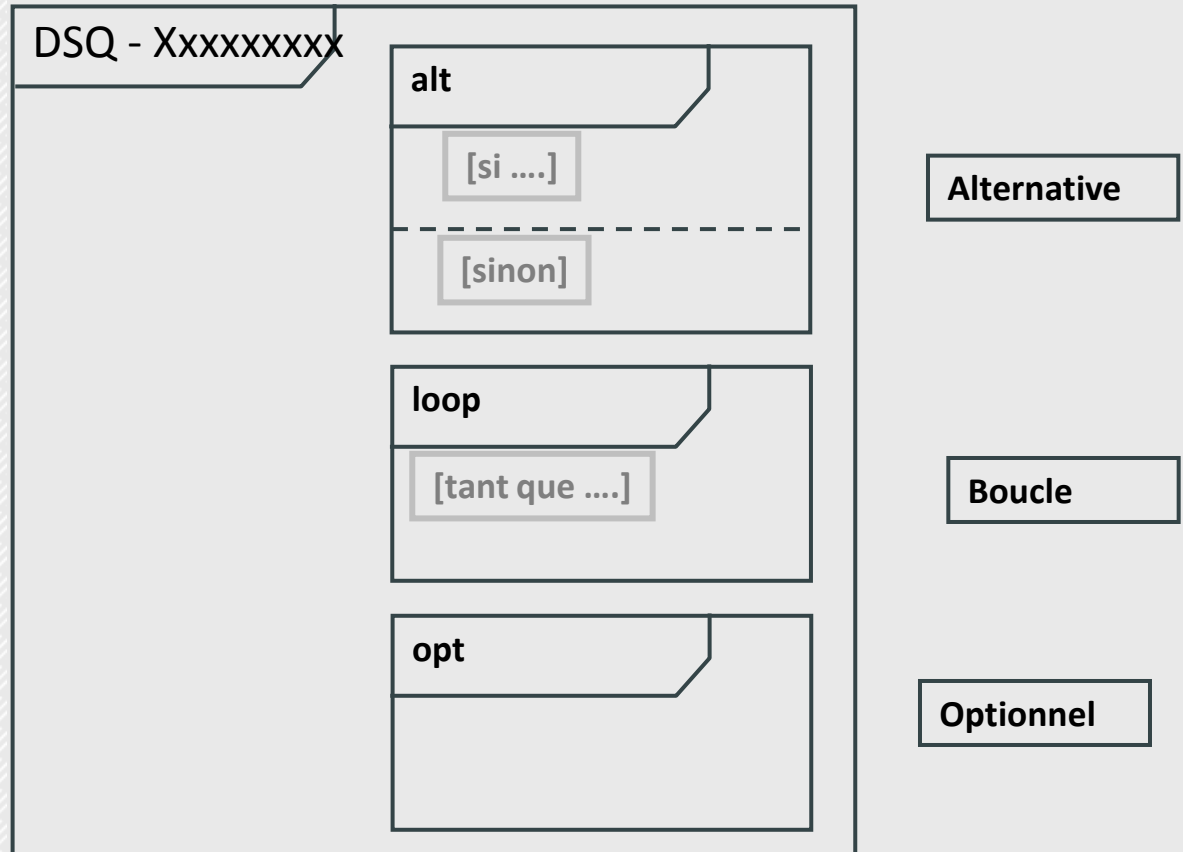
- Message réflexif :

Représentation des Actions internes à l'objet au moyen de message que l'objet s'envoie à lui même

UML : DIAGRAMME DE SEQUENCE

CONCEPT – BLOC:

- Représentation :



UML : DIAGRAMME DE SEQUENCE

CONCEPT – BLOC:

Alt (alternative) : utilisée pour présenter plusieurs branchements possibles. Les conditions sont exprimées [..]

Opt (option) : fragment d'interaction optionnel. Les conditions sont exprimées [..]

Loop boucle : fragment d'interaction répété. Les conditions sont exprimées [..]

Sd déclaration de référence : fragment d'interaction référencé comme un fragment réutilisable

Ref appel à référence : réutilisation d'un fragment référencé

Break arrêt : définition d'un fragment d'interaction correspondant à une exception. . Les conditions sont exprimées [...]

Contrairement à un fragment optionnel ou alternatif etc. ... la séquence tout entière est interrompue à la fin de l'exécution de ce fragment.

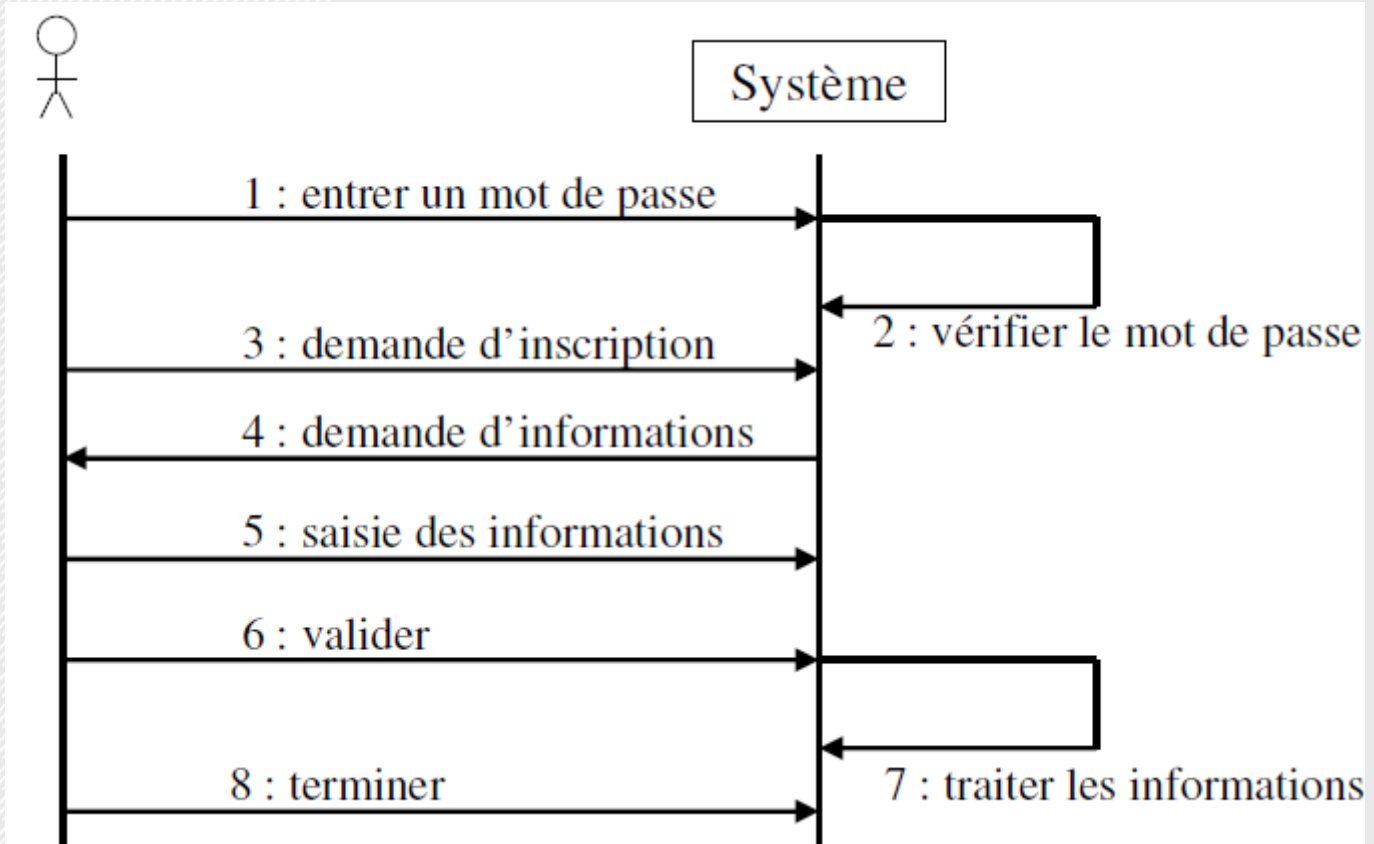
Seq : séquencement faible : délimitation d'une région dans laquelle l'ordre de séquencement n'est pas défini par le positionnement vertical des messages

Strict : séquencement fort : correspondant à la valeur par défaut de tout diagramme de séquence - l'ordre de séquencement est défini et doit être strictement respecté

Critical : déclaration d'une zone d'interaction partageant des ressources concurrentes et ne devant être exécutée que par une seule tâche à la fois

Par : déclaration d'une zone d'interaction dont plusieurs séquences sont exécutées en multitâche (parallélisme)

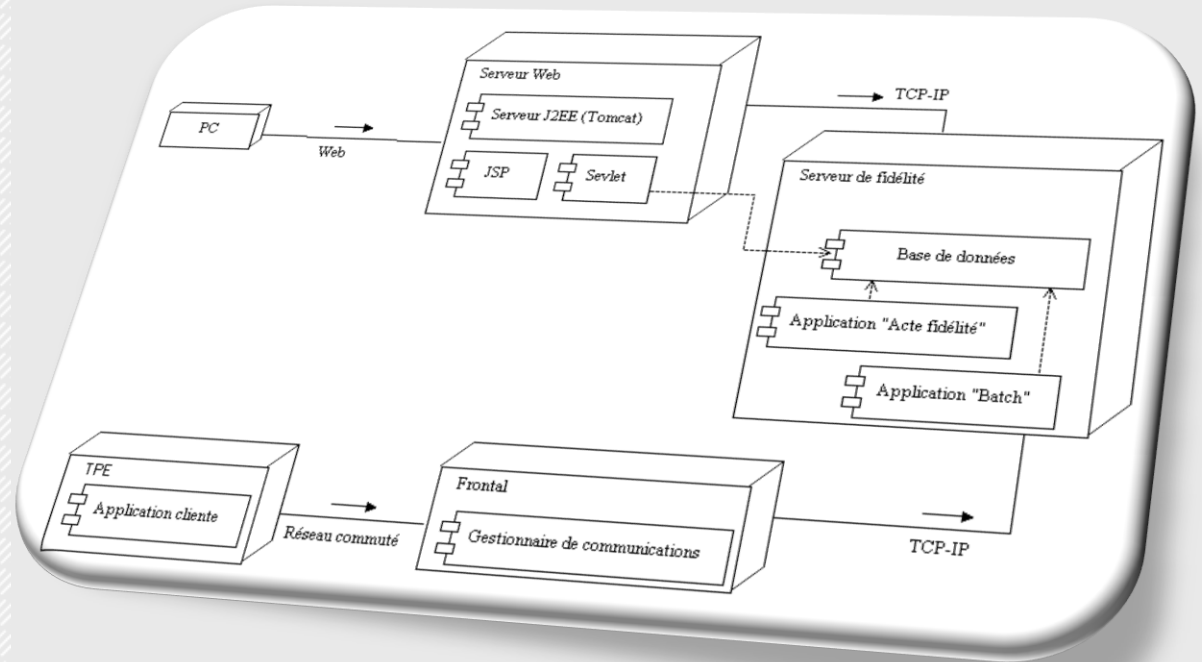
UML : DIAGRAMME DE SEQUENCE



UML : DIAGRAMME DE SEQUENCE

EXERCICES

UML : DIAGRAMME DE COMPOSANTS



UML : DIAGRAMME DE COMPOSANT

DEFINITION :

- Décrit les éléments physiques e leurs relations dans l'environnement de réalisation
 - Montre les dépendances entres les composants logiciels
 - Permet de spécifier l'architecture logicielle dans une environnement de développement donné
- **Composant :**
 - Notion de réutilisabilité
 - Comportement masqué, seules ses interfaces sont représentées

UML : DIAGRAMME DE COMPOSANT

NOTATION :

- **Interface requise** : demi-cercle
- **Interface offerte** : cercle

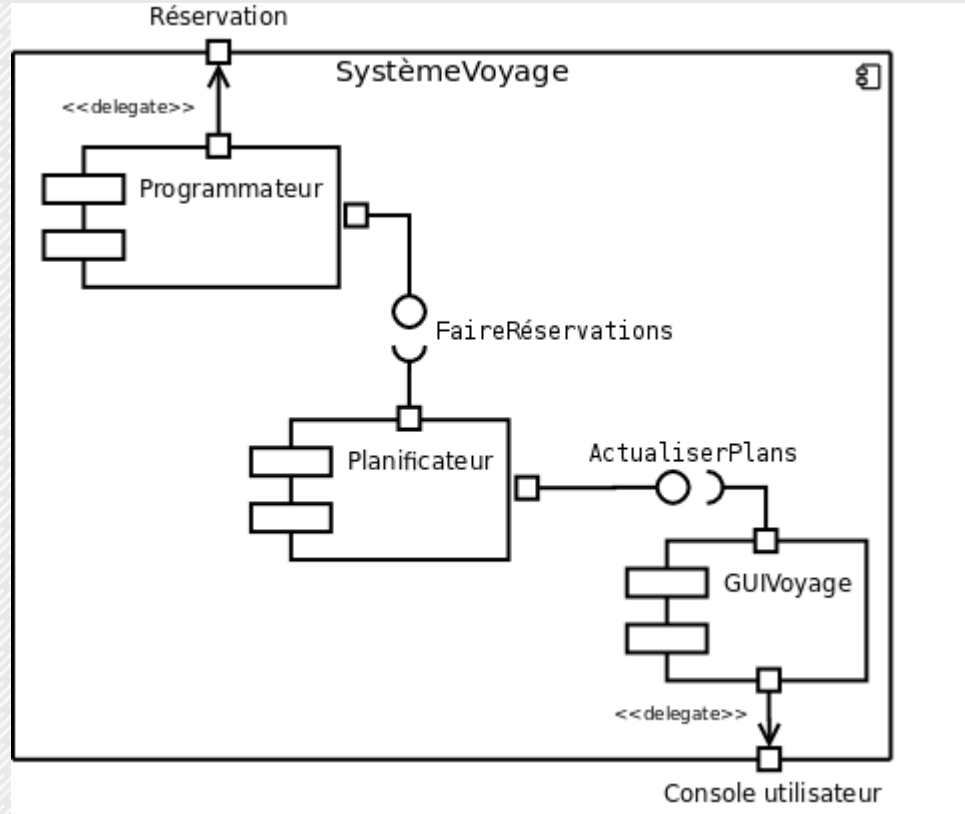


- Représentation alternative :

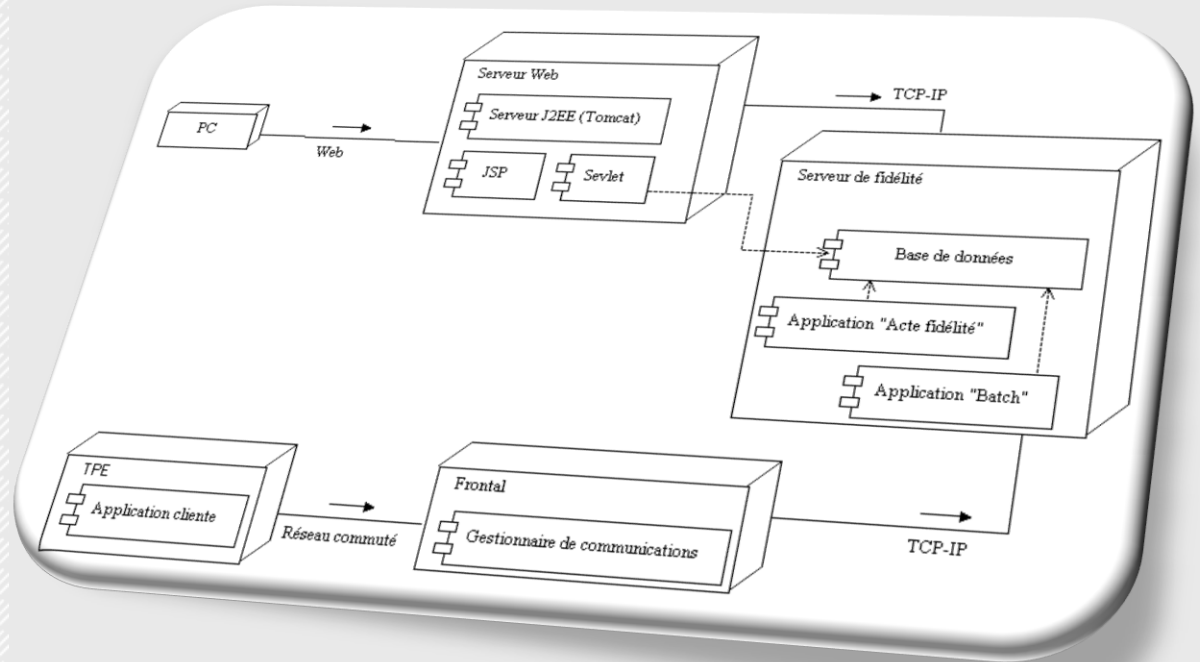


UML : DIAGRAMME DE COMPOSANT

EXEMPLE :



UML : DIAGRAMME DE DEPLOIEMENT



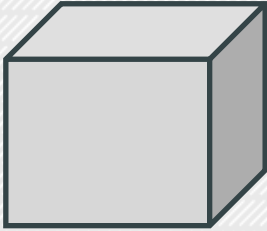
UML : DIAGRAMME DE DEPLOIEMENT

DEFINITION :

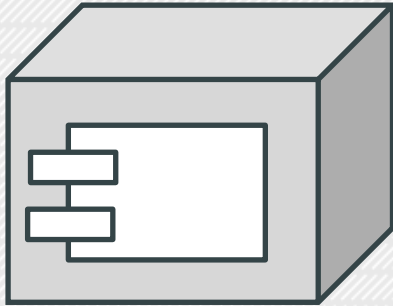
- Représentation de l'architecture matérielle et logicielle du système
- Disposition physique des différents matériels (nœuds) qui entrent dans la composition d'un système
- Répartition des programmes exécutables sur ces matériels
- Lien de dépendance = Lien de communication
- Permet de :
 - Donner la structure d'une plate-forme physique
 - Préciser où se trouvent les processus
 - Montrer comment les objets se créent et se déplacent dans une architecture distribuée

UML : DIAGRAMME DE DEPLOIEMENT

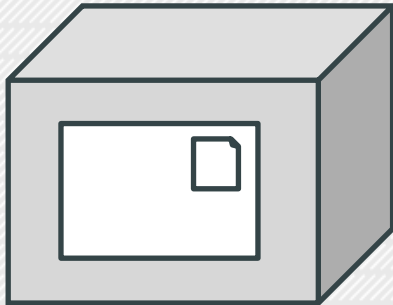
NOTATION :



- **Nœuds**



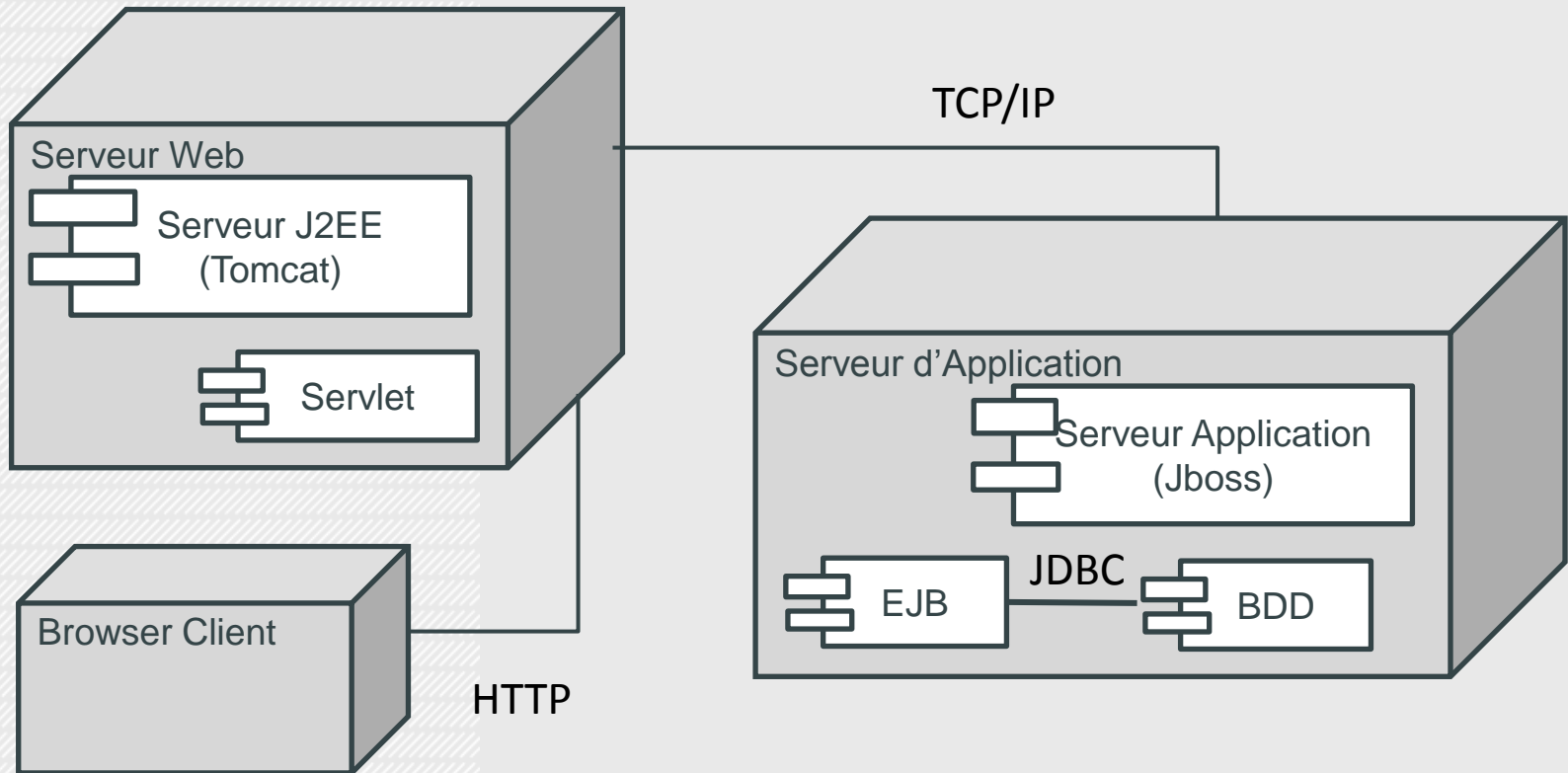
- **Composants**



- **Artefacts** : élément concret existant dans le monde réel

UML : DIAGRAMME DE DEPLOIEMENT

EXEMPLE:



UML : UML V2



UML : UML V2

MOTIVATIONS : Pourquoi une version 2.x ?

1 : MDA / MDD

- **Model Driven Architecture**
- **Model Driven Development**

Construction par modélisation plutôt que par codage

UML : UML V2

MOTIVATIONS : Pourquoi une version 2.x ?

2 : Abstraction

- Elever le niveau d'abstraction
- Développement plus naturel (détaché du code pur)

UML : UML V2

MOTIVATIONS : Pourquoi une version 2.x ?

3 : Modularisation

- Modulariser ce langage en différents sous-langages
- Pas nécessaire d'apprendre toute la syntaxe UML

UML : UML V2

NOUVEAUTES:

- 4 Nouveaux Diagrammes :
 - Diagramme de paquetage (*package diagram*)
 - Diagramme de structure composite (*composite structure diagram*)
 - Diagramme global d'interaction (*interaction overview*)
 - Diagramme de temps (*timing diagram*)

UML : UML V2

RAPPEL DIAGRAMMES

Statiques (structures – données) :

- Diagramme d'objets
Object diagram
- Diagramme de classes
Class diagram
- Diagramme de composants
Component diagram
- Diagramme de déploiement
Deployment diagram
- Diagramme de paquetages
Package diagram
- Diagramme de structure composite
Composite Structure diagram

Dynamiques (comportement) :

- Diagramme de cas d'utilisation
Use case diagram
- Diagramme de collaboration
(communication)
Communication diagram
- Diagramme de séquence
Sequence diagram
- Diagramme d'états-transition
State Machine diagram
- Diagramme d'activités
Activity diagram
- Diagramme global d'interaction
Interaction Overview diagram
- Diagramme de temps
Timing diagram

UML : UML V2

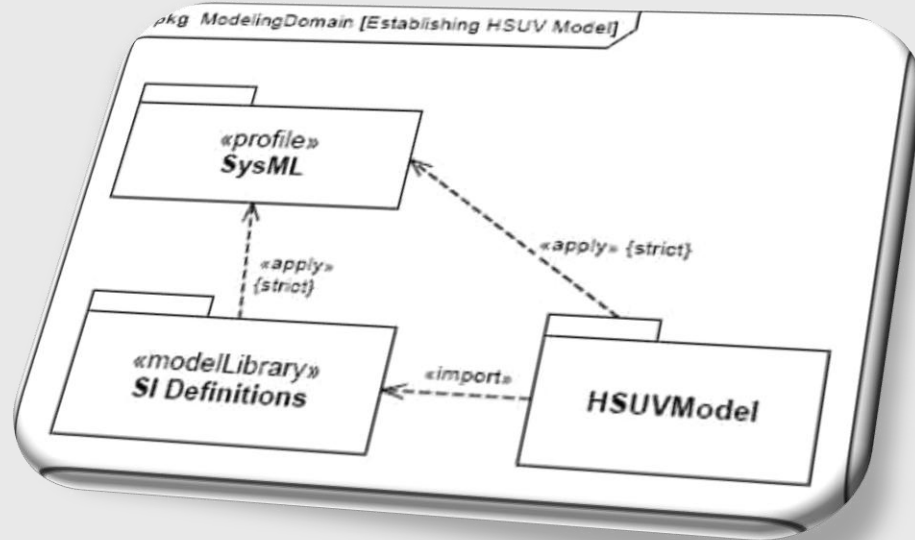
MODIFICATIONS :

- **Diagramme de collaboration** : devient Diagramme de **communication**
- **Diagramme de collaboration** : Ajout de chronologie sur les messages (via numérotation)
- **Diagramme de classe** : Abandon multiplicités discontinues (ex : [2,4])
- **Diagramme d'états-transitions** : Réunion des termes Action et Activités, sous une même représentation : « do » (non vu)
- **Diagramme de composants** : Utilisation d'interfaces (vu)
- **Diagramme de composants, déploiement** : nouvelle notation :
- **Diagramme de séquence** : Ajout de cadre(bloc) d'interaction (vu)

- **Diagramme d'activité** : Ajout notations de temps, signal et pin (vu)



UML : DIAGRAMME DE PAQUETAGE

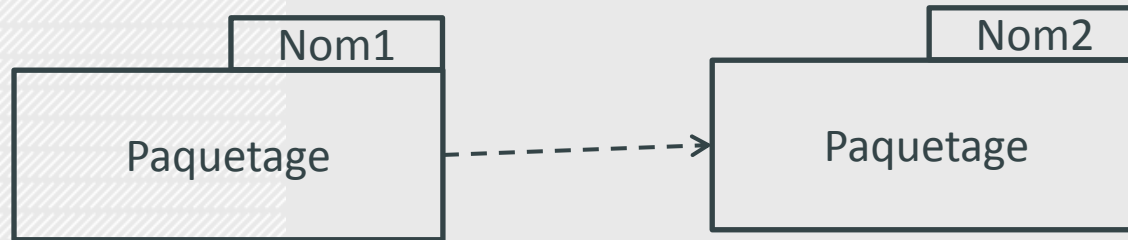


UML : DIAGRAMME DE PAQUETAGE

DEFINITION :

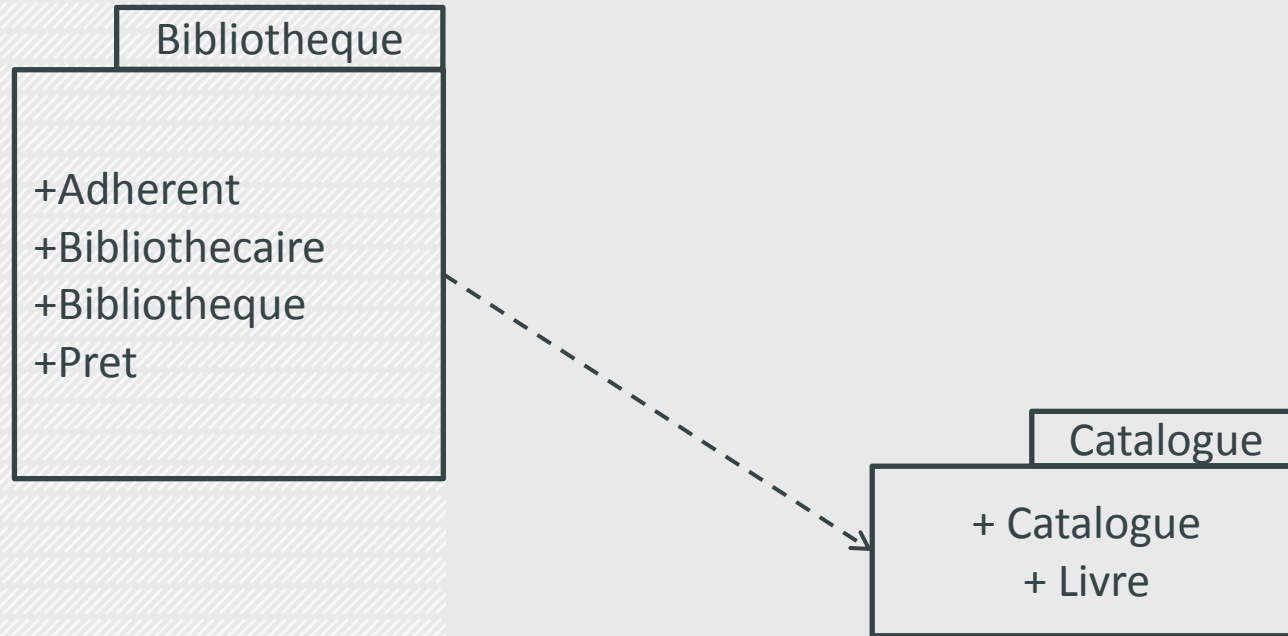
- Représenter la structure hiérarchique et modulaire
- **Permet de regrouper des éléments de modélisation**
- Possibilité de paqueter n'importe quel élément d'UML

NOTATION :

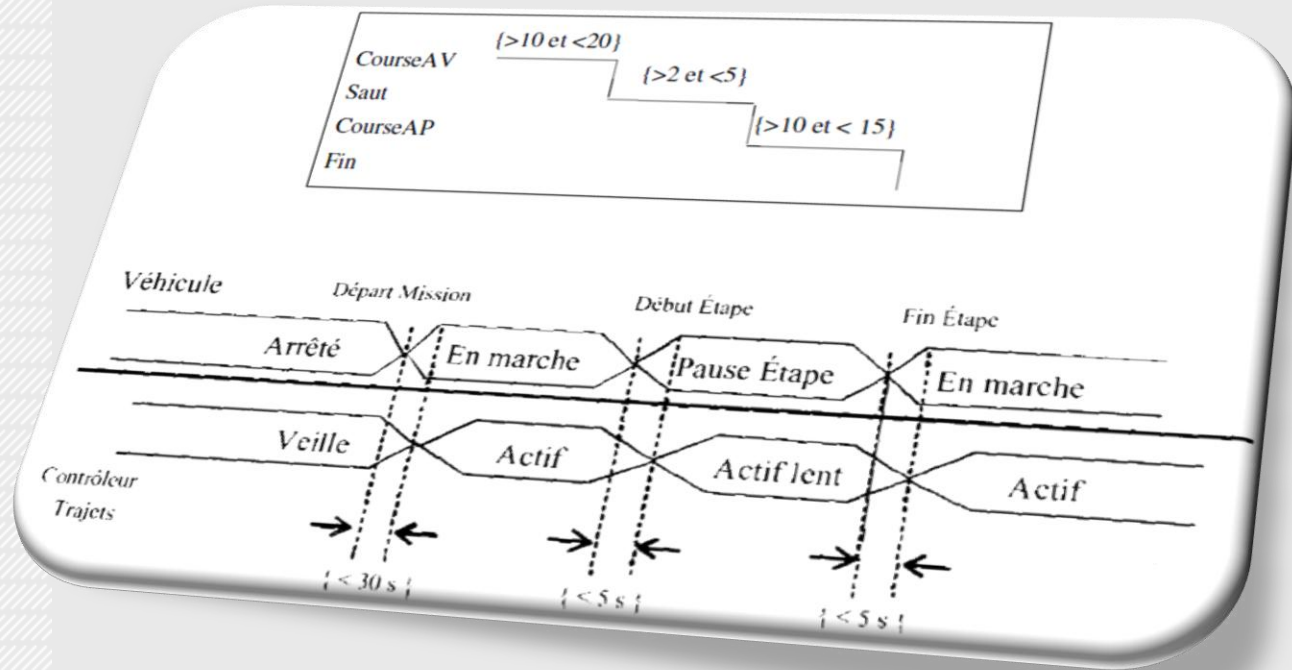


UML : DIAGRAMME DE PAQUETAGE

EXEMPLE :



UML : DIAGRAMME DE TEMPS



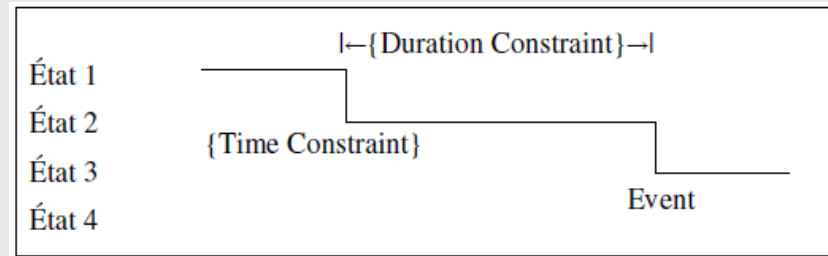
UML : DIAGRAMME DE PAQUETAGE

DEFINITION :

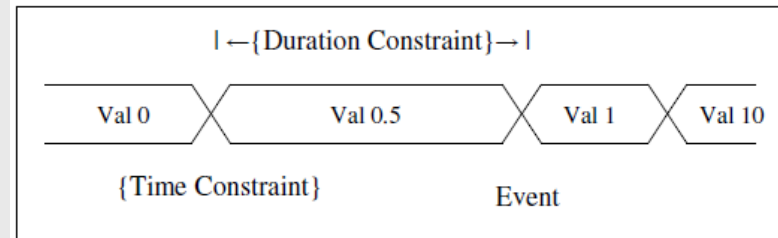
- Représenter l'interaction entre objets :
Accent mis sur le temps

NOTATIONS :

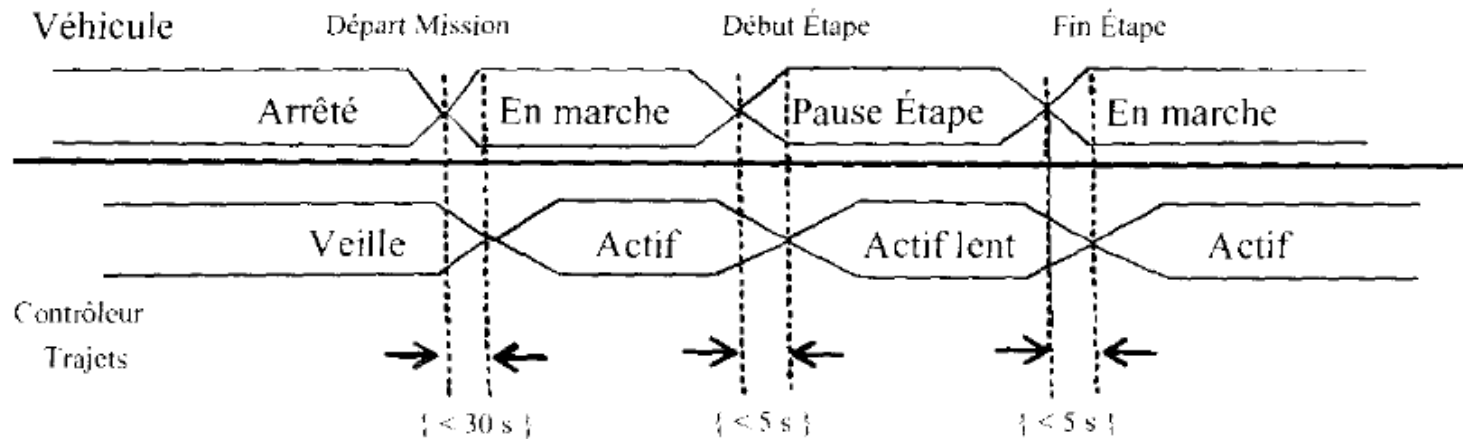
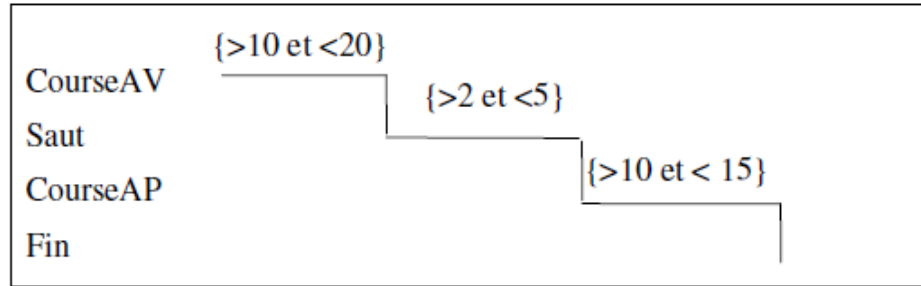
- Evolution de l'état de l'élément



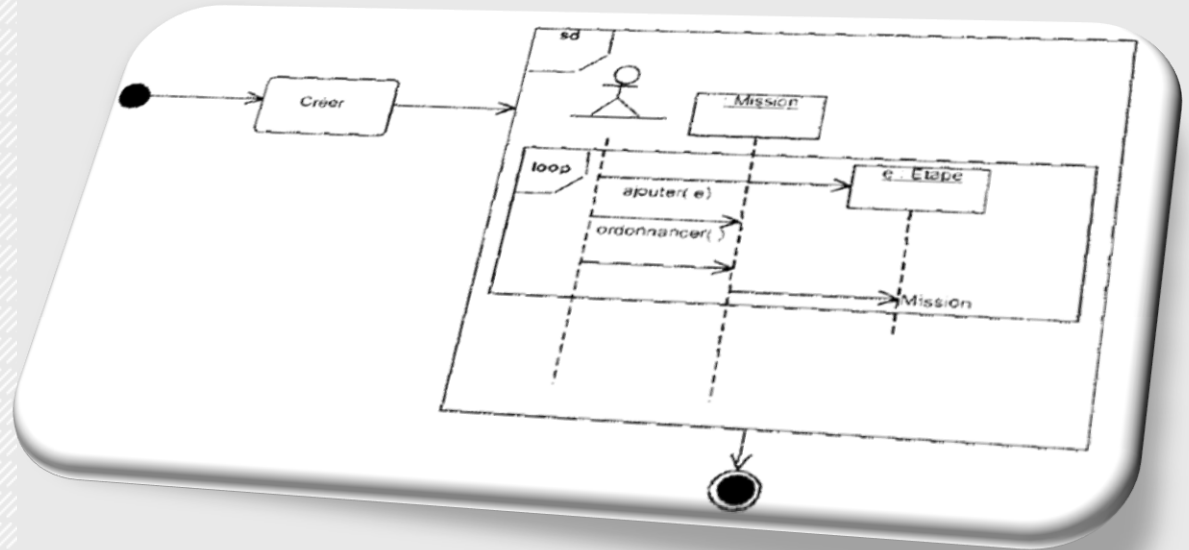
- Evolution de la valeur de l'élément



UML : DIAGRAMME DE PAQUETAGE



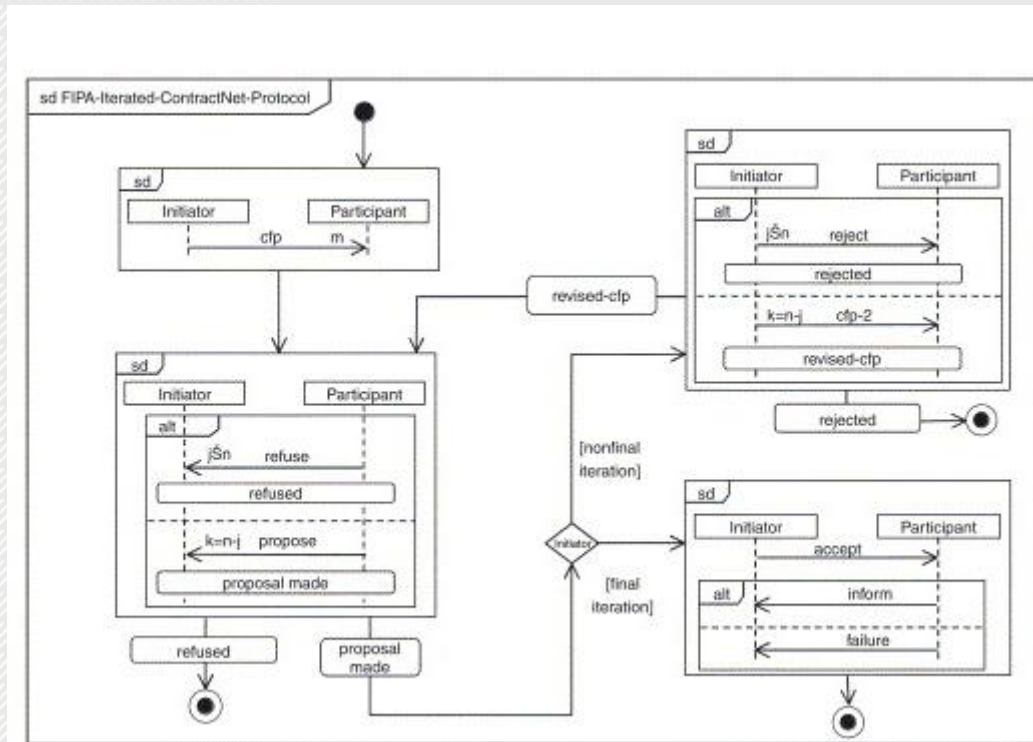
UML : DIAGRAMME GLOBAL D'INTERACTIONS



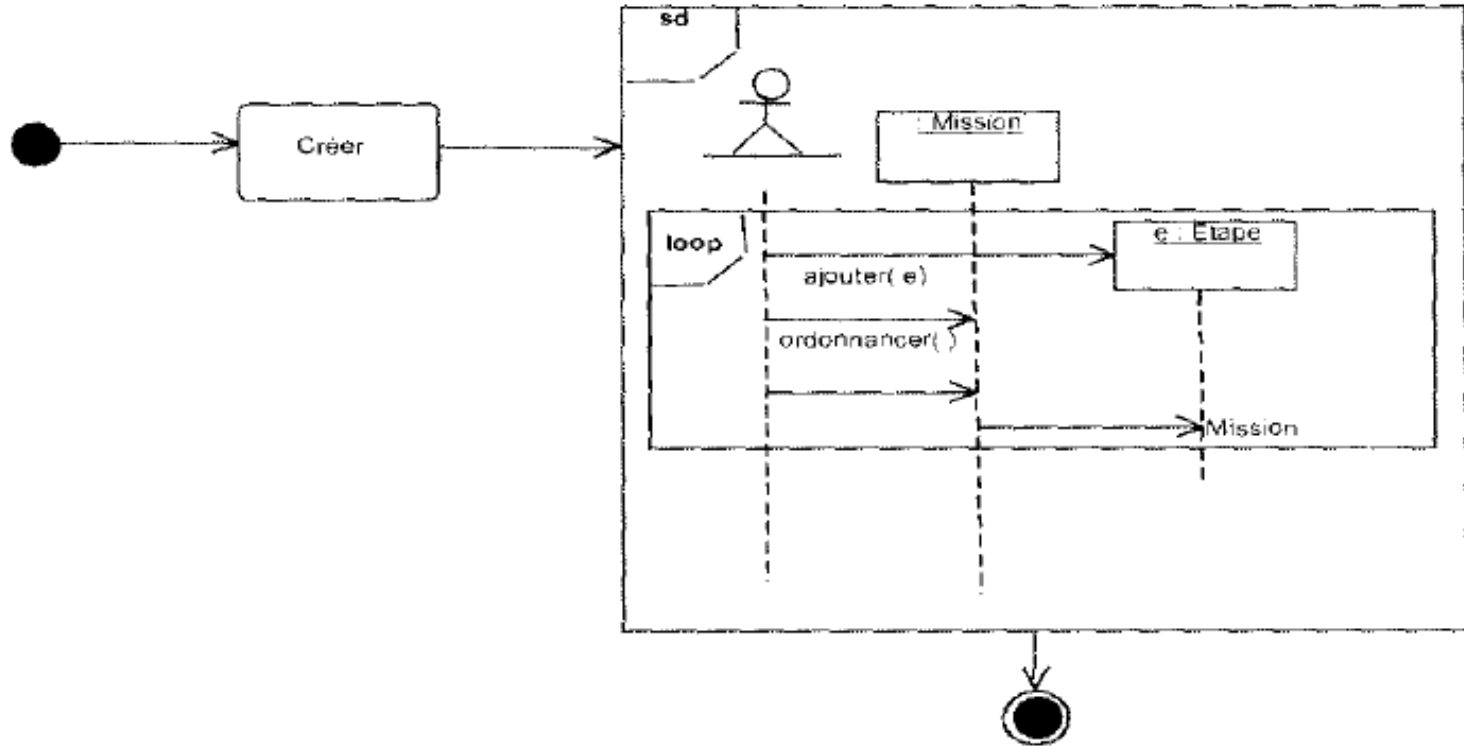
UML : DIAGRAMME GLOBAL D'INTERACTIONS

DEFINITION :

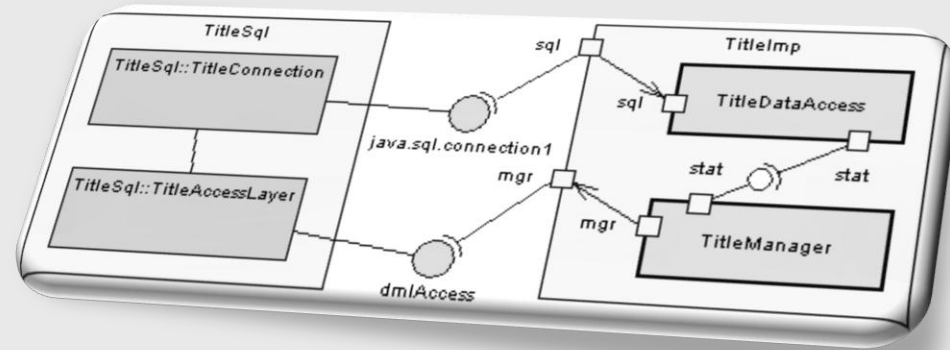
- Mélange de Diagramme d'activité et Diagramme de séquence



UML : DIAGRAMME GLOBAL D'INTERACTIONS



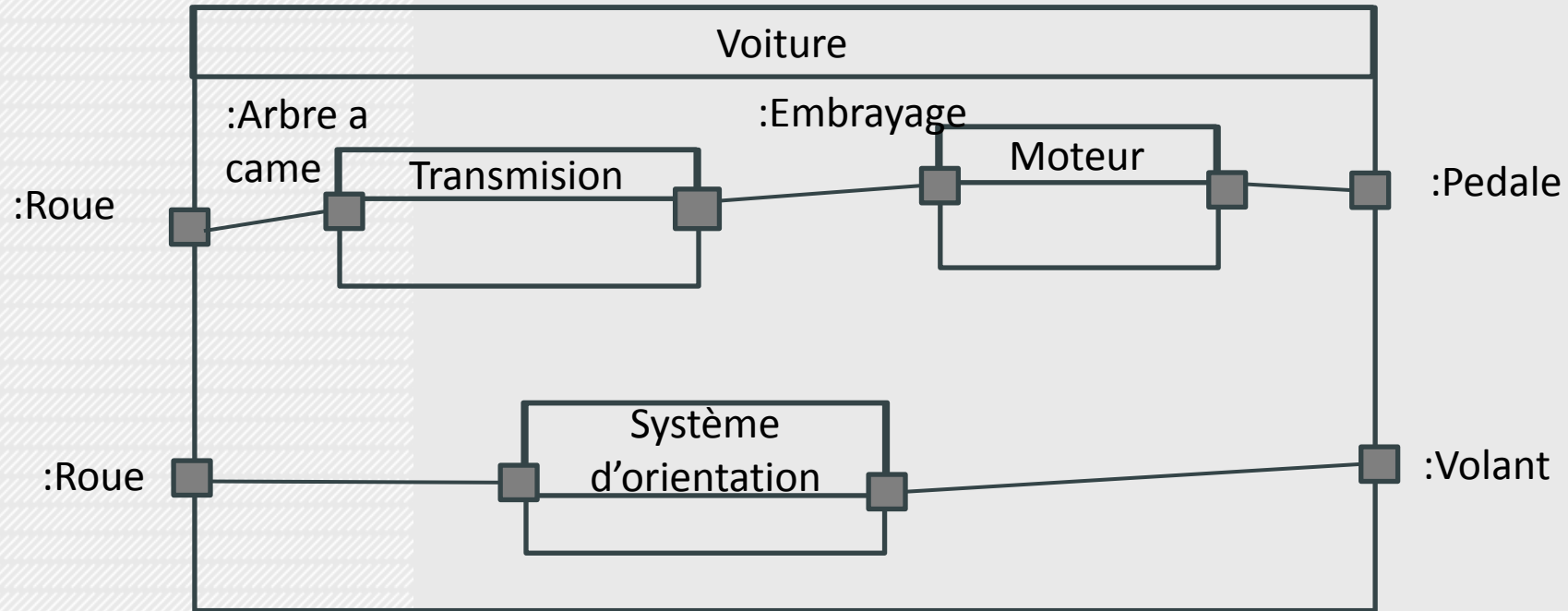
UML : DIAGRAMME DE STRUCTURE COMPOSITE



UML : DIAGRAMME DE STRUCTURE COMPOSITE

DEFINITION :

- Décrire la composition d'un objet complexe au moment de son exécution



UML : TABLEAU RECAPITULATIF



UML : DIAGRAMME GLOBAL D'INTERACTIONS

Cycle\Diagramme	Découverte des besoins	Analyse	Conception
Classes		X	
Packages		X	
Objets		X	
Structure Composite		X	X
Cas d'utilisation	X		
Séquence	X		X
Collaboration	X	X	X

UML : DIAGRAMME GLOBAL D'INTERACTIONS

Cycle\Diagramme	Découverte des besoins	Analyse	Conception
Etats		X	
Temps			X
Activité	X	X	
Global Interaction	X	X	X
Composants			X
Déploiement			X