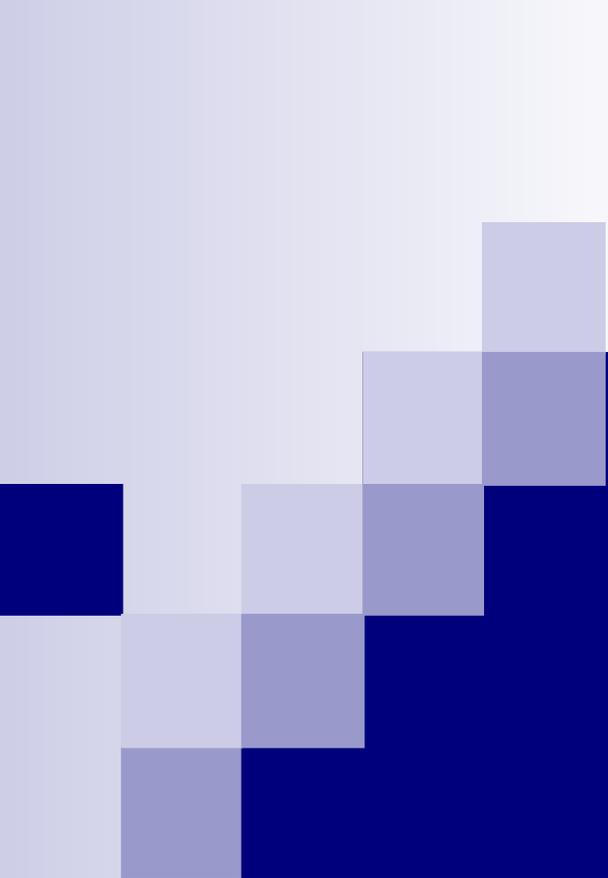


# Partie II : UML

# Plan

- Partie 2 : UML
  - 1 - Présentation d'UML
  - 2 - Les diagrammes de cas d'utilisation
  - 3 - Les diagrammes de classes et d'objets
  - 4 - Les diagrammes d'interaction
  - 5 - Les diagrammes de comportement
  - 6 - Les diagrammes physiques
  - 7 - Mise en oeuvre d'UML



## II.1. Présentation d'UML (*Unified Modeling Language*)

# Présentation d'UML

## ■ Historique

- ◆ Fin des années 80 : compétition des méthodes d'analyse et de conception OO

Booch : particulièrement adaptée au design et à l'implémentation

OOSE (Object Oriented Software Engineering, Jacobson) :  
expression des besoins

OMT-2 (Object Modelling Technique, Rumbaugh) : analyse et applications orientées-données

- ◆ 1994 : Rumbaugh rejoint Booch chez Rational
- ◆ 1995 : Jacobson rejoint Rational
- ◆ 14 novembre 1997 : UML adopté par l'OMG (Object Management Group)

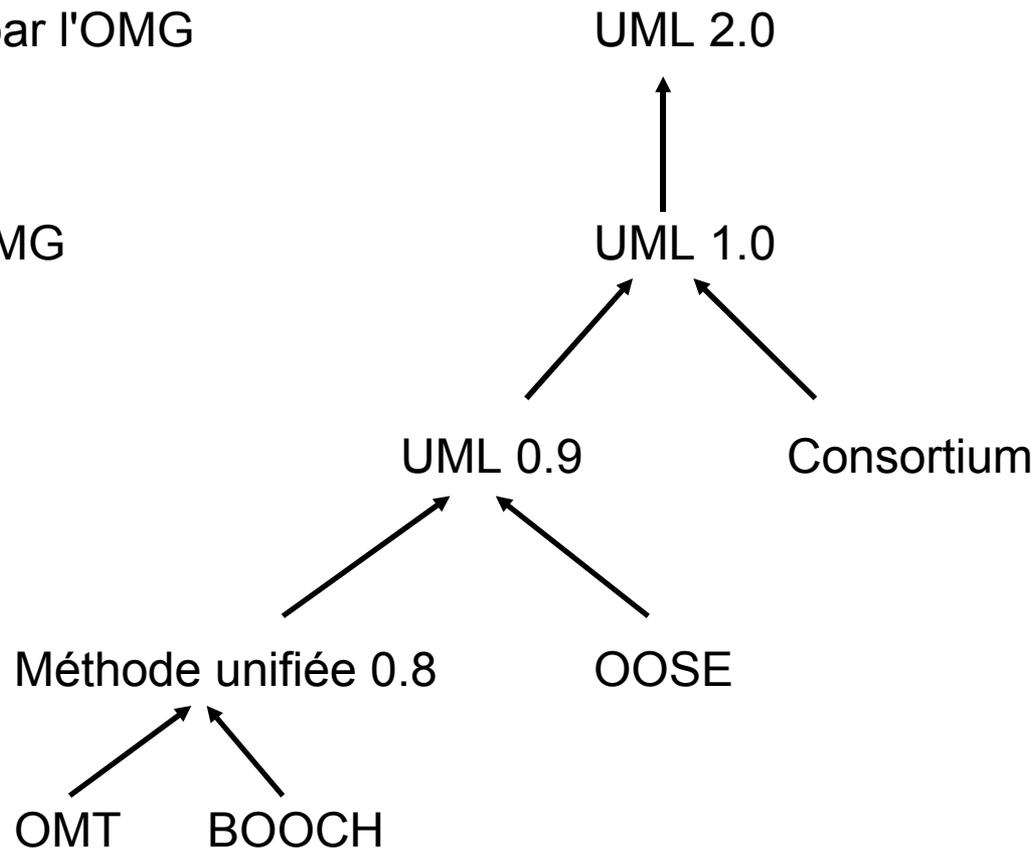
# Présentation d'UML

Standardisation par l'OMG  
depuis 2003

Soumission à l'OMG  
en Janvier 97

Version bêta  
en Juin 96

Draft en 95



NB : UML peut se substituer aux méthodes Booch, OMT et OOSE  
sans perte d'informations.

# Présentation d'UML

## ■ Qu'est-ce qu'UML

« UML est un langage pour visualiser, spécifier, concevoir et documenter les artefacts d'un système à base logicielle »

- 1 Langage : lexique (graphique), syntaxe (diagrammes), sémantique
- 2 Visualiser : représentation graphique
- 3 Spécification : précis, complet, non-ambigu
- 4 Construction : translation vers des langages de programmation
- 5 Documentation : des besoins aux tests

# Présentation d'UML

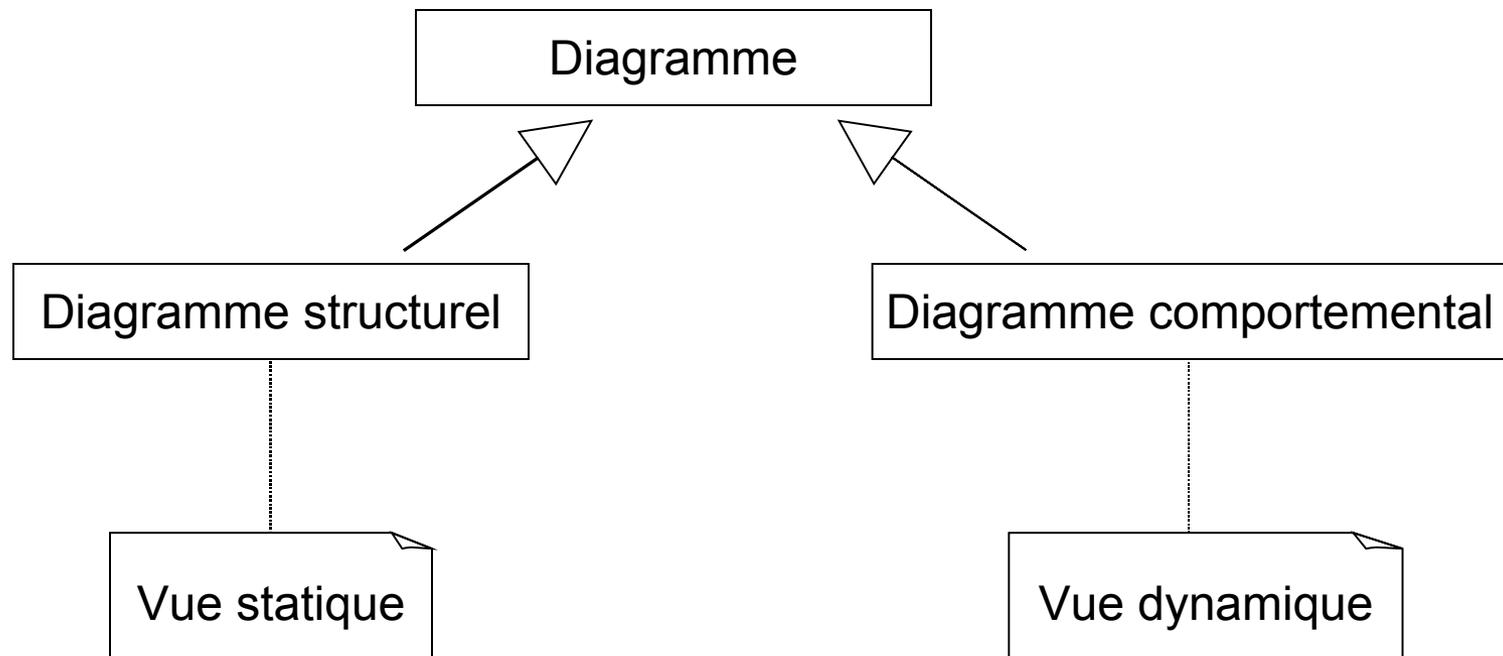
- Qu'est-ce que n'est pas UML

Ce n'est **pas** une méthode objet

... c'est un langage de modélisation objet

# Présentation d'UML

- Les diagrammes d'UML 1/3



# Présentation d'UML

## ■ Les diagrammes d'UML 2/3

### ■ Diagramme structurel

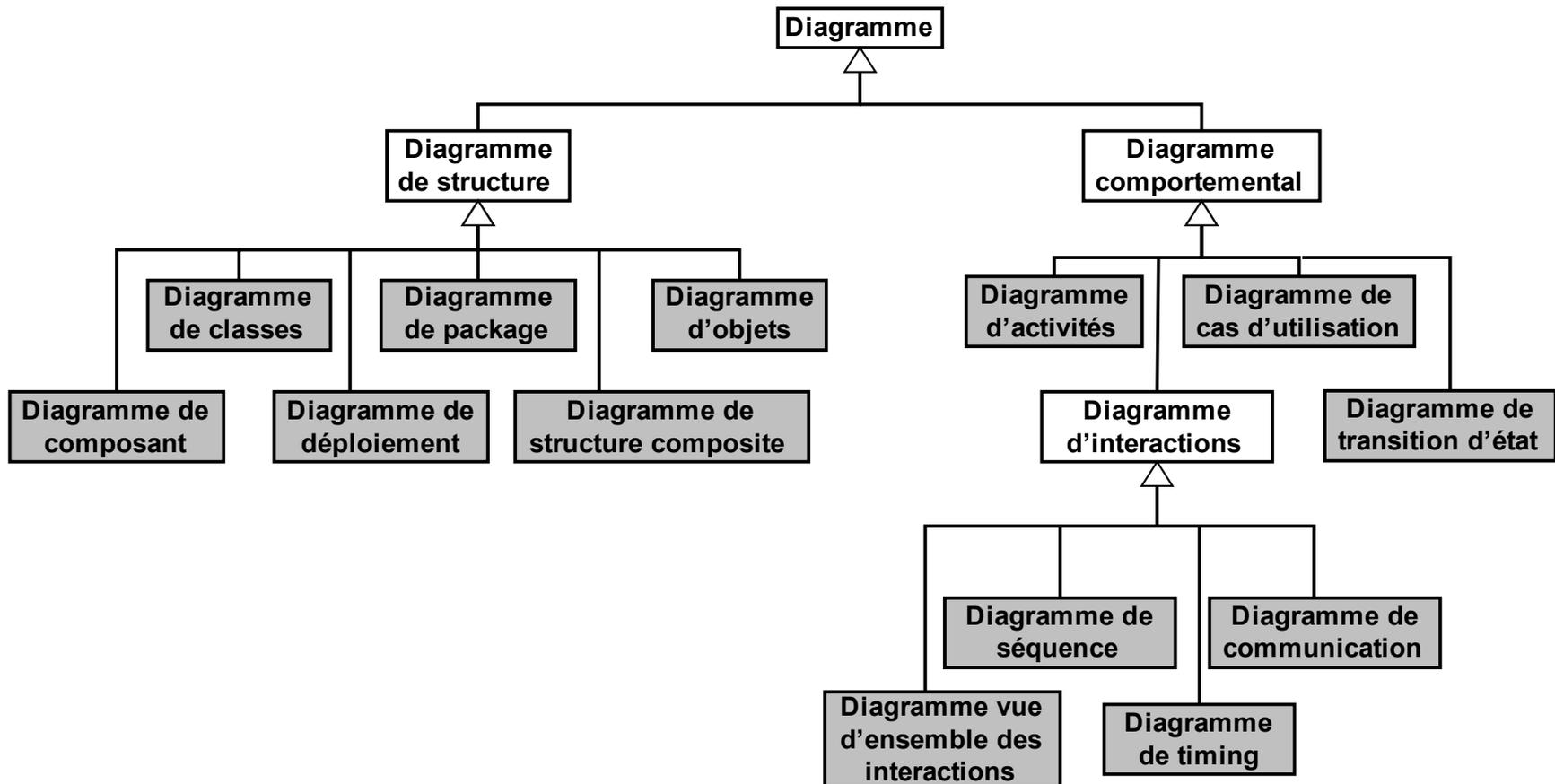
- 1 d'objets
- 2 de classes
- 3 de composants
- 4 de déploiement

### ■ Diagramme comportemental

- 1 de cas d'utilisation
- 2 de collaboration
- 3 de séquence
- 4 d'activités
- 5 d'états-transitions

# Présentation d'UML

## ■ Les diagrammes d'UML 3/3



# Présentation d'UML

## ■ La phase d'analyse

- ◆ Décrire les **cas d'utilisation**.
- ◆ Pour chaque cas d'utilisation, réaliser de un à  $n$  diagrammes d'interactions (les **diagrammes de séquence** en premier pour statuer sur les fonctionnalités avec le client ; puis, passer aux **diagrammes de collaboration** pour continuer l'analyse avec l'équipe projet).
- ◆ À chaque diagramme de collaboration correspond une ébauche de **diagramme de classes**. Préciser lors de la création d'une classe à quel package elle appartient.
- ◆ Faire la synthèse des diagrammes de classes pour un package donné.
- ◆ Pour chaque classe du diagramme de classes, faire un **diagramme d'états-transitions** (optionnel).

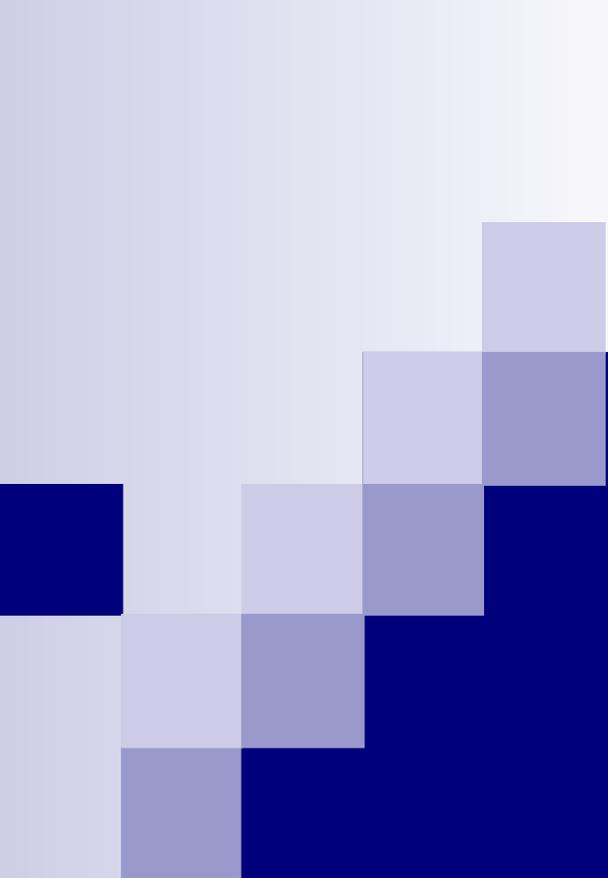
# Présentation d'UML

- Les apports de la modélisation visuelle
  - ◆ Meilleure appréhension des besoins
  - ◆ Facilite la compréhension du problème
  - ◆ Facilite la communication entre les personnes (client, experts du domaine, analystes, concepteurs, ...)
  - ◆ Support pour le raisonnement
  - ◆ Améliore la lisibilité des schémas de conception
  - ◆ Prépare la documentation et les programmes
  - ◆ Facilite la maintenance

# Présentation d'UML

## ■ Guides pour appliquer la méthode

- ◆ Recueillir les besoins de l'utilisateur final
- ◆ Adopter le point de vue de l'utilisateur final
- ◆ Penser à la réutilisation
- ◆ Ne préciser que les caractéristiques utiles des classes
- ◆ Généralement dans le cahier des charges, les noms sont des classes ou des attributs de classes et les verbes sont des méthodes
- ◆ Raffiner la modélisation en éliminant les redondances dues aux synonymes, les informations dérivées qui peuvent être déduites, et en s'efforçant de ne pas introduire de détails d'implémentation



## II.2. Diagrammes de cas d'utilisation

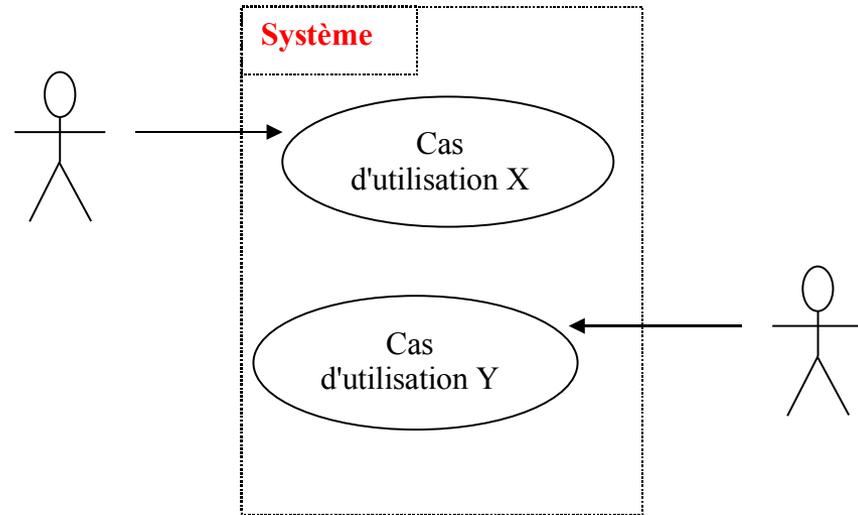
# Diagrammes de cas d'utilisation

- Définition et caractéristiques
  - ◆ Ils se limitent aux préoccupations réelles des utilisateurs
  - ◆ Ils identifient les utilisateurs du système et leur interaction avec le système
  - ◆ Donne une vue externe du comportement du système
  - ◆ Décomposition du système en termes de cas d'utilisation et d'acteurs
  - ◆ Utile pour inventorier les fonctionnalités du système

# Diagrammes de cas d'utilisation

## ■ Exemple

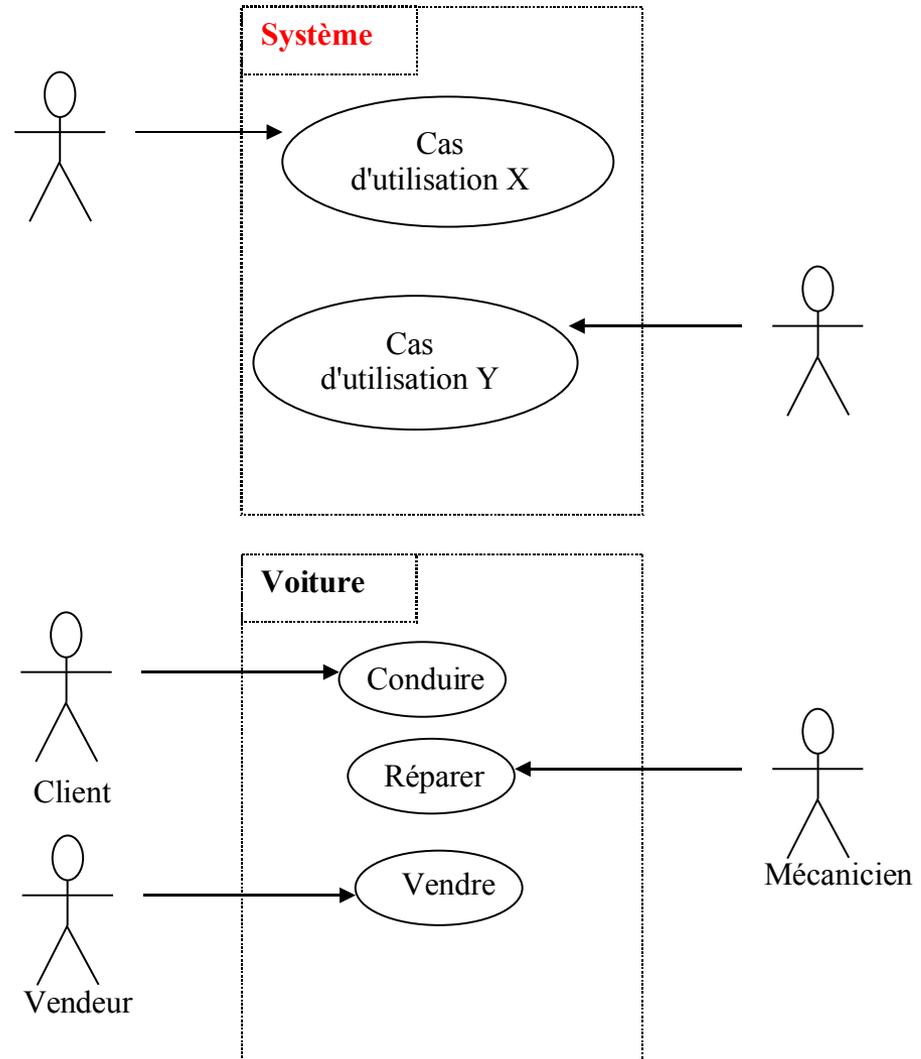
Les **cas d'utilisation** décrivent, sous la forme d'actions et de réactions, le comportement d'un système du point de vue de l'utilisateur, encore appelé **acteur**.



# Diagrammes de cas d'utilisation

## ■ Exemple

Les **cas d'utilisation** décrivent, sous la forme d'actions et de réactions, le comportement d'un système du point de vue de l'utilisateur, encore appelé **acteur**.



On recense, de la sorte, l'ensemble des fonctionnalités d'un système en examinant les besoins fonctionnels de chaque acteur.

# Diagrammes de cas d'utilisation

## ■ Objectif

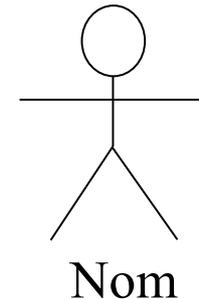
- ◆ Comprendre et structurer les besoins du client
    - Ne pas chercher l'exhaustivité mais clarifier, filtrer et organiser les besoins
    - Une fois identifiés et structurés, ces besoins :
      - Définissent le contour du système à modéliser
      - Permettent d'identifier les fonctionnalités principales du système
  - ◆ Permet une meilleure compréhension du système
  - ◆ Interface entre tous les acteurs du projet
- ➔ C'est le premier modèle à concevoir.
- ➔ Ce diagramme joue un rôle central, il est capital de bien le définir

# Diagrammes de cas d'utilisation

## ■ Éléments de base

### ■ Acteur

- 1 Ne fait pas partie du système
- 2 Quelqu'un ou quelque chose qui interagit avec le système
- 3 En relation avec au moins un cas d'utilisation
- 4 Un acteur peut jouer plusieurs rôles



### ■ Cas d'utilisation

- 1 Représente une fonctionnalité du système



# Diagrammes de cas d'utilisation

## ■ Éléments de base

### Il existe 2 grandes catégories d'acteurs :

- ◆ les **acteurs principaux**. Cette catégorie regroupe les personnes qui utilisent les fonctions principales du système. Dans le cas d'un distributeur de billets, il s'agit des clients.
- ◆ les **acteurs secondaires**. Cette catégorie regroupe les personnes qui effectuent des tâches administratives ou de maintenance. Dans le cas d'un distributeur de billets, il s'agit de la personne qui recharge la caisse du distributeur.

### On distingue aussi :

- ◆ le **matériel externe**. Cette catégorie regroupe les dispositifs matériels autres que les ordinateurs comme les périphériques. Dans le cas d'un distributeur de billets, il s'agit de l'imprimante, du lecteur de carte, de la trieuse de billets.
- ◆ les **autres systèmes**. Cette catégorie regroupe les systèmes avec lesquels le système interagit. Dans le cas d'un distributeur de billets, il s'agit du groupement bancaire qui gère l'ordinateur central qui relie tous les distributeurs.

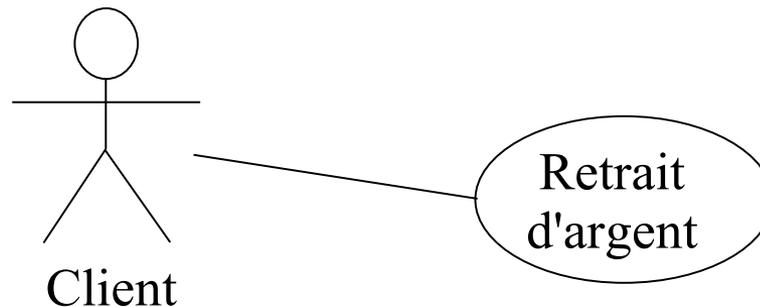
# Diagrammes de cas d'utilisation

- Les relations
  - Relation de **communication** avec le système
  - Relation d'**utilisation** entre cas
  - Relation d'**extension** entre cas

# Diagrammes de cas d'utilisation

## ■ Relation « Communique »

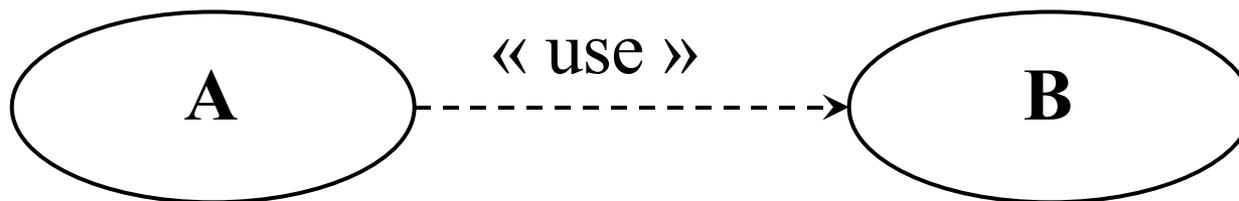
- 1 Relation non directionnelle entre un acteur et un cas d'utilisation
- 2 Exprime les fonctionnalités primaires du système



# Diagrammes de cas d'utilisation

## ■ Relation « Utilise »

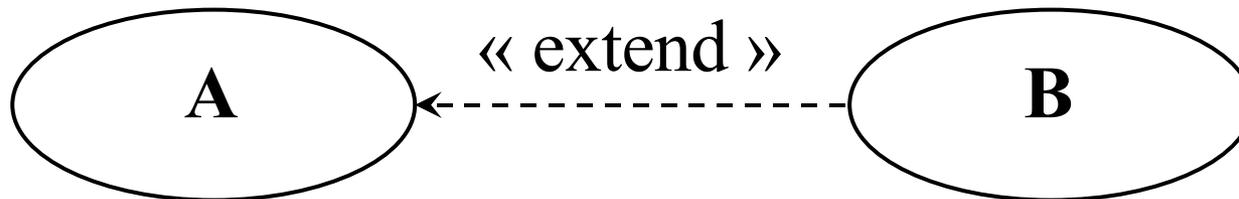
- 1 Le comportement de B est inclus dans le comportement de A
- 2 La fonctionnalité B est **nécessaire** pour réaliser la fonctionnalité A
- 3 Permet d'exprimer une fonctionnalité commune à plusieurs fonctionnalités (factorisation)



# Diagrammes de cas d'utilisation

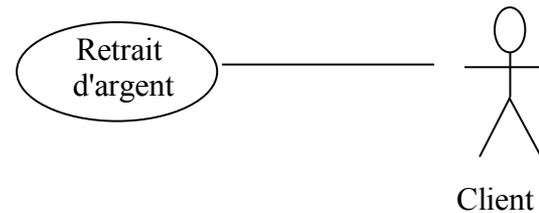
## ■ Relation « Étend »

- 1 Équivalent à l'héritage entre classes
- 2 B hérite du comportement de A, et le spécialise
- 3 B hérite des relations de communication décrites pour A



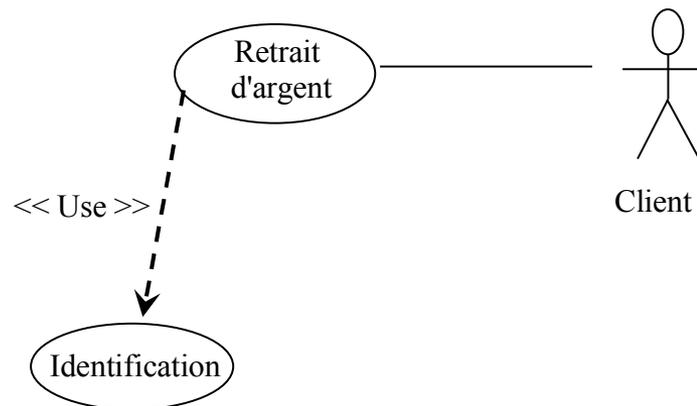
# Diagrammes de cas d'utilisation

## ■ Exemple



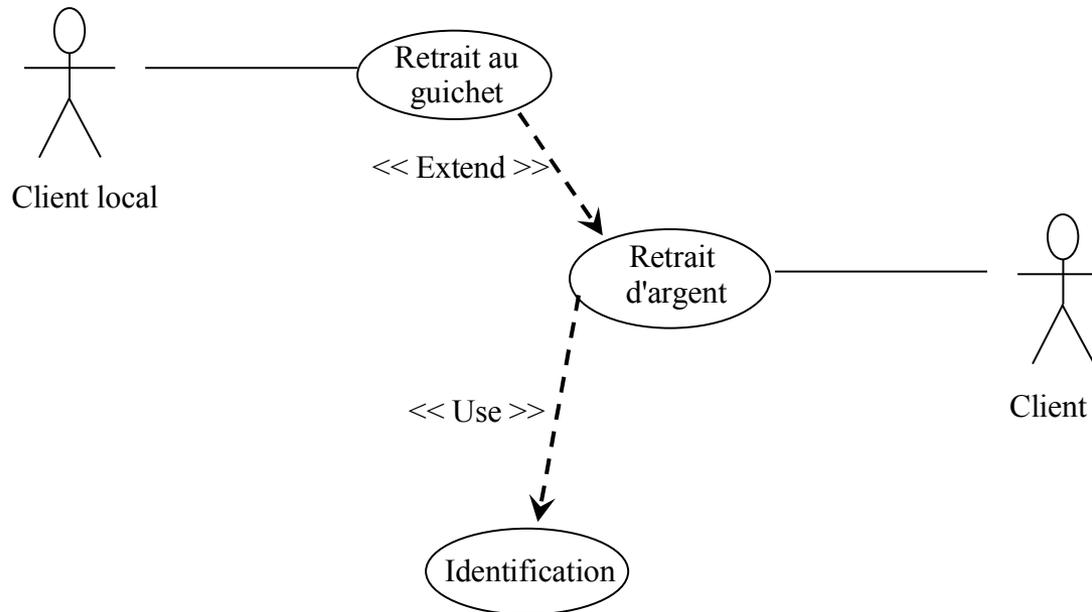
# Diagrammes de cas d'utilisation

## ■ Exemple



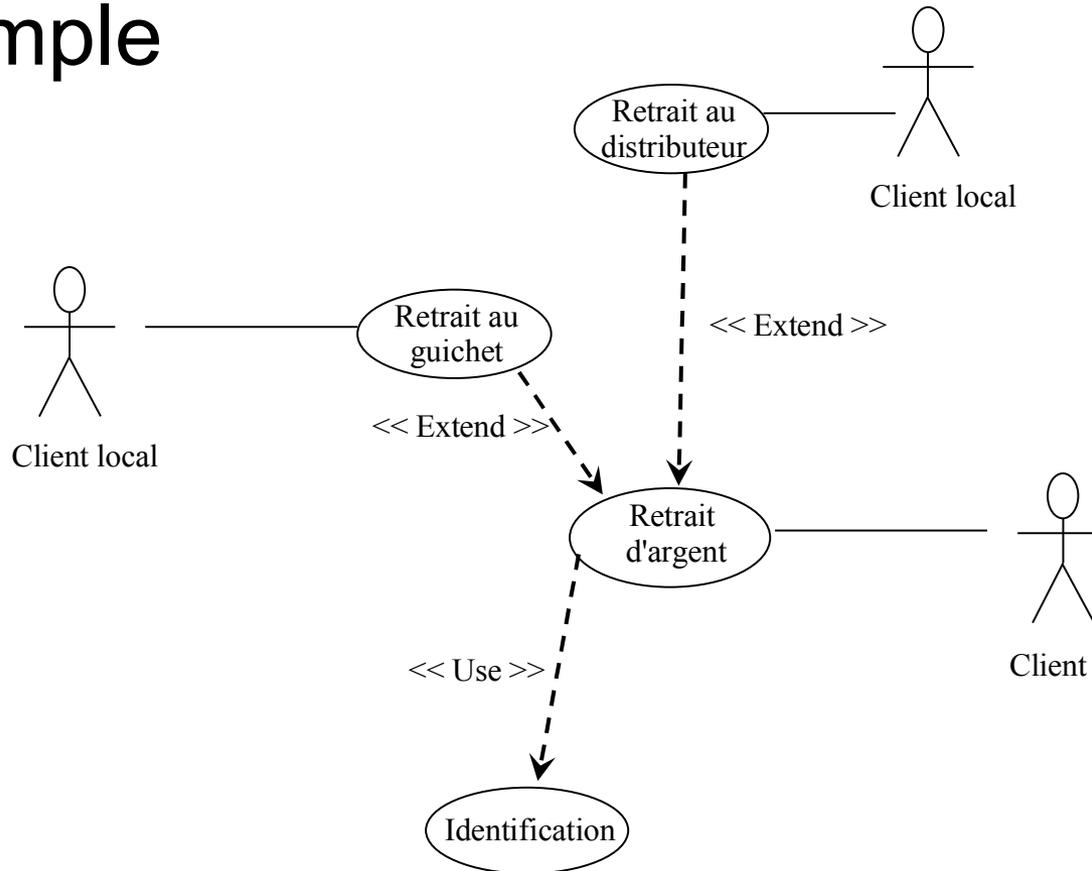
# Diagrammes de cas d'utilisation

## ■ Exemple



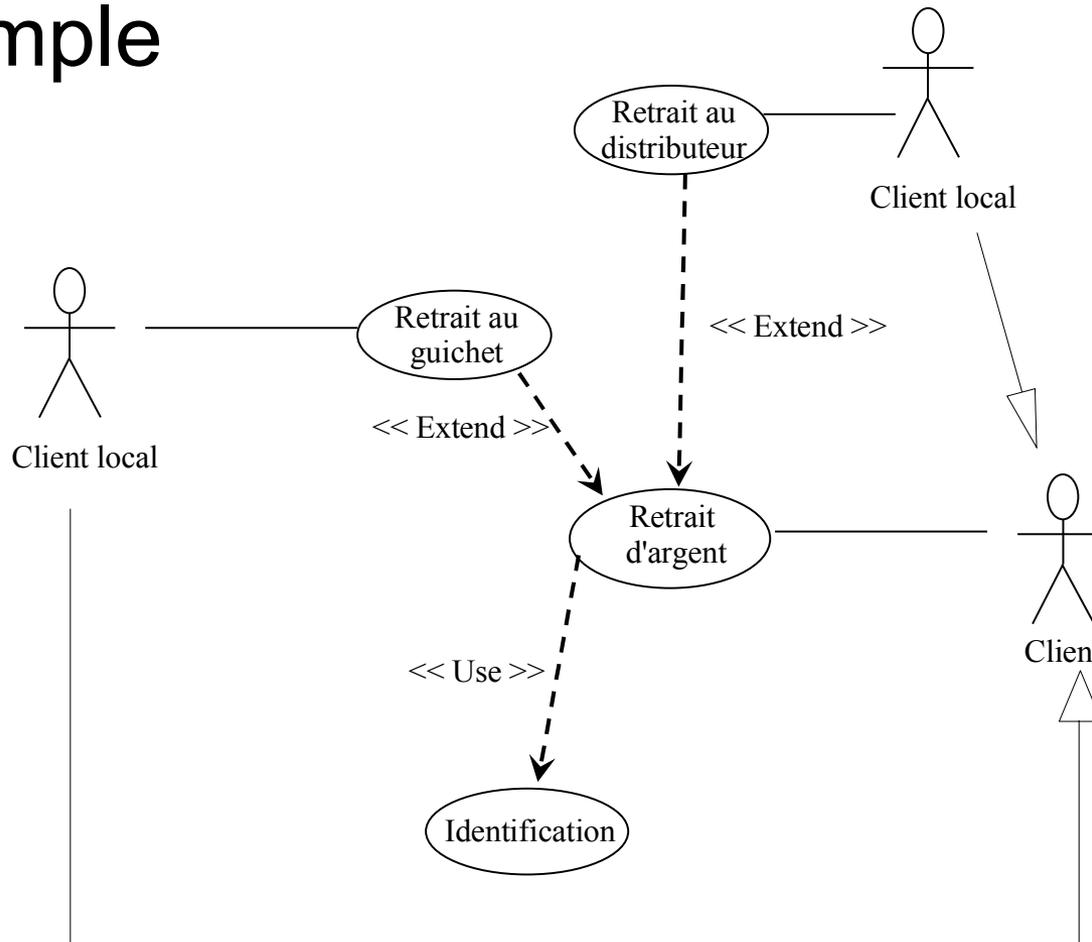
# Diagrammes de cas d'utilisation

## ■ Exemple



# Diagrammes de cas d'utilisation

## ■ Exemple



# Diagrammes de cas d'utilisation

## ■ Elaboration des cas d'utilisation

### Quand ?

Utilisation en phase amont du projet

Modélisation du système

Test et validation

### Comment ?

Un cas d'utilisation doit être simple

Nombre d'acteurs limités

# Diagrammes de cas d'utilisation

## ■ Démarche

- 1 Lister les acteurs
- 2 Déterminer le rôle de chaque acteur
- 3 Déterminer les cas d'utilisation
- 4 Identifier le flot d'évènements auxquels le système doit réagir
- 5 Structurer les cas
- 6 Finaliser un ou plusieurs diagrammes par package

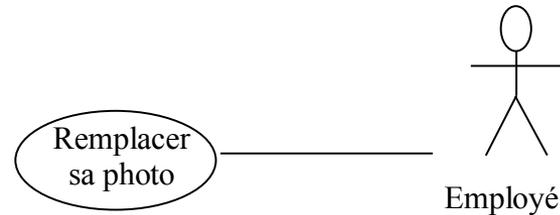
# Diagrammes de cas d'utilisation

## ■ Description d'un cas d'utilisation

### Plan type

- 1 Nom du cas
- 2 Description
- 3 Acteurs principaux et secondaires
- 4 Pré-conditions
- 5 Post-conditions
- 6 Scénario nominal
- 7 Scénario(s) alternatif(s)
- 8 Règles - Remarques

# Diagrammes de cas d'utilisation



- Cas d'utilisation : *Remplacer sa photo*

→ Un employé souhaite changer sa photo stockée dans l'annuaire de son organisation

- Acteur : Employé

- Pré-condition : L'employé est connu du système

- Post-condition : La nouvelle photo remplace l'ancienne

- Scénario nominal

1. Le système communique un formulaire d'identification
2. L'employé renseigne les champs et valide le tout
3. Le système transmet la photo de l'employé
4. L'employé sélectionne un autre photo qu'il transmet au système
5. Le système remplace l'ancienne photo par la nouvelle
6. Le système confirme le succès de l'opération

# Diagrammes de cas d'utilisation

- Extensions
  - 1 à 4, L'employé peut annuler l'opération :  
Le cas s'arrête
  - 3a, l'employé n'est pas connu du système :  
Retour en 1
  - 3b, l'employé n'a pas de photo :  
Le système transmet l'image d'un pingouin
  - 4, le système ne reconnaît pas le format de l'image transmise par l'employé :  
Le système en informe l'employé  
Retour en 3
- Question ouverte

Faut-il se préoccuper du poids de l'image transmise par l'employé ?  
(du point de vue « système de réception et système de stockage)

# Diagrammes de cas d'utilisation

## ■ Exercice : distributeur de boisson

### Caractéristiques principales du distributeur :

Plusieurs choix possibles de boissons : chaudes ou froides

Prix dépendant de la boisson choisie

Rend la monnaie

### Caractéristiques secondaires du distributeur :

Peut tomber en panne

Doit être entretenu

### Utilisation typique :

Introduire la monnaie

Faire un choix de boisson

Distribuer la boisson

Rendre la monnaie

### **Travail a effectuer**

1. Identifier les acteurs et le système
2. Créer un diagramme général
3. Créer un diagramme détaillé
4. Présenter un cas d'utilisation :  
*servir à boire*