

# Modélisation objet avec UML

C. BENSARI

- Introduction au génie logiciel
- Qualité d'un logiciel
- La modélisation
- Méthodes de modélisation
- La notion d'objet
- Concepts importants de l'approche objet
- La modélisation objet avec UML

# Introduction au génie logiciel

- Le Génie Logiciel (GL) est un domaine de recherche qui a pour objectif de répondre aux deux problèmes connus des logiciels :
  - la fiabilité
  - difficulté de réaliser des logiciels satisfaisant le cahier de charges **dans les délais**
- L'objectif du GL est d'optimiser le coût de développement du logiciel.
- Parmi les causes d'augmentation des coûts :
  - **Difficultés de maintenance**
  - Non fiabilité
  - Non respect des spécifications
  - Non respect des délais

# Introduction au génie logiciel

- Le GL s'intéresse aux différentes phases d'un projet, de la définition du besoin jusqu'à la phase de maintenance
- Les projets relatifs à l'ingénierie logicielle dépassent généralement les 10 000 lignes de code d'où le besoin de former une équipe structurée

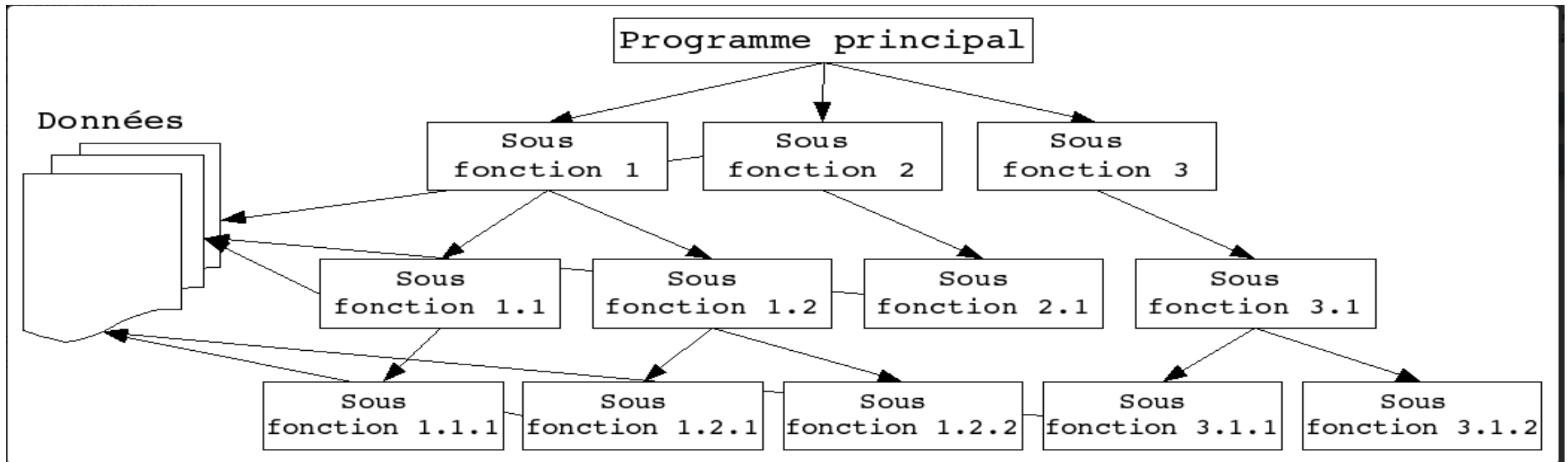
- Beaucoup de facteurs permettent de déterminer la qualité d'un logiciel :
  - **Validité**: respect des cahiers de charges
  - **Fiabilité (robustesse)**: fonctionnement dans des conditions anormales
  - **Extensibilité** : maintenable et évolutif
  - **Réutilisabilité** : réutiliser la totalité ou une partie d'un logiciel dans d'autres applications
  - **Compatibilité** : combinable avec d'autres logiciels

- **Efficacité** : meilleure gestion de ressources matérielles
- **Portabilité** : fonctionnel sous différents environnements matériels et logiciels
- **Vérifiabilité** : facilité de préparation des procédures de test
- **Intégrité** : protections des données, gestion des accès
- **Facilité d'emploi** : facile à apprendre, à utiliser, ..

Ces facteurs sont parfois contradictoires, le choix des compromis doit s'effectuer en fonction du contexte

- Modéliser un système avant sa réalisation permet de mieux comprendre son fonctionnement, maîtriser sa complexité et d'assurer sa cohérence
- Dans le domaine du bâtiment, les plans sont des modèles qui donnent une vue d'ensemble pour construire le bâtiment
- En développement informatique, la modélisation est souvent faite par l'équipe de maîtrise d'oeuvre (MOE)

- **Méthodes fonctionnelles ou structurées**
  - origine : langages procéduraux (impératifs)
  - mettre en évidence les fonctions à assurer
  - proposent une approche hiérarchique descendante et modulaire





- **Méthodes orientées objet**
  - Considère le logiciel comme un ensemble d'objets identifiés et possédant des caractéristiques
  - Une caractéristique est soit un attribut (donnée) caractérisant l'objet soit une entité comportementale de l'objet (une fonction)
  - La fonctionnalité du logiciel émerge alors de l'interaction entre les différents objets qui le constituent

- **Méthodes fonctionnelles vs orientées objet**
  - Dans l'approche fonctionnelle, l'architecture du système est dictée par la réponse au problème (la fonction du système)
  - Dans l'approche orientée objet, l'architecture est dictée par la structure du problème
  - L'approche orientée objet rapproche les données et leurs traitements associés au sein d'un unique objet

- L'approche orientée objet considère un logiciel comme une collection d'objets dissociés, **identifiés** et possédant des caractéristiques (**attributs**, **fonctions/méthodes**)
  - identité : permet de distinguer l'objet des autres objets (même principe qu'une primary key)
  - attribut : une variable stockant une information sur l'état d'un objet
  - methode : c'est l'action/opération réalisée par l'objet. Ces opérations permettent de faire réagir l'objet aux sollicitations extérieures (autres objets)

# Concepts importants de l'approche objet

- **Notion de classe**

- Une classe est un type abstrait de données (équivalent à un type enregistrement vu en algorithmique)
- Elle précise des caractéristiques (attributs et méthodes) communes à toute une famille d'objets
- Elle permet de créer (**instancier**) des objets

- **Notion d'objet**

- Un objet représente une instance d'une classe. On dira aussi qu'un objet d'identité « A » a comme type la classe l'ayant instanciée
- Les objets de la même classe partagent les mêmes caractéristique de la classe

- **Encapsulation**

- Consiste à masquer les détail d'implémentation d'un objet
- Elle définit une interface qui représente la vue externe de l'objet
- L'interface définit les services (fonctions ou attributs) accessibles aux utilisateurs de l'objet

# Concepts importants de l'approche objet

- **Héritage**

- Mécanisme de transmission des caractéristiques d'une classe vers une sous-classe
- Permet de construire des hiérarchies de classes
- Peut être simple ou multiple (C++)
- Evite les duplications

- **Spécialisation et généralisation**

- Une classe qui hérite d'une autre classe est dite une classe spécialisée. Elle permet d'y ajouter des caractéristiques spécifiques
- Des classes spécialisées peuvent être généralisées en une seule classe (classe mère)

# Concepts importants de l'approche objet

- **Polymorphisme**

- Littéralement : multi formes
- Capacité d'une méthode à pouvoir s'appliquer de différentes manières
  - **Overriding** : une classe qui redéfinit une méthode héritée d'une autre classe
  - **Overloading** : Une classe qui définit des méthodes ayant le même nom mais avec des arguments différents
- Il augmente la généricité et la qualité de code

- **Agrégation**

- Définie par une relation entre deux classes spécifiant qu'un objet d'une classe est un composant d'une autre classe
- Elle permet de construire des objets complexes à partir d'objets basiques

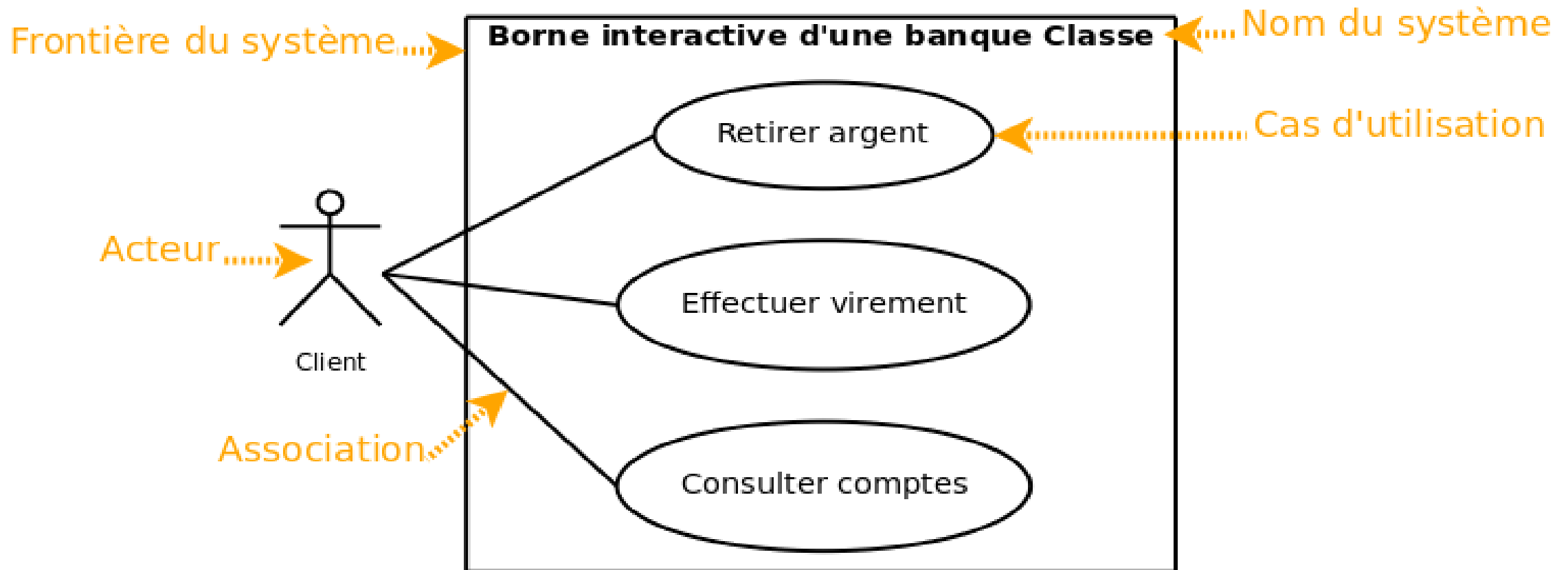
- La modélisation consiste à organiser, documenter des idées et puis organiser leur réalisation en définissant les modules et étapes de la réalisation d'un système (application)
- Le produit d'une modélisation est un modèle
- UML (Unified Modeling Language) est un langage de modélisation graphique (diagrammes) qui permet de représenter et communiquer les divers aspects d'un système d'information

- La version 2.0 de UML définit 13 types de diagrammes pour représenter des vues distinctes des aspects d'un système d'informations
- Les 13 diagrammes sont réparties sur deux catégories :
  - Diagrammes structurels : diagrammes de classes, d'objets, ..
  - Diagrammes comportementaux : diagrammes de cas d'utilisations, de séquences, ..

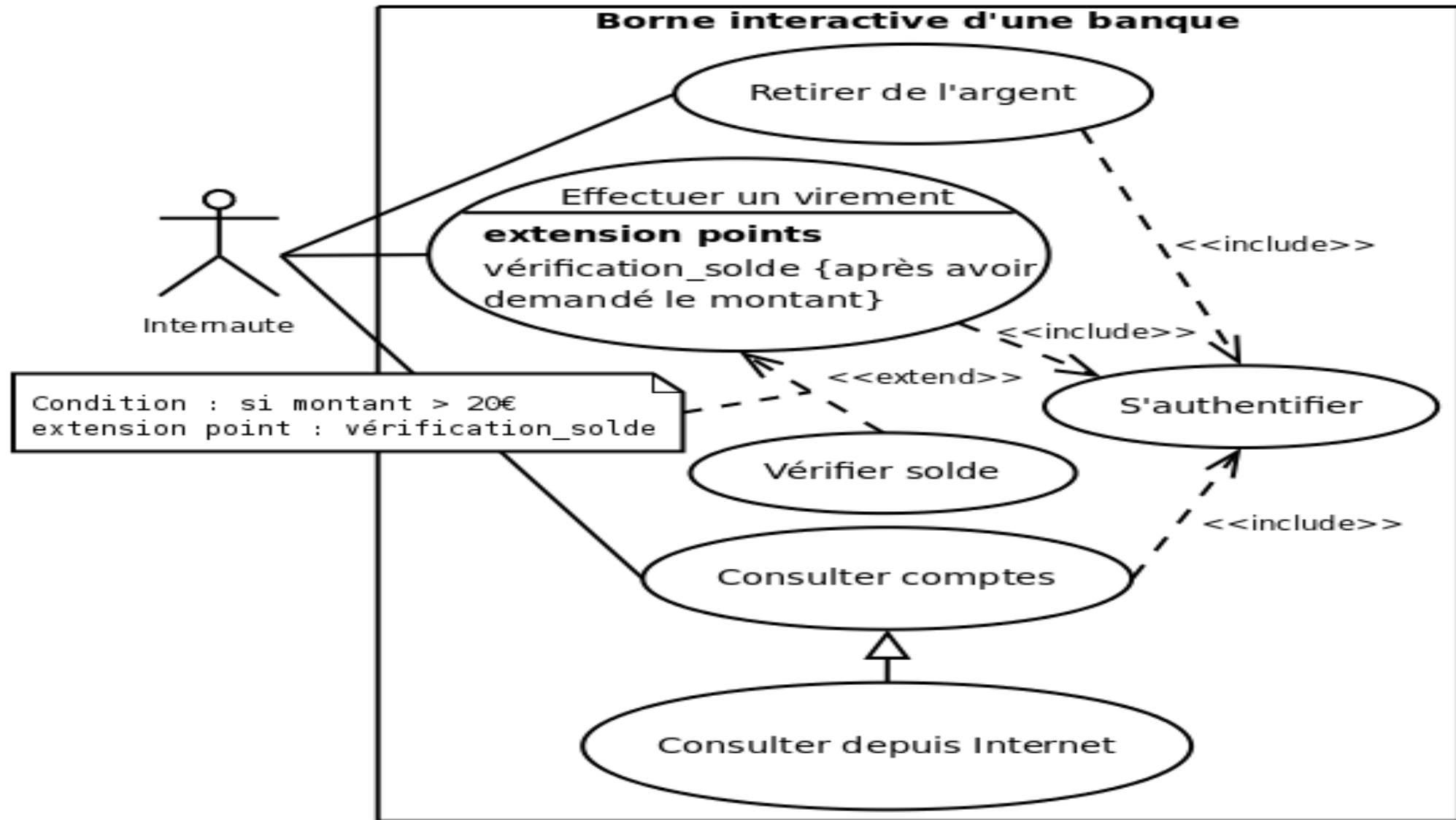


# Diagramme de cas d'utilisation

- C'est le premier diagramme UML à réaliser. Il représente les grandes fonctionnalités nécessaires aux utilisateurs du système

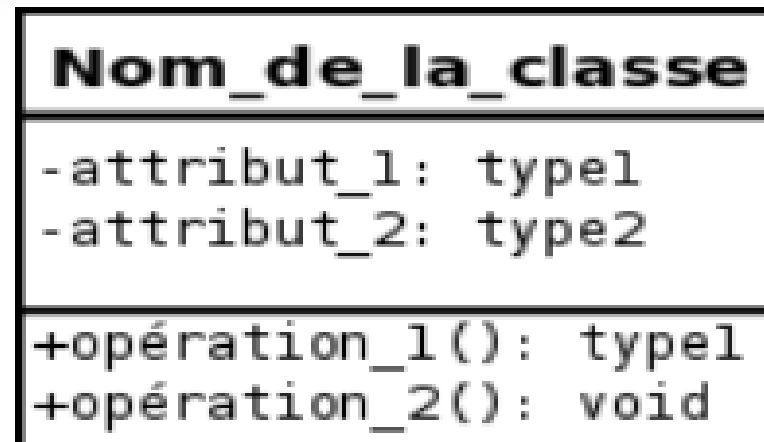


# Diagramme de cas d'utilisation

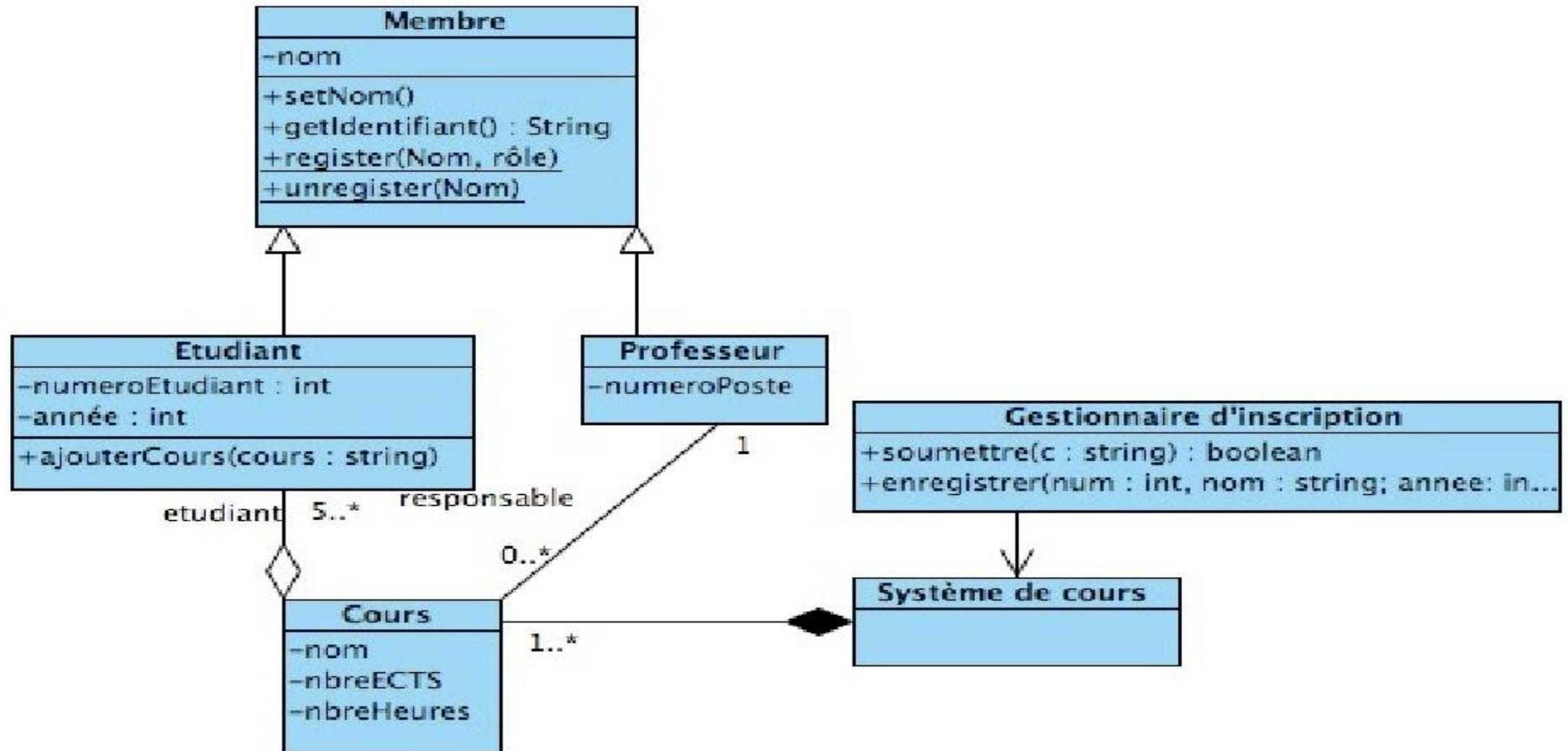


# Diagramme de classes

- C'est le diagramme le plus important dans un développement orienté objet
- Il représente l'architecture conceptuelle du système
- Il décrit les classes que le système utilise ainsi que leurs liens (héritage, agrégation, ..) avec les autres classes



# Diagramme de classes

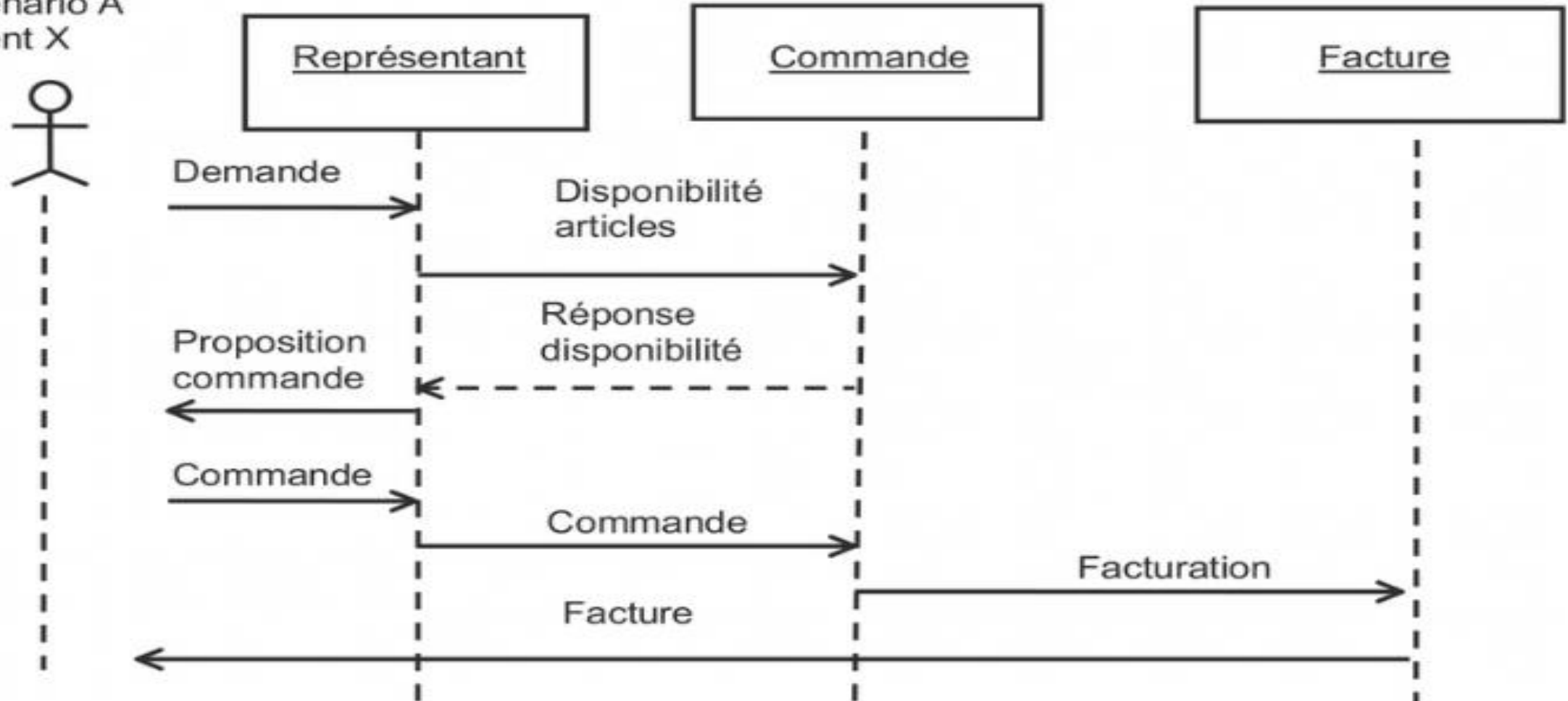


# Diagramme de séquences

- Représente la succession chronologiques des opérations réalisées par l'acteur
- Indique les objets que l'acteur va manipuler et les opérations qui font passer d'un objet à un autre

# Diagramme de séquences

Scénario A  
Client X



- Laurent AUDIBERT, UML2 de l'apprentissage à la pratique

<https://laurent-audibert.developpez.com/Cours-UML/?page=introduction-modelisation-objet>