

INF5153 – Génie logiciel : conception

UML

Jacques Berger

# Objectifs

Retour sur la notation UML

# Prérequis

INF5151

# UML

## Unified Modeling Language

Version actuelle : 2.0

# UML

Offre différents types de diagrammes pour illustrer des problèmes de différentes natures

# UML

UML est un outil de communication

Il est possible d'illustrer une conception très détaillée, mais il peut aussi être adéquat d'omettre certains détails

Tout dépend de ce qu'on cherche à communiquer

# Diagramme de classes

Applicable pour :

Modélisation du domaine

Représenter l'implémentation

Vue statique : ne change pas dans le temps

# Diagramme de classes

Composé de classes et leurs relations

Les types de relation :

Association

Généralisation

Dépendance

Réalisation

Utilisation

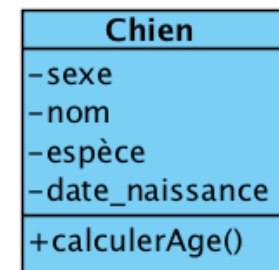


# Diagramme de classes

Une classe : la description d'un concept du domaine ou de la solution

Représentée par un rectangle avec un nom

Peut contenir une liste d'attributs et d'opérations



# Diagramme de classes

Les relations sont des lignes, avec différents motifs, entre les classes

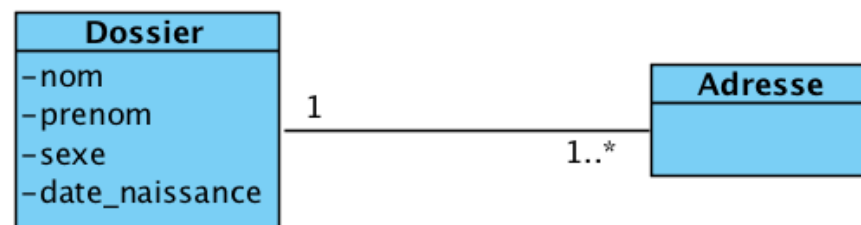
# Diagramme de classes

## Association

Relation logique entre instances d'objets

Vient souvent avec une multiplicité

La relation la plus commune

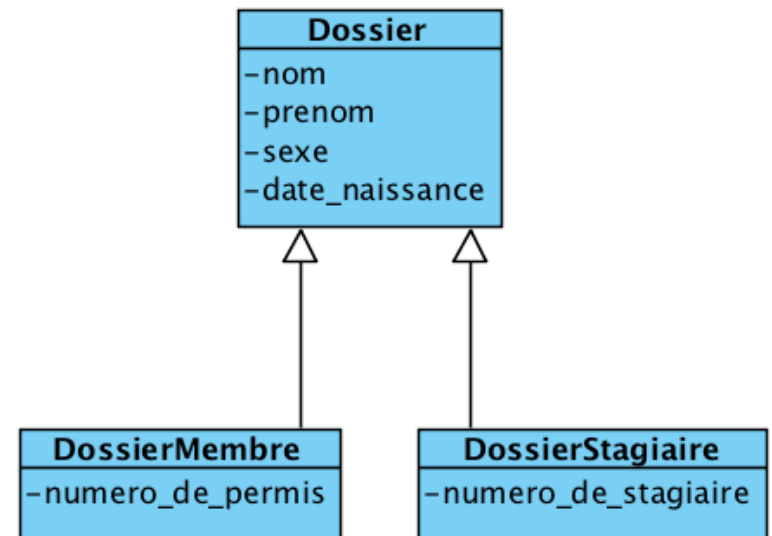


# Diagramme de classes

## Généralisation

Représente un lien d'héritage

La flèche pointe vers la classe de base

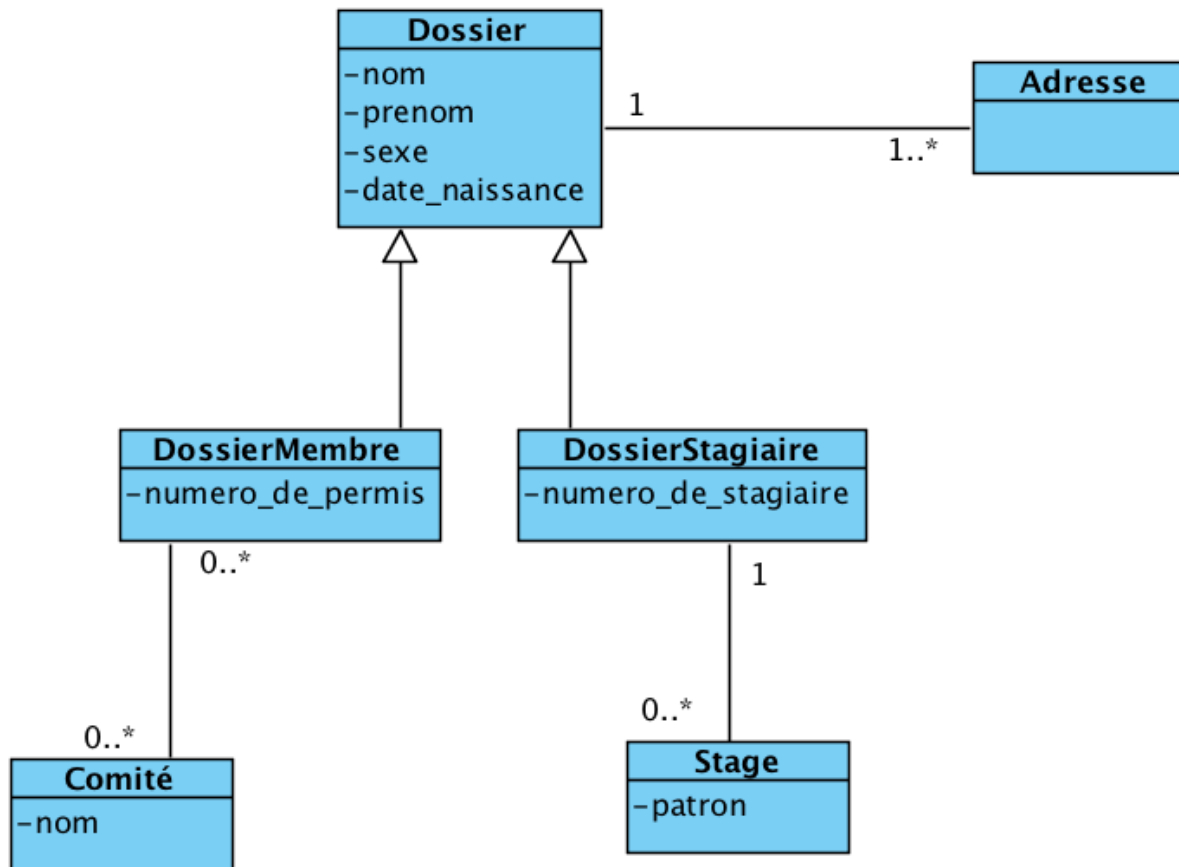


# Diagramme de classes

La réalisation illustre qu'une classe est l'implémentation d'une abstraction



# Diagramme de classes



# Diagramme d'états

Modélise les différents états qu'une instance d'un objet peut avoir et les transitions entre ces états

Chaque état est un moment dans la vie de l'objet

Lorsqu'un événement survient, un changement d'état peut survenir également

# Diagramme d'états

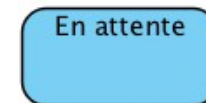
Le point noir représente le point de départ





# Diagramme d'états

Chaque état est représenté par une boîte aux coins arrondis



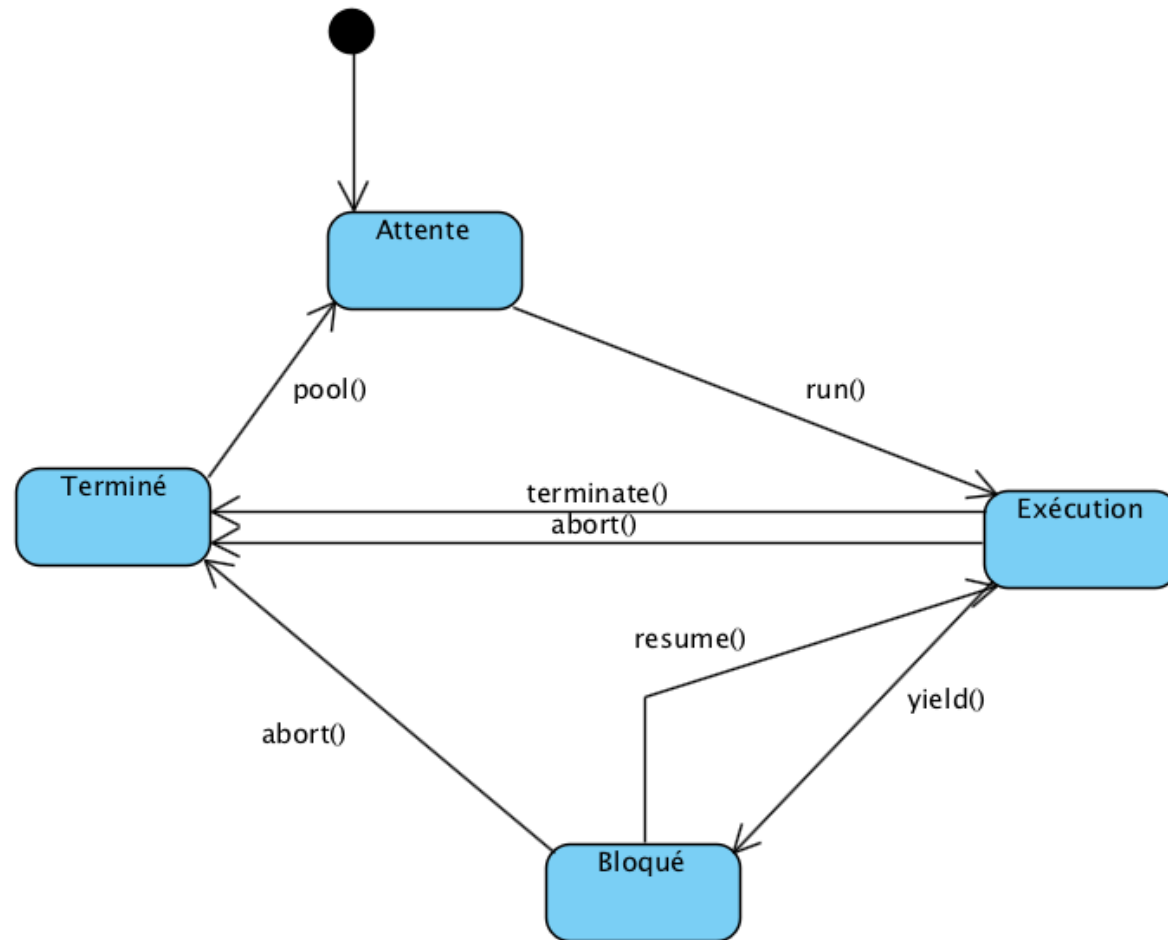
# Diagramme d'états

Une transition est représentée avec une flèche vers le nouvel état

La transition est souvent accompagnée de la méthode ou l'événement qui a provoqué ce changement



# Diagramme d'états



# Diagramme de séquence

Permet d'illustrer les relations entre les objets dans le temps

Présente une séquence de messages

Peut modéliser les étapes d'un processus ou les appels entre les classes

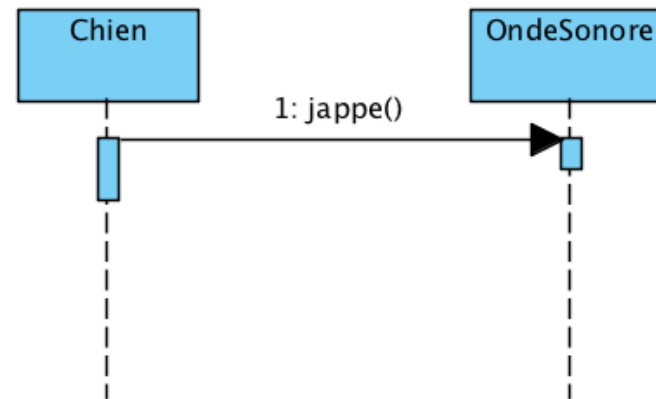
# Diagramme de séquence

Une classe est représentée avec un rectangle et une ligne pointillée en dessous



# Diagramme de séquence

Les messages entre les objets sont des flèches avec une description (nature du message ou méthode invoquée)



# Diagramme de séquence

Peut contenir des boucles, des conditions, de l'exécution parallèle, représenté avec un vaste rectangle

Voir exemple

# Diagramme de communication

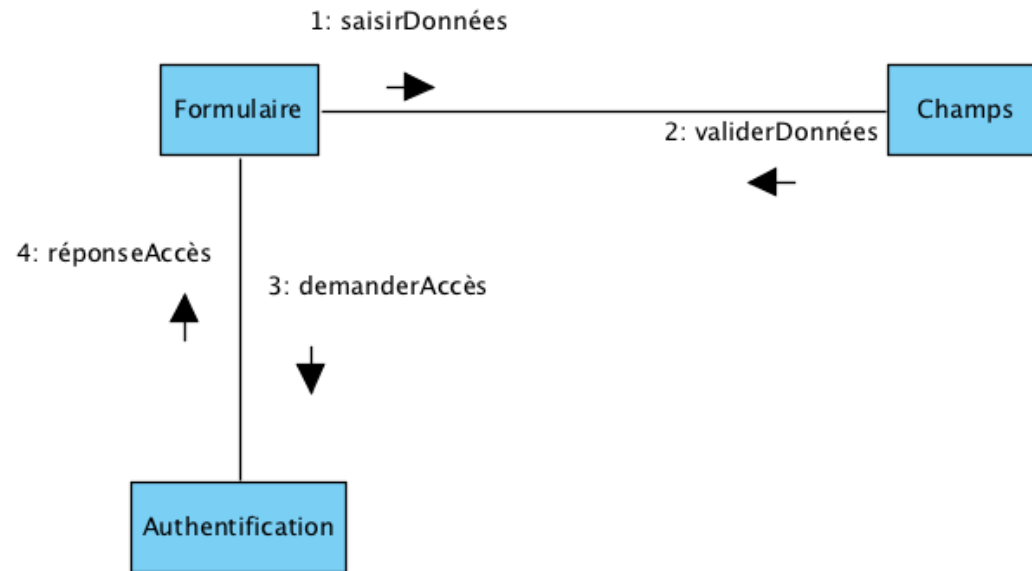
L'équivalent d'un diagramme de séquence avec une disposition différente

Illustre la communication entre les objets

Les opérations sont numérotées pour indiquer l'ordre d'exécution



# Diagramme de communication



# Diagramme de use case

Modélise les fonctionnalités du logiciel, tel que perçues par les éléments extérieurs

Les éléments extérieurs sont des acteurs

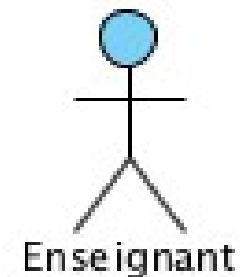
# Diagramme de use case

Un use case est une unité de fonctionnalité

Illustre la participation des acteurs pour un use case donné

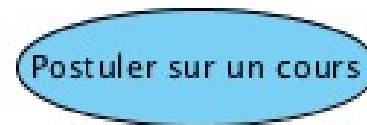
# Diagramme de use case

Un acteur est représenté par un bonhomme allumette



# Diagramme de use case

Le use case est représenté par un ovale

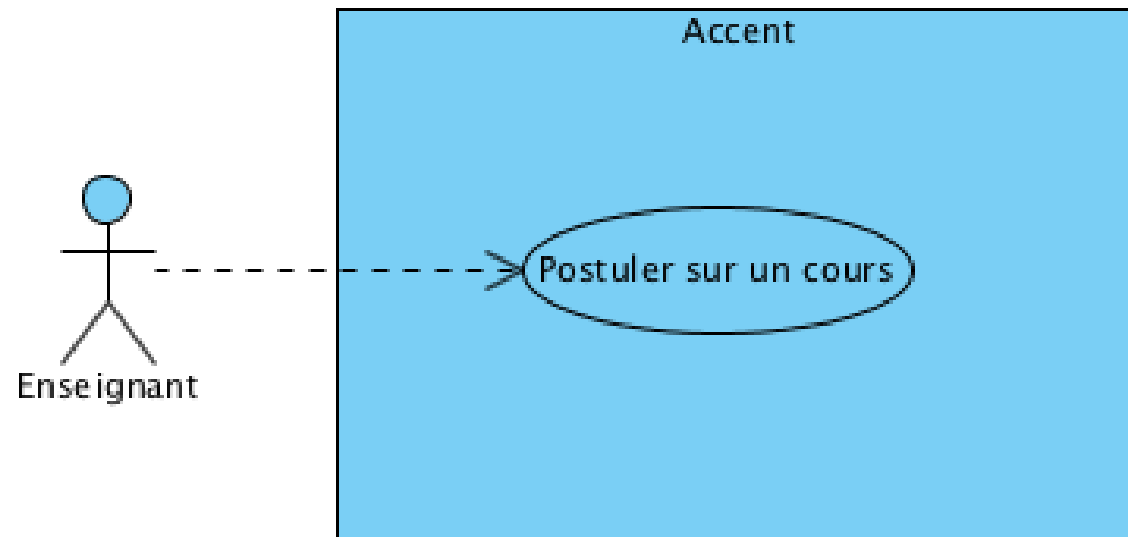


# Diagramme de use case

La participation d'un acteur est représentée par une flèche qui part de l'acteur jusqu'au use case

Le système est représenté par une grande boîte contenant les use cases

# Diagramme de use case



# Diagramme de use case

Voir un exemple



# Plus loin...

The Unified Modeling Language Reference Manual, Second Edition  
James Rumbaugh, Ivar Jacobson, Grady Booch  
Addison-Wesley, 2005