

# UML, Une introduction

## Présentation

Abdoulaye DIALLO  
Redoine EI OUASTI

abdoulaye.diallo@etud.univ-montp2.fr  
redoine.el-ouasti@etud.univ-montp2.fr  
**Université Montpellier II**

December 10, 2014

# Sommaire

## ① Introduction

- UML, C'est quoi?
- Un peu d'histoire
- Pourquoi modéliser?

## ② Les diagrammes

- Diagrammes statiques (UML Structure)
  - diagramme de classes (Class diagram)
  - diagramme d'objets (Object diagram)
- Diagrammes dynamiques (UML Behavior)
  - diagramme de cas d'utilisation (Use case diagram)
  - diagramme d'états-transitions (State machine diagram)
  - diagramme d'activités (Activity diagram)
  - diagramme de séquence (Sequence diagram)

## ③ Conclusion

# UML, C'est quoi?

## Definition

UML ( Unified Modeling Language), ou Langage de modélisation unifié, est un langage de modélisation graphique à base de pictogrammes.

Comme n'importe quel type de projet, un projet informatique nécessite une phase d'**analyse**, suivi d'une étape de **conception**, d'où le rôle d'UML.

# Pourquoi modéliser?

## Definition

Modéliser, c'est décrire de manière visuelle et graphique les besoins et, les solutions fonctionnelles et techniques de votre projet logiciel.

L'UML est avant tout un support de communication performant, qui facilite la représentation et la compréhension de solutions objet. L'UML est bien plus qu'un simple outil qui permet de dessiner des représentations mentales de projet... Il permet de parler un langage commun, normalisé mais accessible pour tous car il s'agit d'une représentation visuel.

# Un peu d'histoire

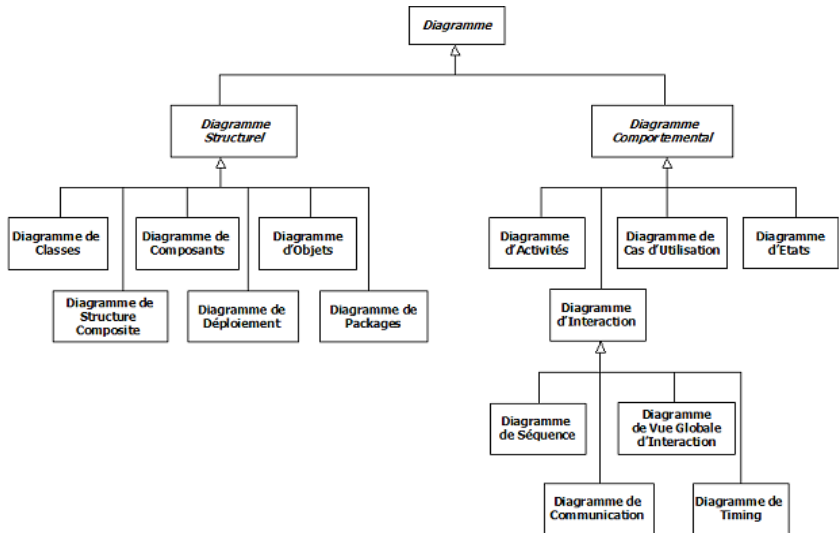
## Naissance

UML est né de la fusion des trois méthodes qui ont le plus influencé la modélisation objet au milieu des années 90 : OMT, Booch et OOSE. Principalement issu des travaux de Grady Booch, James Rumbaugh et Ivar Jacobson.

Fin 1997, UML est devenu une norme OMG (Object Management Group).

UML 1.1 sorti en 1997; 4 révisions du standard (de l'UML1.1 à l'UML1.5 en 2003). Les dernières améliorations (en cours de validation en 2003) étant conséquentes, l'UML passe à une nouvelle version : l'UML 2.0

La dernière version diffusée par l'OMG est UML 2.4.1, disponible depuis août 2011.

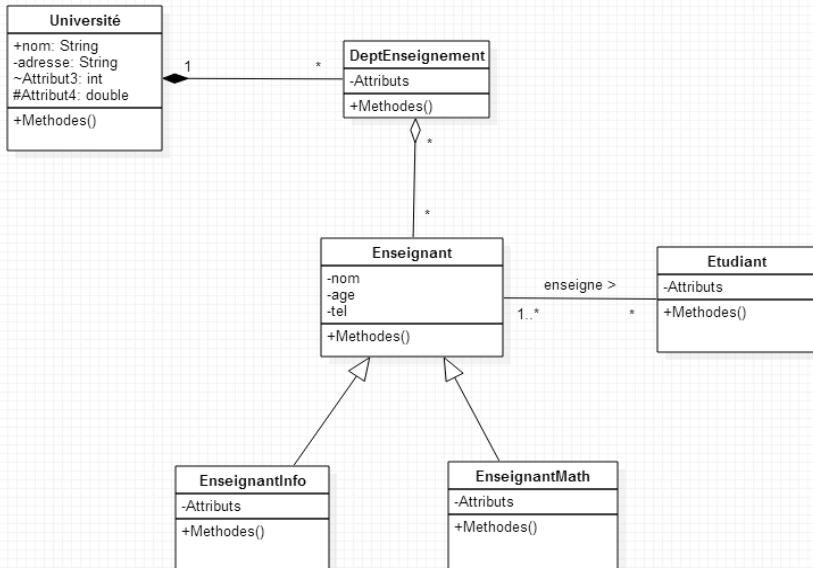


## Diagramme de classes

Représente l'architecture conceptuelle du système.  
considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation.

Caractéristiques:

- Encapsulation, visibilité
- attributs et méthodes
- Relations entre classes
  - association (multiplicité)
  - Agrégation et composition
  - Héritage

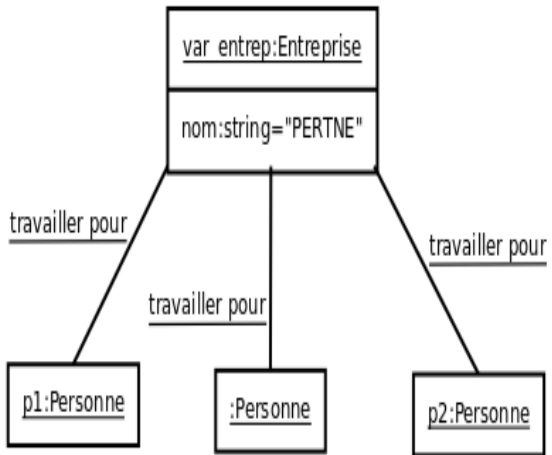
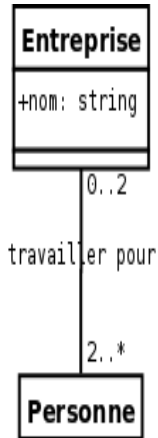




## Diagramme d'objets

représente des objets (i.e. instances de classes) et leurs liens (i.e. instances de relations) pour donner une vue figée de l'état d'un système à un instant donné.

Il est, par exemple, utilisé pour vérifier l'adéquation d'un diagramme de classes à différents cas possibles.

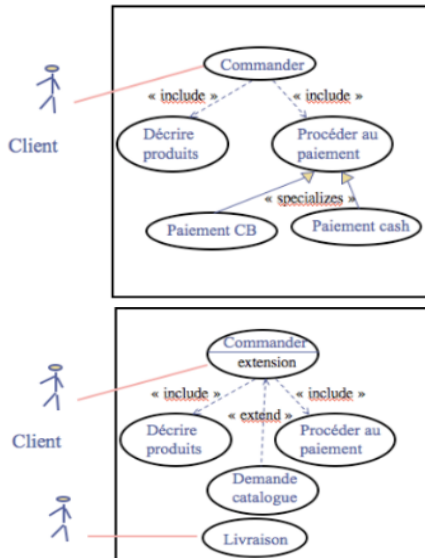


## Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation capture le comportement d'un système, d'un sous- système, d'une classe ou d'un composant tel qu'un utilisateur extérieur le voit.

Éléments des diagrammes de cas d'utilisation:

- Acteur
- Cas d'utilisation
- Relation entre cas d'utilisation



## Diagramme d'états-transitions (ou machine à état)

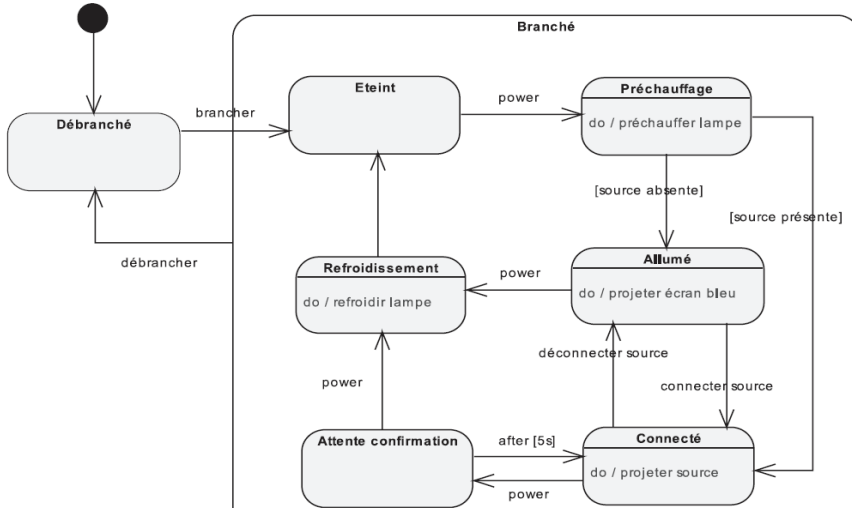
Permettent de décrire les changements d'états d'un objet ou d'un composant, en réponse aux interactions avec d'autres objets/composants ou avec des acteurs.

La vision globale du système n'apparaît pas sur ce type de diagrammes puisqu'ils ne s'intéressent qu'à un seul élément du système indépendamment de son environnement.

Caractéristiques:

- Notion d'états (initial et final)
- Notion d'événement
- Transition
- États composites

sm VideoProjecteur (4è)



## Diagramme d'activité

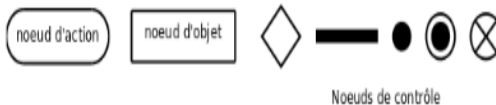
Ils permettent ainsi de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation.

Une activité représente une exécution d' un mécanisme, un déroulement d'étapes séquentielles.

Le passage d'une activité à une autre est matérialisé par une transition

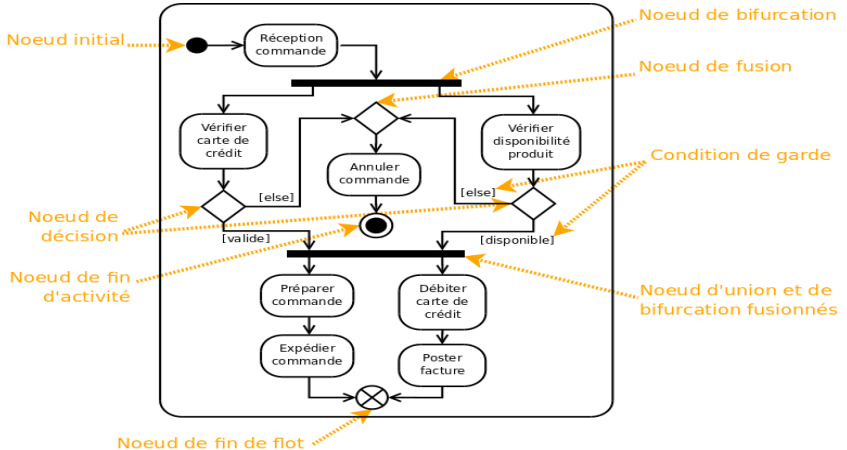
Caractéristiques

- Noeuds d'activités:



- Transition, couloir d'activité

# exemple





## Attention

Les diagrammes d'activités sont relativement proches des diagrammes d'états-transitions dans leur présentation, mais leur interprétation est sensiblement différente; machine à états décrit l'Enchaînement de tous les états d'objet(état, événement, transition).

## Diagramme de Séquence

représente la succession chronologique des opérations réalisées par un acteur.

Il indique les objets que l'acteur va manipuler et les opérations qui font passer d'un objet à l'autre

### Caractéristiques

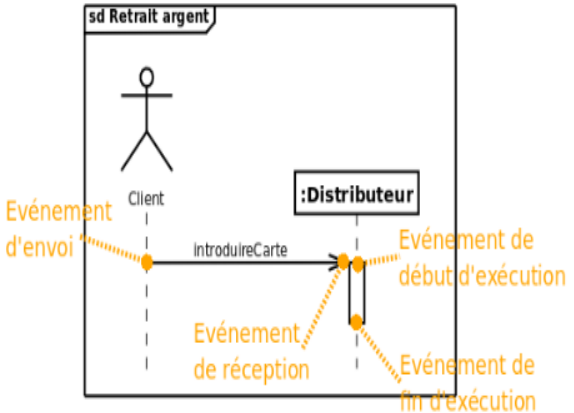
- ligne de vie (le temps s'écoule de haut en bas)
- messages : envoi d'un signal, opération, instantiation



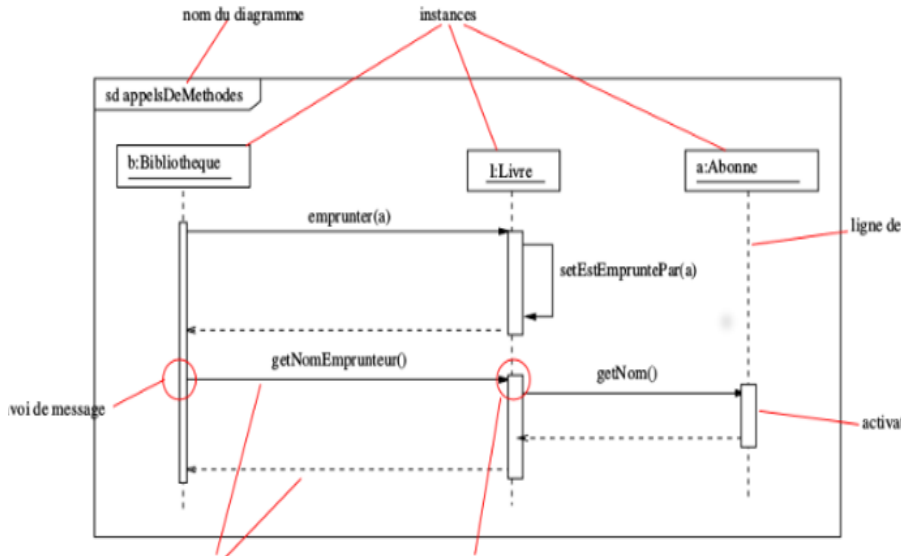
*messages asynchrones*



*Messages synchrones*



## Événement et messages



# Conclusion

UML est un langage formel et normalisé.

Il permet le gain de précision (et aussi de temps), encourage l'utilisation d'outils et constitue à cet effet un gage de stabilité.

UML est un support de communication performant.

Il cadre l'analyse et facilite la compréhension de représentations abstraites complexes. Son caractère polyvalent et sa souplesse en font un langage universel.



**Merci  
Pour  
Votre  
Attention  
!**