

# UML : introduction

**Achref El Mouelhi**

Docteur de l'université d'Aix-Marseille  
Chercheur en programmation par contrainte (IA)  
Ingénieur en génie logiciel

[elmouelhi.achref@gmail.com](mailto:elmouelhi.achref@gmail.com)



**UNIFIED  
MODELING  
LANGUAGE**



Avant de commencer la construction, il faut établir un plan.

La réalisation d'une application peut passer par plusieurs étapes :

- **Définition des besoins** : Identifier les attentes et contraintes.
- **Analyse** : Étudier le problème et définir les cas d'utilisation.
- **Conception** : Modéliser l'architecture et les détails techniques.
- **Développement** : Implémenter le code source.
- **Tests** : Vérifier la conformité aux exigences.
- **Validation** : Confirmer la satisfaction des besoins.
- **Déploiement** : Mettre l'application en production.
- **Maintenance** : Corriger et améliorer le système.

La réalisation d'une application peut passer par plusieurs étapes :

- **Définition des besoins** : Identifier les attentes et contraintes.
- **Analyse** : Étudier le problème et définir les cas d'utilisation.
- **Conception** : Modéliser l'architecture et les détails techniques.
- **Développement** : Implémenter le code source.
- **Tests** : Vérifier la conformité aux exigences.
- **Validation** : Confirmer la satisfaction des besoins.
- **Déploiement** : Mettre l'application en production.
- **Maintenance** : Corriger et améliorer le système.

Ces étapes peuvent être adaptées selon la méthodologie utilisée (cycle en V, Agile, etc.).

## Où est **UML** dans tout ça ?

UML permet de **modéliser** toutes les étapes du développement d'une application de l'analyse au déploiement (en utilisant plusieurs diagrammes).

- **Analyse** : Diagrammes de cas d'utilisation pour capturer les besoins.
- **Conception** : Diagrammes de classes et d'objets pour modéliser la structure.
- ...

## UML : Unified Modeling Language

- Un langage de modélisation unifié.
- Pas un langage de programmation.
- **Indépendant** de tout langage de programmation.
- Basé sur des **notations graphiques**.
- Constitué de **plusieurs diagrammes** permettant de modéliser le système sous divers angles.
- Une norme maintenue par l'**OMG** (Object Management Group : organisation mondiale créée en 1989 pour standardiser le modèle objet)

## Et un graphe ?

En mathématiques, c'est un outil composé de :

- un ensemble de sommets, et
- un ensemble d'arêtes (arcs) reliant les sommets

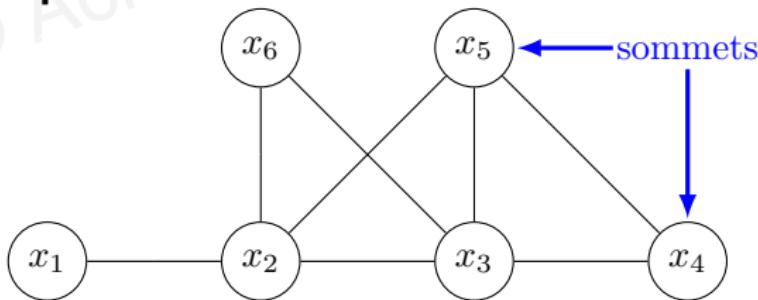
© Achref EL HADJ

## Et un graphe ?

En mathématiques, c'est un outil composé de :

- un ensemble de sommets, et
- un ensemble d'arêtes (arcs) reliant les sommets

## Exemple de graphe

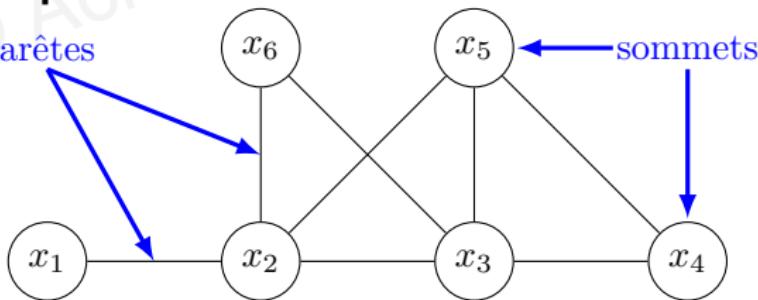


## Et un graphe ?

En mathématiques, c'est un outil composé de :

- un ensemble de sommets, et
- un ensemble d'arêtes (arcs) reliant les sommets

## Exemple de graphe



Avant **UML** : plusieurs méthodes orientées objet (entre 1970 et 1995)

- **Booch** (présentée par **Grady Booch**)
- **OMT** (introduite par **James Rumbaugh**)
- **OOSE** (proposée par **Ivar Jacobson**)
- **OOA, OOD, HOOD...**

## Début d'**UML**

- En 1995, Booch, Rumbaugh et Jacobson commencent à travailler sur une méthode unifiée (**Unified Method**)
- En 1996, création d'un consortium de partenaires pour travailler sur la définition d'**UML**
- En 1997, normalisation de la méthode **UML 1.1** par l'**OMG**

## Différente version d'UML

- UML 1.1 : 1997
- UML 1.2 : 1998
- UML 1.3 : 1999
- UML 1.4 : 2001
- UML 1.5 : 2003
- UML 2.0 : 2005
- UML 2.1 : 2006
- UML 2.2 : 2009
- UML 2.3 : 2010
- UML 2.4 : 2011
- UML 2.5 : 2015
- UML 2.5.1 : 2017

## Remarques

- 14 diagrammes depuis **UML 2.3**
- classés en deux catégories
  - 7 diagrammes de structure (statiques) : permettent de décrire la structure d'un système selon plusieurs points de vue différents (classes, composants, nœuds, objets, packages...)
  - 7 diagrammes de comportement (dynamiques) : permettent de décrire le comportement d'un système de plusieurs points de vue différents (temporel, changement d'état...)

## Diagrammes de structure (statiques)

- Diagramme de classes (class diagram)
- Diagramme d'objets (object diagram)
- Diagramme des paquets (package diagram)
- Diagramme de composants (component diagram)
- Diagramme de déploiement (deployment diagram)
- Diagramme de structure composite (composite structure diagram)
- Diagramme de profils (profile diagram)

## Diagrammes de comportement (dynamiques)

- Diagramme de cas d'utilisation (use-case diagram)
- Diagramme états-transitions (state machine diagram)
- Diagramme d'activité (activity diagram)
- Diagramme de séquence (sequence diagram)
- Diagramme de communication (communication diagram)
- Diagramme global d'interaction (interaction overview diagram)
- Diagramme de temps (timing diagram)

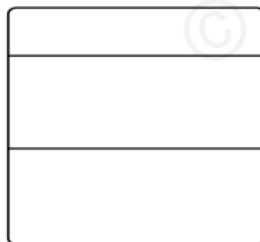
## Notations communes

- Classeur : a une forme rectangulaire et permet de représenter plusieurs éléments dans de différents diagrammes **UML**
- Package (paquetage) : est un regroupement d'éléments de système ou de diagrammes
- Stéréotype : annotation entourée par <<nomAnnotation>> permettant d'ajouter une précision sur l'élément annoté

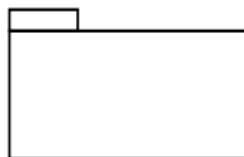
© Achref

## Notations communes

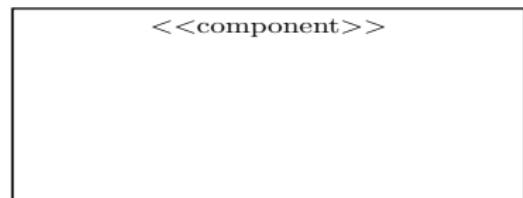
- Classeur : a une forme rectangulaire et permet de représenter plusieurs éléments dans de différents diagrammes **UML**
- Package (paquetage) : est un regroupement d'éléments de système ou de diagrammes
- Stéréotype : annotation entourée par <<nomAnnotation>> permettant d'ajouter une précision sur l'élément annoté



Classeur



Package



Stéréotype

## Les flèches en UML

- Association bidirectionnelle
- Association unidirectionnelle
- - - - - → Dépendance
- Héritage
- - - - - ⇒ Implémentation
- ◊ Agrégation
- ◆ Composition

## Quelques logiciels pour faire la modélisation **UML**

- Power Designer (payant - version d'essai 30 jours)
- **StarUML** (payant depuis octobre 2024 - version d'essai 30 jours)
- BoUML
- Visual Paradigm (payant - version d'essai 30 jours)
- Astah (payant - version d'essai 30 jours)
- Outil en ligne : <https://www.draw.io/>
- ArgoUML (Open source)
- PlantUML