

Qu'est-ce que UML ?

UML (*Unified Modeling Language*) est **un langage de modélisation graphique à base de pictogrammes**. Il est utilisé pour **le développement logiciel en orientée objet**. [Wikipedia]

UML est une notation permettant de modéliser un problème de façon standard. Ce langage est né de la fusion de plusieurs méthodes existantes auparavant, et est devenu désormais la référence en terme de modélisation objet, à un tel point que sa connaissance est souvent nécessaire pour obtenir un poste de développeur objet. [commentcamarche.net]



Qu'est-ce qu'un modèle ?

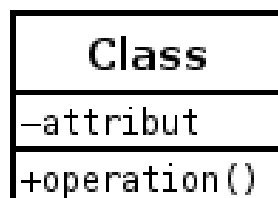
La modélisation consiste **à créer une représentation simplifiée d'un problème** : le **modèle**. Le modèle constitue ainsi une représentation possible du système pour un point de vue donné. [commentcamarche.net]

La modélisation comporte deux composantes :

- L'**analyse**, c'est-à-dire **l'étude du problème et des besoins (quoi faire ?)**
- la **conception**, soit **l'élaboration d'une solution au problème et aux besoins (comment faire ?)**

À quoi sert UML ?


UML est utilisé pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement d'un logiciel orienté objet. UML permet d'obtenir une modélisation indépendante des langages et des environnements.



Grâce aux outils de modélisation UML, il est également possible de **générer automatiquement une partie de code** (génération de squelette de code par exemple en langage Java, C++, PHP, ...) à partir des divers documents réalisés.

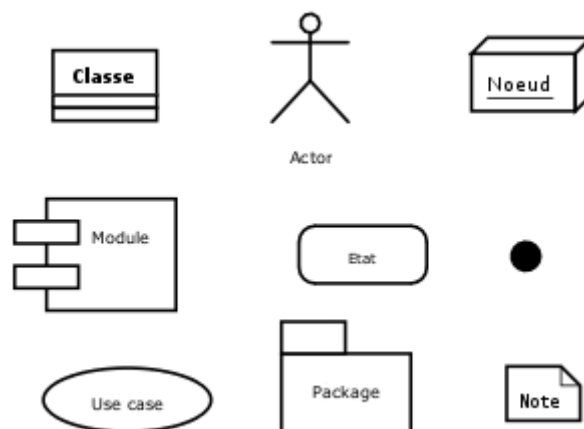
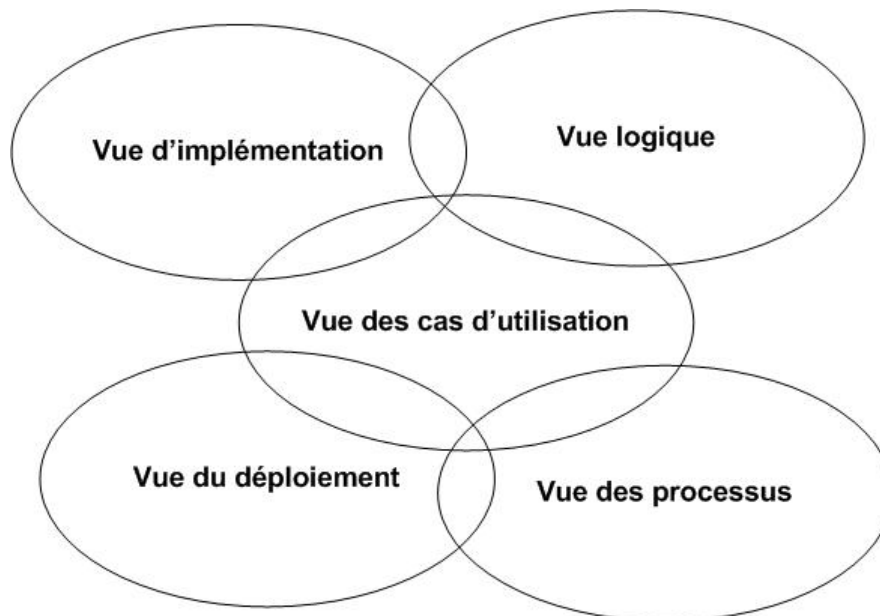
Qu'est-ce qui compose UML ?

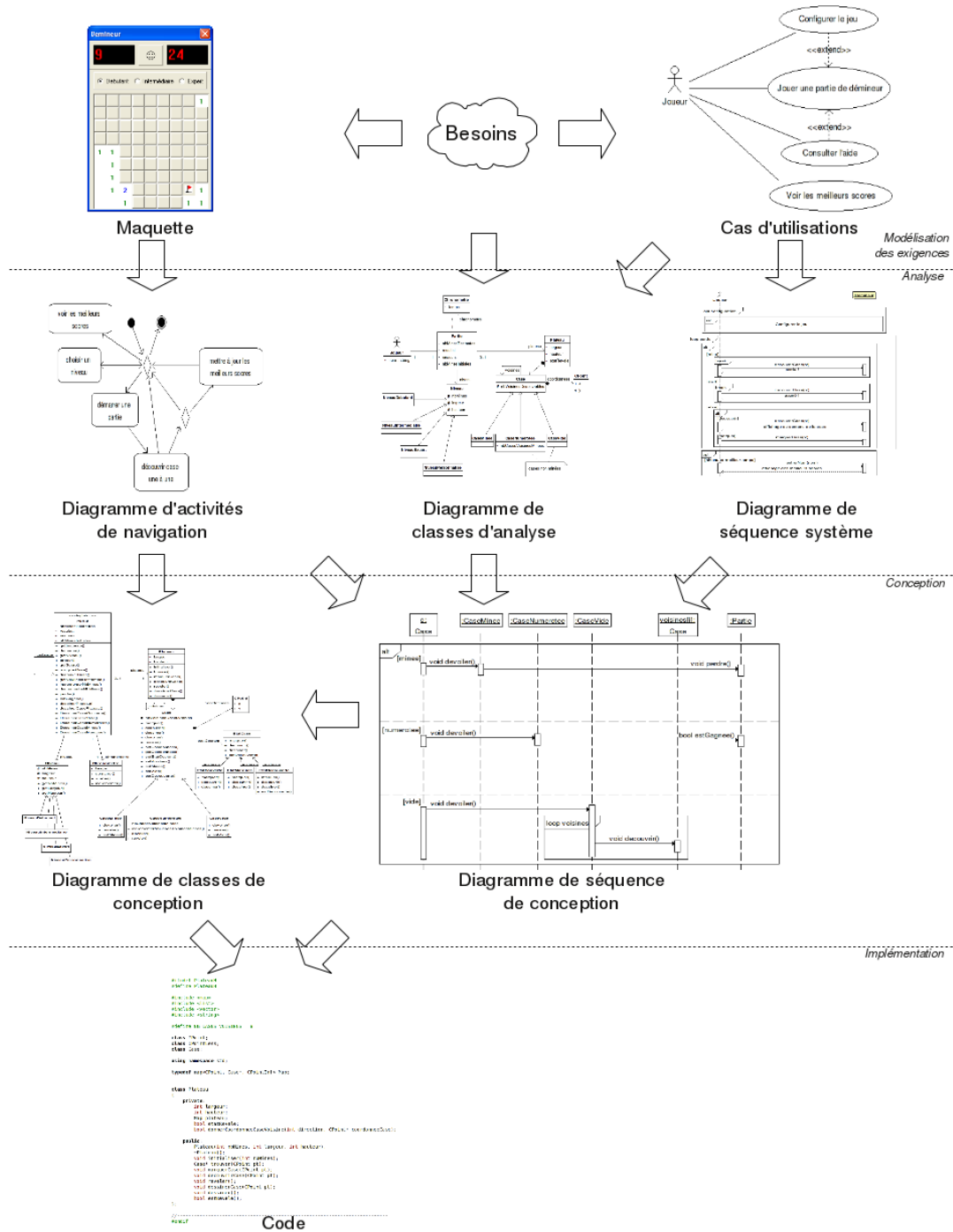
UML 2.3 propose **13 types de diagrammes** (9 diagrammes en UML 1.3).

 UML n'étant pas une méthode, leur utilisation est laissée à l'appréciation de chacun. Des méthodologies, telles que l'*Unified Process* (UP), axent l'analyse en tout premier lieu sur les diagrammes de cas d'utilisation (*Use Case*).

UML se décompose en plusieurs sous-ensembles :

- **Les vues** : elles décrivent le système d'un point de vue donné, qui peut être organisationnel, dynamique, temporel, architectural, logique, etc. En combinant toutes ces vues, il est possible de définir (ou retrouver) le système complet.
- **Les diagrammes** : ils décrivent le contenu des vues. Le diagramme de classes est généralement considéré comme l'élément central d'UML.
- **Les modèles d'élément** : ils sont les briques des diagrammes UML.





Outils logiciels

Il existe de nombreux outils logiciels de modélisation UML (bouml, umbrello, ...). Malheureusement, aucun d'entre eux ne respecte strictement chacune des versions de UML, particulièrement UML2 : beaucoup de ces outils introduisent des notations particulières non conformes, très peu supportent les différents types de diagrammes définis par le standard.

Analogie

Pour illustrer au mieux ce qu'est un modèle, Grady Booch¹ a établi un parallèle entre le développement logiciel et la construction BTP. Cette analogie est judicieuse, car les plans de construction d'un immeuble reflètent bien les idées d'anticipation, de conception et de documentation du modèle. Chaque plan développe par ailleurs un point de vue particulier suivant les corps de métier. Par exemple, le plan des circuits d'eau et le plan des passages électriques concernent le même immeuble mais sont nécessairement séparés. Enfin, chaque plan se situe à un niveau d'abstraction et de détail différent suivant l'usage que l'on désire en faire. Ainsi, le plan de masse permet d'anticiper les conséquences de l'implantation de l'immeuble sur son environnement, exactement comme le modèle de contexte. Viennent ensuite des plans de construction d'un étage, analogues aux modèles de conception. Notons cependant que l'anticipation ne permet pas de prendre en compte les besoins changeants des utilisateurs, l'hypothèse de départ étant justement que ces besoins doivent être définis une bonne fois pour toutes. Or, dans bien des cas ces besoins évoluent au fil du projet : c'est pourquoi il est important de gérer le changement et d'admettre la nécessité de continuer à faire vivre les modèles. Le processus de modélisation du logiciel doit être « adaptatif » et non pas prédictif, contrairement à ce qui se fait dans le BTP !

1. Voir, par exemple, le *Guide de l'utilisateur UML*, G. Booch, J. Rumbaugh, I. Jacobson, Eyrolles 2000.

Qu'est-ce qu'un « bon » modèle ?

A est un bon modèle de B si A permet de répondre de façon satisfaisante à des questions prédéfinies sur B (d'après D.T. Ross).

Un bon modèle doit donc être construit :

- au bon niveau de détail ;
- selon le bon point de vue.

Pensez à l'analogie avec une carte routière. Pour circuler dans Toulouse, la carte de France serait de peu d'utilité. Par contre, pour aller de Toulouse à Paris, la carte de la Haute-Garonne ne suffit pas... À chaque voyage correspond la « bonne » carte !