

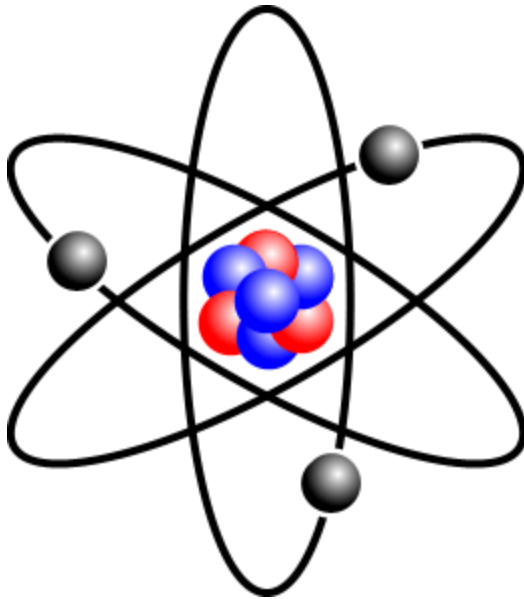
## Chapitre 2

# Modélisation UML

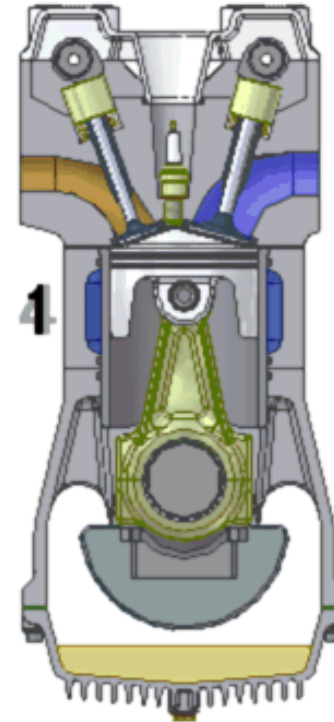
# UML: Un langage de modélisation

- Spécifier un modèle,
- Construire un modèle,
- Communiquer un modèle,
- Documenter un modèle.

# Des modèles spécifiques à chaque discipline



© Indolences / CC-BY-SA 3.0



© UtzOnBike / CC-BY-SA 3.0

# Construire un modèle

- Pour communiquer la structure et le comportement d'un système,
- Pour visualiser, analyser et contrôler l'architecture d'un système,
- Pour comprendre un système,
- Pour gérer le risque lié au développement d'un système.

# Caractéristiques d'un modèle

- Fournit une simplification de la réalité,
- Facilite les manipulations formelles ou automatiques,
- Permet de subdiviser un problème en sous-problèmes plus faciles à comprendre,
- Élève le niveau d'abstraction.

# Quelques types de modèles pour le logiciel

- Modèle procédural,
- Modèle entité-relation,
- Modèle orienté objet,
- Modèle logique,
- Modèle par contraintes.

# Principes de modélisation

- Pour les problèmes complexes, plusieurs modèles presque indépendants sont généralement nécessaires,
- Le choix d'un modèle influence profondément la solution apportée à un problème,
- Chaque modèle d'un problème devrait correspondre à un niveau de détail distinct,
- Les bons modèles sont connectés à la réalité.

# Caractéristiques d'un modèle orienté objet

## Éléments majeurs:

- Abstraction
- Encapsulation
- Héritage
- Polymorphisme

## Éléments mineurs:

- Typage fort
- Concurrency
- Persistance



# Langage de modélisation vs processus

- UML est un langage de modélisation.
- UML n'est pas un processus, c'est un outil qui peut être utilisé dans le cadre d'un processus.
- La méthode présentée dans le livre de Larman s'appelle le « Rational Unified Process » (RUP).
- Ce processus a été développé par Booch, Rumbaugh et Jacobson.

# Un modèle pour plusieurs intervenants

- Le modèle fournit de l'information à plusieurs personnes:
  - Utilisateurs,
  - Clients,
  - Ingénieurs systèmes,
  - Analystes,
  - Concepteurs,
  - Programmeurs,
  - Chefs de projets,
  - etc.
- Il doit donc être compréhensible.

# Points de vue architecturaux

vocabulaire  
fonctionnalité

Logique

Implantation

gestion de  
configuration  
mise en place  
de système

comportement

Scénario

performance  
débit

Processus

Déploiement

topologie de  
système  
répartition  
livraison  
installation

## Point de vue LOGIQUE

### Décomposition orientée objet

- Décomposition en objets et classes,
- Regroupement en paquetages,
- Connexion par héritage, associations, etc,
- Accent sur l'abstraction, l'encapsulation, l'uniformité,
- Réalisation des scénarios.

## Point de vue PROCESSUS

### Décomposition de l'exécution

- Décomposition en tâches et processus,
- Regroupement des groupes de processus,
- Communication,
- Information sur les caractéristiques suivantes:
  - Disponibilité,
  - Fiabilité,
  - Intégrité,
  - Performance,
  - Contrôle.

## Point de vue IMPLANTATION

### Décomposition statique des modules et sous-systèmes

- Décomposition en modules et niveaux,
- Regroupement de modules en paquetages,
- Organisation des sous-systèmes en niveaux pour:
  - Réduire le couplage et la visibilité,
  - Augmenter la robustesse.
- Information sur les caractéristiques suivantes:
  - Facilité de développement,
  - Potentiel de réutilisation,
  - Gestion de configuration.

## Point de vue DÉPLOIEMENT

### Relation Matériel-Logiciel

- Décomposition en noeuds d'exécution,
- Rôle d'un noeud,
- Interconnectivité, topologie,
- Information sur les caractéristiques suivantes:
  - Performance,
  - Disponibilité,
  - Installation,
  - Maintenance.

## Point de vue SCÉNARIO

Comportement du système pour les utilisateurs

- Illustration des autres vues,
- Regroupe le comportement du système selon:
  - Priorité: critique, important, accessoire,
  - Risques à circonscrire,
  - Options disponibles,
  - Couverture de l'architecture,
  - Autres objectifs tactiques et contraintes.