

Introduction à l'UML

Unified Modeling Language

W.Soussi B.Rima

L2 CMI Informatique

HLIN408, 5 mars 2017

1 Modèles et Langages de Modélisation

- Un Modèle
- Un Langage de Modélisation

2 L'UML

- Qu'est-ce que c'est l'UML
- L pour Langage
- Un peu d'histoire
- Le MOF (Meta Object Facility)

3 Modèles et diagrammes d'UML

- Modèles d'UML
- Diagrammes d'UML
- Diagrammes du modèle structurel

4 Outils et logiciels pour UML

Un Modèle

Modèles et Langages de Modélisation

Definition

Représentation abstraite et simplifiée d'une réalité.

Un Modèle

Modèles et Langages de Modélisation

Definition

Représentation abstraite et simplifiée d'une réalité.

Utilité

Idéalisation d'une réalité complexe et accentuation des détails importants.
Description d'un problème et essai d'en trouver une solution.

Un Langage de Modélisation

Modèles et Langages de Modélisation

Definition

Système formel permettant la modélisation d'une réalité.

Un Langage de Modélisation

Modèles et Langages de Modélisation

Definition

Système formel permettant la modélisation d'une réalité.

Utilité

Définir les entités (caractéristiques, comportements, associations) composant un système complexe.

Un Langage de Modélisation

Modèles et Langages de Modélisation

Definition

Système formel permettant la modélisation d'une réalité.

Utilité

Définir les entités (caractéristiques, comportements, associations) composant un système complexe.

Remarque

Différent d'un langage de programmation, avec lequel il permet la mise en oeuvre d'un logiciel.

Qu'est-ce que c'est l'UML

L'UML

- **UML (Unified Modeling Language)** est un langage de modélisation semi-formel et graphique utilisé pour :
 - ① spécifier
 - ② visualiser
 - ③ documenterles artefacts d'un système.

Qu'est-ce que c'est l'UML

L'UML

- **UML (Unified Modeling Language)** est un langage de modélisation semi-formel et graphique utilisé pour :

- ① spécifier
- ② visualiser
- ③ documenter

les artefacts d'un système.

- Un « artefact » est n'importe quel produit résultant de la réalisation d'un projet : fichiers sources, exécutables, documentations, fichiers de configurations etc.

Qu'est-ce que c'est l'UML (2)

L'UML

- Indépendant de la portée du projet.

Qu'est-ce que c'est l'UML (2)

L'UML

- Indépendant de la portée du projet.
- Indépendant du langage de programmation utilisé.

Qu'est-ce que c'est l'UML (2)

L'UML

- Indépendant de la portée du projet.
- Indépendant du langage de programmation utilisé.
- Composant d'un processus de développement et non pas le processus lui-même.

Un langage semi-formel

L'UML

- Le langage UML est dit graphique parce qu'il est basé sur des diagrammes.

Un langage semi-formel

L'UML

- Le langage UML est dit graphique parce qu'il est basé sur des diagrammes.
- Le langage UML est dit semi-formel parce qu'il est présenté par des diagrammes et par le langage naturel.

Un langage semi-formel

L'UML

- Le langage UML est dit graphique parce qu'il est basé sur des diagrammes.
- Le langage UML est dit semi-formel parce qu'il est présenté par des diagrammes et par le langage naturel.
- Un diagramme d'UML est constitué d'un ensemble d'éléments ayant une représentation graphique.

Un langage semi-formel

L'UML

- Le langage UML est dit graphique parce qu'il est basé sur des diagrammes.
- Le langage UML est dit semi-formel parce qu'il est présenté par des diagrammes et par le langage naturel.
- Un diagramme d'UML est constitué d'un ensemble d'éléments ayant une représentation graphique.
- Ce n'est pas une notation graphique mais un vrai langage ayant des règles de syntaxe et de sémantique bien définies.

Un langage semi-formel (2)

L'UML

Example



- SYNTAXE : La flèche indiquant le sens de l'association est facultative.
- SÉMANTIQUE : La flèche spécifie le sens de l'interprétation de l'association.

Un peu d'histoire

L'UML

- Fin des années 80 : plusieurs langages de modélisation objet → un obstacle pour la diffusion du paradigme objet.

Un peu d'histoire

L'UML

- Fin des années 80 : plusieurs langages de modélisation objet → un obstacle pour la diffusion du paradigme objet.
- 1994 : RSC (Rational Software Corporation) fusionna les langages OOD (Booch) et OMT (Rumbaugh).

Un peu d'histoire

L'UML

- Fin des années 80 : plusieurs langages de modélisation objet → un obstacle pour la diffusion du paradigme objet.
- 1994 : RSC (Rational Software Corporation) fusionna les langages OOD (Booch) et OMT (Rumbaugh).
- 1995 : acquisition de l'entreprise d'Ivar Jacobson par RSC → intégration de son langage OOSE aux deux autres (OOD et OMT).

Un peu d'histoire

L'UML

- Fin des années 80 : plusieurs langages de modélisation objet → un obstacle pour la diffusion du paradigme objet.
- 1994 : RSC (Rational Software Corporation) fusionna les langages OOD (Booch) et OMT (Rumbaugh).
- 1995 : acquisition de l'entreprise d'Ivar Jacobson par RSC → intégration de son langage OOSE aux deux autres (OOD et OMT).
- 1996 : Booch, Rumbaugh et Jacobson : chargés de la création de l'UML par RSC.

Un peu d'histoire (2)

L'UML

- 1997 : 1^{ère} normalisation de UML par l'OMG (Object Management Group) : UML 1.0

Un peu d'histoire (2)

L'UML

- 1997 : 1^{ère} normalisation de UML par l'OMG (Object Management Group) : UML 1.0
- les versions d'UML les plus importantes adoptées par OMG :
 - 1 2005 : UML 2.0
 - 2 2007 : UML 2.1.2
 - 3 2013 : UML 2.5 beta 2

Le MOF (Meta Object Facility)

L'UML

- Architecture standardisée par OMG pour décrire formellement l'UML.

Le MOF (Meta Object Facility)

L'UML

- Architecture standardisée par OMG pour décrire formellement l'UML.
- Quatre niveaux de modélisation : M0, M1, M2 et M3.

Le MOF (Meta Object Facility)

L'UML

- Architecture standardisée par OMG pour décrire formellement l'UML.
- Quatre niveaux de modélisation : M0, M1, M2 et M3.
- Chaque niveau est une instance d'un élément du niveau supérieur :
 - ① Un élément de M0 est la réalité à décrire.

Le MOF (Meta Object Facility)

L'UML

- Architecture standardisée par OMG pour décrire formellement l'UML.
- Quatre niveaux de modélisation : M0, M1, M2 et M3.
- Chaque niveau est une instance d'un élément du niveau supérieur :
 - 1 Un élément de M0 est la réalité à décrire.
 - 2 Un élément de M1 est un modèle décrivant une réalité en M0.

Le MOF (Meta Object Facility)

L'UML

- Architecture standardisée par OMG pour décrire formellement l'UML.
- Quatre niveaux de modélisation : M0, M1, M2 et M3.
- Chaque niveau est une instance d'un élément du niveau supérieur :
 - 1 Un élément de M0 est la réalité à décrire.
 - 2 Un élément de M1 est un modèle décrivant une réalité en M0.
 - 3 Un élément de M2 est un méta-modèle (ici l' UML) décrivant un modèle en M1.

Le MOF (Meta Object Facility)

L'UML

- Architecture standardisée par OMG pour décrire formellement l'UML.
- Quatre niveaux de modélisation : M0, M1, M2 et M3.
- Chaque niveau est une instance d'un élément du niveau supérieur :
 - ① Un élément de M0 est la réalité à décrire.
 - ② Un élément de M1 est un modèle décrivant une réalité en M0.
 - ③ Un élément de M2 est un méta-modèle (ici l' UML) décrivant un modèle en M1.
 - ④ Un élément de M3 est un méta-méta-modèle (ici le MOF) décrivant un méta-modèle en M2.

Le MOF (Meta Object Facility)

L'UML

- Architecture standardisée par OMG pour décrire formellement l'UML.
- Quatre niveaux de modélisation : M0, M1, M2 et M3.
- Chaque niveau est une instance d'un élément du niveau supérieur :
 - ① Un élément de M0 est la réalité à décrire.
 - ② Un élément de M1 est un modèle décrivant une réalité en M0.
 - ③ Un élément de M2 est un méta-modèle (ici l' UML) décrivant un modèle en M1.
 - ④ Un élément de M3 est un méta-méta-modèle (ici le MOF) décrivant un méta-modèle en M2.
- Un élément de M3 est défini comme une instance de lui-même.

- UML décrit un système informatique selon quatre points de vue correspondant chacun à un « *modèle* » :

- UML décrit un système informatique selon quatre points de vue correspondant chacun à un « *modèle* » :
 - 1 modèle structurel/statique : décrire les types d'objets et leurs relations.

- UML décrit un système informatique selon quatre points de vue correspondant chacun à un « *modèle* » :
 - 1 modèle structurel/statique : décrire les types d'objets et leurs relations.
 - 2 modèle dynamique : stimuli des objets et leurs réponses.

- UML décrit un système informatique selon quatre points de vue correspondant chacun à un « *modèle* » :
 - 1 modèle structurel/statique : décrire les types d'objets et leurs relations.
 - 2 modèle dynamique : stimuli des objets et leurs réponses.
 - 3 modèle d'utilisation : fonctionnalités des objets.

- UML décrit un système informatique selon quatre points de vue correspondant chacun à un « *modèle* » :
 - ① modèle structurel/statique : décrire les types d'objets et leurs relations.
 - ② modèle dynamique : stimuli des objets et leurs réponses.
 - ③ modèle d'utilisation : fonctionnalités des objets.
 - ④ modèle d'implémentation : les composants, fichiers base de données, projection sur le matériel, etc...

- UML décrit un système informatique selon quatre points de vue correspondant chacun à un « *modèle* » :
 - ① modèle structurel/statique : décrire les types d'objets et leurs relations.
 - ② modèle dynamique : stimuli des objets et leurs réponses.
 - ③ modèle d'utilisation : fonctionnalités des objets.
 - ④ modèle d'implémentation : les composants, fichiers base de données, projection sur le matériel, etc...
- chaque modèle est une représentation abstraite d'une réalité fournissant une image simplifiée du monde réel selon un point de vue.

- chaque modèle d'UML contient des « *diagrammes* », décrivant chacun des aspects particuliers du modèle :

- chaque modèle d'UML contient des « *diagrammes* », décrivant chacun des aspects particuliers du modèle :
 - 1 modèle structurel/statique : diagrammes de classes, d'instances.

- chaque modèle d'UML contient des « *diagrammes* », décrivant chacun des aspects particuliers du modèle :
 - 1 modèle structurel/statique : diagrammes de classes, d'instances.
 - 2 modèle dynamique : diagrammes de séquences, d'états, d'activités.

- chaque modèle d'UML contient des « *diagrammes* », décrivant chacun des aspects particuliers du modèle :
 - 1 modèle structurel/statique : diagrammes de classes, d'instances.
 - 2 modèle dynamique : diagrammes de séquences, d'états, d'activités.
 - 3 modèle d'utilisation : diagramme de cas d'utilisation.

- chaque modèle d'UML contient des « *diagrammes* », décrivant chacun des aspects particuliers du modèle :
 - 1 modèle structurel/statique : diagrammes de classes, d'instances.
 - 2 modèle dynamique : diagrammes de séquences, d'états, d'activités.
 - 3 modèle d'utilisation : diagramme de cas d'utilisation.
 - 4 modèle d'implémentation : diagrammes de composants.

Diagramme d'instances

- décrire les objets du domaine modélisé.
- décrire les liens entre les objets.

Diagrammes du modèle structurel

Modèles et diagrammes d'UML

Diagramme d'instances

- décrire les objets du domaine modélisé.
- décrire les liens entre les objets.

Diagramme de classes

Une abstraction des diagrammes d'instances : des classes en relation regroupant chacune des objets ayant des caractéristiques communes.

Diagrammes du modèle structurel (2)

Modèles et diagrammes d'UML

Syntaxe des attributs en UML

« [visibilité] [/] nom [:type] [[multiplicité]] [= valeurParDéfaut]
[{propriétés...}] »

visibilité : {+, -, #, ~}

/ : le fait qu'un attribut soit dérivé

nom : le nom de l'attribut (partie **obligatoire** de la syntaxe)

type : le domaine de valeurs de l'attribut

multiplicité : le nombre de valeurs que peut prendre l'attribut

valeurParDéfaut : la valeur que possède l'attribut à l'origine

propriétés : des propriétés précisant le comportement de l'attribut,
(*{constant}* pour indiquer qu'il est "read-only" par exemple)

Diagrammes du modèle structurel (3)

Modèles et diagrammes d'UML

Syntaxe des opérations en UML

« [visibilité] nom [(liste-paramètres)] [: typeRetour] [{propriétés}] »

visibilité : {+, -, #, ~}

nom : le nom de l'opération (partie **obligatoire** de la syntaxe)

(liste-paramètres) : les paramètres que peut avoir une opération

typeRetour : le type de retour de l'opération

propriétés : des propriétés précisant le comportement de l'opération,
(*{query}*) pour indiquer que l'opération ne modifie pas
l'instance courante sur laquelle elle est appliquée par exemple)

Diagrammes du modèle structurel (4)

Modèles et diagrammes d'UML

Syntaxe des paramètres des opérations en UML

« [direction] nom : type [multiplicité] [= valeurParDéfaut] {propriétés} »

direction : {in, out, inout}

nom : le nom du paramètre (partie **obligatoire** de la syntaxe)

type : le domaine de valeurs du paramètre (partie **obligatoire** de la syntaxe)

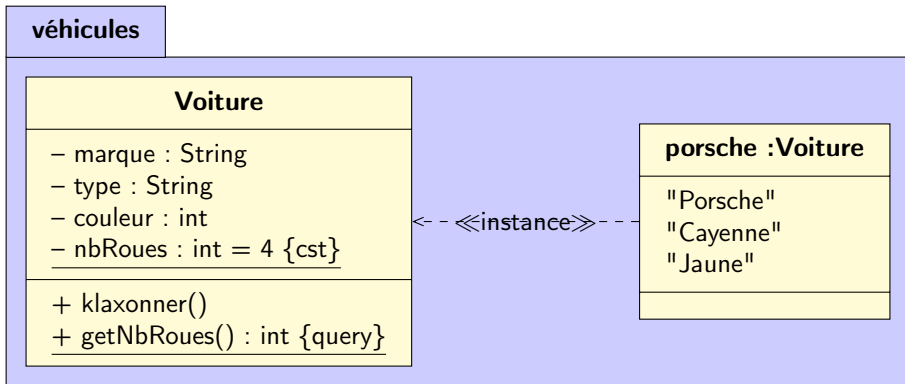
multiplicité : le nombre de valeurs que peut prendre le paramètre

valeurParDéfaut : la valeur que possède le paramètre à l'origine

propriétés : des propriétés précisant le comportement du paramètre, (*idem* que les attributs)

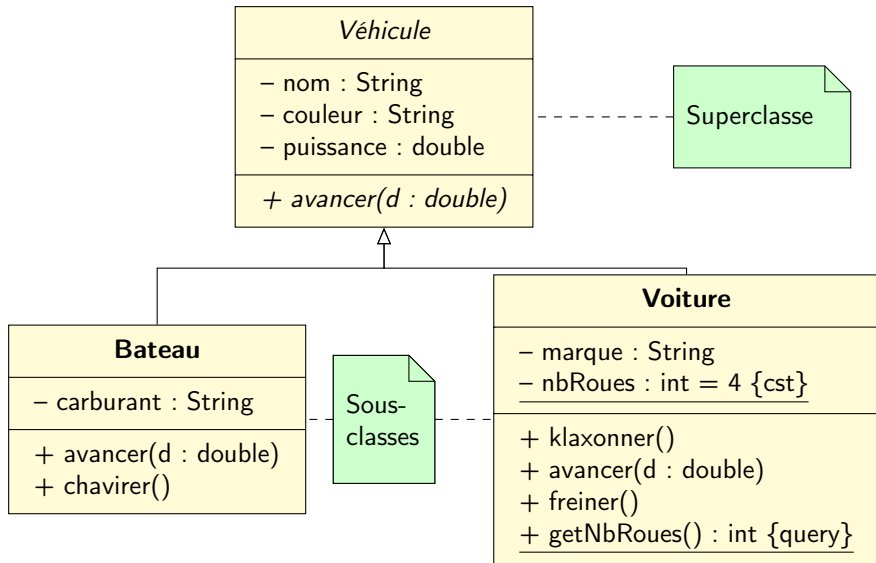
Diagrammes du modèle structurel (Exemples)

Modèles et diagrammes d'UML



Diagrammes du modèle structurel (Exemples)

Modèles et diagrammes d'UML



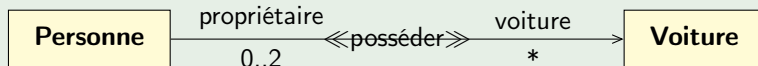
Diagrammes du modèle structurel (Les associations)

Modèles et diagrammes d'UML

Definition

Une association est une abstraction de liens identifiés entre des instances de classes.

Example



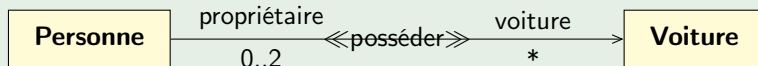
Diagrammes du modèle structurel (Les associations)

Modèles et diagrammes d'UML

Definition

Une association est une abstraction de liens identifiés entre des instances de classes.

Example



Remarque

L'association est une notion de modélisation qui est absente en programmation.

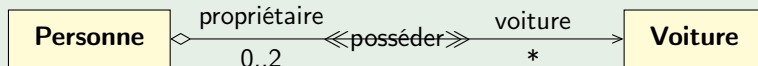
Diagrammes du modèle structurel (Les associations)

Modèles et diagrammes d'UML

Agrégation

L'agrégation est une association entre un ensemble d'éléments et un autre élément.

Exemple



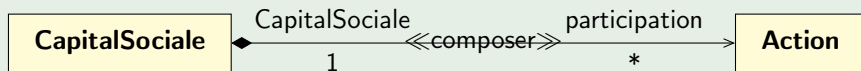
Diagrammes du modèle structurel (Les associations)

Modèles et diagrammes d'UML

Composition

La composition est une agrégation où l'ensemble d'éléments, les composants, appartiennent à un seul élément, nommé composite.

Exemple



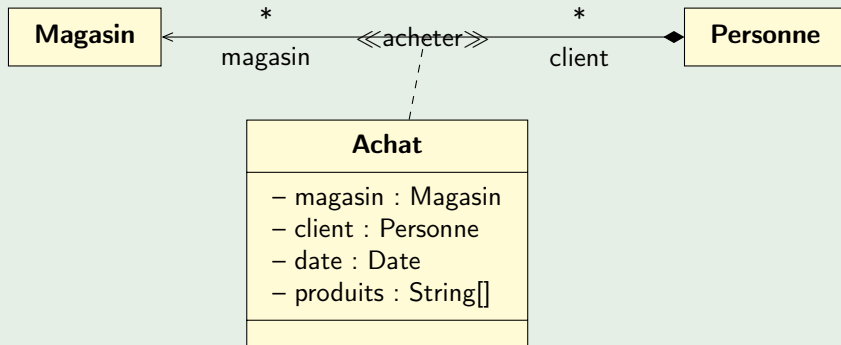
Diagrammes du modèle structurel (Les associations)

Modèles et diagrammes d'UML

Classe d'association

Une classe d'association est une classe qui contient les attributs de lien d'une association.

Exemple



Exemples

Outils et logiciels pour UML

- Il existe plein de logiciels et de services en ligne pour :

Exemples

Outils et logiciels pour UML

- Il existe plein de logiciels et de services en ligne pour :
 - ① Dessiner en UML.

- Il existe plein de logiciels et de services en ligne pour :
 - ① Dessiner en UML.
 - ② Générer d'UML à partir de code source.

- Il existe plein de logiciels et de services en ligne pour :
 - ① Dessiner en UML.
 - ② Générer d'UML à partir de code source.
 - ③ Générer du code source à partir d'UML.

- Il existe plein de logiciels et de services en ligne pour :
 - ① Dessiner en UML.
 - ② Générer d'UML à partir de code source.
 - ③ Générer du code source à partir d'UML.
 - ④ etc.
- Parmi les outils les plus couramment utilisés :
 - ① draw.io (<https://www.draw.io/>)

- Il existe plein de logiciels et de services en ligne pour :
 - ① Dessiner en UML.
 - ② Générer d'UML à partir de code source.
 - ③ Générer du code source à partir d'UML.
 - ④ etc.
- Parmi les outils les plus couramment utilisés :
 - ① draw.io (<https://www.draw.io/>)
 - ② Umbrello (<https://umbrello.kde.org/>)

- Il existe plein de logiciels et de services en ligne pour :
 - ① Dessiner en UML.
 - ② Générer d'UML à partir de code source.
 - ③ Générer du code source à partir d'UML.
 - ④ etc.
- Parmi les outils les plus couramment utilisés :
 - ① draw.io (<https://www.draw.io/>)
 - ② Umbrello (<https://umbrello.kde.org/>)
 - ③ tikz-uml : un package en LaTeX pour faire des diagrammes d'UML (<http://perso.ensta-paristech.fr/~kielbasi/tikzuml/>)

- Il existe plein de logiciels et de services en ligne pour :
 - ① Dessiner en UML.
 - ② Générer d'UML à partir de code source.
 - ③ Générer du code source à partir d'UML.
 - ④ etc.
- Parmi les outils les plus couramment utilisés :
 - ① draw.io (<https://www.draw.io/>)
 - ② Umbrello (<https://umbrello.kde.org/>)
 - ③ tikz-uml : un package en LaTeX pour faire des diagrammes d'UML (<http://perso.ensta-paristech.fr/~kielbasi/tikzuml/>)
 - ④ ObjectAid : un plugin eclipse pour générer de l'UML (<http://www.objectaid.com/home>)

- **Un Modèle** : Représentation abstraite et simplifiée d'une réalité.
- **Un Langage de Modélisation** : Système formel permettant la modélisation d'une réalité.
- **UML** : Langage de modélisation semi-formel graphique.
- **Un Diagramme d'UML** : Une entité graphique en UML décrivant des aspects particuliers d'un modèle UML.
- Parmi les diagrammes d'UML : **le diagramme de classes**, **d'instances**, et le **diagramme de cas d'utilisation**.
- Parmi les outils permettant de manipuler l'UML : **ObjectAid** pour eclipse et **tikz-uml** en LaTeX.



Marianne Huchard, Clémentine Nebut.

Modélisation et programmation par Objets 1 HLIN406, 12 janvier 2017



Gian Piero Favini

Introduzione à l'UML.

www.cs.unibo.it/gabbri/MaterialeCorsi/1.introUML.favini.pdf, 2007.