



UNIFIED MODELING LANGUAGE

Jérémy PERROUAULT



PRÉSENTATION UML

Introduction UML

UML

Modèle qui simplifie la réalité

Permet de

- Visualiser le système comme il est (ou comme il devrait être, idéalement)
- De valider le modèle avec le client
- De spécifier les structures (données) et le comportement (fonctionnalités)
- Fournir un guide
- Documenter le système

UML, CE LANGAGE

UML n'est pas une méthode

UML est un langage de modélisation objet

UML a été adopté par toutes les méthodes objets

UML est dans le domaine public

UML est une norme

UML

3 blocs de base pour construire

- Les entités
- Les relations
- Les diagrammes

Règles à observer pour utiliser ces blocs

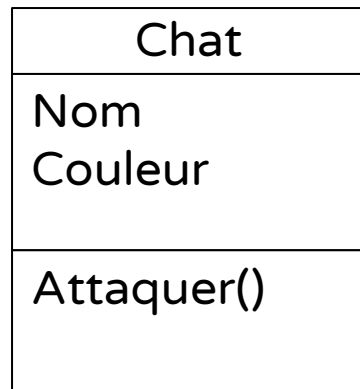
- Règles sémantiques
- Règles de présentation

Les mécanismes communs

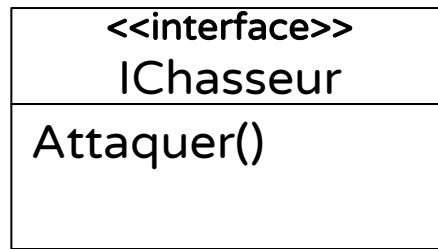
- Spécification
- Présentation
- Extension des modèles

LES ENTITÉS

Classe



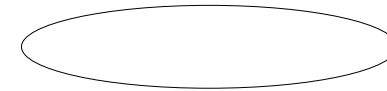
Interface



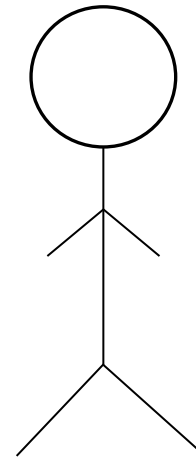
Collaboration



Cas d'utilisation

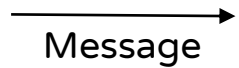


Acteur



LES COMPORTEMENTS

Appel



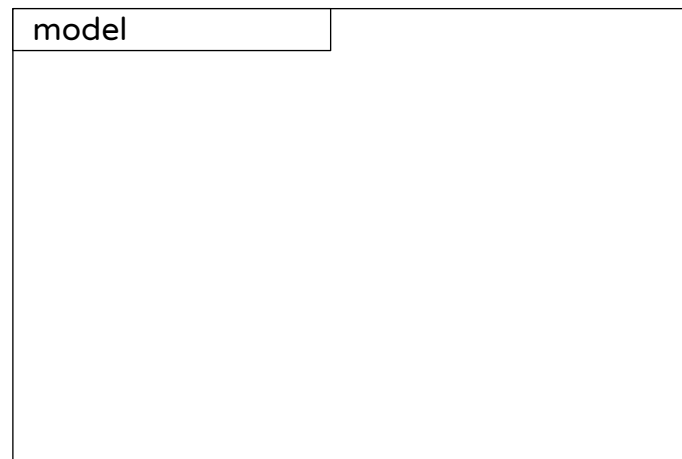
Etat

utilisé

Activité

recherche

Regroupement



Notes

Un chat a attaqué

LES RELATIONS



Dépendance (Dépendance entre entités)



Association (Relation entre entités)

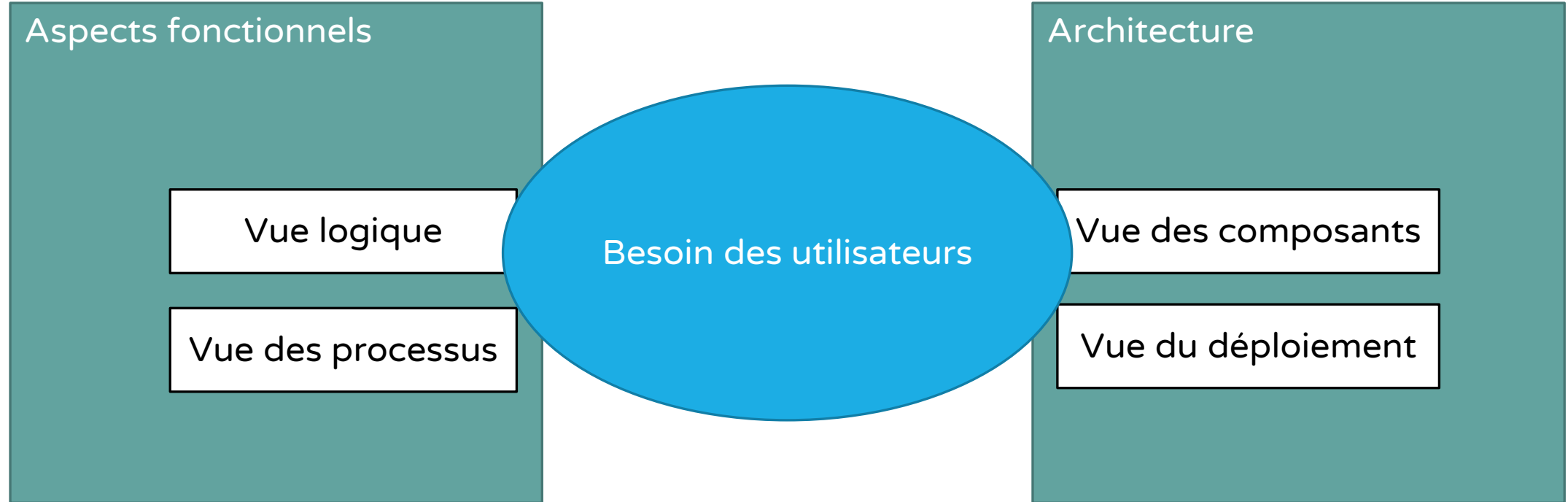


Héritage (Une classe hérite d'une classe mère)



Implémentation (Une classe implémente une interface)

LES 4+1 VUES D'UN SYSTÈME



LES DIAGRAMMES

Diagrammes de structure (ou statiques)

- Diagramme de classes
 - Représente les classes et interfaces
- Diagramme d'objets
 - Représente les instances de classe (les objets)
- Diagramme de composants
 - Représente les composants d'un point de vue physique (fichiers, bases de données)
- Diagramme de déploiement
 - Représente des éléments matériels (périphérique, machine, réseau)
- Diagramme de paquets
 - Regroupe des définitions
- Diagramme de structure composite
 - Représentation sous forme de « boîte blanche » les relations entre composants d'une classe
- Diagramme de profils
 - Spécialisation et personnalisation du domaine particulier d'un meta-modèle de référence

LES DIAGRAMMES

Diagrammes de comportement

- Diagramme de cas d'utilisations
 - Représente les possibilités d'interactions entre le système et les acteurs
- Diagramme états-transitions
 - Représente sous forme de machine à états finis le comportement du système ou des composants
- Diagramme d'activités
 - Représente sous forme de flux d'activités le comportement du système

LES DIAGRAMMES

Diagrammes d'interaction (dynamiques)

- Diagramme de séquence
 - Représente de façon séquentielle le déroulement des traitements et des interactions entre acteurs
- Diagramme de communication
 - Représente de façon simplifiée un diagramme de séquence en se concentrant sur les messages échangés
- Diagramme global d'interaction
 - Représente les enchaînements possibles entre les scénarios de séquence
- Diagramme de temps
 - Représente les variations d'une données au cours du temps

LES DIAGRAMMES

Diagramme	Etape projet
Diagramme de cas d'utilisation	Spécifications, cahier des charges, cahier fonctionnel
Diagramme de séquences	
Diagramme d'activités	
Diagramme de classes	Conception architecturale, cahier technique
Diagramme d'objets	
Diagramme de communication	
Diagramme de déploiement	

LES DIAGRAMMES

Diagramme	Etape projet
Diagramme de packages	Besoins de l'utilisateur
Diagramme de cas d'utilisation	
Diagramme de classes	Vue logique
Diagramme d'objets	
Diagramme de séquences	Vue des processus
Diagramme d'activités	
Diagramme de collaboration	
Diagramme d'états-transitions	
Diagramme global d'interaction	
Diagramme de temps	
Diagramme de structure composite	Vue des composants
Diagramme de composants	
Diagramme de déploiement	Vue de déploiement



DIAGRAMME DE CLASSES

Les diagrammes de classe

DIAGRAMME DE CLASSES

Une classe permet de représenter quelque chose

- Définition de ses attributs, des différentes informations
- Définition de ses actions, ses fonctionnalités (comme des sous-programmes)

▪ Exemple :

▪ Une personne a un nom, un prénom, un age, et peut marcher

- Ses attributs sont « nom », « prenom », « age »
- Ses actions sont « marcher »

▪ Une réunion a un nom, une date, un lieu, une liste des personnes, et peut être démarrée ou annulée

- Ses attributs sont « nom », « date », « lieu », « liste de personnes »
- Ses actions sont « demarrer », et « annuler »

DIAGRAMME DE CLASSES

Chaque attribut et action d'une classe a une « visibilité »

- Elle peut être publique, notée « + »
- Elle peut être privée, notée « - »
- Elle peut être protégée, notée « # »
 - Dans ce cas, l'information est visible uniquement par les classes de la même hiérarchie (classes filles -> classe mère)

Chaque attribut est d'un type

Chaque action peut (éventuellement) retourner un type

- On note ces types par « : type »

DIAGRAMME DE CLASSES

NomClasse
Liste Des Attributs
Liste() Des() Actions()

Personne
- nom : string - prenom : string - age : int
+ marcher() : void

Reunion
- nom : string - date : datetime - lieu : string - personnes : list
+ demarrer() : void + annuler() : void

+	public
-	privé
#	protégé

DIAGRAMME DE CLASSES

Sur le diagramme de classe, on peut ajouter des stéréotypes

- Préciser que c'est une interface ou classe abstraite, par exemple

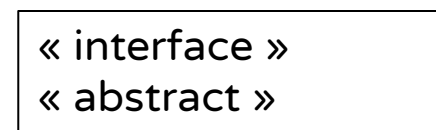
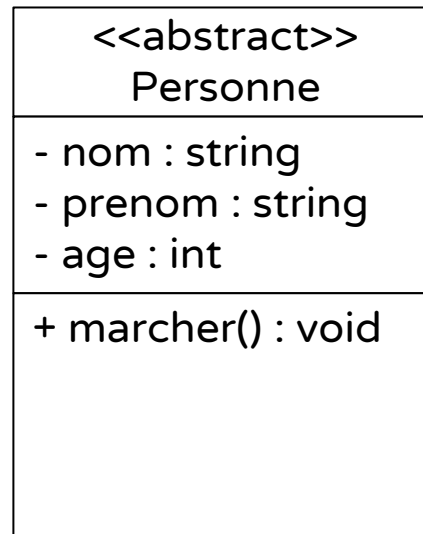
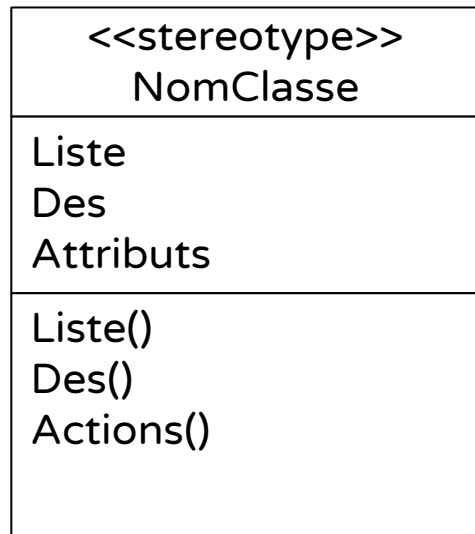
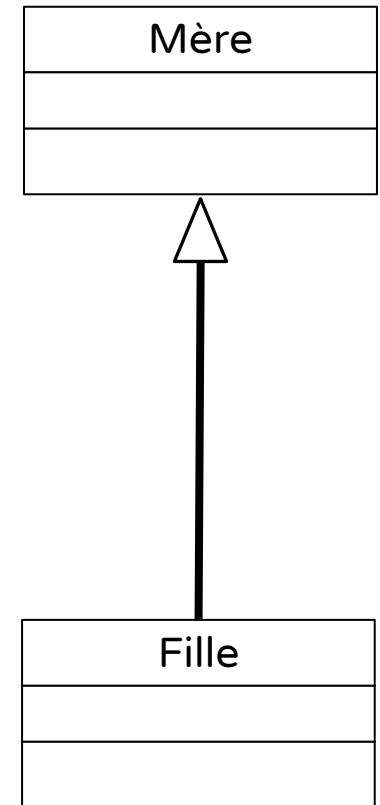


DIAGRAMME DE CLASSES — HÉRITAGE

Un hérite entre une classe mère et une classe fille



EXERCICE

Modéliser les classes Animal et Chat dans un diagramme de classe

- Chat est un Animal
- Un animal a un nom et un age, et il peut manger
- Un chat peut griffer

DIAGRAMME DE CLASSES — RELATIONS

Agrégation

- Relation qui indique que la classe A a un attribut de la classe B
- Schématisé par un losange vide

Composition

- Agrégation forte
- La classe A ne peut pas exister sans la classe B
- Schématisé par un losange plein

DIAGRAMME DE CLASSES — RELATIONS

La Personne a une relation agrégée avec le Chat

- Elle peut exister sans le Chat, avoir plusieurs Chats

Le Chassis a une relation de composition avec la Voiture

- Le Chassis est un élément indissociable de la Voiture
- La voiture a besoin du chassis
- Si le chassis disparaît, la voiture aussi



DIAGRAMME DE CLASSES — ASSOCIATIONS

Connexion entre 2 classes

Distinguer association classique avec ou sans sens de lecture

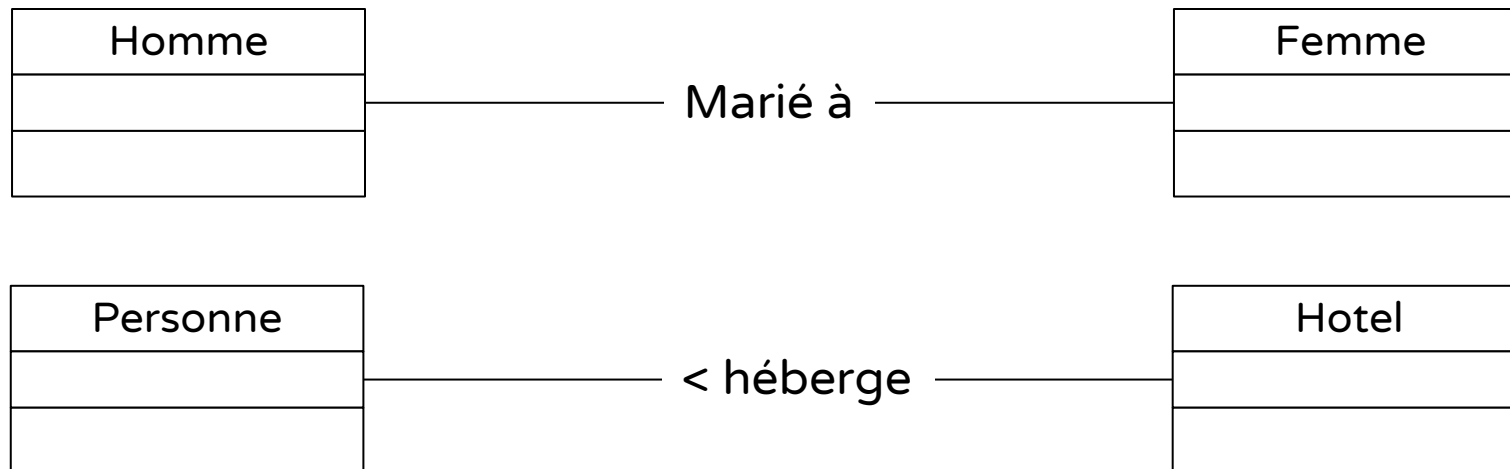


DIAGRAMME DE CLASSES – RÔLES

Préciser le rôle d'une classe vis-à-vis de l'autre

- Pas besoin de le faire lorsque le nom est trivial pour ne pas alourdir le schéma inutilement



DIAGRAMME DE CLASSES — CARDINALITÉS

Nombre d'éléments participant à la relation

x exactement X (entier > 0)

- 7, 11

x..y de x à y (entier $y \geq x$)

- 1..1, 1..10, 0..n

* (ou n) de 0 à plusieurs

- 0..*, 0..n

x..* (ou x..n) de x à plusieurs ($x \geq 0$)

- 0..*, 5..n, 10..*

DIAGRAMME DE CLASSES — CARDINALITÉS

Un hôtel héberge plusieurs personnes (0 à plusieurs)

Une personne est hébergée dans un seul hôtel (à la fois)

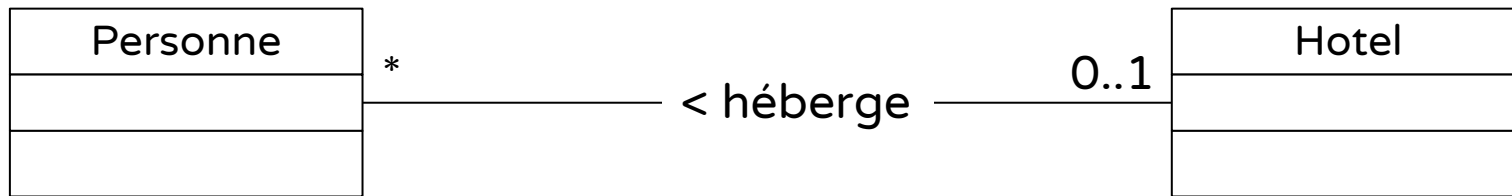


DIAGRAMME DE CLASSES — UNIDIRECTIONNELLE

Par défaut, une association est navigable dans les deux sens

La réduction à une direction « unidirectionnelle » indique que les instances d'une classe ne connaissent pas les instances d'une autre

Un électeur connaît le candidat, pas l'inverse n'est pas vrai

- Donc, un candidat n'aura pas d'attribut électeur ou liste d'électeurs
- Mais un électeur aura un attribut candidat

