UML introduction

Alix Ducros et Valentin Lachand
Département Informatique
Université Claude Bernard Lyon 1

Basé sur les cours d'Aurélien Tabard et Yannick Prié

Plan

1. Introduction au langage de modélisation UML
   - points de vue et diagrammes
   - cas d'utilisation, analyse, conception, implémentation
2. Les cas d'utilisations
   - Présentation standard des CU
   - Rédaction de cas d'utilisation
3. Notion de classes et objets et leur diagramme
   - introduction aux classes, aux objets
   - notion de relation, de composition et d'héritage
   - recherche d'un diagramme de classes à partir du cahier des charges
4. Modèle dynamique
   - diagramme de séquences, de collaboration, d'état et d'activité
   - réalisation des cas d'utilisation par les diagrammes de séquences
   - réalisation des cas d'utilisation par les diagrammes de collaboration

UML, un aperçu

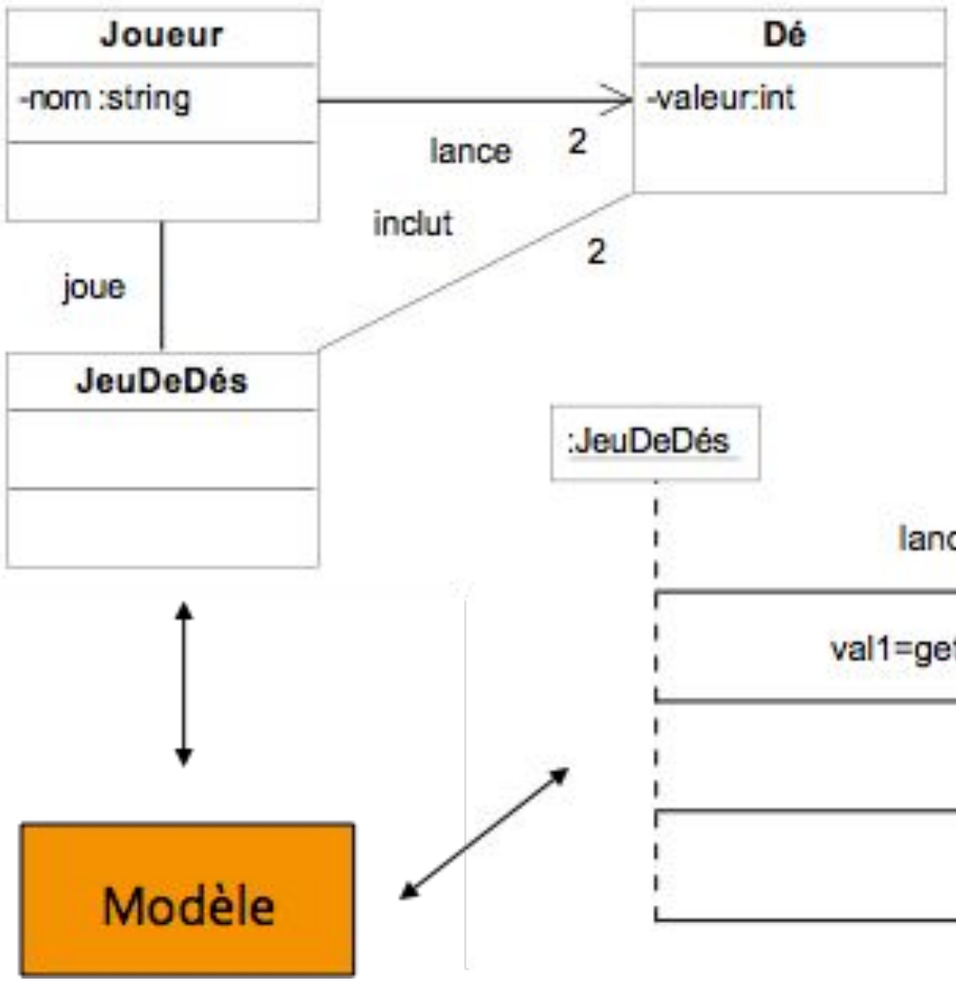


diagramme de classe,
vue statique

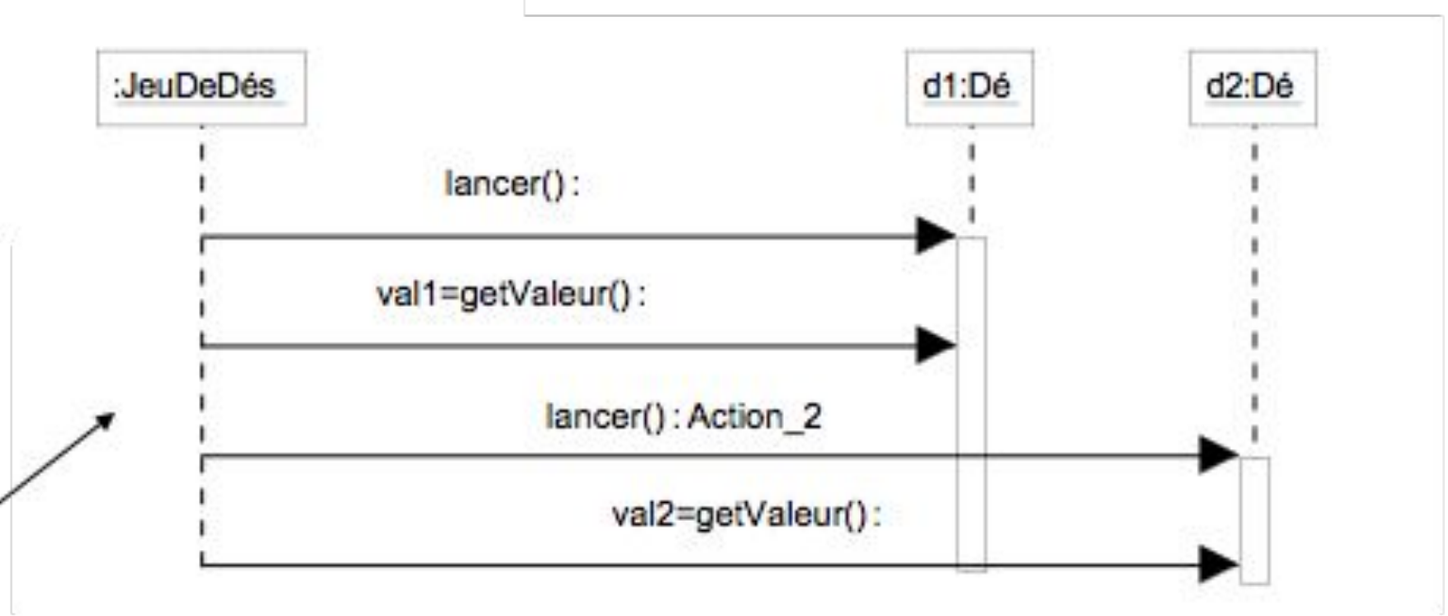
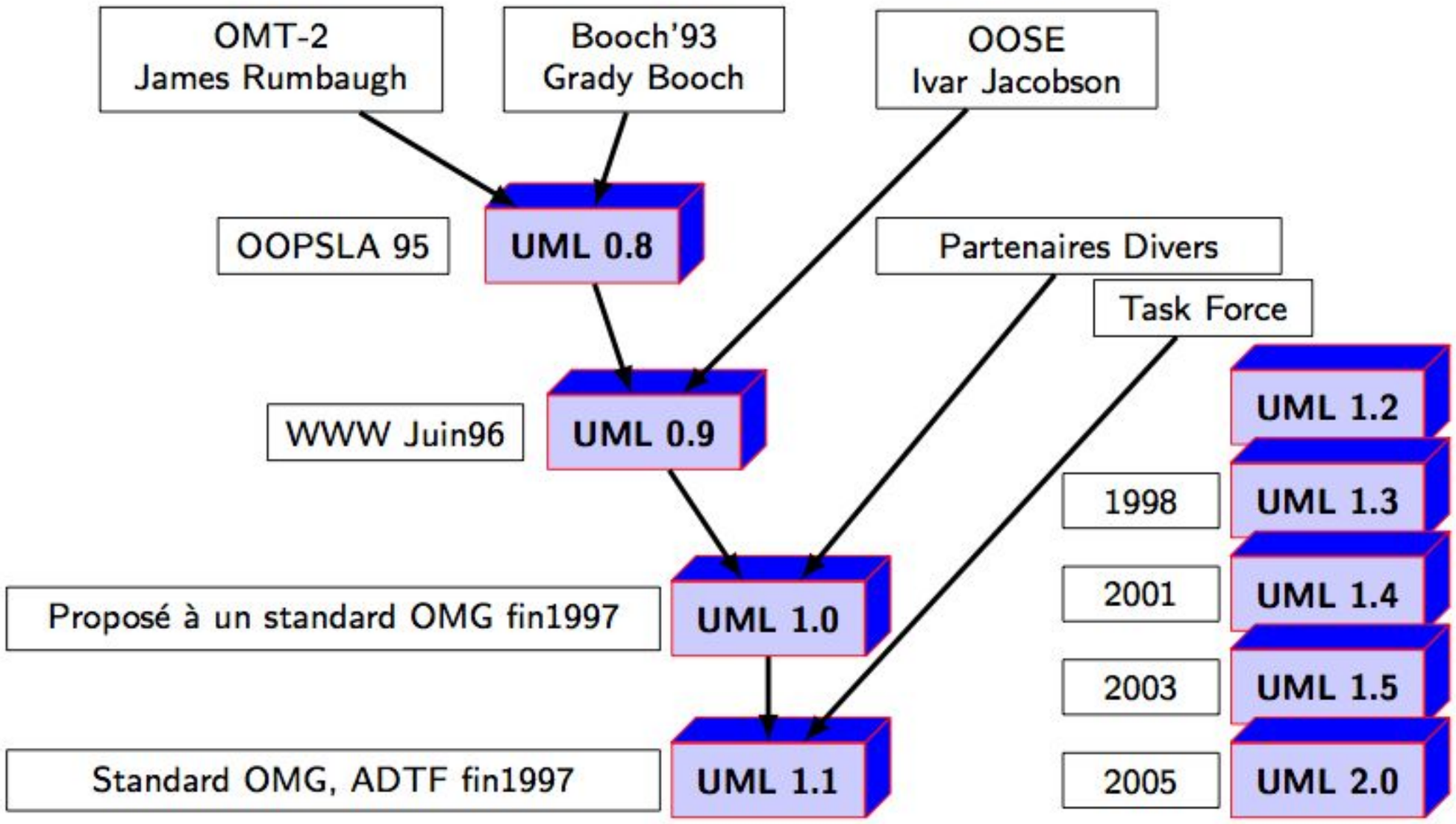


diagramme de séquences,
vue dynamique

Une histoire concise



Unified Modeling Language

- Combinaison de principes à succès
  - modélisation de données, de l'activité, objet, en composants…
- Objectif
  - visualiser / spécifier / construire / documenter les artefacts de la conception d'une application
- La norme elle-même
  - méta-modèle et familles de diagrammes
- Utilisation
  - pas de méthode préconisée
  - pas de spécification technologique

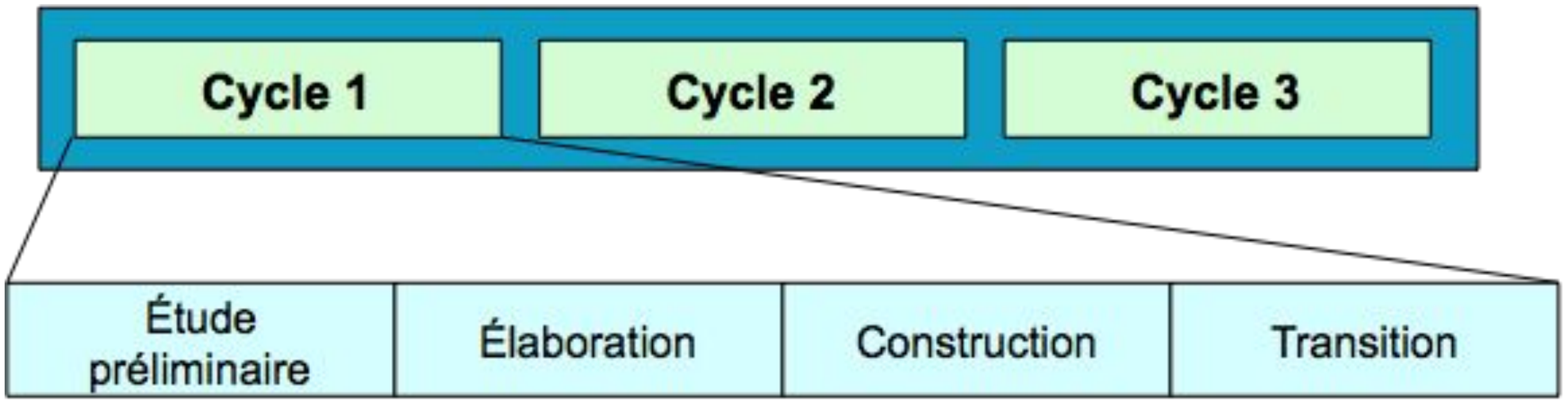
Pourquoi modéliser

Un modèle est une simplification de la réalité qui permet de mieux comprendre le système à développer.

Il permet :

- De visualiser le système comme il est ou devrait être ;
- De valider le modèle vis à vis des clients ;
- De spécifier les structures de données et le comportement du système ;
- De fournir un guide pour la construction du système ;
- De documenter le système et les décisions prises.

UML dans le processus



Cas d’utilisations

Diagrammes statiques

Diagrammes dynamiques

Principes de modélisation

1. Le modèle doit être connecté au réel ;
2. Un modèle peut être exprimé avec différents niveaux de précision ;
3. Un simple modèle n'est pas suffisant, il y a plusieurs façons de voir un système, par ex. dans le bâtiment :
   - plan de masse ;
   - vue de face, de coté, ...
   - plan des niveaux ;
   - plan électrique ;
   - plan de plomberie ;
   - plan des calculs de construction ;

Un language

- UML **n’est pas** une méthode ;
- UML est un **langage** de modélisation objet ;
- UML a été adopté par toutes les méthodes objet ;
- UML est dans le domaine public, c’est une **norme**.

Un langage pour :

- visualiser
  - chaque symbole graphique a une sémantique.
- spécifier
  - de manière précise et complète, avec un minimum d’ambiguïté.
- construire
  - les classes, les bases de données peuvent être générées automatiquement.
- documenter
  - les différents diagrammes, notes, contraintes, exigences seront présentés dans un document.

Exemple - Messagerie instantanée

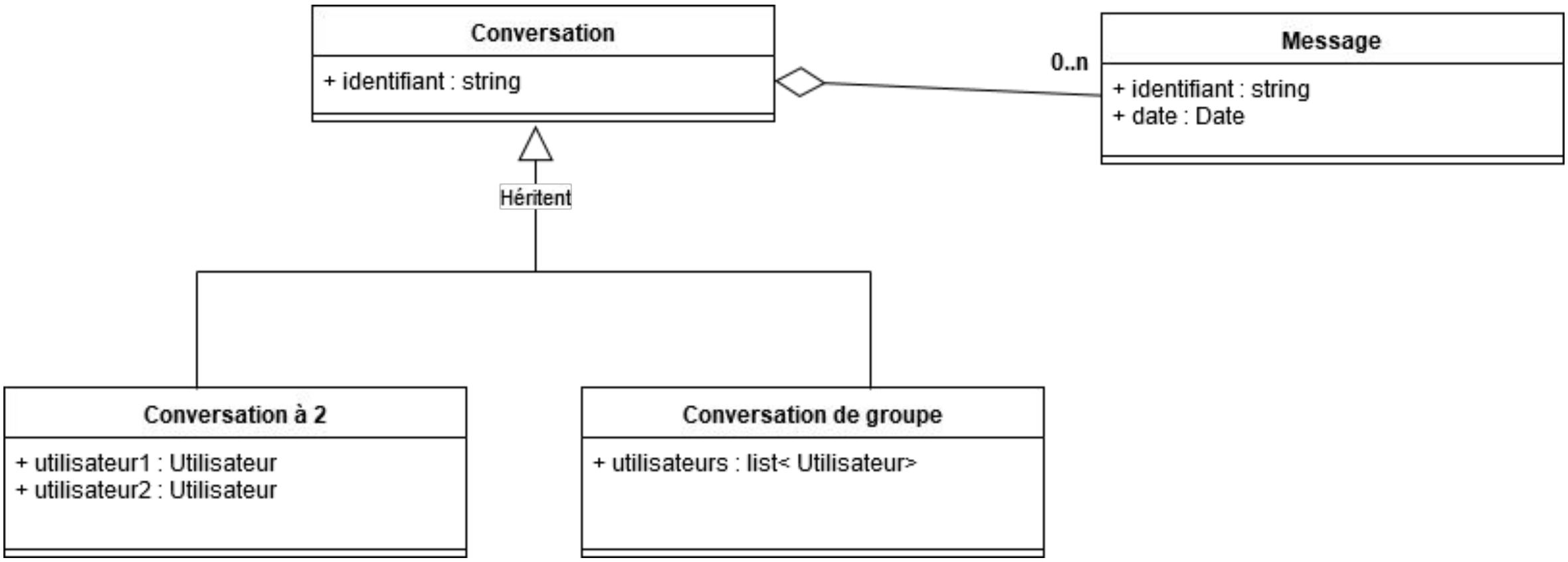
Eléments de modélisation et sémantique

- Eléments de modélisation
  - classe «Utilisateur»
  - attribut « identifiant » associé au type « string »
  - la classe « Utilisateur » peut avoir d’autres attributs
  - etc.

Eléments de modélisation, de visualisation et sémantique

- Eléments de modélisation
  - forment le modèle lui-même
- Syntaxe de UML :
  - comment composer les éléments de modélisation dans les diagrammes
- Sémantique de UML
  - ce qu’ils signifient en termes objets
  - une classe peut avoir des instances
  - une instance a des attributs
  - une instance peut être valeur d’attribut
  - etc.

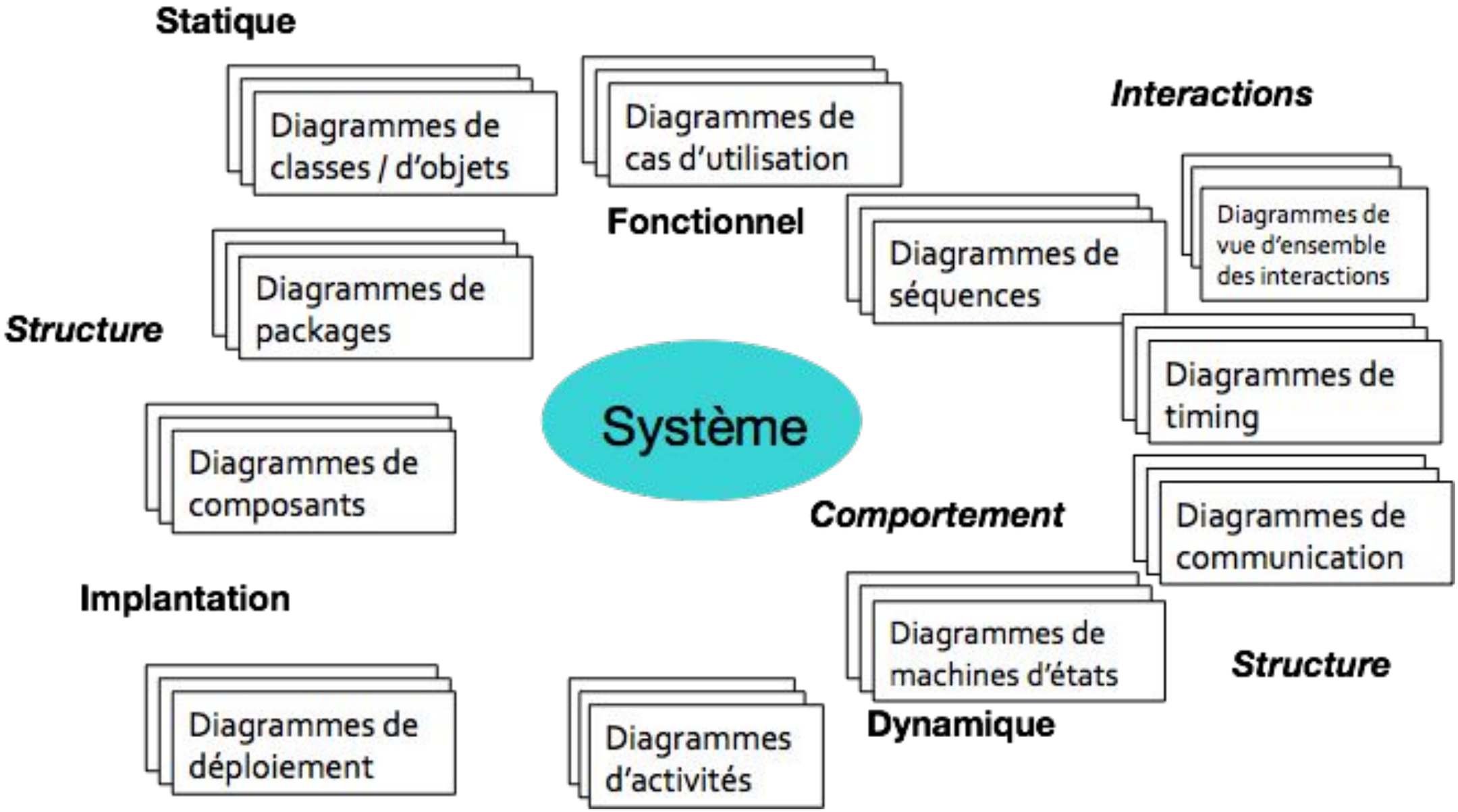
Crash course - Modélisation Objet



Objectifs d'UML

- **Montrer les limites d'un système** et ses fonctions principales (pour les utilisateurs) à l'aide des cas d'utilisation et des acteurs
- **Illustrer** les réalisations de CU à l'aide de diagrammes d'interaction
- **Modéliser la structure statique** d'un système à l'aide de diagrammes de classes, associations, contraintes
- **Modéliser la dynamique**, le comportement des objets à l'aide de diagrammes de machines d'états
- **Révéler l'implantation physique** de l'architecture avec des diagrammes de composants et de déploiement
- Possibilité d'étendre les fonctionnalités du langage avec des stéréotypes
- Un langage utilisable par l'homme et la machine : permettre la génération automatique de code, et la rétro-ingénierie

Panorama d'UML



Modèles, vue et diagrammes UML

- Modèle
  - abstraction d'un système, composée d'un ensemble d'éléments de modèle
  - ce qui est construit par et ce qui est perçu au travers des diagrammes (par le concepteur, le lecteur)
- Vue
  - projection d'un modèle suivant une perspective qui omet les éléments non pertinents pour cette perspective. Elle se manifeste dans des diagrammes
  - ex. : vue statique, vue fonctionnelle…
- Diagramme
  - présentation graphique d'éléments de visualisation représentant des éléments de modèle (graphe)
  - ex. : diagramme de classes, de séquences…

3 modes d’utilisation d’UML

- Esquisse
  - conception / communication
  - incomplétude
- Plan
  - exhaustivité
  - outils bidirectionnels
- Programmation
  - Model Driven Architecture / UML exécutable
  - implantation automatique
  - réaliste ?

Focus sur
les diagrammes

Focus sur
le méta-modèle

Conception et UML

- Différentes façons de voir UML : différentes façons de penser
  - la conception
  - l’objectif et l’efficacité d'un processus de génie logiciel
- donc
  - essayer de comprendre le point de vue de l’auteur pour chaque publication / site sur UML
- UML n’est pas une méthode… mais des principes de conception orientée objet sont sous-jacents
  - aux diagrammes
  - aux façons de les présenter
- donc
  - difficile de présenter uniquement les diagrammes
  - on parlera aussi de méthode, de bonnes pratiques

UML et la règle

- Deux types de règles pour l'utilisation d'UML
  - *normatives*, comment il faut faire, comité d'experts : normes
  - *descriptives*, comment les gens font, usages, modes : conventions d'utilisation
  - peuvent être en contradiction avec la norme (surtout pour UML2)
- Règles
  - utiliser le sous-ensemble d'UML qui vous convient
  - droit de supprimer n'importe quel élément d'un diagramme
  - droit d'utiliser n'importe quel élément d'un diagramme dans un autre
    - ce qui compte pour les auteurs d'UML, c'est le méta-modèle, pas les diagrammes
  - liberté de dessiner ce que l'on veut
    - surtout en mode esquisse, sur papier ou au tableau

Généralité sur les diagrammes

Aurélien Tabard - Université Claude Bernard Lyon 1

Classifier

Classifier

- pour regrouper en familles des éléments similaires d’un modèle pour ne pas multiplier les symboles différents dans les diagrammes

Ornements textuels

- associés à des éléments du modèle
- certains mots-clé sont prédéfinis par UML

Notation

- << mot-clé >>
- ex. <<use case>> << interface >> << class >>

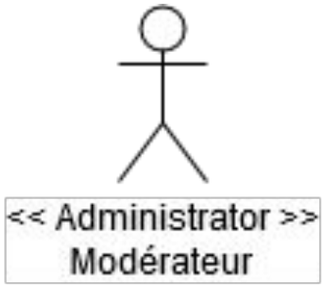
Classifier

<< class >>
Utilisateur
+ identifiant : string
+ envoyerMessage(Message): boolean

Stéréotypes

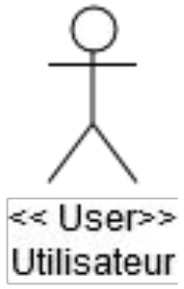
- Permet d'étendre le vocabulaire UML en dérivant des nouveaux éléments à partir d'éléments existants
  - par ajout de valeur étiquetées (nom=valeur)
- Notation : << stéréotype >>

Stéréotype


<< Administrator >>
Modérateur

<< Owner>>
Marc Zouquerberg

<< Administrator >>
Administrateur


<< User>>
Utilisateur

<< Administrator >>
Agent de la DGSI

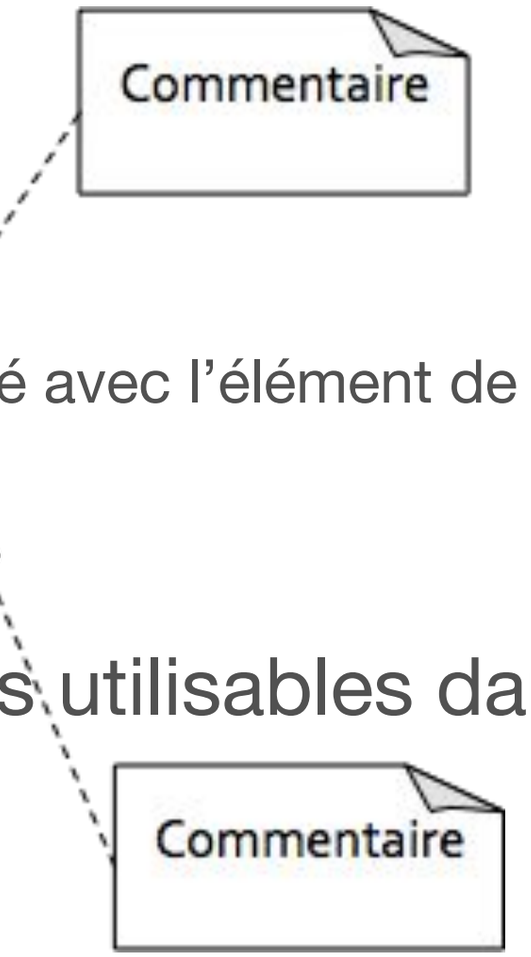
Contraintes

- Relation sémantique quelconque
  - concernant un ou plusieurs éléments du modèle
  - définissant des propositions devant être maintenues à Vrai pour garantir la validité du système modélisé
- Notation : {contrainte}
  - contenu formel ou informel
  - à côté des éléments concernés
  - ex. {frozen}, {jamais détruit !}, {x - y < 10}
- Certaines sont prédéfinies
  - ex. xor, ordered
- D’autres créées par l’utilisateur
  - langue, pseudo-code, OCL…

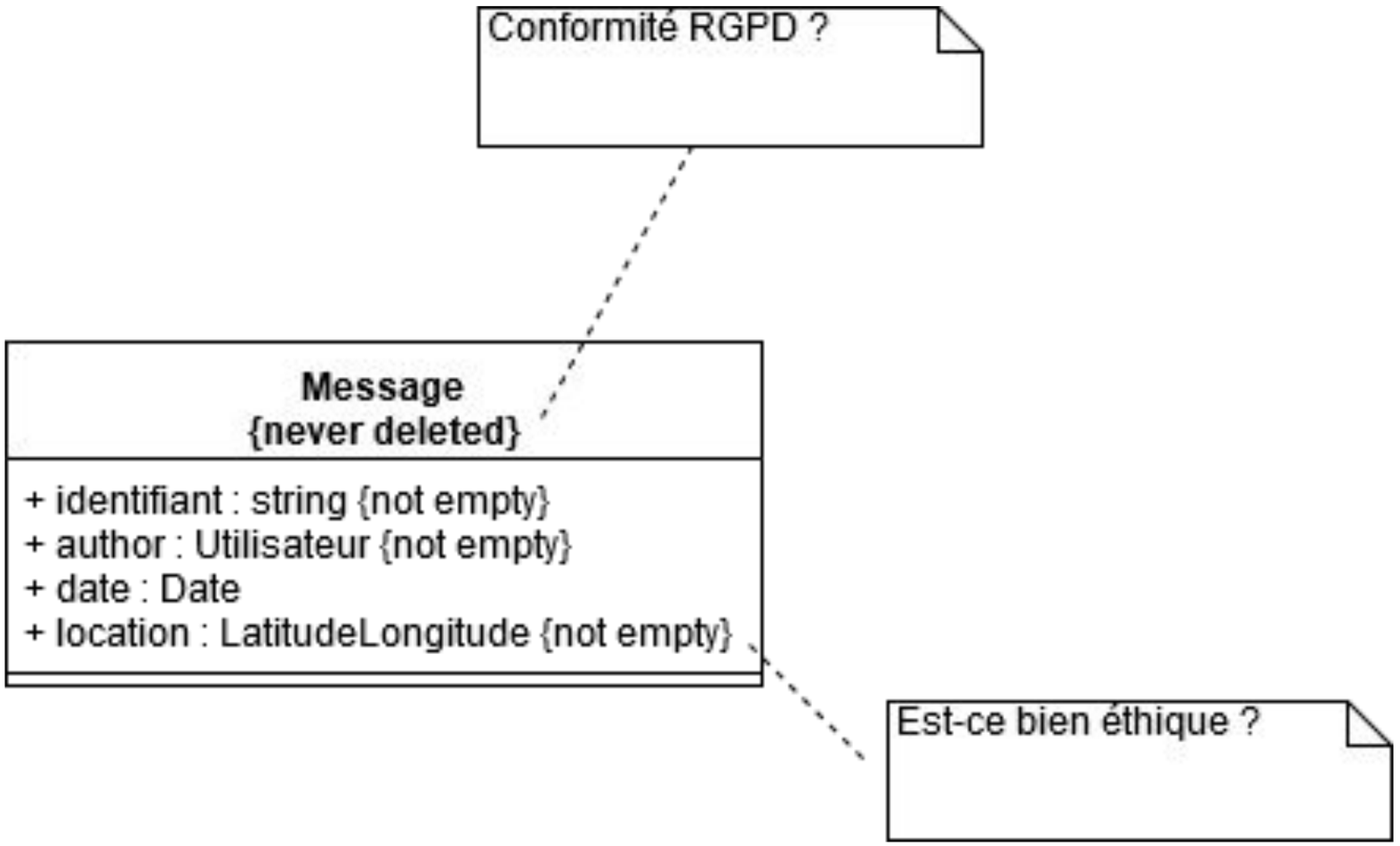
Contraintes

Commentaires

- Commentaire
  - annotation quelconque associée à un élément du modèle
  - pas de sémantique pour le modèle
- Notation : note
  - rectangle avec coin replié, lien pointillé avec l'élément de visualisation concerné
  - cercle en bout de ligne : plus précis
- Il existe des mots-clé prédéfinis utilisables dans les commentaires
  - ex. « besoin », « responsabilité »



Commentaires



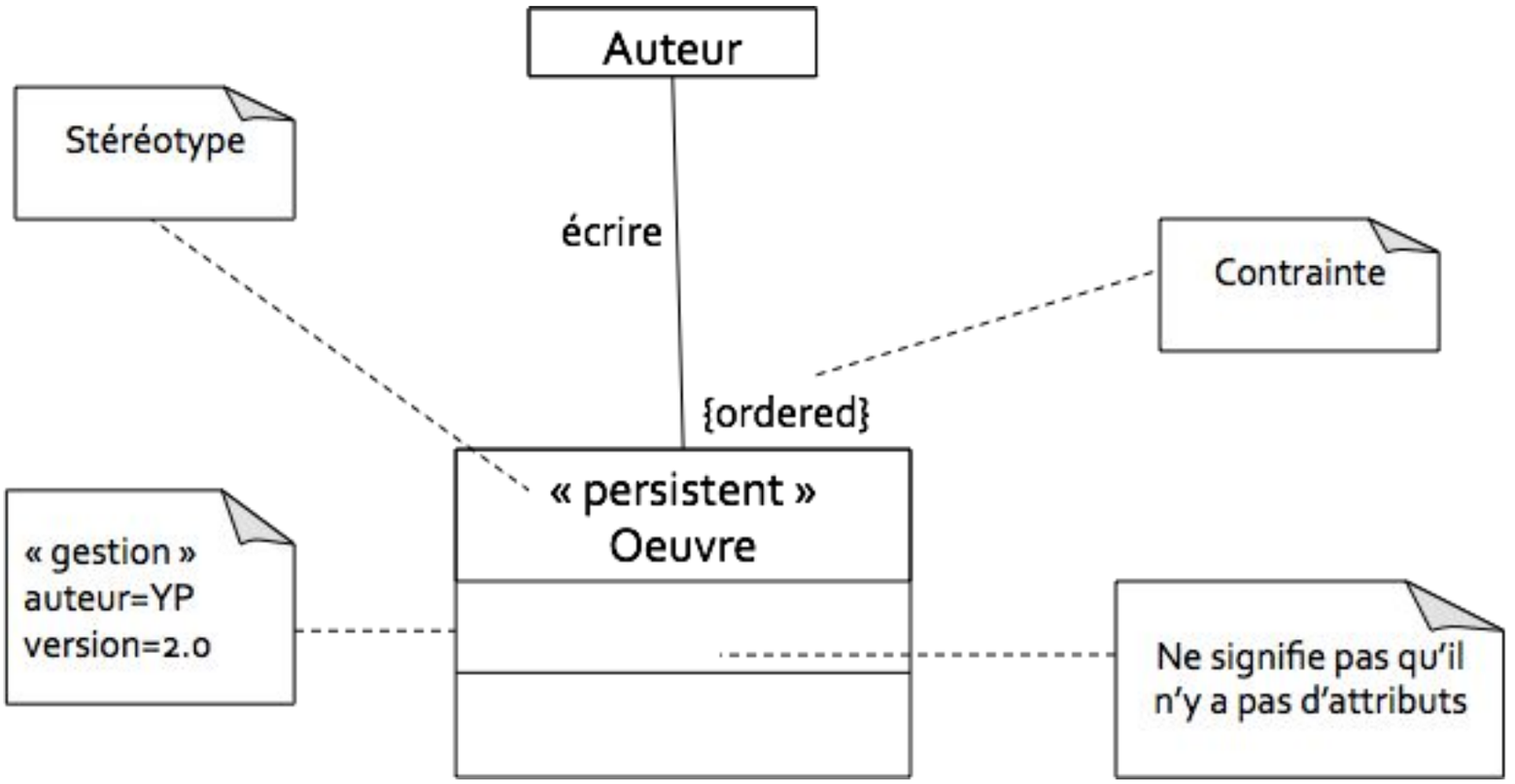
Dépendances

- Relation sémantique faible
  - relation d'utilisation unidirectionnelle entre deux éléments
- Notation
  - flèche pointillée de l'élément source vers l'élément cible, éventuellement stéréotype / mot-clé pour préciser le type de dépendance

Le module de connexion dépend du module de gestion des comptes

- Conséquences :
  - Toute modification dans le module Gestion des Comptes peut avoir des conséquences sur le fonctionnement du module Connexion
  - Une modification dans le module Connexion n'a au vu de cette dépendance pas de conséquence sur le module de Gestion des Comptes

En résumé

- Diagramme = association d’éléments de dessin
  - formes nœuds
  - relation de graphe
  - formes conteneurs
  - texte
- Sémantique graphique importante
  - graphe
  - contenant / contenu
  - Proximité
- Liberté pour le reste (positions…)
- N'importe quelle information peut être supprimée dans un diagramme
  - attention : pas de déduction due à l'absence d'un élément

Exemple général