

UML – Unified Modeling Language

1/4 : introduction

Yannick Prié

UFR Informatique – Université Claude Bernard Lyon 1

M1 MIAGE – SIMA / M1 Informatique MIF17

2008-2009

Objectifs de ce cours

- Pourquoi UML ?
- Petite histoire d'UML
- Principes généraux
- Quelques diagrammes

UML en un transparent

- UML = Unified Modelling Language
- Unification
 - de nombreux langages de modélisation graphique OO des années 1990,
 - de diagrammes et de principes de modélisation à succès
- Défini par l'OMG (Object Management Group)
- Un métamodèle et divers diagrammes dont
 - diagrammes de classes
 - diagrammes de séquences
- Utilisé pour concevoir, comprendre, échanger, à propos de systèmes d'information

Plan

- Histoire d'UML
- Diagrammes et modélisation
- Généralités sur les diagrammes

Un foisonnement de méthodes

- Fin 80 / début 90
 - orientation de plus en plus marquée vers l'objet
- Conséquence naturelle, mise en place de méthodes
 - OOD : Object Oriented Design (Booch, 1991)
 - HOOD : Hierarchical Object Oriented Design (Delatte et al., 1993)
 - OOA : Object Oriented Analysis (Schlaer, Mellor, 1992)
 - OOA/OOD : (Coad, Yourdon, 1991)
 - OMT : Object Modeling Technique (Rumbaugh, 1991)
 - OOSE : Object Oriented Software Engineering (Jacobson, 1992)
 - OOM : Object Oriented Merise (Bouzeghoub, Rochfeld, 1993)
 - Fusion (Coleman et al., 1994)
- Bilan
 - de nombreuses méthodes (>50)
 - ayant des avantages et des inconvénients
 - des concepts assez proches, des notations différentes

Vers une unification

- 1994
 - tentative de normalisation de l'OMG, sans effet
 - Rumbaugh (OMT) rejoint Booch (OOD) chez Rational Software
 - objectif : créer une méthode en commun (méthode unifiée)
- 1995
 - présentation de la version 0.8 de la méthode
 - arrivée de Jacobson (OOSE) chez Rational
- 1996
 - implication de l'OMG (sous pression des industriels pour favoriser l'interopérabilité des modèles)
 - langage unifié UML 0.9 (Unified Modeling Language),
- 1997
 - UML 1.0 sort chez Rational
 - UML 1.1 adopté par l'OMG comme standard officiel

Evolutions d'UML

- 1997-2003
 - adoption par les entreprises
 - UML 1.1 à UML1.5 : modifications/améliorations
- 2005
 - UML 2.0
 - quelques nouveaux diagrammes
 - changements importants au niveau du méta-modèle, pour permettre d'utiliser UML pour la programmation
- 2007
 - UML 2.1.1
 - À suivre

Unified Modeling Language

- Combinaison de principes à succès
 - modélisation de données (E/A), 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

Plan

- Histoire d'UML
- Diagrammes et modélisation
- Généralités sur les diagrammes

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

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)
 - conforme au méta-modèle UML
- **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...

Exemples de diagrammes

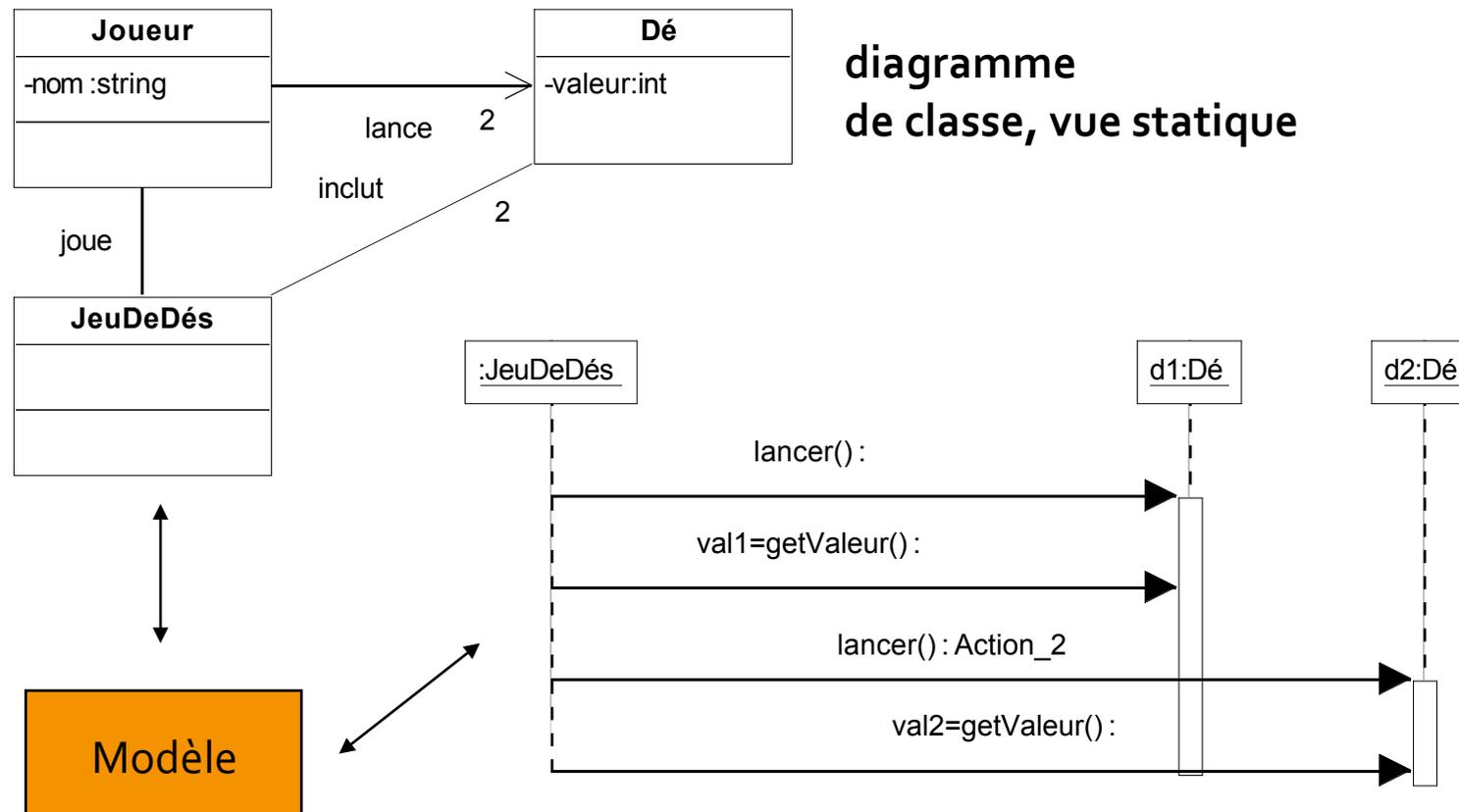
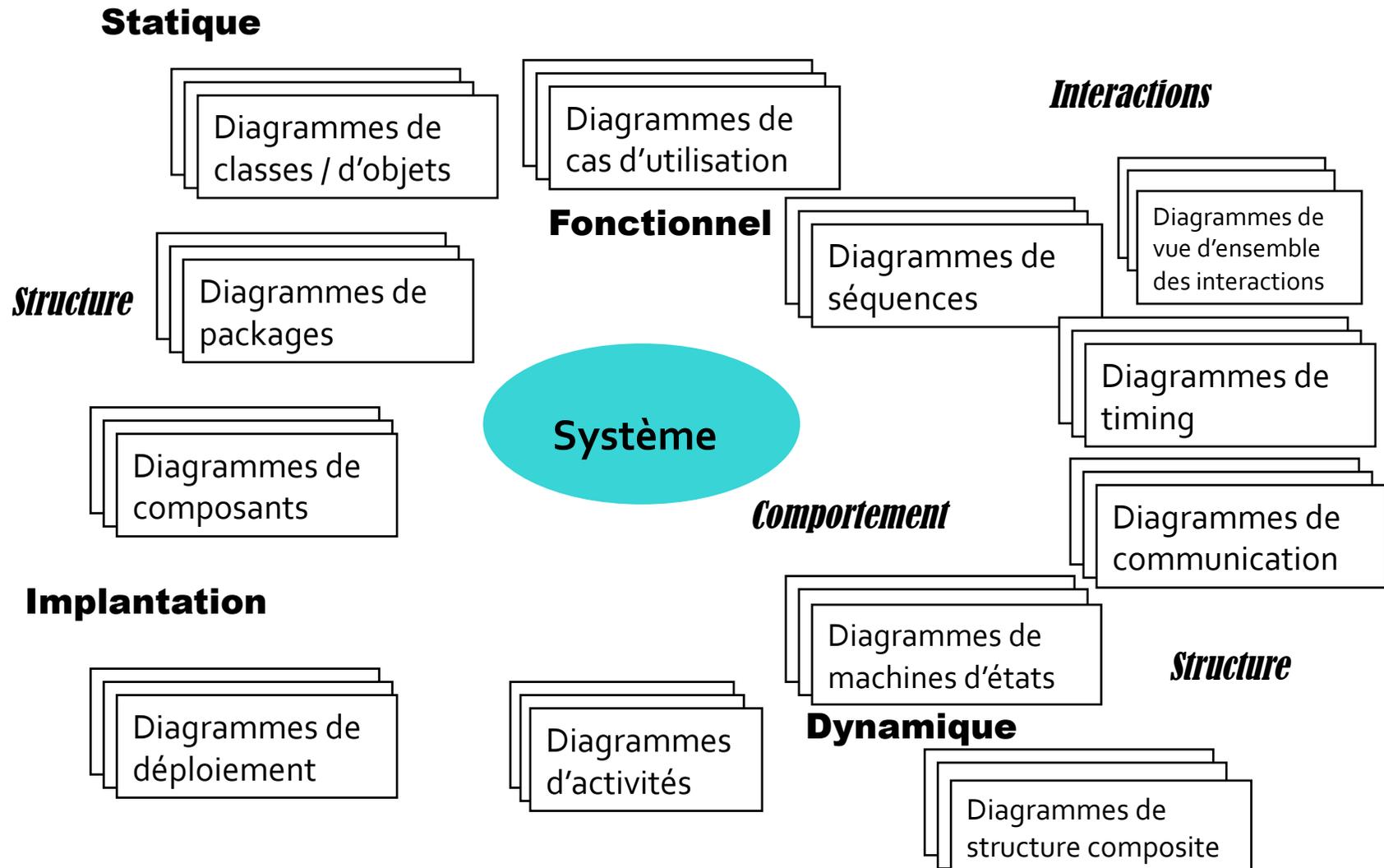


diagramme de classe, vue statique

diagramme de séquences, vue dynamique

Panorama des diagrammes



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

Généralités sur les méthodes OO

- Grandes caractéristiques
 - itératives (vs cascade)
 - analyse et conception tout au long du projet, pas seulement au début
 - centrées sur les cas d'utilisation
 - besoins réels
 - centrées sur l'architecture
- Découpage d'un projet en activités
 - besoins : comprendre dans quoi s'insère le système et ce qu'il doit faire
 - analyse : fonctionnement du système à haut niveau
 - conception : fonctionnement logiciel
 - réalisation : codage
 - tests, déploiement...

À propos de ces cours

- Présentation de UML 2, quelques points de UML 1
 - faites attention à la syntaxe quand vous rencontrez un diagramme
- Présentation = synthèse de nombreuses lectures
 - mixe syntaxe et utilisation
 - synthèse personnelle des bonnes pratiques présentées
- Présentation d'UML non exhaustive
 - ce cours contient beaucoup de choses utiles
 - pour plus de précisions : livres de référence
 - pour la description exacte (syntaxe et sémantique) : <http://www.omg.org/uml>
- UML et le web
 - beaucoup de sites web parlent d'UML
 - on trouve du bon et du moins bon

UML et la règle

- Deux types de règles pour l'utilisation d'UML
 - normatives
 - comment il faut faire, comité d'experts : norme
 - descriptives
 - comment les gens font, usages, modes : conventions dans l'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

Plan

- Histoire d'UML
- Diagrammes et modélisation
- Généralités sur les diagrammes

Mots-clé

- Objectif
 - regrouper en familles des éléments similaires d'un modèle
 - pour ne pas multiplier les symboles différents dans les diagrammes
- Ornaments textuels
 - associés à des éléments du modèle
 - certains mots-clé sont prédéfinis par UML
- Notation
 - « mot-clé »
 - ex. « abstract »

Valeurs étiquetées

- Objectif
 - attacher une information arbitraire à un élément de modélisation
- Paires (nom,valeur)
 - associées à des éléments de diagramme
- Notation : nom=valeur
 - ex. : auteur=YP, version=1.3

Stéréotypes

- Étendre le méta-modèle
 - définir des profils de valeurs étiquetées pour des éléments de modélisation
 - plus formels que les mots-clé
- Notation : « stéréotype »
 - ex. « gestion » avec des valeurs étiquetées associées si nécessaire
- Certains sont prédéfinis par UML
- Possibilité d'associer une icône
 - forme visuelle déterminée
 - ex. : pour « control »

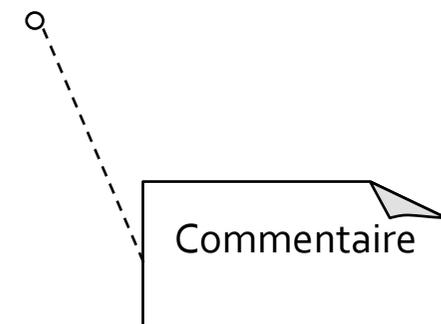


Contraintes

- 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...

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**
 - ex. « besoin », « responsabilité »



Dépendance

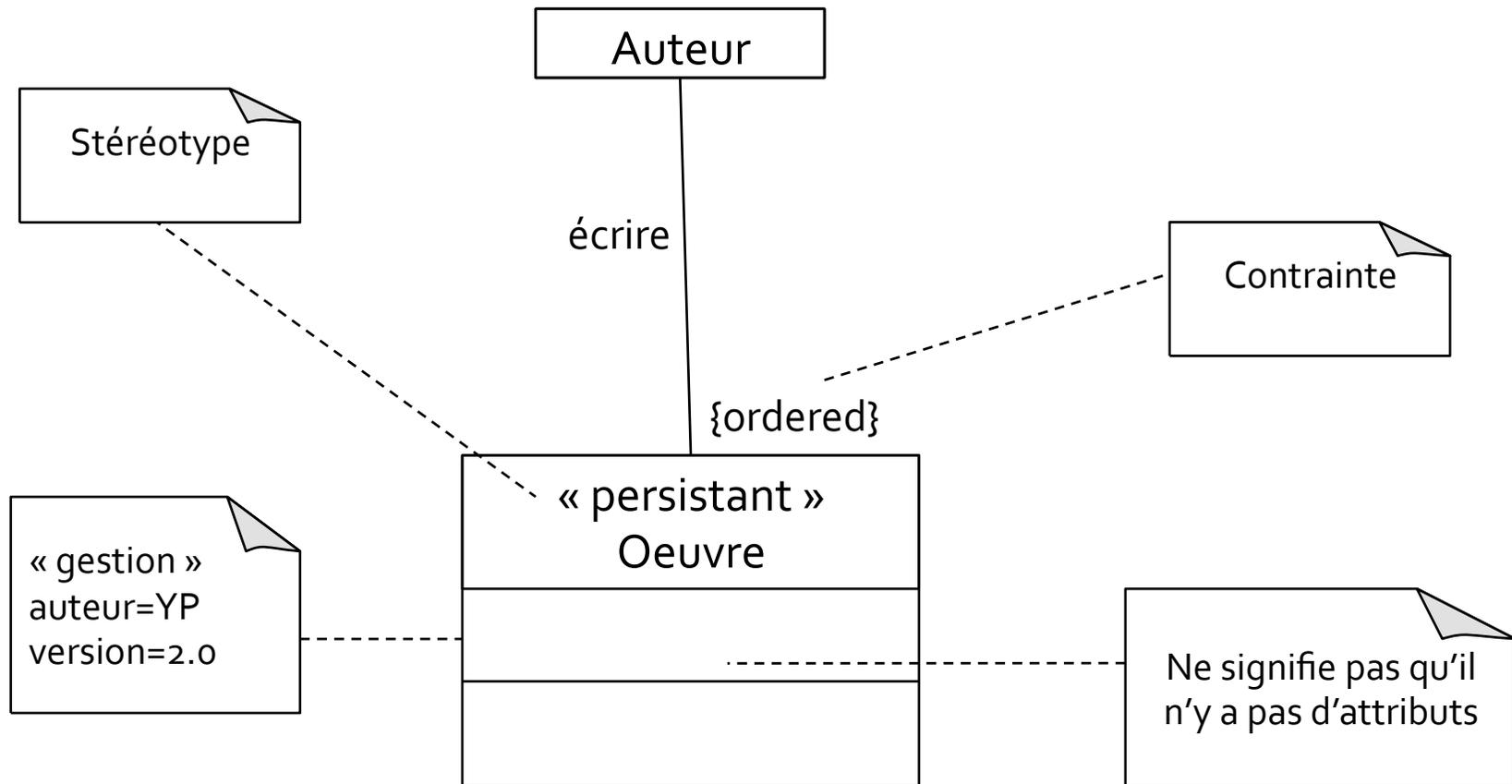
- Relation sémantique faible
 - relation d'utilisation unidirectionnelle entre deux éléments
 - relation sémantique non structurelle entre client et fournisseur
- Notation
 - flèche pointillée de l'élément source vers l'élément cible, éventuellement stéréotype



Diagrammes

- Diagrammes
 - éléments de dessin dont on dispose
 - formes nœuds et relation de graphe
 - formes conteneurs
 - texte
- Principe
 - la sémantique d'UML impose de conserver
 - graphe
 - contenant / contenu
 - proximité
 - liberté pour le reste (positions...)
 - n'importe quelle information peut être supprimée dans un diagramme
 - pas de déduction due à l'absence d'un élément

Exemple général



A suivre

- UML 2/4 : diagrammes statiques
- UML 3/4 : diagrammes dynamiques et d'interactions
- UML 4/4 : concepts avancés