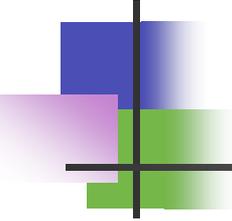




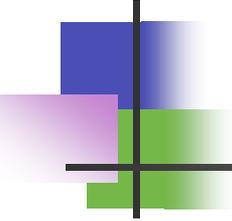
Chapitre1 : Introduction à UML

M.BOUABID, 09-2013



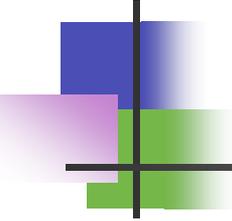
Plan

- Pourquoi modéliser?
- Définition d'UML
- Historique d'UML
- Sommaire des diagrammes d'UML 2
- Utilisation des diagrammes d'UML 2
- Architecture 4+1 et UML
- Outil d'aide à la conception



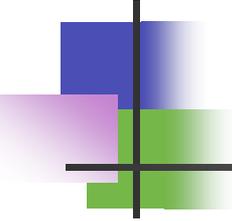
Modéliser

- Un modèle
 - C'est une représentation simplifiée d'une entité du monde réel en vue de la décrire, de l'expliquer
 - Exemple : plan
- Langage de modélisation
 - Langage graphique pour représenter, communiquer les divers aspects d'un système



Pourquoi modéliser? (1/2)

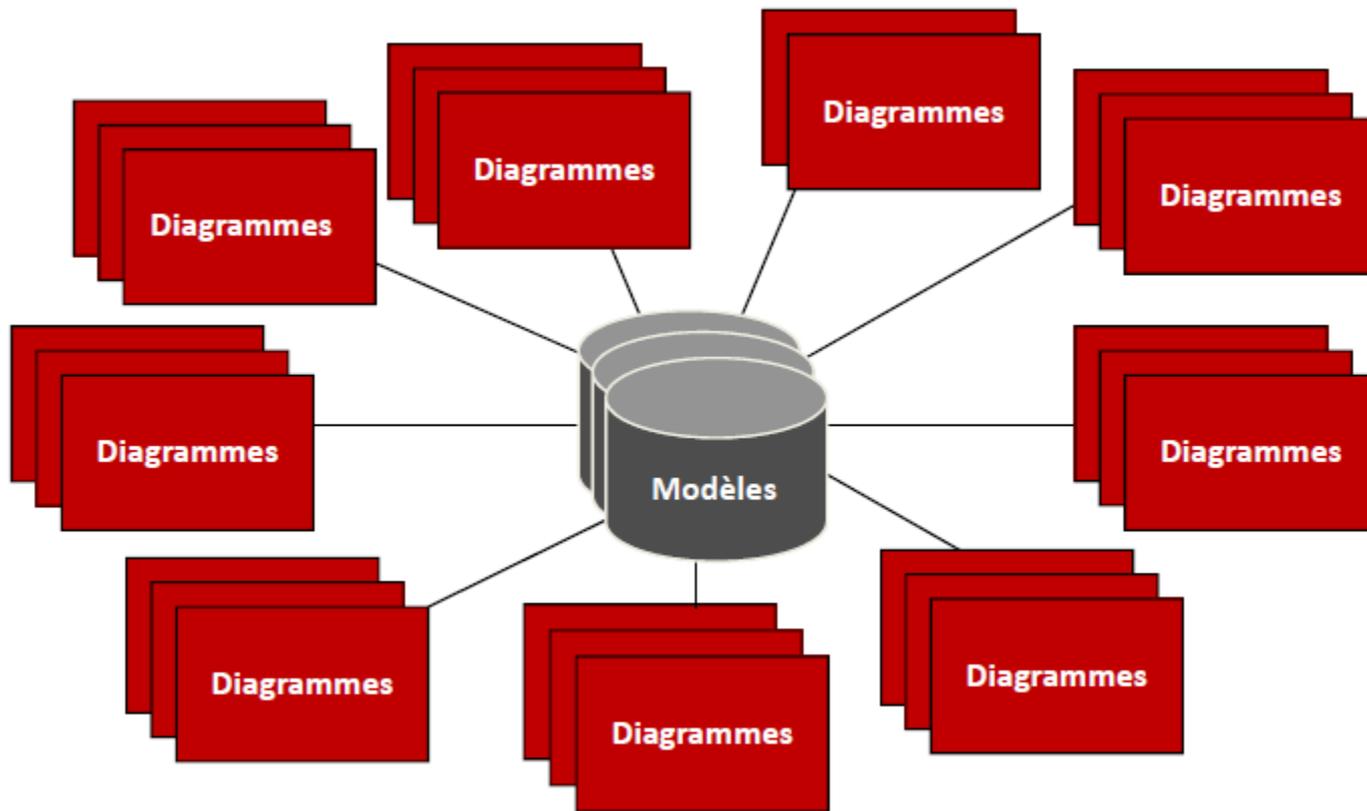
- Les modèles ont différents usages
 - Ils optimisent l'organisation des systèmes
 - La modélisation de la structure d'une entreprise en départements et services permet d'avoir une vision simplifiée du système et d'en assurer une meilleure gestion



Pourquoi modéliser? (2/2)

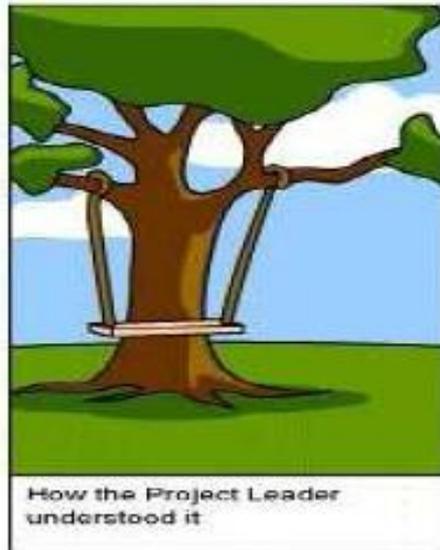
- Ils permettent de se focaliser sur des aspects spécifiques d'un système sans s'embarrasser des données non pertinentes
 - S'intéresser à une vue statique sans se soucier des autres vues
 - Ils permettent de décrire avec précision et complétude les besoins sans connaître les détails du système
- Un modèle est composé de plusieurs diagrammes

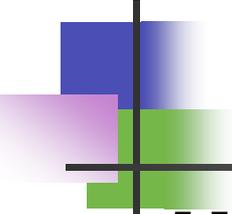
Modèles vs. Diagrammes



Langage de modélisation

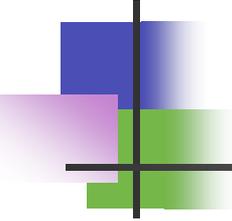
- L'expression d'un modèle se fait dans un langage de modélisation
 - Le physicien utilise les mathématiques
 - L'informaticien utilise l'UML





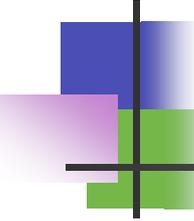
Qu'est ce qu'UML? (1/2)

- **UML** : Unified Modeling Language (Langage de Modélisation Unifié)
 - Il est apparu en 1997, UML est issu de l'unification de nombreux langages de modélisation graphiques OO
 - Il est contrôlé par OMG « Object Management Group »
 - Mission OMG: faciliter l'interopérabilité



Qu'est ce qu'UML? (2/2)

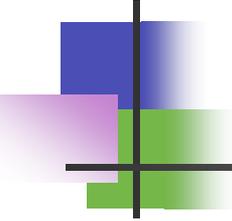
- « UML est une famille de notations graphiques s'appuyant sur un métamodèle unique qui permet de décrire et de concevoir des systèmes logiciels, OO en particulier » (Fowler, 2004)
 - Notation: représentation graphique et la syntaxe du langage de modélisation
 - Métamodèle: un diagramme qui décrit la notation



Méthodologie de développement

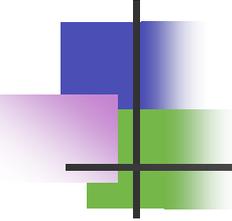
- Une méthodologie de développement définit à la fois:
 - Un langage de modélisation
 - La méthode à suivre lors de la conception

- UML
 - une notation (langage de modélisation)



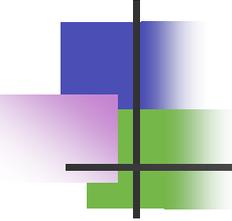
Historique d'UML (1/2)

- L'industrie commence à utiliser massivement les langages de programmation OO
 - Le besoin de de « penser objet » indépendamment du langage d'implémentation
 - Le besoin d'unifier les langages de modélisation en un standard unique



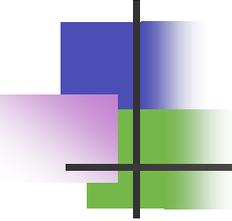
Historique d'UML (2/2)

- 1995: Booch & Rumbaugh présentent à la conférence OOPSLA Unified Method 0.8
- 1996: Jacobson rejoint le groupe et ils améliorent la méthode unifiée (version 0.9)
- 1997: Rational software, employeur du trio, publie la documentation d'UML 1.0
- 1997: OMG adopte UML et publie une version améliorée 1.1
- La version actuelle est 2.3



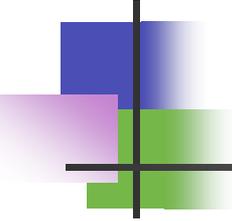
UML 1 dans les projets (1/2)

- UML 1 est utilisé lors de la phase de l'analyse et de la conception préliminaire du système
 - Il sert à spécifier les fonctionnalités attendues du système (diagramme des cas d'utilisation et de séquence) et à décrire l'architecture (diagramme de classe) c'est tout!
 - La description de la partie comportementale (diagramme d'activités et d'états) est peu utilisée



UML 1 dans les projets (2/2)

- En résumé
 - UML 1 s'utilise seulement durant les phases préliminaires
- Un des objectifs d'OMG est de proposer un paradigme guidé par des modèles décrivant:
 - Le codage
 - La gestion de la qualité
 - Les tests et vérifications
 - La production de la documentation



Briques de base d'UML (1/2)

- Les éléments

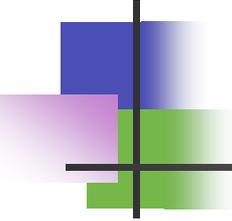
- Ce sont les abstractions essentielles au modèle.

- Les relations

- Les relations expriment les liens existants entre les différents éléments.

- Les diagrammes

- Un diagramme est une représentation visuelle de l'ensemble des éléments qui constituent le système
- Ils servent à visualiser un système sous différents angles (utilisateur, administrateur par ex.)



Briques de base d'UML (2/2)

- Dans les systèmes complexes, un diagramme ne fournit qu'une vue partielle du système
 - L'ensemble des diagrammes réunis permet d'obtenir une vue globale du système à concevoir
 - Chaque diagramme va permettre de modéliser ou spécifier une vue (spécificité) du système à concevoir

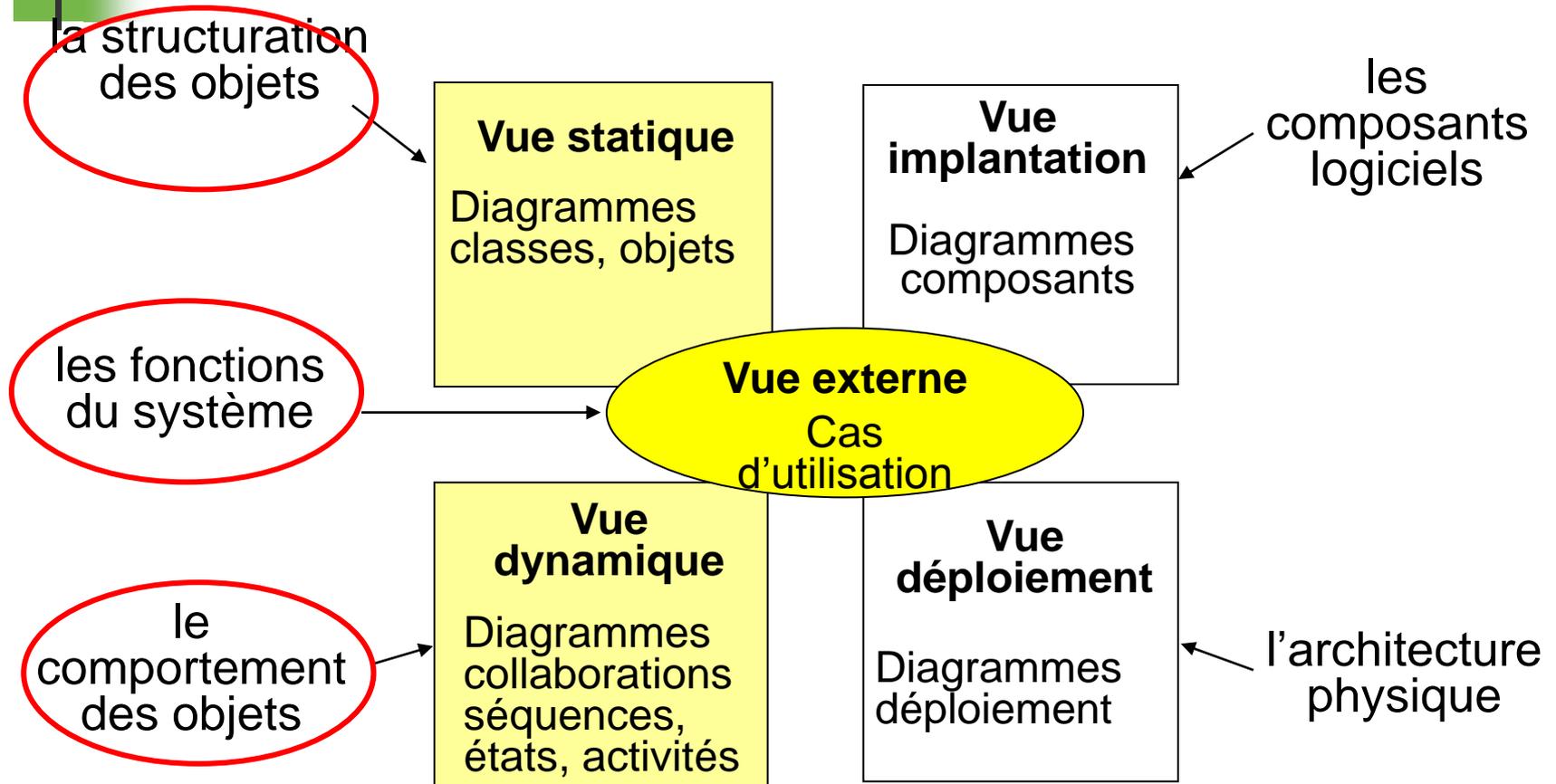
Diagrammes d'UML 2

	Diagrammes	Historique
<input checked="" type="checkbox"/>	Activité	UML 1
<input checked="" type="checkbox"/>	Cas d'utilisation	UML 1
<input checked="" type="checkbox"/>	Classes	UML 1
<input checked="" type="checkbox"/>	Communication	Diagramme de collaboration d'UML 1
<input checked="" type="checkbox"/>	Composants	UML 1
<input checked="" type="checkbox"/>	Déploiement	UML 1
	Vue d'ensemble des interactions	UML 2
<input checked="" type="checkbox"/>	États-transition	UML 1
<input checked="" type="checkbox"/>	Objets	Officieux dans UML 1
<input checked="" type="checkbox"/>	Packages	Officieux dans UML 1
<input checked="" type="checkbox"/>	Séquence	UML 1
	Structure composite	UML 2
	Timing	UML 2

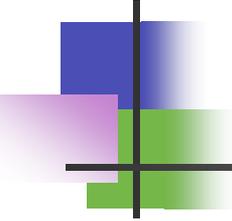
Utilisation des diagrammes

	Recueil des besoins	Analyse	Conception	Déploiement
Cas d'utilisation	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Séquence	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Communication	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Activité	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Classes		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Objets			<input checked="" type="checkbox"/>	
États-transitions			<input checked="" type="checkbox"/>	
Composants				<input checked="" type="checkbox"/>
Déploiement				<input checked="" type="checkbox"/>
Vue d'ensemble des interactions	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Timing	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Structure composite				<input checked="" type="checkbox"/>
Packages		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Architecture 4+1 (1/2)



Concepts utilisables tout au long du projet :
analyse → conception → implantation



Architecture 4+1 (2/2)

- Vue utilisateur
 - Description du modèle vu par les acteurs du système
 - Besoins attendus pour chaque acteur
 - Le **QUOI** et le **QUI**
- Vue logique
 - Définition du système vu de l'intérieur
 - **COMMENT** satisfaire les besoins des acteurs
- Vue d'implémentation
 - Dépendances entre les modules
- Vue de comportement
 - Vue temporelle et technique
 - Mise en œuvre des notions de tâches concurrentes, synchronisation...
- Vue de déploiement
 - Position géographique et architecture physique de chaque élément
 - Le **OÙ**

Outil d'aide à la conception

- Outil: StarUML version 5.0 (Open Source)

- http://sourceforge.net/project/showfiles.php?group_id=152825&package_id=169190&release_id=437438



- Outil: ArgoUML (Open Source)

- <http://argouml.tigris.org/>

