

ANALYSE ET CONCEPTION ORIENTÉE OBJET (UML)

ANNEE DE FORMATION
2019 - 2020

Animateur : Belkassem ECHCHADLI

Introduction à la modélisation

PLAN DE L'UNITÉ

- Pourquoi modéliser
- Les principes de la modélisation
- Anciennes méthodes de conception objet
- Historique d'UML
- Définitions
- Les vues d'UML
- Les diagrammes d'UML
- Les outils de modélisation UML

Pourquoi modéliser

- Un modèle est une simplification de la réalité qui permet de mieux comprendre le système à développer (Construction, réseau inf., site Web).
- Il permet de:
 - Visualiser le système comme il est ou comme il devrait l'être.
 - valider le modèle vis-à-vis des clients.
 - Spécifier les structures de données et le comportement du système.
 - Fournir un guide pour la construction du système.
 - Documenter le système et les décisions prises.
 - Réduire les coûts de développement.
 - Réduire les délais de mise en production.

Les principes

- Le choix du type de modèle que nous créons a une influence profonde sur la manière d'attaquer le problème et de former une solution.
- Tout modèle peut être exprimé à différents niveaux de précision.
- Les meilleurs modèles sont bien reliés à la réalité.
- Un seul modèle ne suffit pas. Tout système un peu compliqué sera mieux appréhendé à travers un ensemble de modèles presque indépendants.

ANCIENNES MÉTHODES DE CONCEPTION OBJET

- **OMT:** (par Rumbaugh) **O**bject **M**odeling **T**echniques a pour **vues**:
 - Statiques: ne modifie pas l'objet.
 - Dynamiques: peut modifier l'objet.
 - Fonctionnelles: séquences d'appels de fonctions d'une utilisation.
- **OOD:** (par Boosh) **O**riented **O**bject **D**esign a pour vues:
 - Logiques: décomposition logicielle.
 - Physiques: décomposition matérielle.
- **OOSE:** (par Jacobson) **O**riented **O**bject **S**oftware **E**ngineering a des vues:
 - basées sur le cycle de vie logiciel.
 - Analyse.
 - Conception.
 - Réalisation/implémentation.
 - Test/maintenance.

DÉFINITIONS

- UML: (**U**nified **M**odeling **L**anguage) est un langage unifié de modélisation.
- UML est un langage graphique (visuel).
- UML est un langage de modélisation objet.
- UML n'est pas une démarche.
- UML a été adopté par toutes les méthodes Objet.

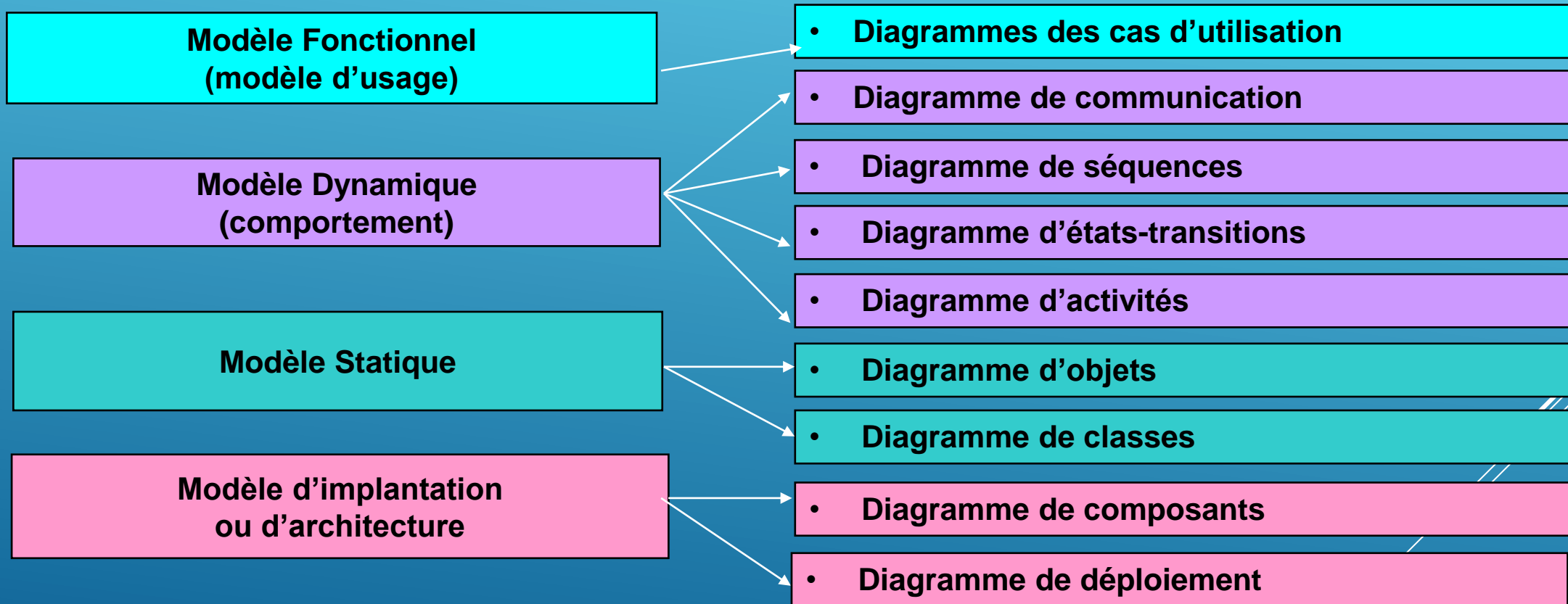
LES VUES D'UML

5 différentes vues:

- **La vue logique:** c'est la définition du système vu de l'intérieur. Elle explique comment satisfaire les besoins des acteurs (**c'est le comment**).
- **La vue des cas d'utilisations:** c'est une vue externe qui représente les besoins attendus par les utilisateurs (**c'est le qui et le quoi**).
- **La vue des processus:** c'est la vue temporelle et technique, elle représente l'architecture et la topologie (**c'est le quand**).
- **La vue d'implémentation:** c'est une vue de bas niveau composée de modules logiciels.
- **La vue de déploiement:** utilisée pour représenter l'architecture et la disposition du matériel (**c'est le où**).

LES DIAGRAMMES D'UML (1/5)

● Il existe 9 diagrammes en **UML 1.1**, et 13 diagrammes en **UML 2.0** répartis en 4 modèles:



LES DIAGRAMMES D'UML (2/5)

● Diagrammes des cas d'utilisation:

Ils répondent à la question "**qu'est-ce que les utilisateurs attendent ?**", c'est-à-dire les relations entre les acteurs et les fonctionnalités du système d'information.

● Diagramme de classes:

Il représente l'ensemble des éléments qui constituent le monde réel et les relations qui existent entre eux.

● Diagramme d'objets:

Il représente les objets et les liens qui les relient, permet de préciser un aspect particulier du diagramme de classe.

LES DIAGRAMMES D'UML (3/5)

- **Diagramme d'activités:** décrit les phases d'évolution du système et modélise les actions effectuées sur ce système.
- **Diagramme d'interaction:**
 - **Diagramme de communication :** montre les relations sémantiquement faibles entre les objets. Spécifie les structures de données et le comportement du système.
 - **Diagramme de séquence:** représente les messages échangés entre les objets qui s'enchaînent de façon séquentielle.

LES DIAGRAMMES D'UML (4/5)

- **Diagramme d'états/transitions:** définit les règles d'évolution, soit le cycle de vie des objets d'une classe.
- **Diagramme de composants:** montre le découpage du système en unités pouvant être distribuées (logiciels).
- **Diagramme de déploiement:** répartition du matériel: machines, systèmes d'exploitation et les liens réseaux entre ces machines.

LES DIAGRAMMES D'UML (5/5)

Les nouveaux diagrammes d'UML 2:

- **Diagramme "interaction overview" ou synthèse des interactions:** c'est un mélange des diagrammes de séquence et d'activités.
- **Diagramme de timing:** décrit les contraintes temporelles sur l'évolution du système.
- **Diagramme des packages:** montre les dépendances des éléments au niveau de la compilation.
- **Diagramme des structures composites:** Comme le diagramme des packages, il montre les dépendances des éléments, mais au niveau de l'exécution.

Merci De Votre Attention

Animateur : Belkassem ECHCHADLI