Cours 3 – Introduction à UML

MODULE INTRODUCTION AU GÉNIE LOGICIEL

MOSTEFAI MOHAMMED AMINE – BATATA SOFIANE

COPYRIGHT © ESI 2015-2016

Objectifs du Cours

Sensibilisation au besoin de modélisation

Introduction au standard UML

Découverte des différents diagrammes UML

Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

Plan du Cours

Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

Modélisation

SECTION 1

Qu'est-ce qu'un modèle ?

Un modèle est une représentation du monde réel

Les modèles représentent le système selon des degrés différents de détails

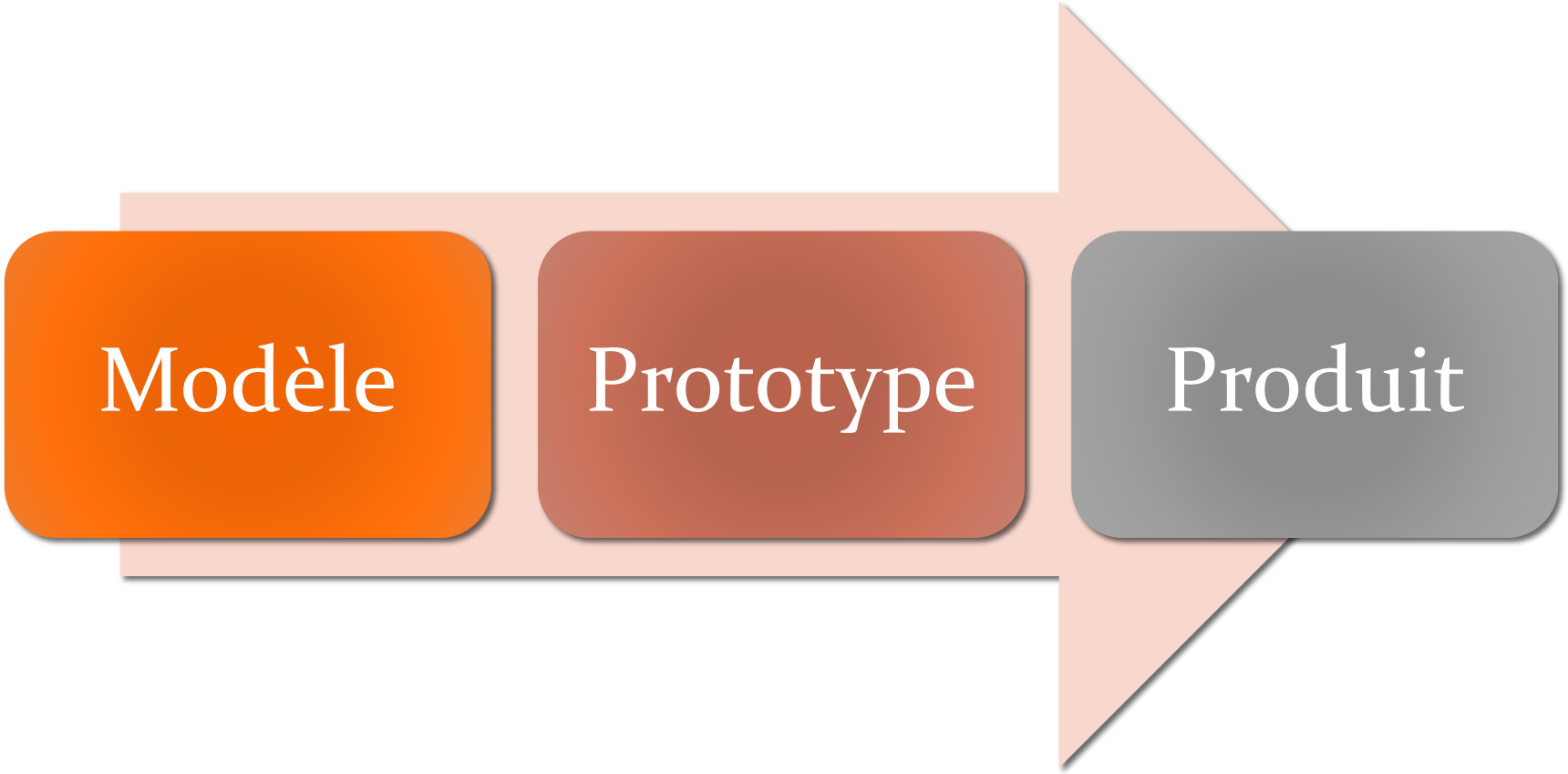
Le modèle est une abstraction

Une description ou une analogie qui permet d'observer un élément difficile à observer directement

Utilise une notation graphique et simple

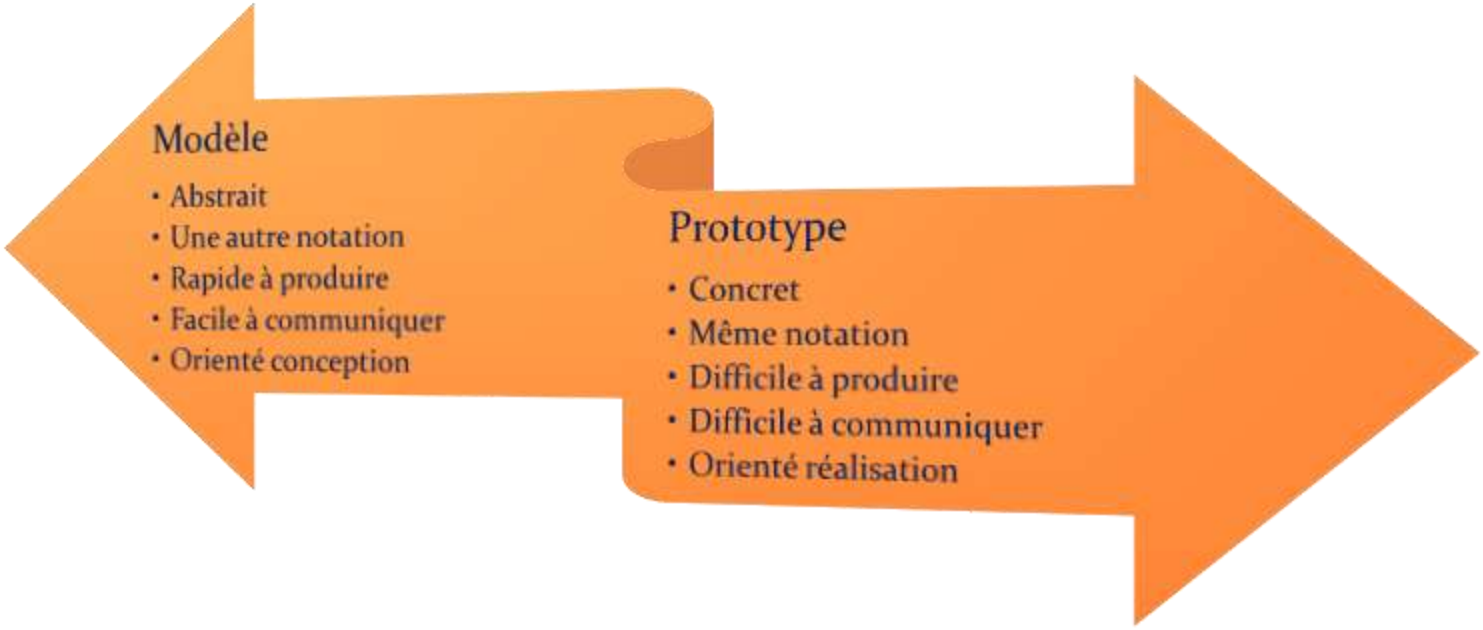
Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

Cycle



Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

Modèle vs Prototype

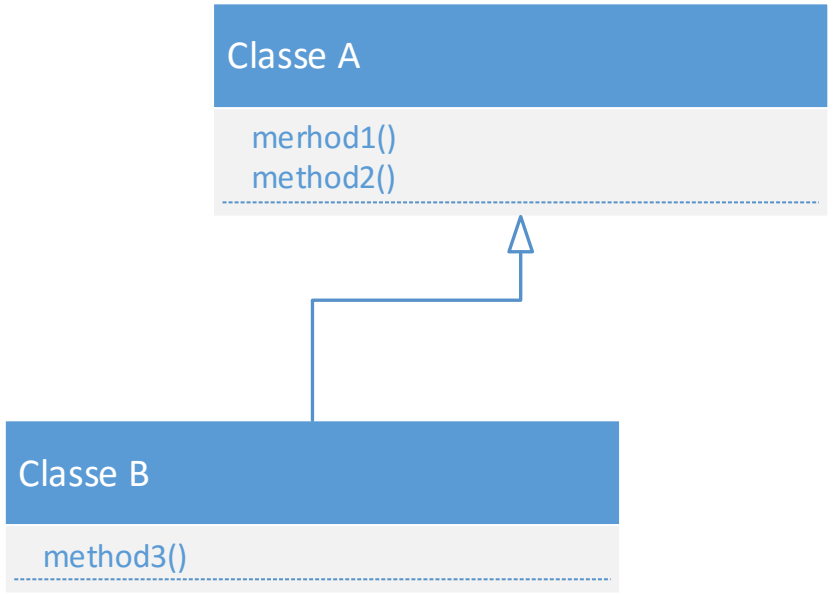


Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

Modèle vs Prototype ou Produit

Modèle



Prototype ou Produit

```
class ClassA{
 .....
}

class ClassB extends ClassA
{
 ......
}
```

Modélisation

- L'action de concevoir un modèle dans un langage de modélisation dédié

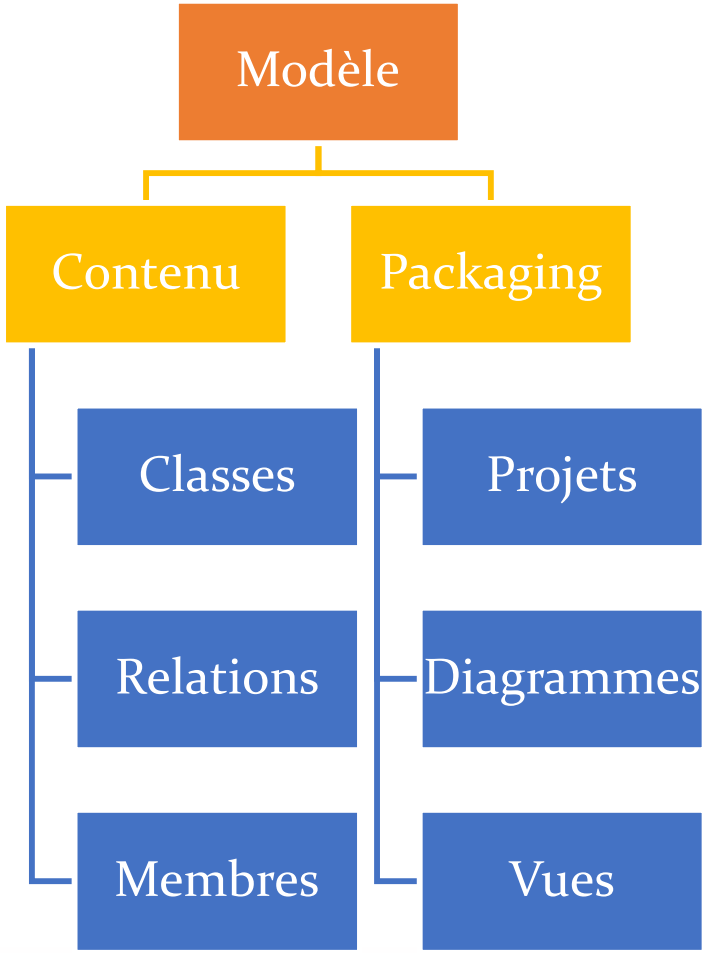
Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

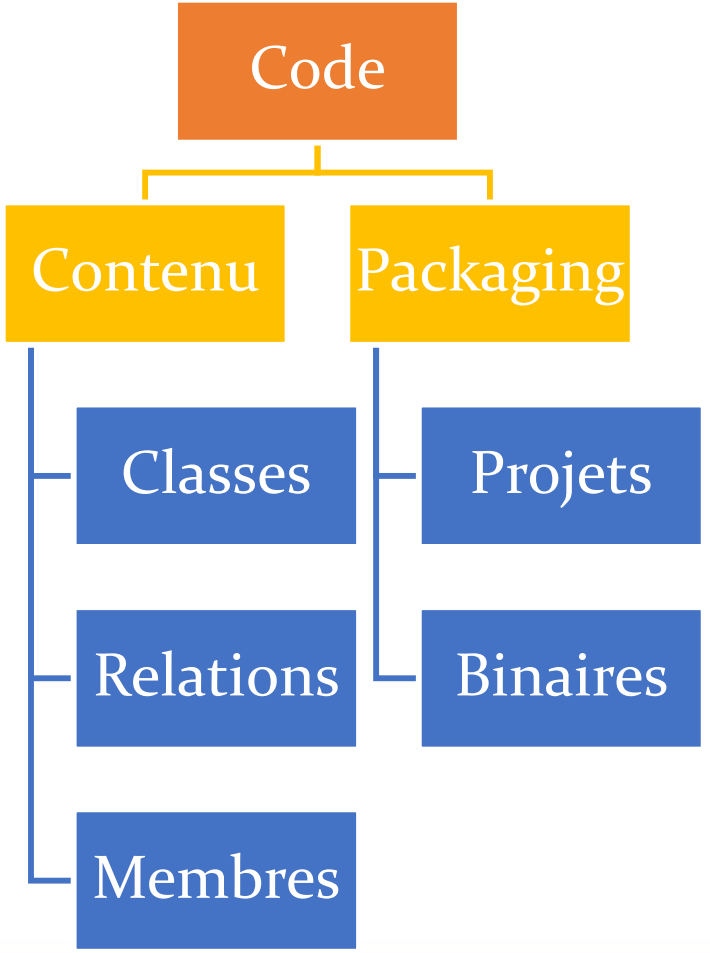
Pourquoi

- Représenter efficacement l'entité modélisée
- Réduire les coûts
- Permettre des visions différentes
- Faciliter l'observation
- Utiliser une notation graphique et simple
- Faciliter la communication
- Simplifier les aspects complexes
- Uniformiser le langage

Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

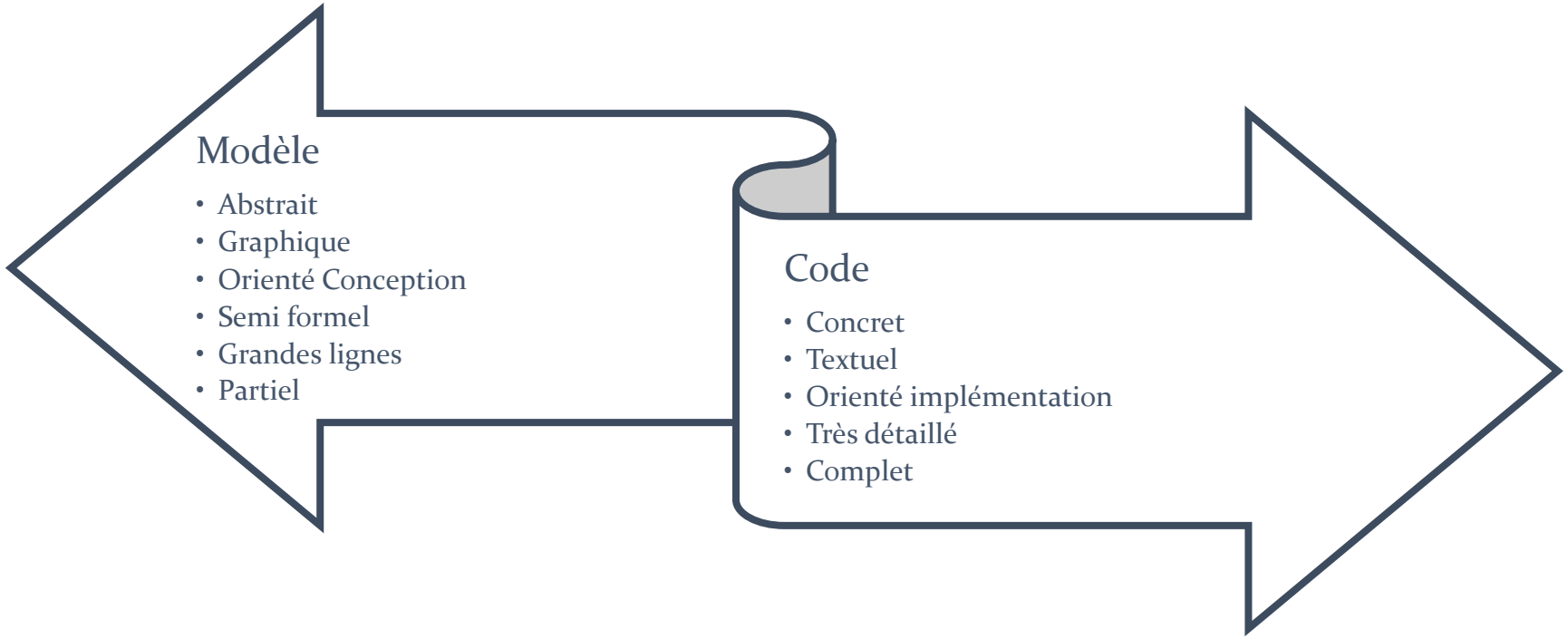
Contenu





Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

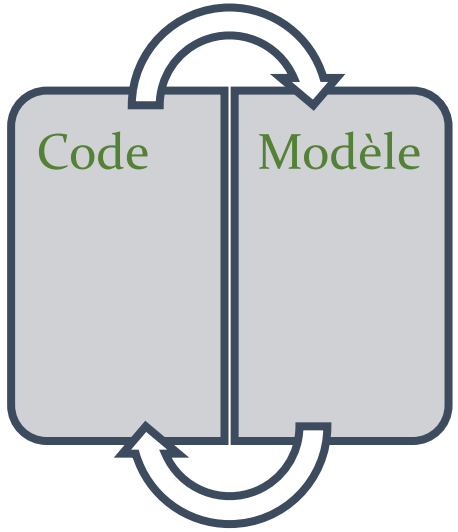
Modèle vs Code



Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

Transformations

Reverse Engineering



Génération de Code

Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

Modélisation

SECTION 1 – DÉBAT (10 MNS)

Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

Introduction à UML

SECTION 2

Origine

Il existait plusieurs méthodes de modélisation et il y avait un besoin de standardisation

UML est la fusion entre les travaux de plusieurs spécialistes de modélisation

UML a été standardisé par l'OMG

UML est un langage de modélisation, pas une méthode

Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

Pourquoi UML ?

UML est graphique

UML est simple

UML est un standard

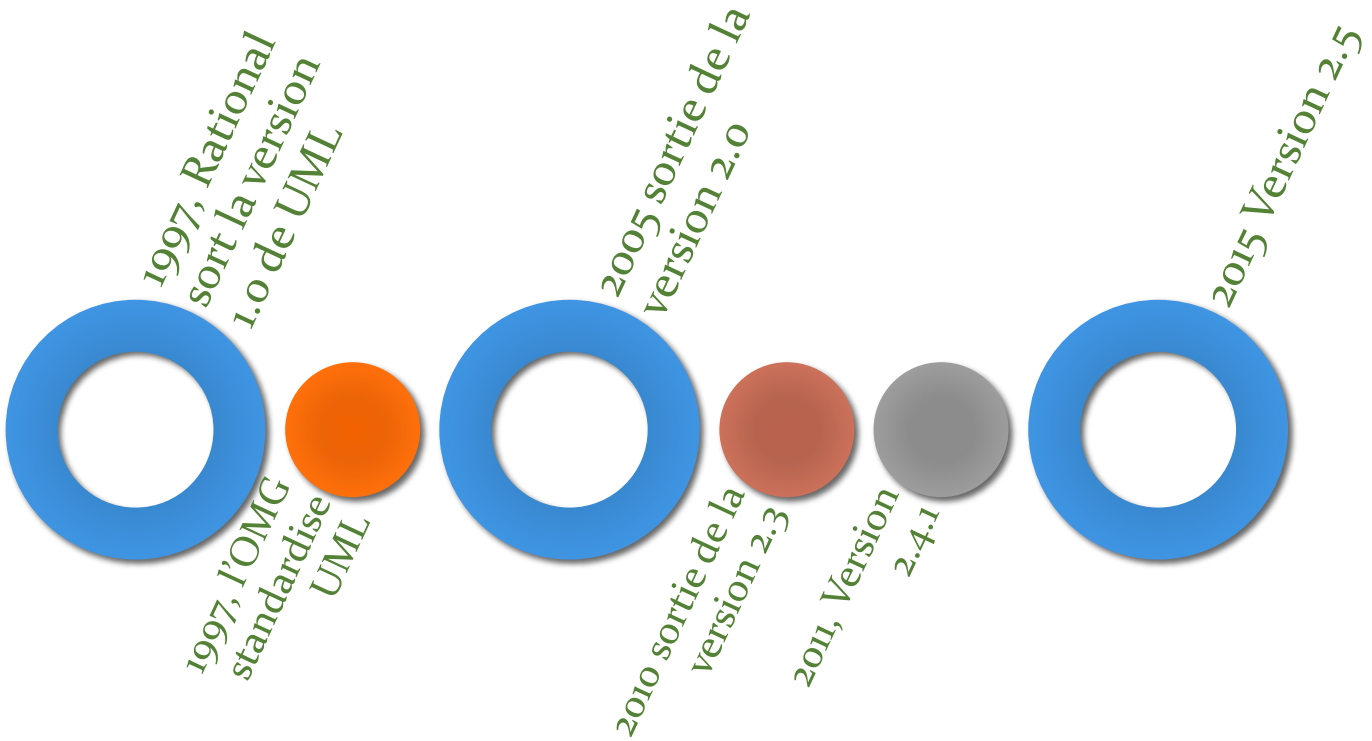
Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

Historique

- Durant les années 70, la communauté a senti que les techniques d'analyse et de conception sont aussi importantes que le développement lui-même
- Au début des années 80, la POO se transforme d'un produit de laboratoire en de vrais produits, Smalltalk et C++ en étaient les premiers tests.
- Plusieurs méthodes d'analyse OO apparurent dont OMT de Rubmaugh, OOSE de Jacobson et OOAD de Booch
- 1995, Grady et Booch sortent la version 0.8 de UML
- 1996, Jacobson rejoint Grady et Booch au sein de Rational et travaille sur la version future de UML

Versions



Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

Outils

- Il existe une pléthore d'outils UML sur le marché : payant, gratuits et open source
- Outils payants : suite rational, Sparx Enterprise Architect
- Outils gratuits / opensource : StarUML, ArguoUML
- Intégration avec les IDE : Eclipse, Visual Studio

Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

Introduction à UML

SECTION 2, DÉBAT 05 MNS

Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

Diagrammes UML

SECTION 3

Diagrammes

- Le diagramme permet de visualiser un modèle selon un angle de vue
- Le diagramme est une vue partielle du modèle
- Un modèle peut contenir 0, 1 ou N diagrammes
- Un modèle contient plusieurs types de diagramme
- Chaque type de diagramme cible un aspect particulier du GL
- Un modèle peut contenir plusieurs diagrammes du même type
- Chaque diagramme est utilisé dans une ou plusieurs étapes du cycle de vie du logiciel
- Un diagramme doit être compact, lisible et expressif
- Un élément peut se trouver dans plusieurs diagrammes en même temps et éventuellement du même type
- UML 2.5 contient 22 diagrammes

Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

Types de diagrammes

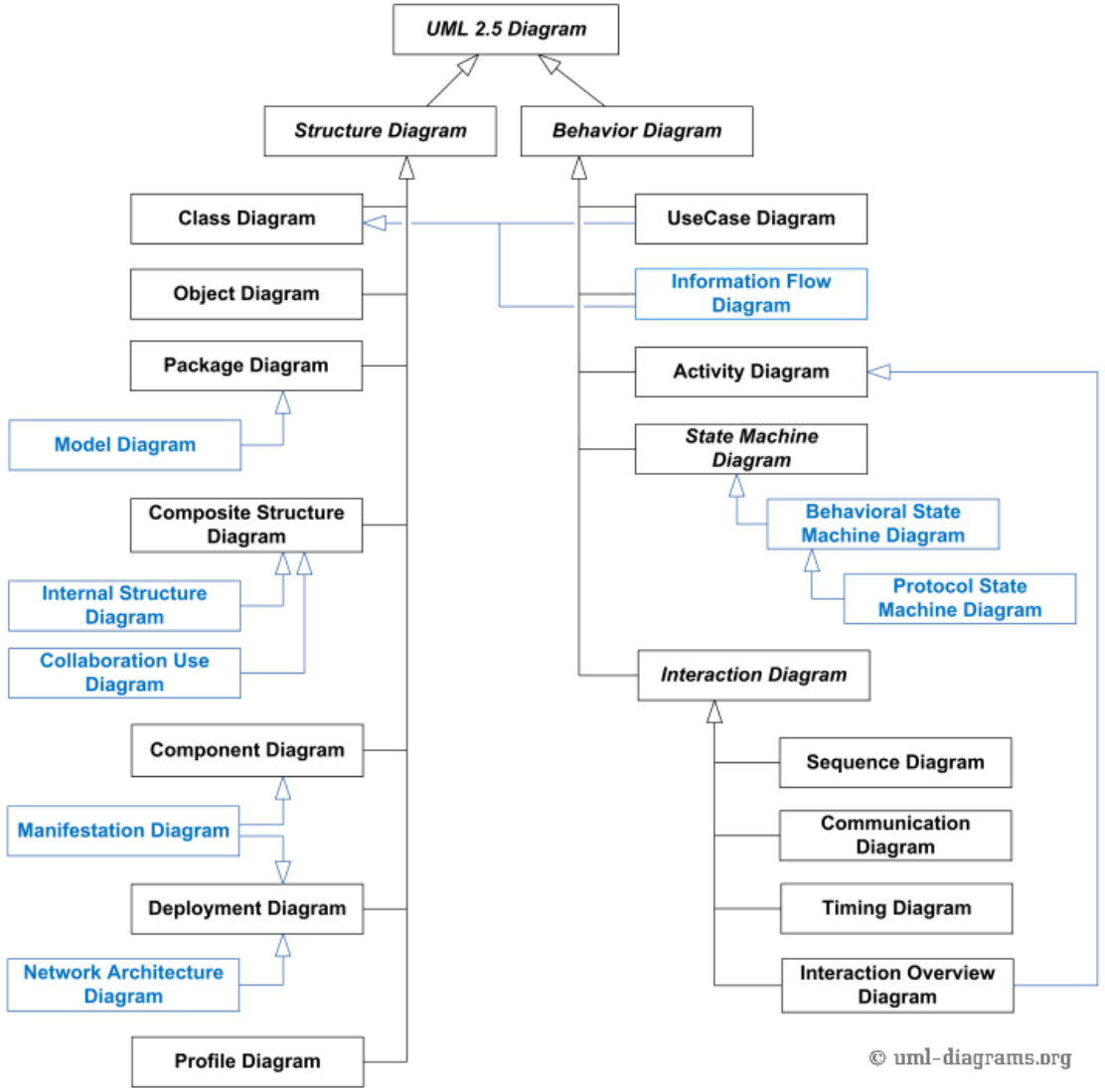
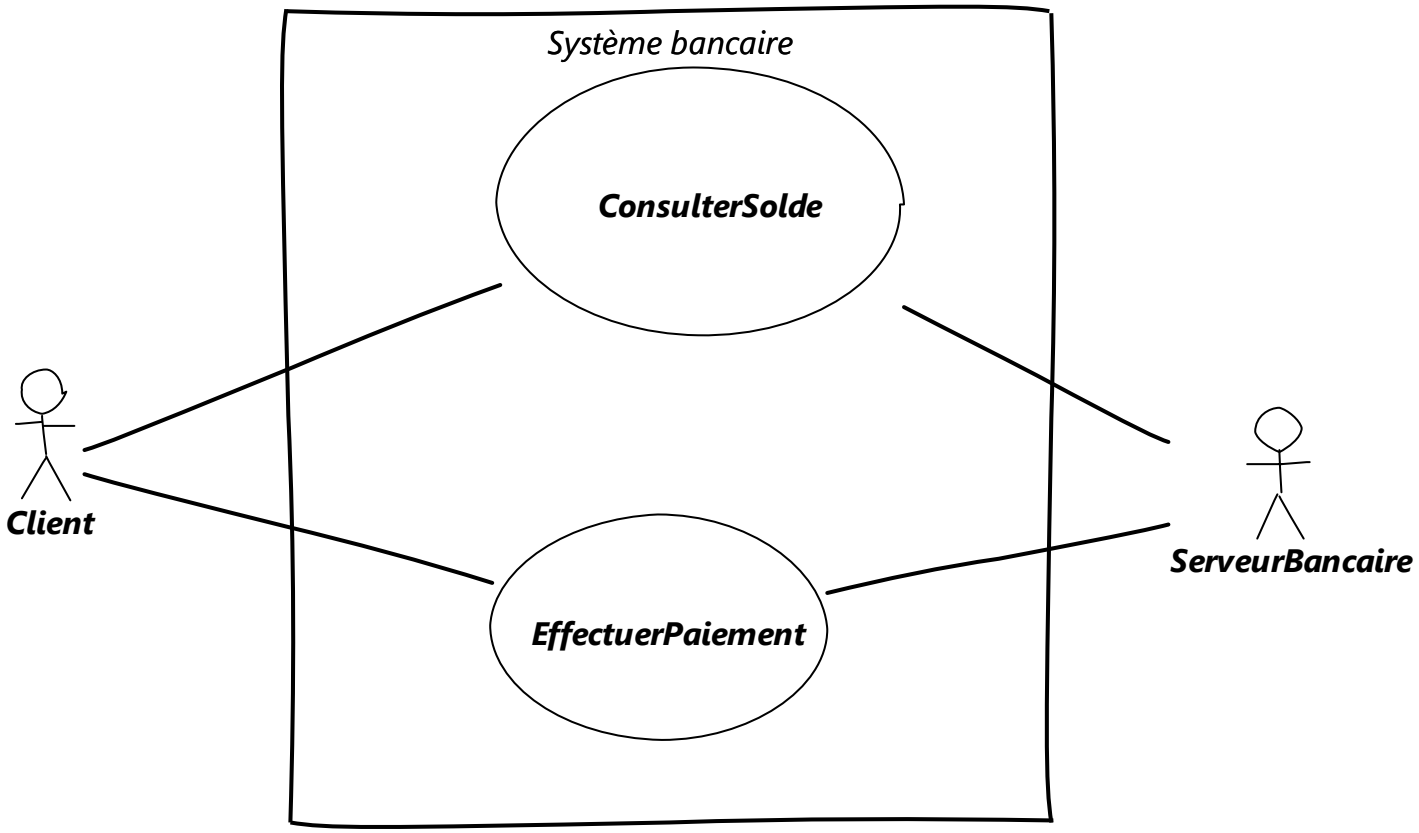


Diagramme de cas d'utilisation (DCU)

- Fournit une vision du système en terme d'acteurs et leurs objectifs
- Le but du DCU est de déterminer quelles sont les fonctions effectuées par chaque acteur

Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

Diagramme de cas d'utilisation (DCU)



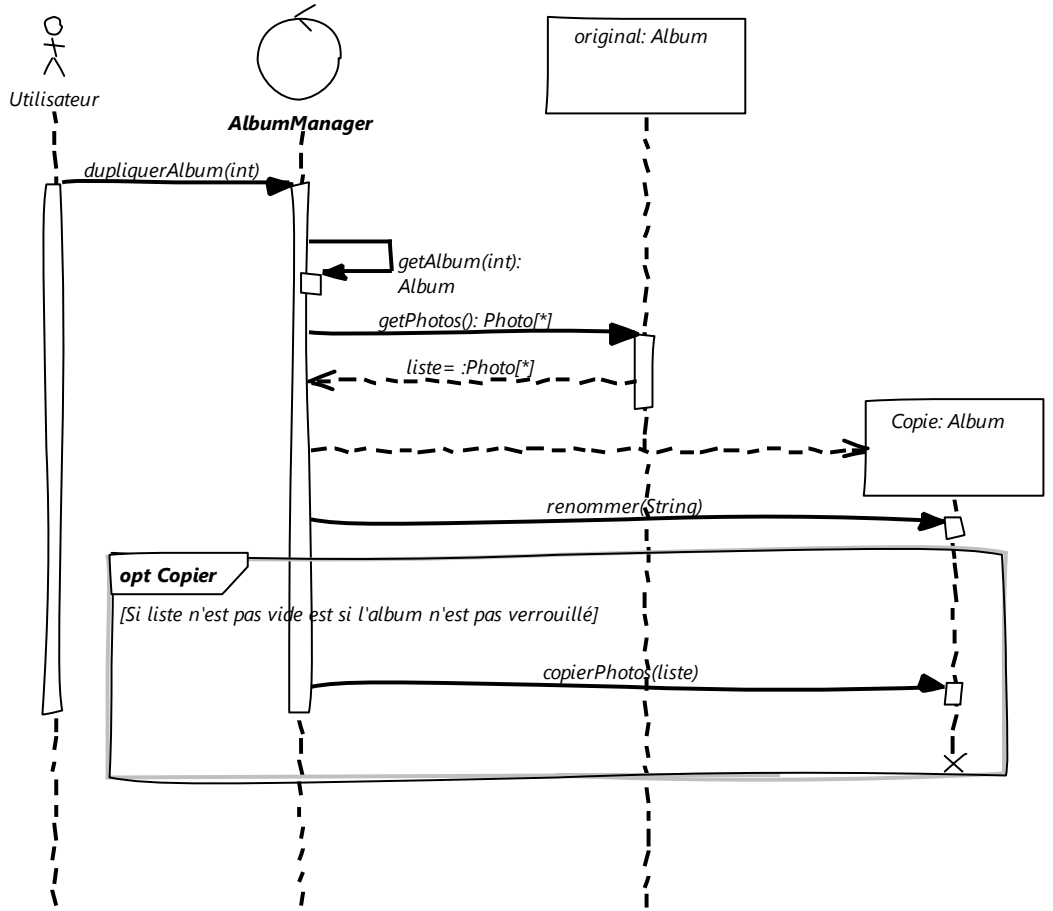
Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

Diagramme de séquence (DSQ)

- Un diagramme d'interaction qui présente comment chaque processus interagit avec l'autre et dans quel ordre.
- Le DSQ illustre les interaction sur un axe temporel donné.
- Le DSQ liste les objets impliqués dans l'interaction afin d'atteindre un certain objectif.
- Ces diagrammes sont associés aux diagrammes de cas d'utilisation.

Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

Diagramme de séquence (DSQ)



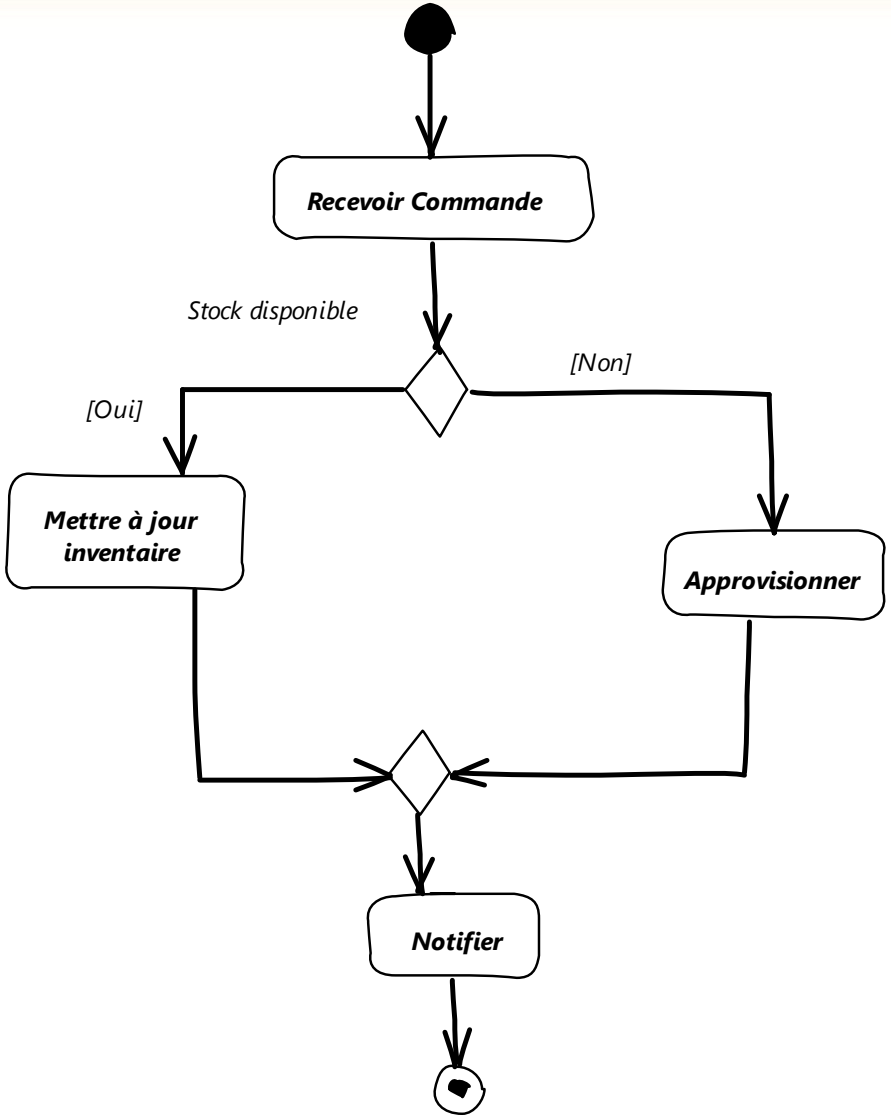
Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

Diagramme d'activité

- Représentation graphique de workflows d'activités
- Décrit les opération étape par étape
- Support de fux conditionnel et itératif

Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

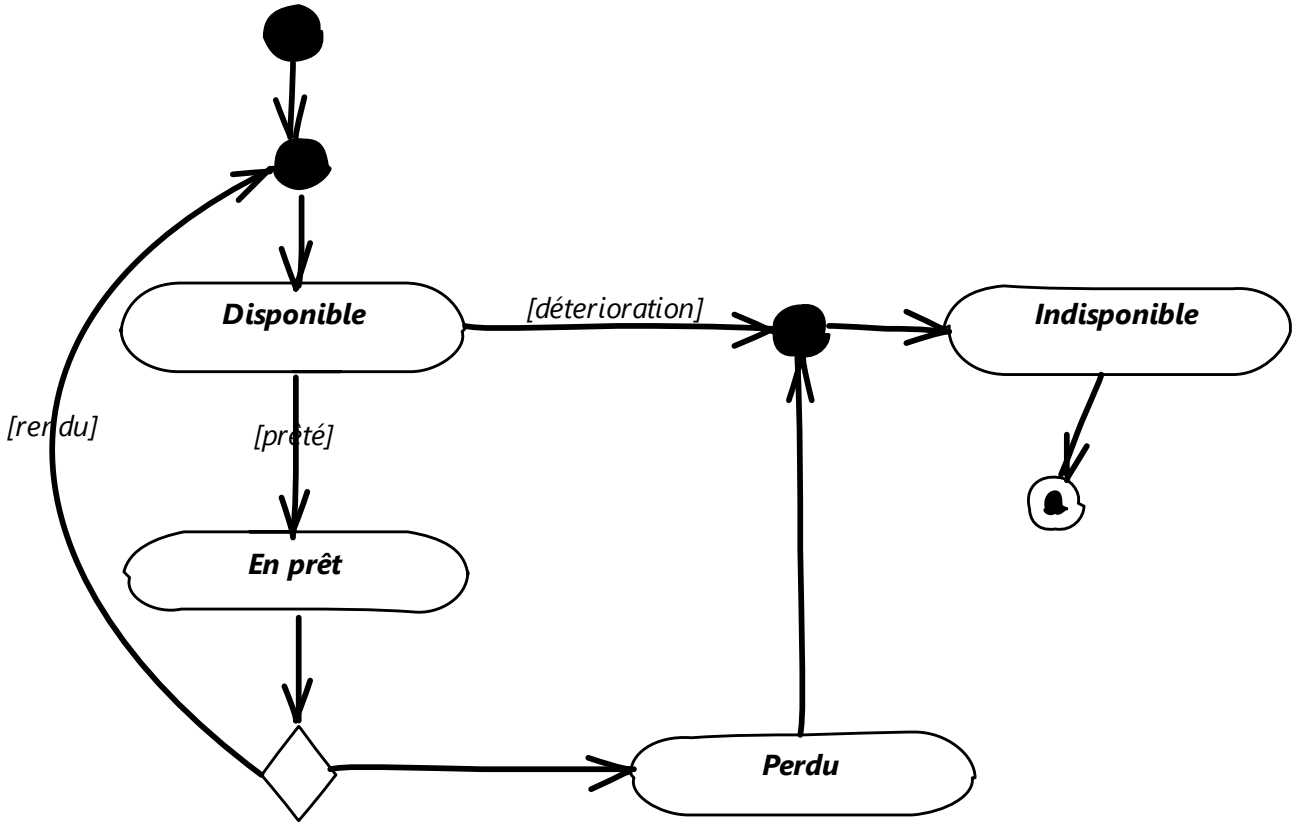
Diagramme d'activité



Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

Diagramme d'état

- Décrit le cycle de vie d'un système ou d'une entité



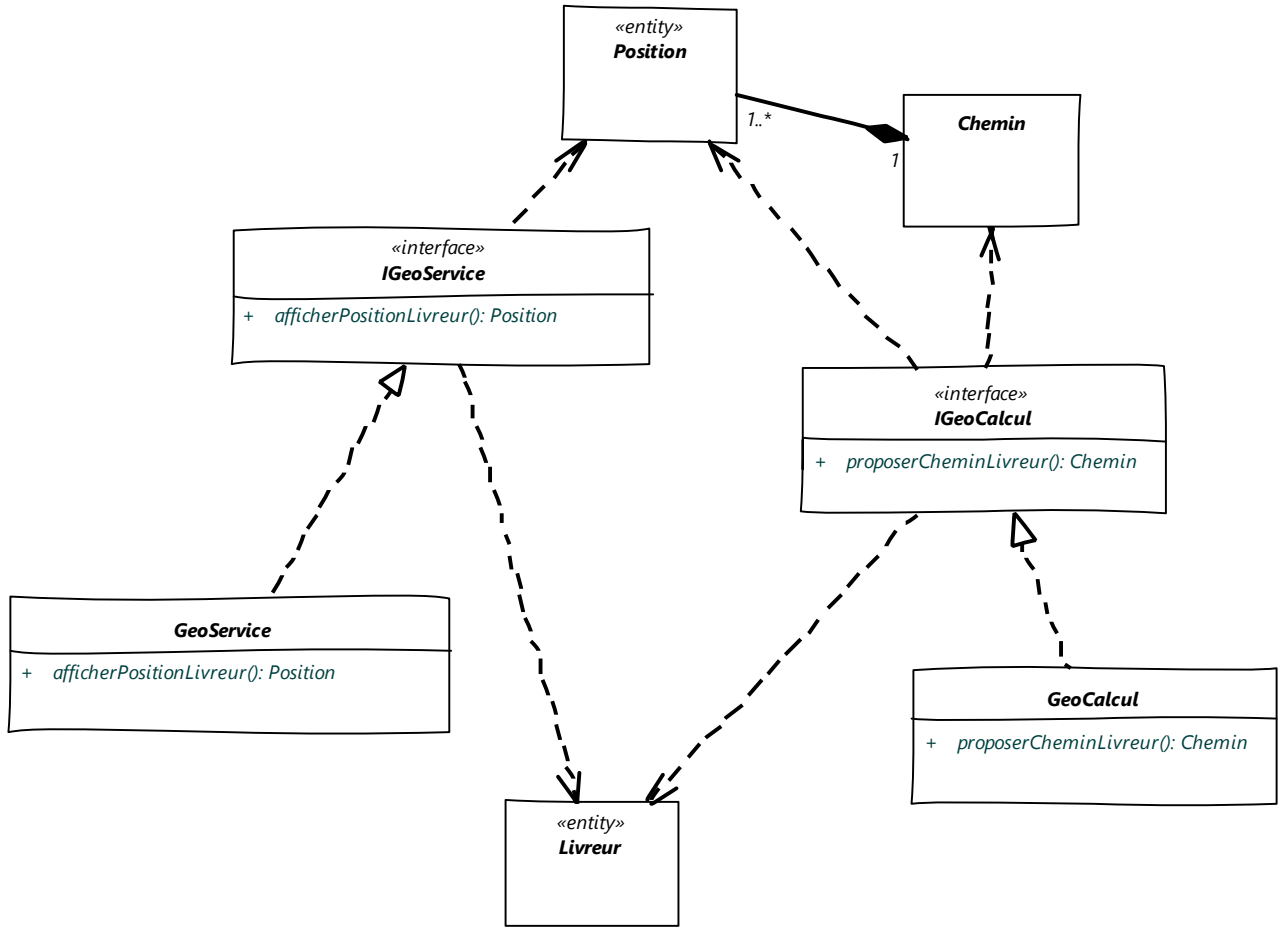
Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

Diagramme de classes

- Souvent considéré comme le diagramme le plus important
- Définit les classes, leurs attributs et leurs relations
- Décrit la conception du système
- Utilisé aussi pour les taxonomies de domaine

Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

Diagramme de classes



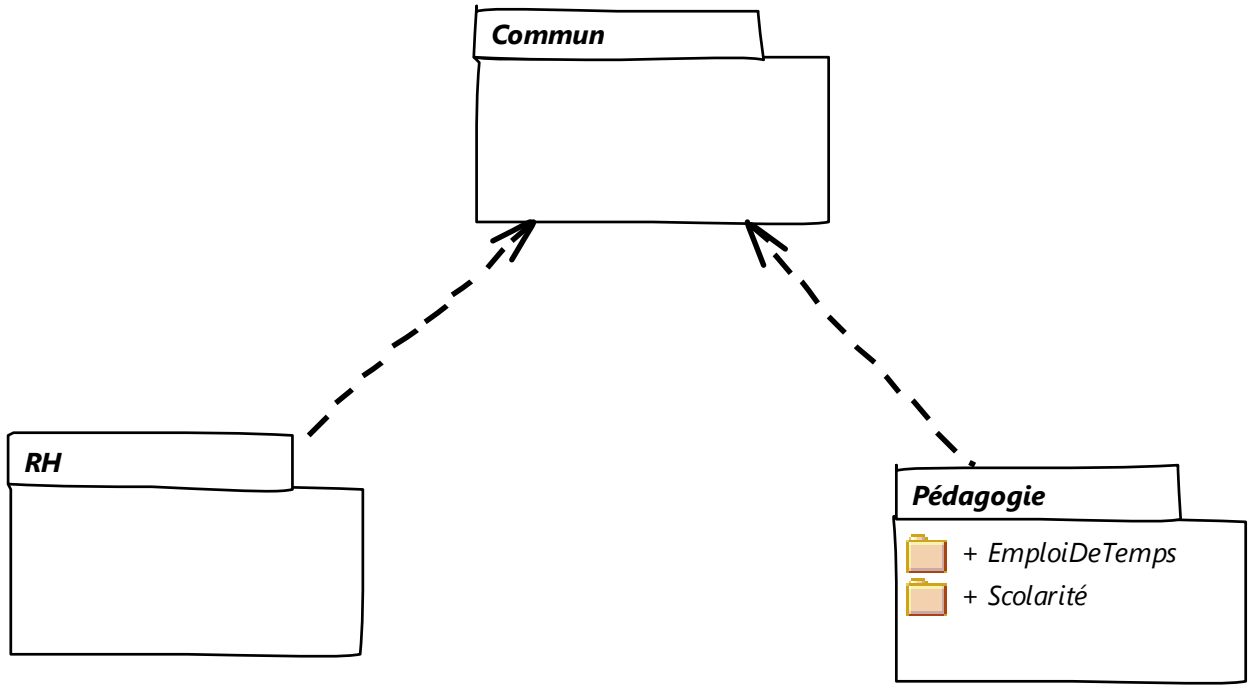
Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

Diagramme de paquets

- Un paquet est un groupement d'éléments UML
- Le paquet peut fournir un espace de nom pour les éléments qu'il contient
- Tous les éléments UML peuvent être groupés dans des packages (pas uniquement les classes)
- Le diagramme de paquets définit les dépendances entres les paquets constituant un modèle

Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

Diagramme de paquets



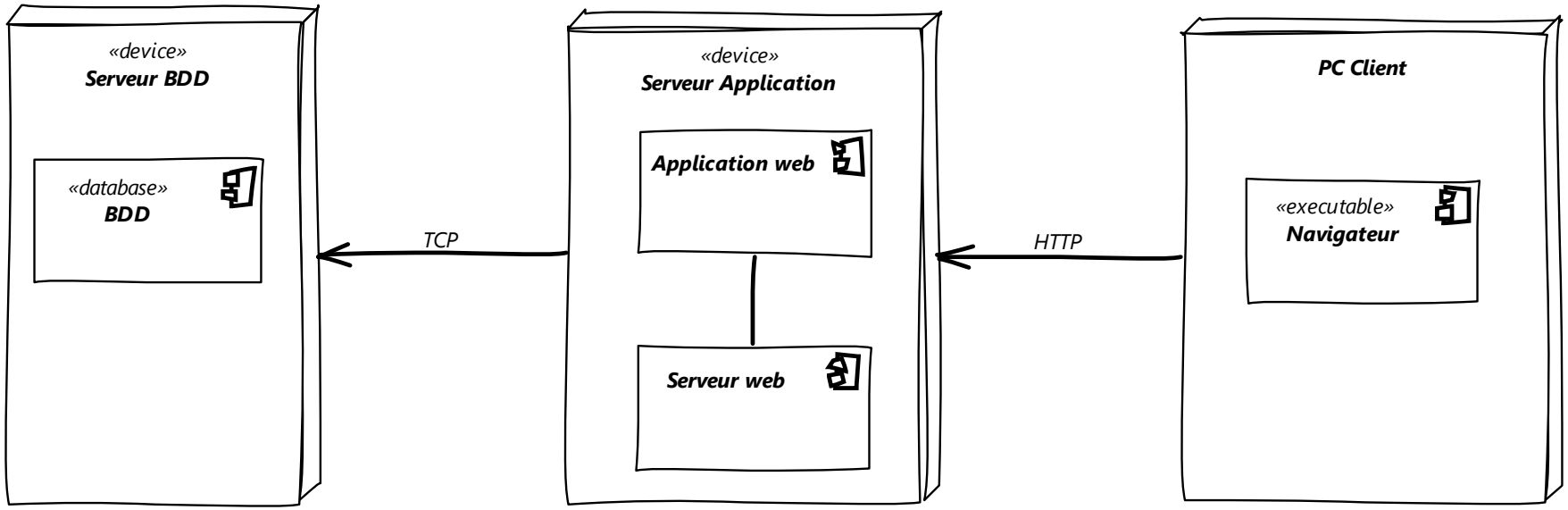
Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

Diagramme de déploiement

- Le diagramme de déploiement définit le « déploiement » physique des entités et des systèmes
- Par exemple, un système logiciel composé de plusieurs sous-systèmes logiciels et plusieurs serveurs physiques : définition où chaque sous-système va être déployé

Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

Diagramme de déploiement



Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

Diagrammes UML

SECTION 3, DÉBAT 05 MNS

Mostefai Mohammed Amine – Batata Sofiane | Copyright © ESI 2015-2016

Bibliographie

- Software Engineering Right Edition, Ian Sommerville, Addison Wesley, 2007
- Software Development and Professional Practice, John Dooley, APress, 2010
- Software Development Life Cycle (SDLC), Togi Berra, course session 2
- Rational Unified Process - Best Practices for Software
- Development Teams, IBM / Rational, 1998