

Analyse et modélisation

Séance 6: Introduction, UML 2 en action

420-A56-GG

Fortement inspiré du livre « UML 2 en action »

Objectifs

- Langage UML
- 2 Track Unified Process

Introduction

- UML est un langage : *Unified Modeling Language* traduit« langage de modélisation unifié»
- Un modèle est une abstraction de la réalité

www.uml.org et en français <http://uml.free.fr/>

Introduction

Nous utiliserons UML pour :

- capturer les besoins des utilisateurs
- analyser ces besoins
- concevoir en vue d'un développement objet

Problématique

- Complexité croissante des SI
 - Définir des méthodes.
- 50 méthodes en 1994.
 - Notation et processus spécifiques.
- UML a ouvert dès lors le terrain
 - Unification des meilleurs pratiques.

Processus (définition)

Séquence
d'étapes, en
partie
ordonnées

Objectif

obtention d'un
système logiciel
ou évolution
d'un système
existant qui
satisfasse le
client

Contraintes

Délais
Coûts

Processus unifié

- Processus unifié qui suit deux chemins:

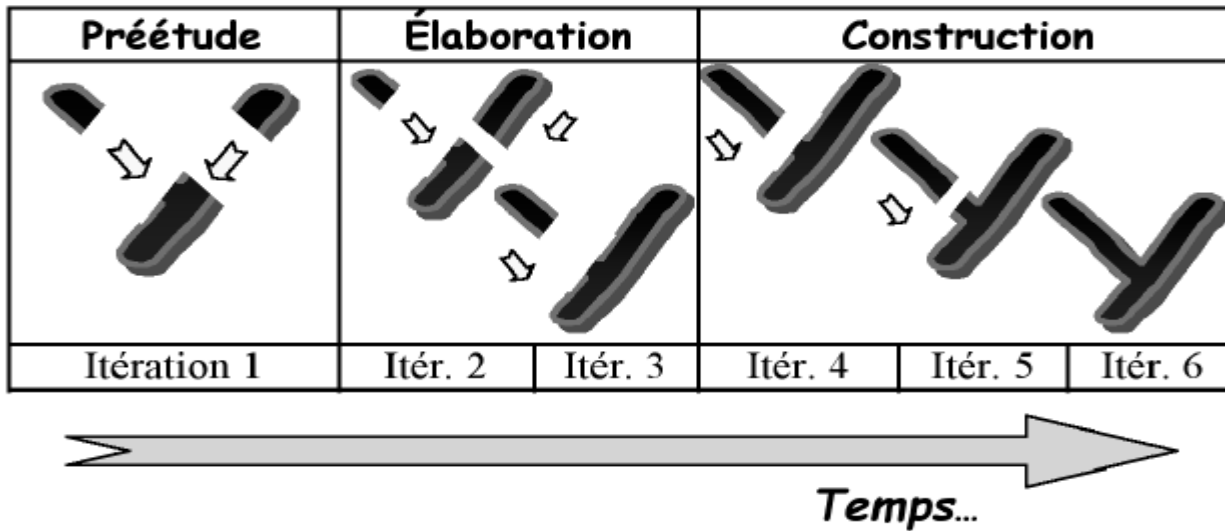
- Fonctionnels
- Techniques

Le SI est soumis à deux natures de contraintes

- Processus unifié

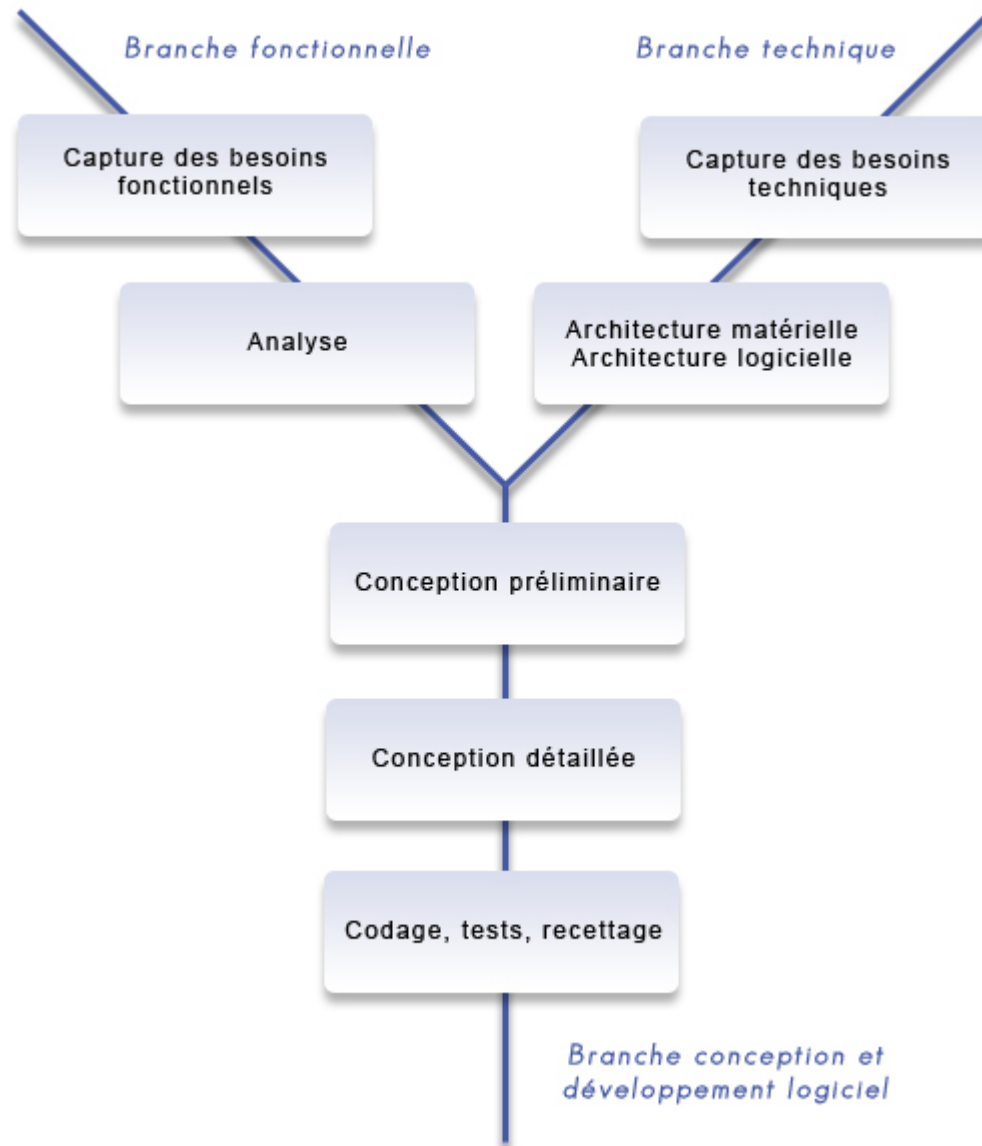
- Itératif et incrémental
- Piloté par les risques
- Construit autour de la création et de la maintenance de modèles
- Orienté composant – réutilisation
- Orienté utilisateur

Définitions



- Itération
 - Séquence distincte d'activités avec un plan de base et des critères d'évaluation, qui produit un release(interne ou externe) ou version.
- Incrément
 - Différence entre deux *releases* produites à la fin de deux itérations successives - (différences entre les versions)
- Exemple de répartition des efforts par activité

2 Track Unified Process



Processus de modélisation avec UML

- UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire les besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue.

Les diagrammes d'UML 2

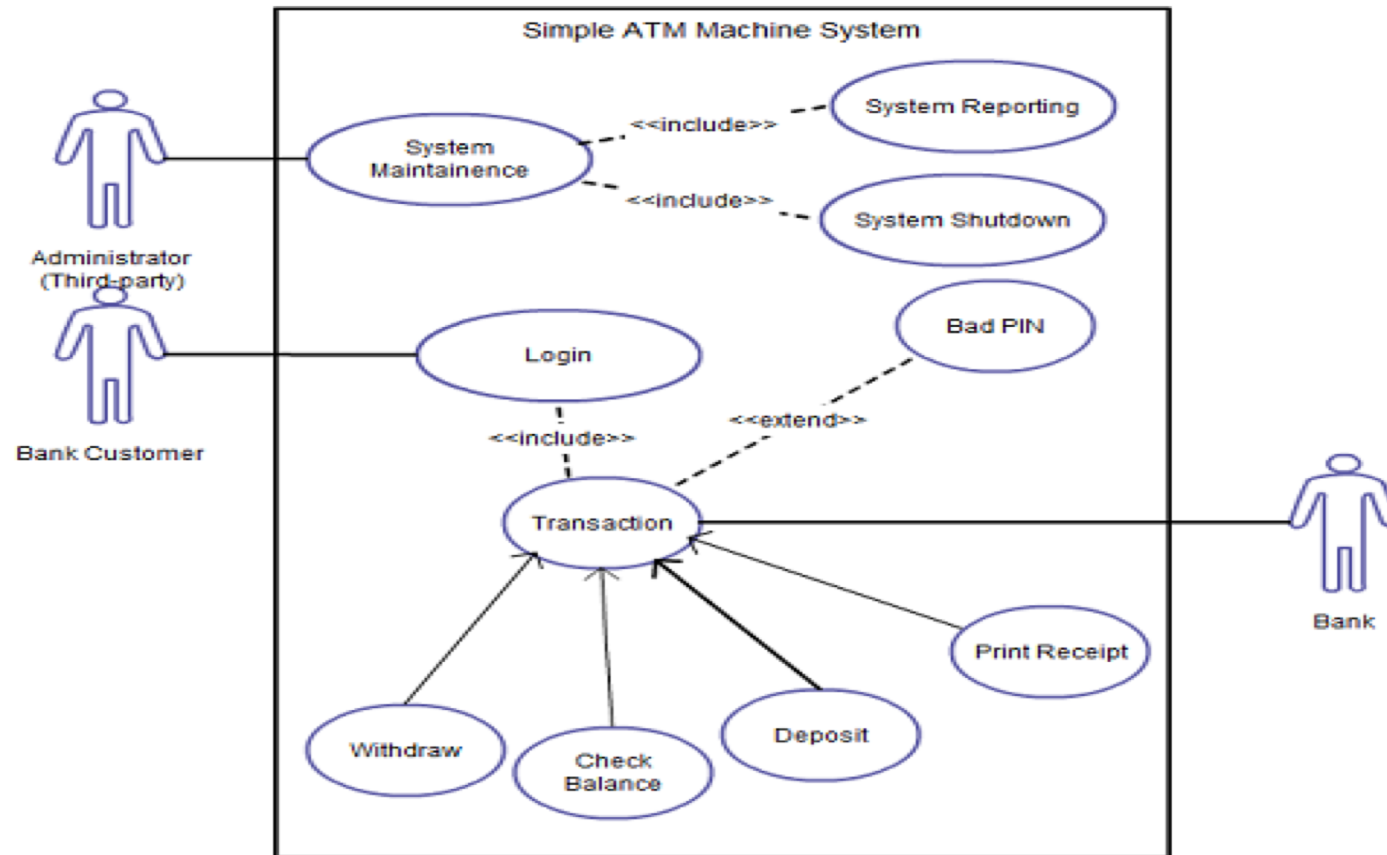
- 13 dans le livre
- 7 Diagrammes structurels ou statiques
- Diagrammes comportementaux dont les diagrammes d'interactions

Diagrammes structurels ou statiques

- Diagramme de cas d'utilisation
- Diagramme de classes (*Class diagram*)
- Diagramme d'objets (*Object diagram*)
- Diagramme de composants (*Component diagram*)
- Diagramme de déploiement (*Deployment diagram*)
- Diagramme des paquetages (*Package diagram*)
- Diagramme de structure composite (*Composite Structure Diagram*)
(UML 2.x et +)

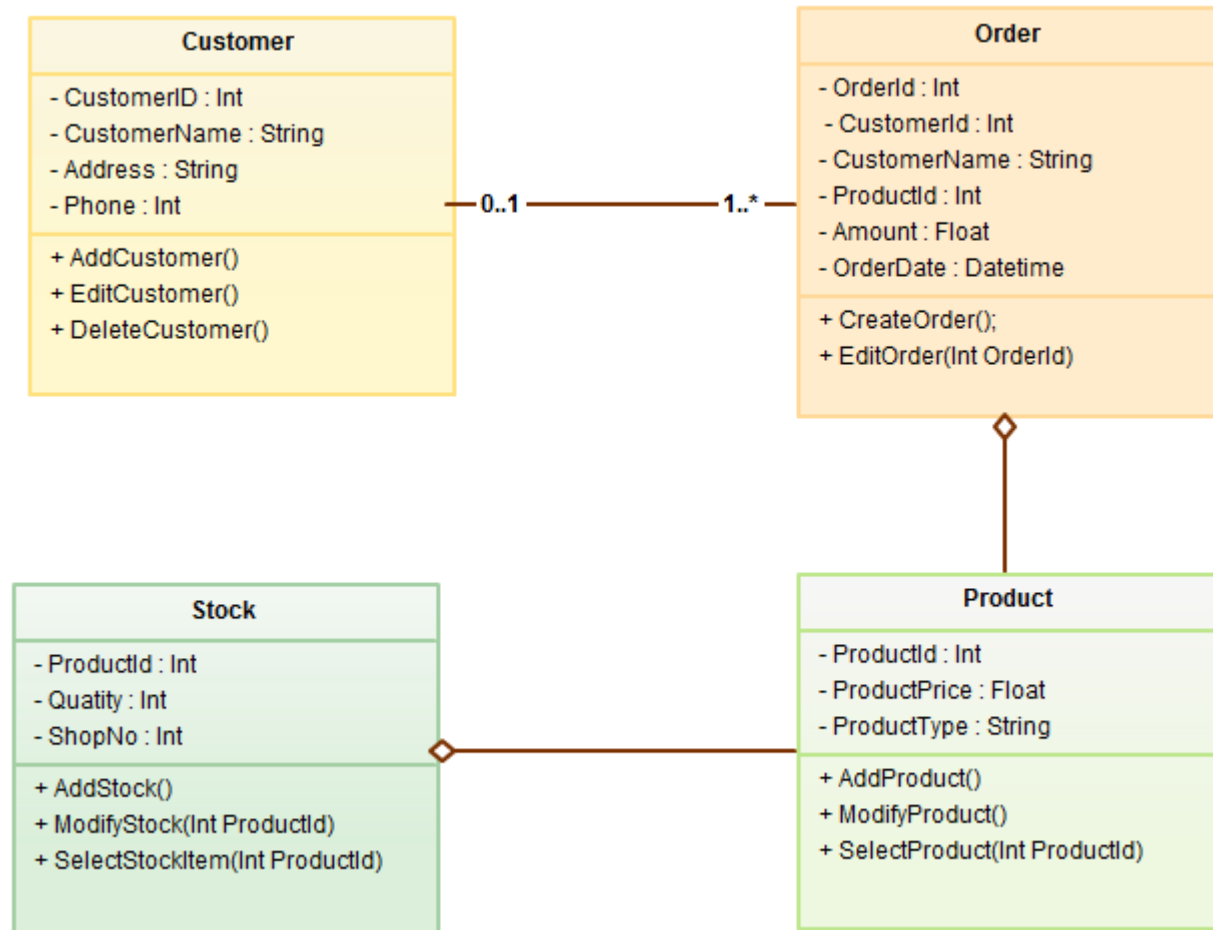
Diagrammes structurels ou statiques

- Diagramme de **cas d'utilisation** (*Use Case Diagram*), représente la structure des fonctionnalités du système.



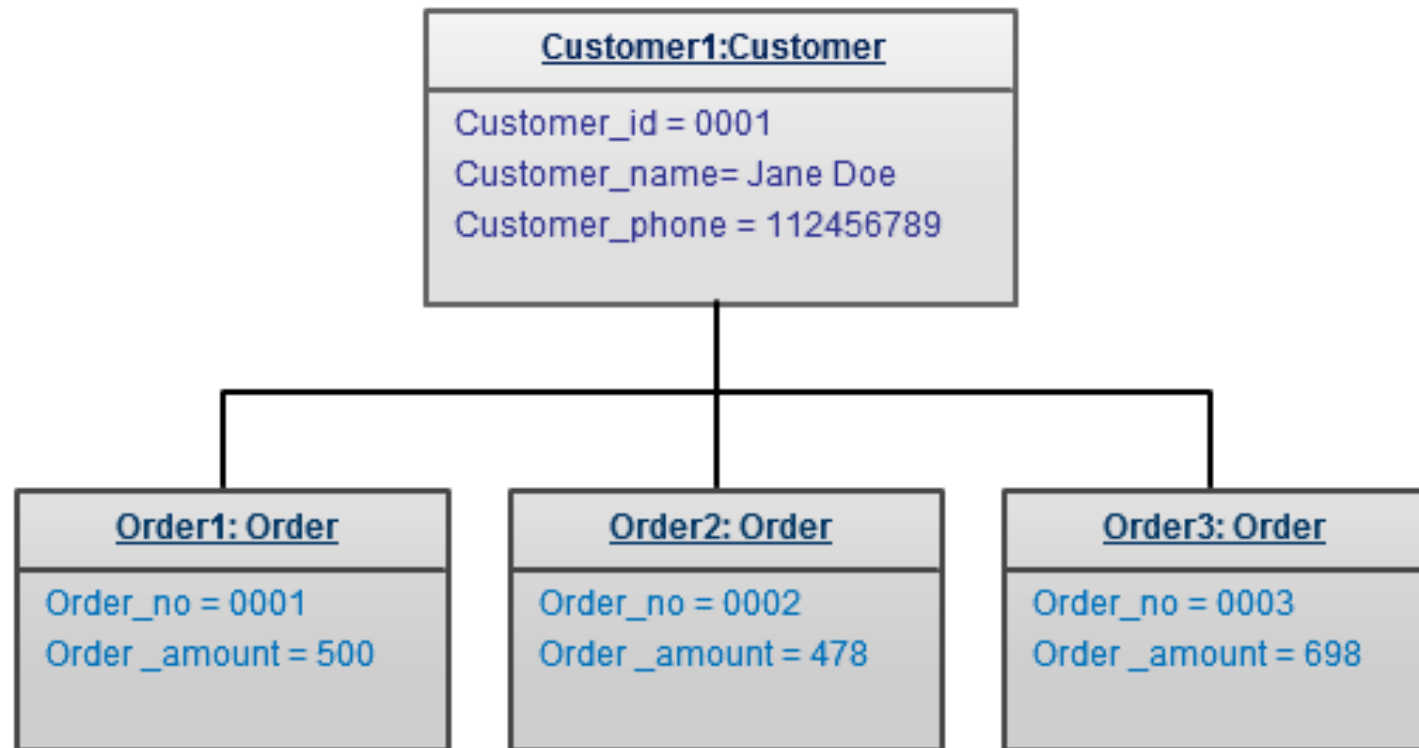
Diagrammes structurels ou statiques

- Le diagramme de **classes** (*Class diagram*), qui contient :
 - Les classes
 - Les interfaces
 - Les relations entre celles-ci



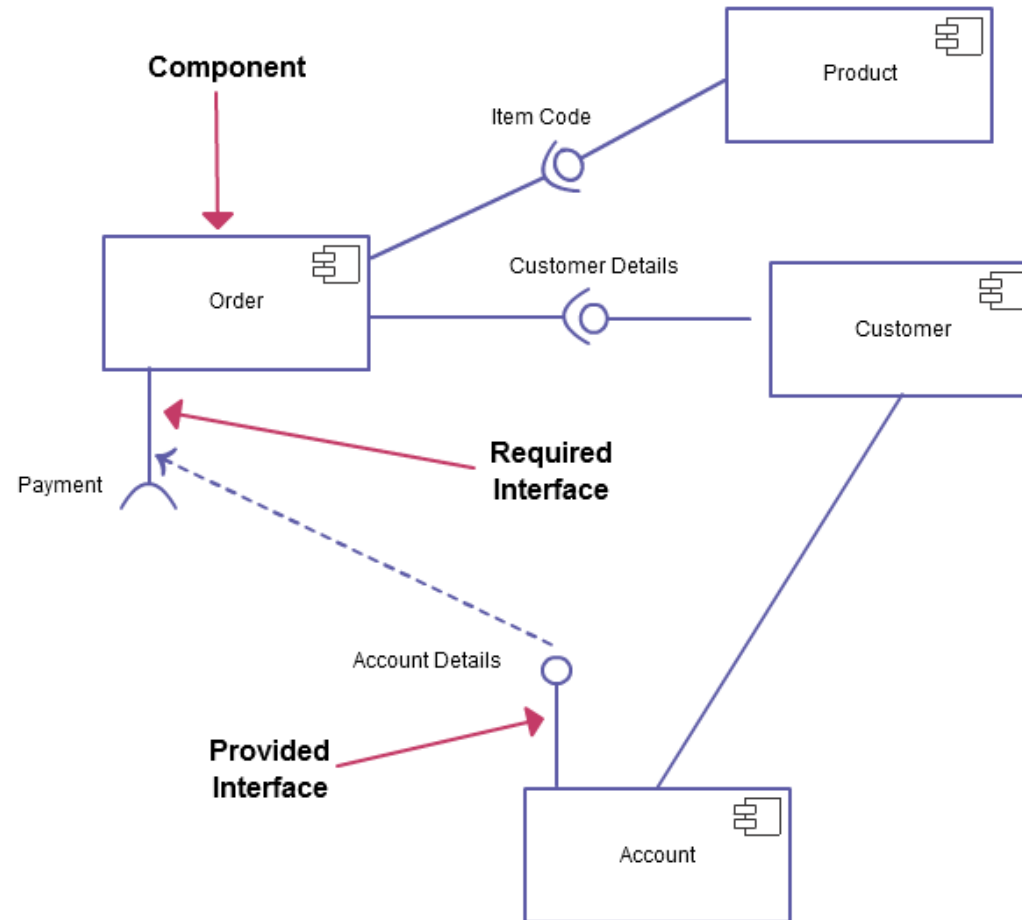
Diagrammes structurels ou statiques

- Le diagramme **d'objets** (*object diagram*), qui représente les instances des objets



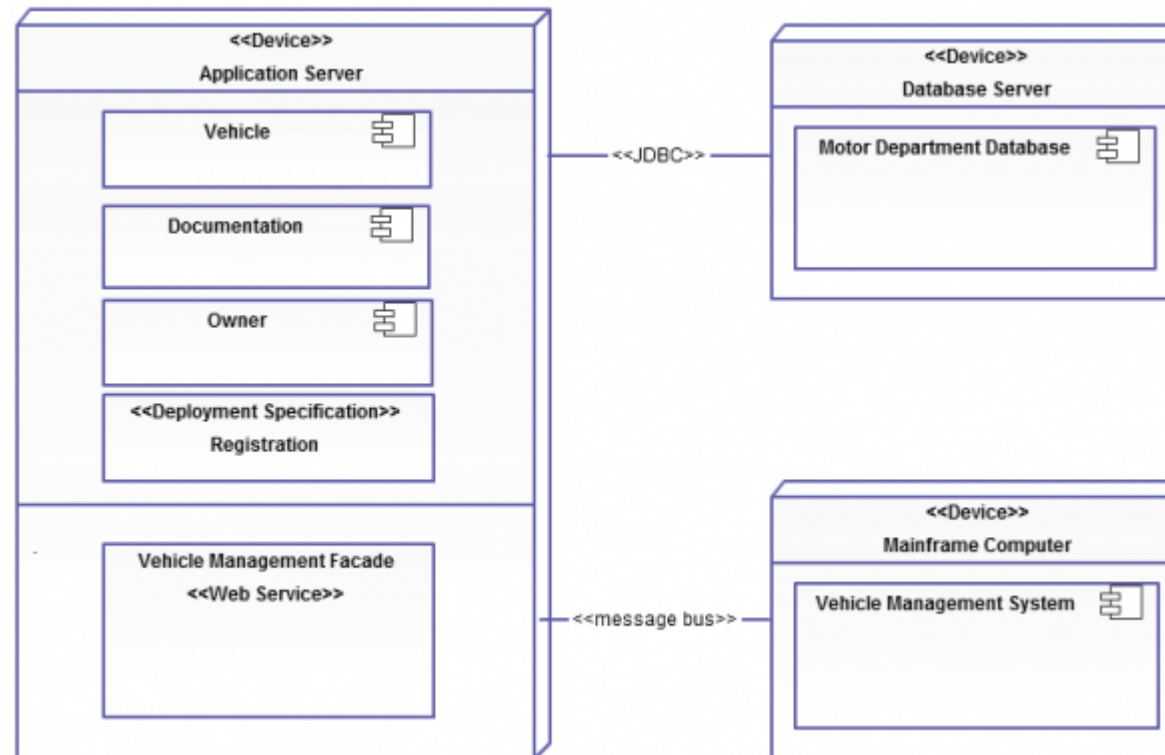
Diagrammes structurels ou statiques

- Diagramme de composants (*component diagram*), qui illustre les composants du système (fichiers, librairies (bibliothèques), bases de données, des executables, des applications...)



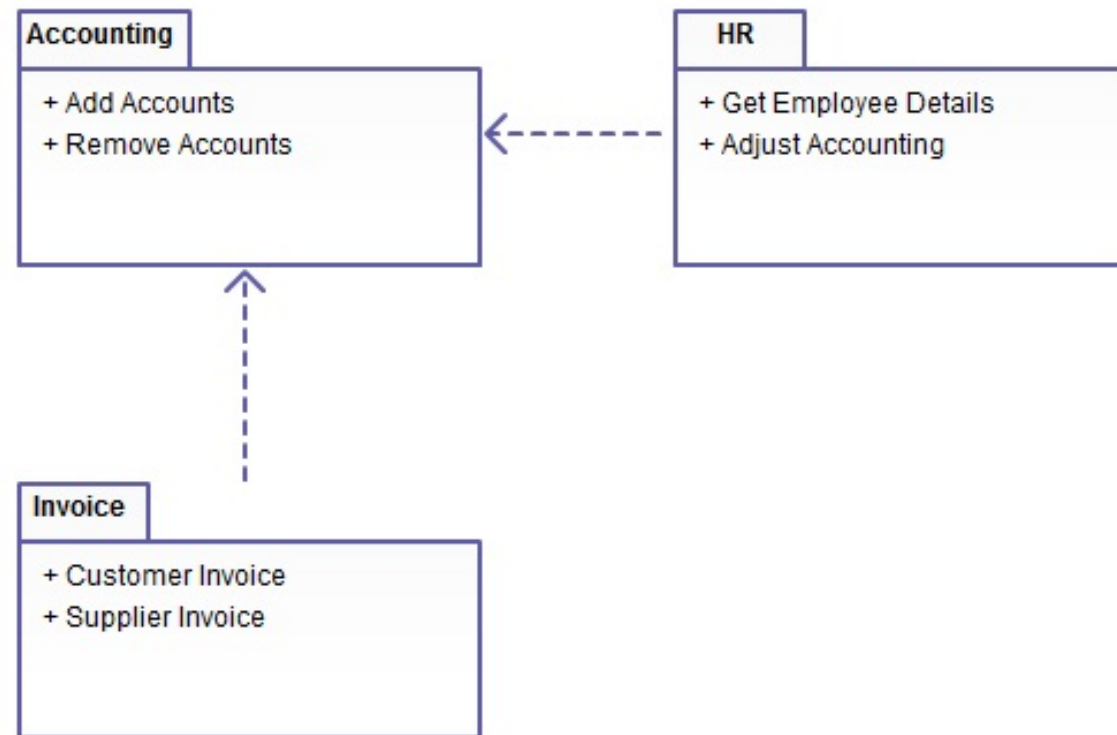
Diagrammes structurels ou statiques

- Le diagramme de **déploiement** (*deployment diagram*), pour un déploiement sur réseau (plusieurs machines): illustrent les relations entre différents serveurs et services.



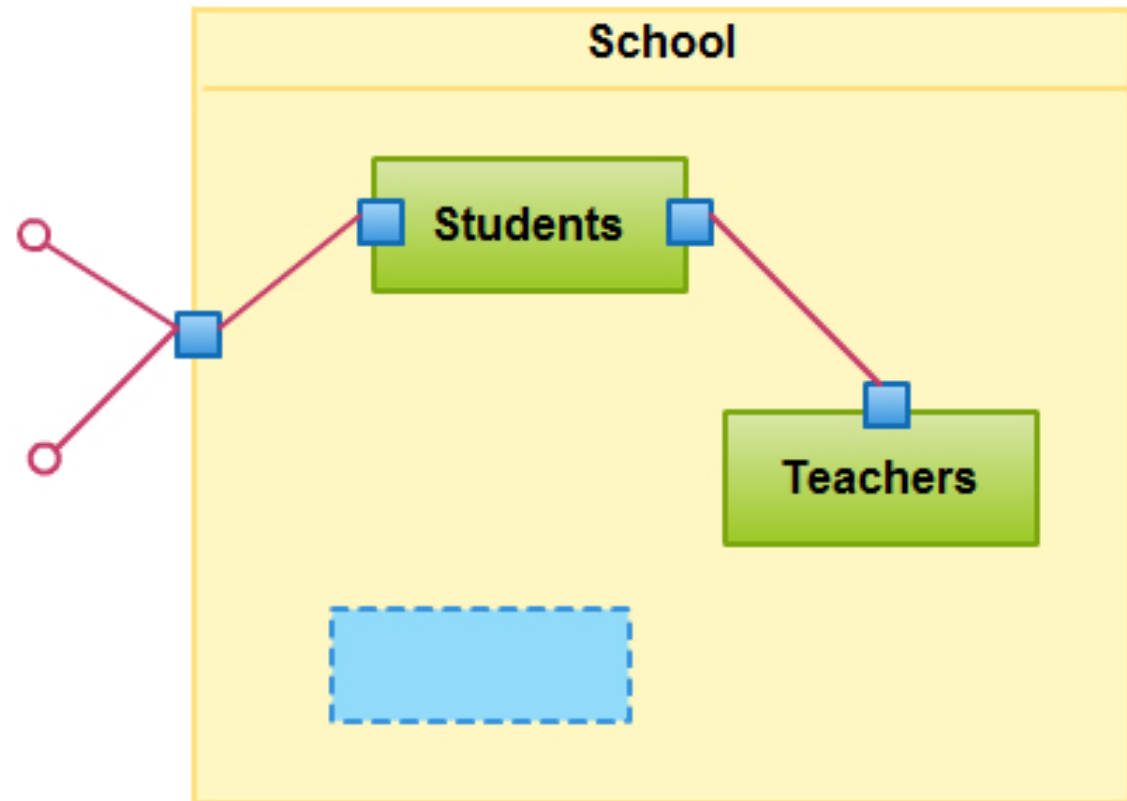
Diagrammes structurels ou statiques

- Le diagramme des **paquetages** (*Package diagram*), qui illustre les relations existant entre les paquetages (ou espaces de noms) dans le système



Diagrammes structurels ou statiques

- Diagramme de **structure composite** (*composite structure diagram*), qui montre la structure interne d'une classe

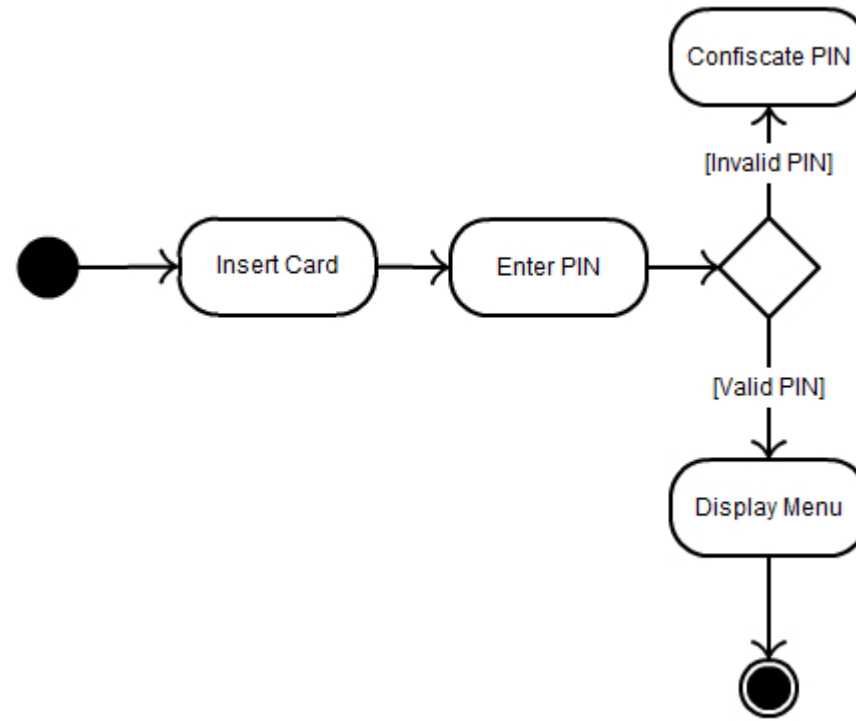


Diagrammes comportementaux

- Diagramme d'activité (*Activity Diagram*)
- Diagramme d'états (*State Diagram*)
- Diagrammes d'interactions

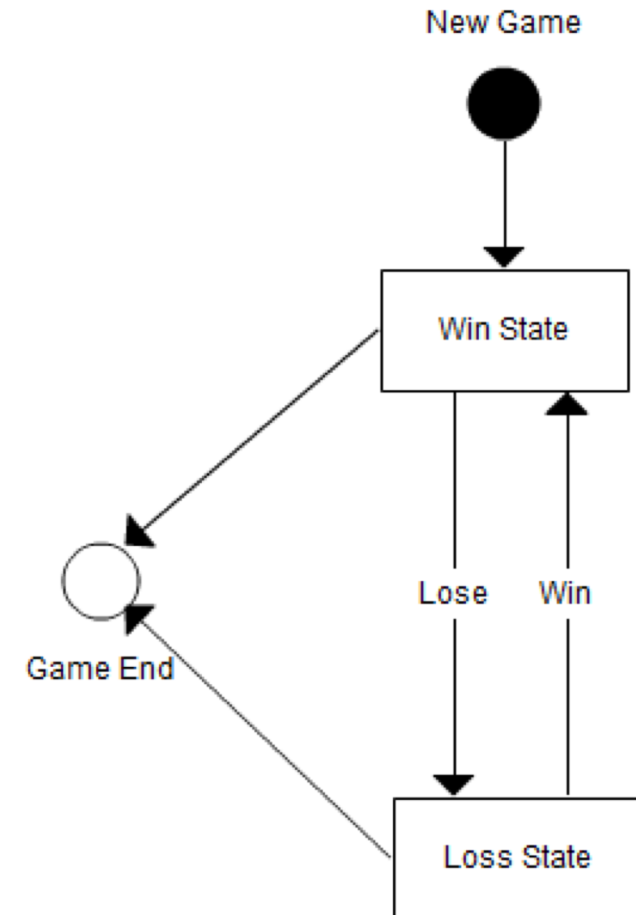
Diagrammes comportementaux

- Diagramme d'activité (*Activity Diagram*), représente les règles d'enchaînement des activités dans le système



Diagrammes comportementaux

- **Diagramme d'états** (*State Diagram*), qui est similaire au diagramme d'activité et qui représente sous forme de machine à états finis le cycle de vie commun aux objets d'une même classe

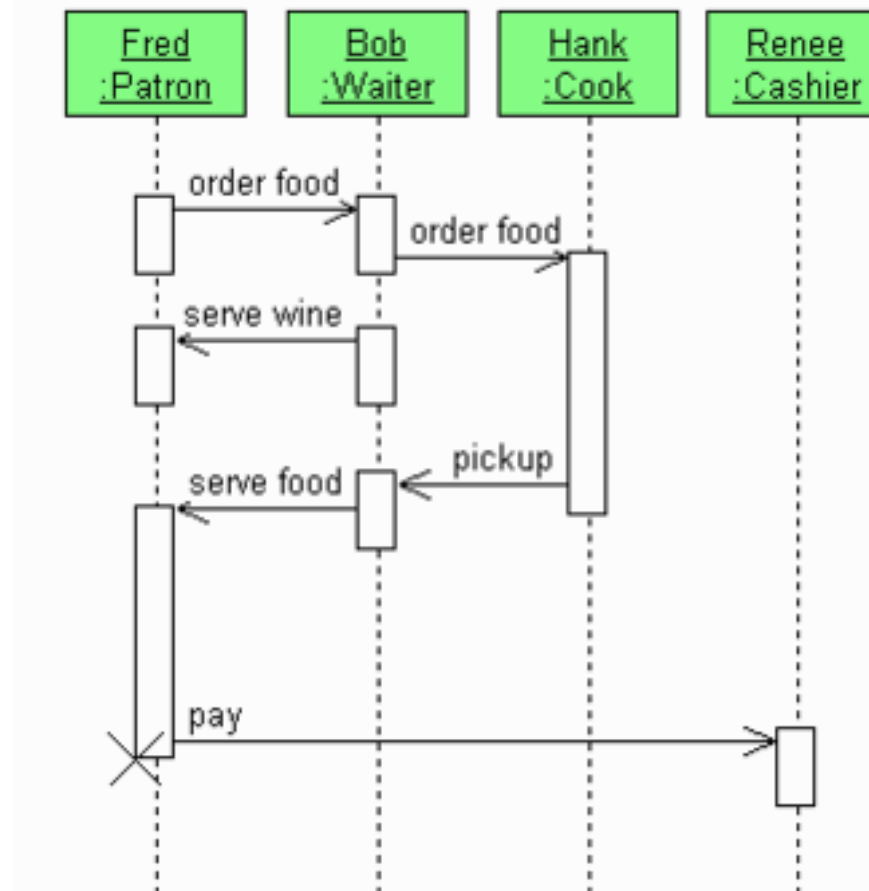


Diagrammes d'interaction

- Diagramme de séquence (*Sequence Diagram*)
- Diagramme de communication (*Communication Diagram*) (UML 2.x et +)
- Diagramme global d'interaction (*Interaction Overview Diagram*) (UML 2.x et +)
- Diagramme de temps (*Timing Diagram*) (UML 2.3 et +)

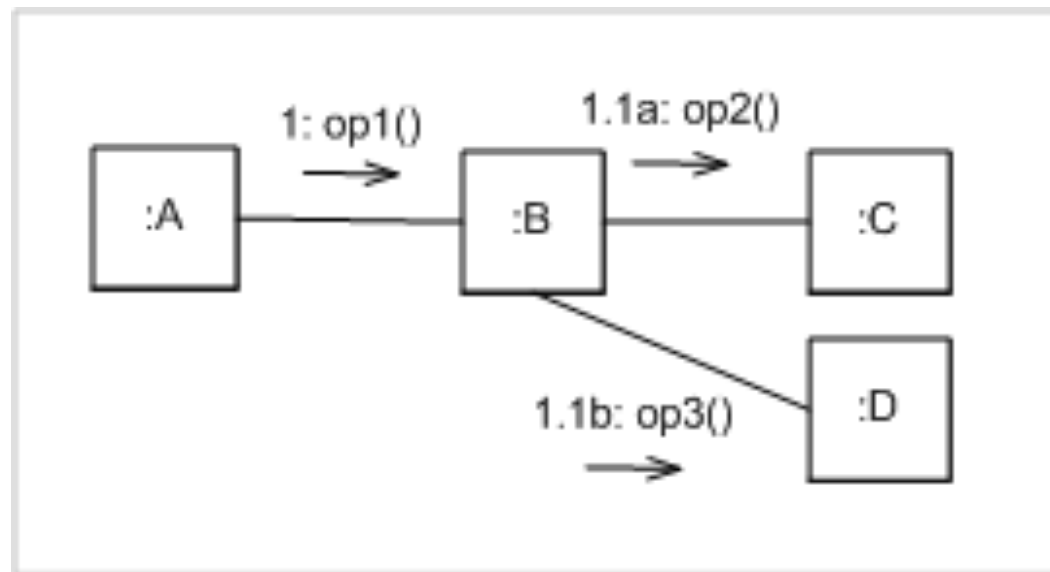
Diagrammes d'interaction

- Diagramme de **séquence** (*Sequence Diagram*) : représente les interactions entre les acteurs et le système selon un ordre chronologique (en lien avec les scénarios d'utilisation)



Diagrammes d'interaction

- Diagramme de **communication** (*Communication Diagram*) : représente les échanges entre objets



Diagrammes d'interaction

- Diagramme global d'interaction (*Interaction Overview Diagram*) il propose d'associer les notations du diagramme d'activités avec celles du diagramme de séquences

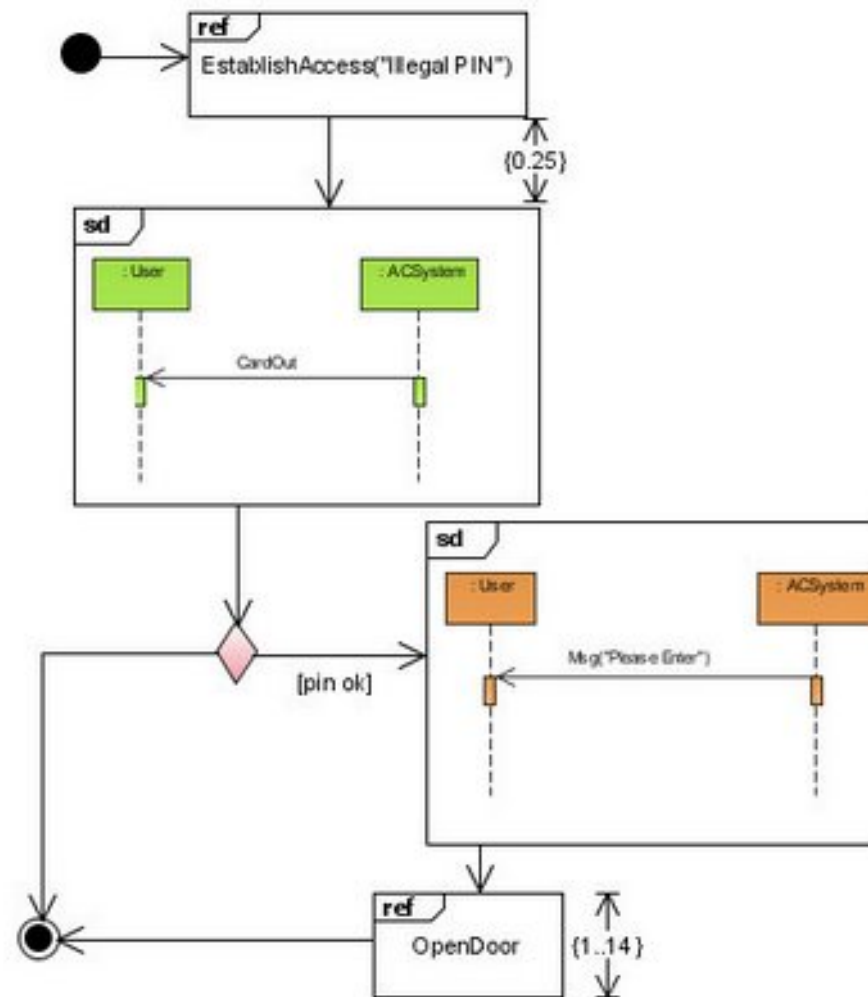


Diagramme de temps

- (*Timing Diagram*) : décrit les contraintes temps réel qu'exigent certaines interactions entre objets.

