

IFT 2255

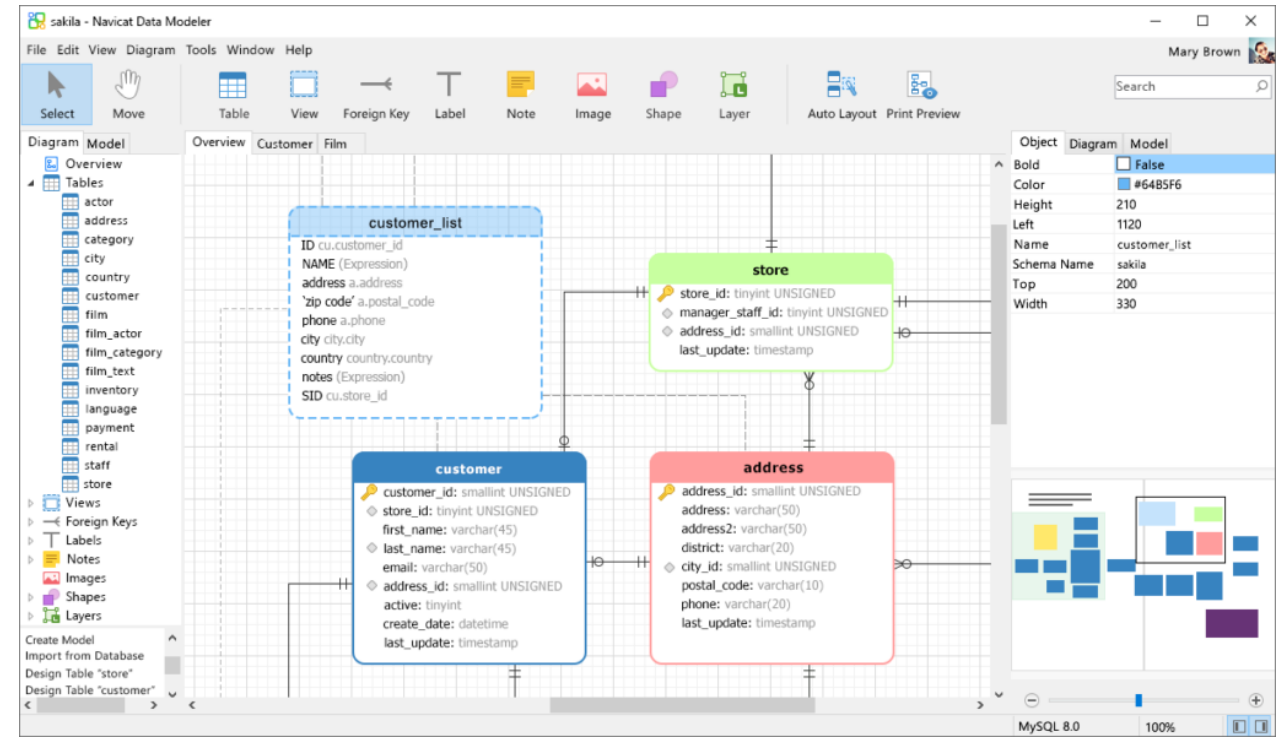
UNIFIED MODELING LANGUAGE™



Modélisation
UML

Louis-Edouard LAFONTANT

Modélisation en génie logiciel



- Activité consistant à créer une **représentation simplifiée (modèle)** d'un phénomène ou d'un système
- **Modèle** est une **abstraction** du système
- Modèle permet d'étudier **structure** et **fonctionnement** du système

Modèle

- Approfondir compréhension du système
 - Raffinement de l'analyse et de la conception
 - Représentation d'un système existant : **rétro-ingénierie** (*reverse engineering*)
- Réduire la **complexité** du problème par abstraction
- Réunir et visualiser un ensemble de détails choisis
- Favoriser la communication au sein de l'équipe
- Documenter
- Plus le formalisme est précis, plus on peut l'utiliser pour **générer** l'implémentation





UML

Collection de langages graphiques
offrant différentes vues d'un même système

Unified Modeling Language (UML)

- **Norme** de l'OMG (Object Management Group) pour la modélisation OO et **standard** ISO
- Langage majoritairement **graphique** (diagrammes) et non formel
- Moyen de communication qui facilite la **représentation** et la **compréhension** du logiciel (basé sur OO)
- Notations pour décrire les **exigences**, la **conception** et le **déploiement**
- **Extensible** (via les profiles)
- **Abstrait**: indépendant des langages de programmations, domaines d'application ou processus de développement





Attention! UML n'est pas...



Un langage de programmation

- Peut être utilisé pour générer le code dans un langage donné
- La modélisation est une activité d'abstraction

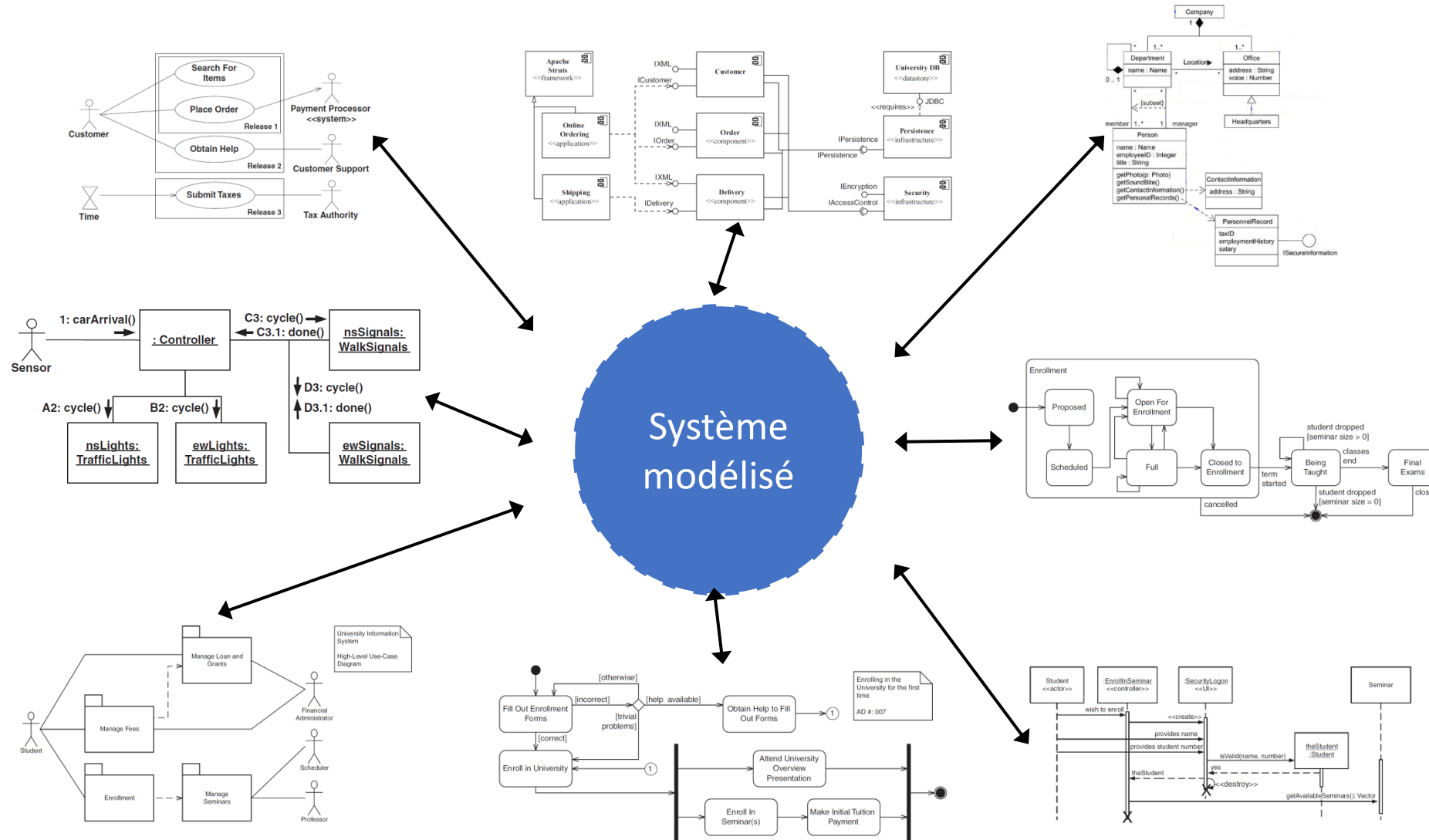
OU



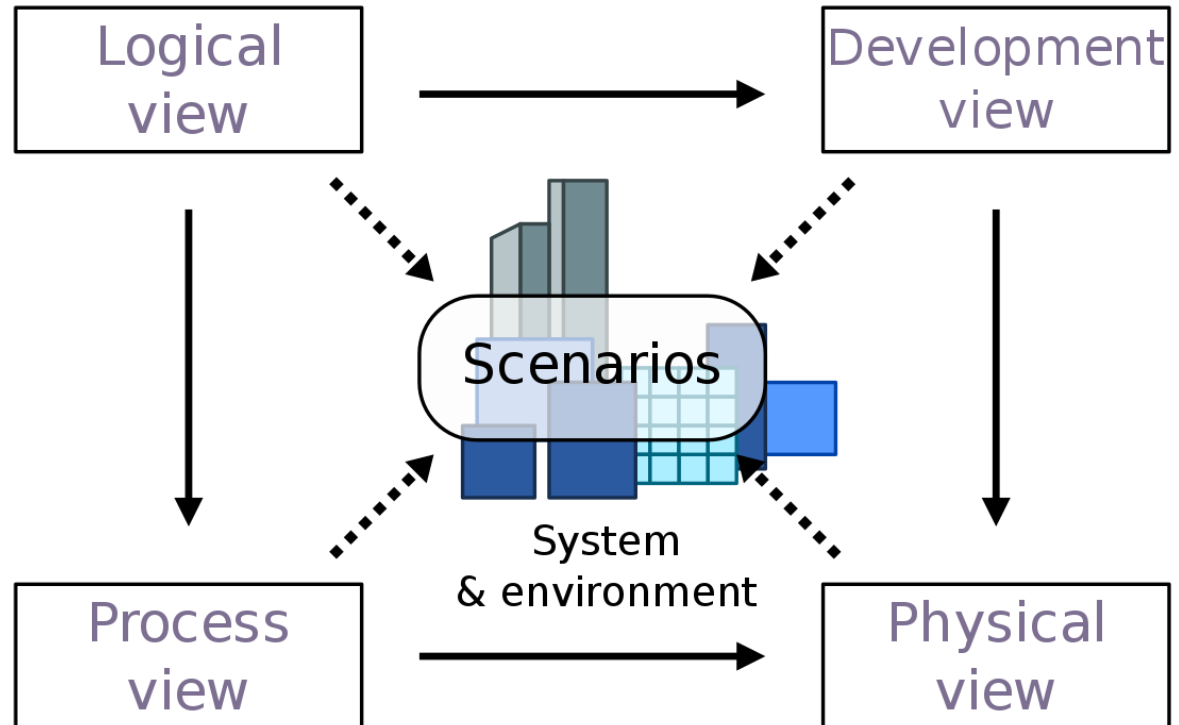
Un processus

- Le processus unifié est souvent utilisé quand on modélise en UML

Différentes vues sur le même système

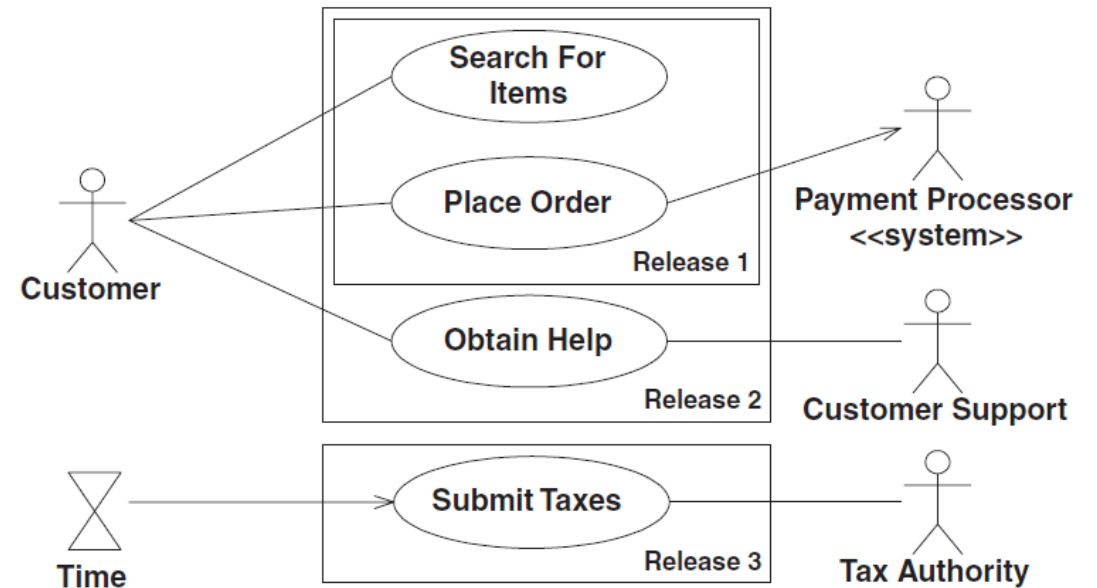


Modèle 4+1



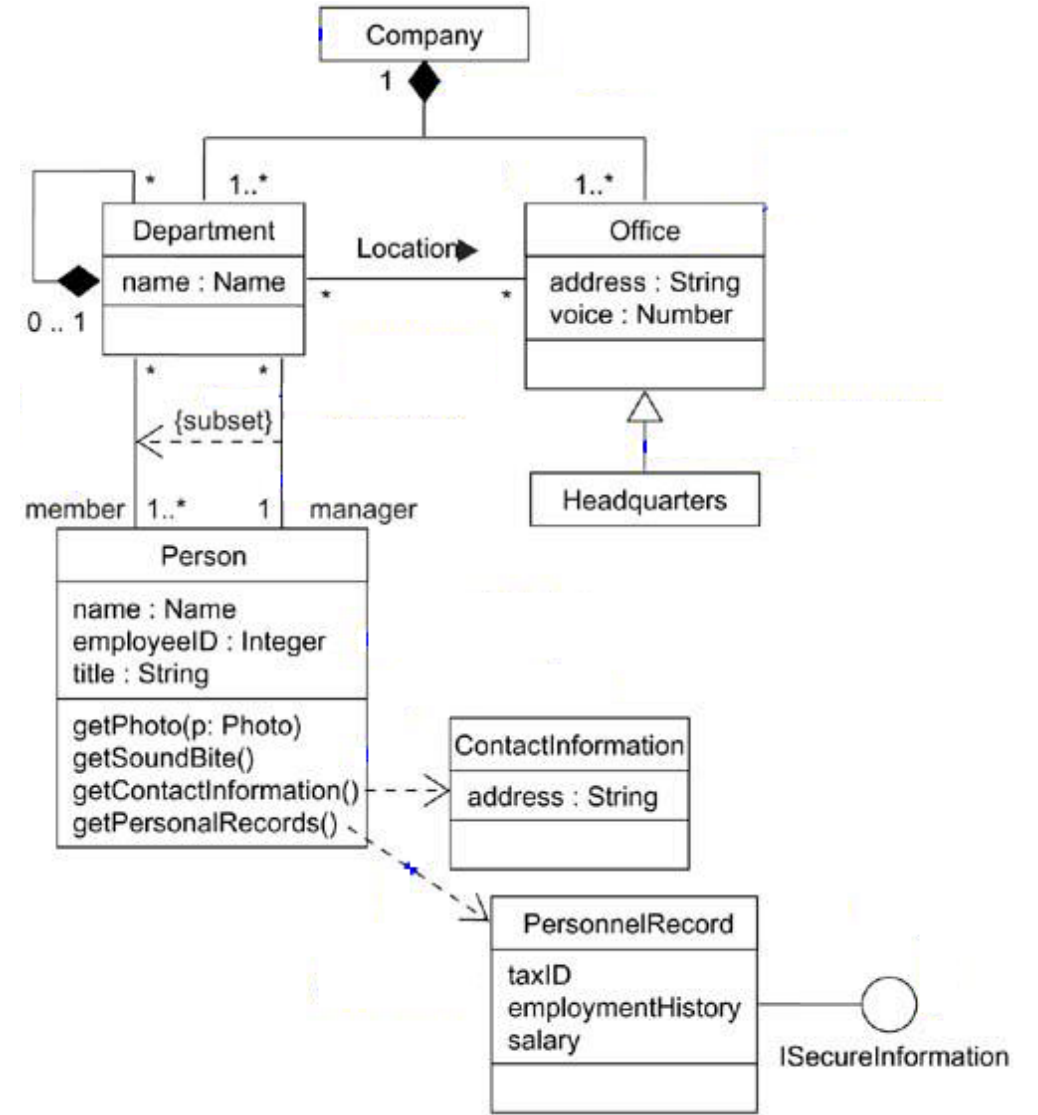
Vue des scénarios

- Ce qui tient tout ensemble
- Cohérence du système, validité
 - Tests
- Diagramme UML impliqué
 - Diagramme de CU
 - Scénarios écrits des CUs



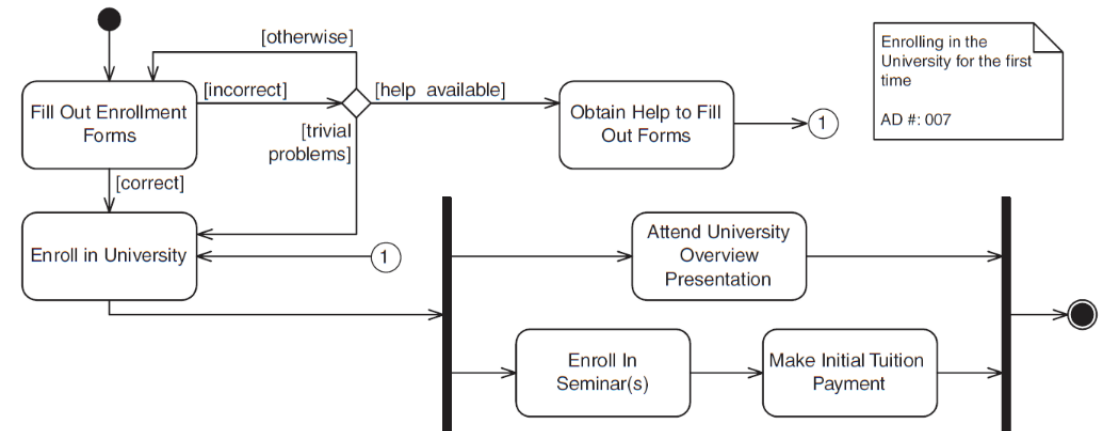
Vue logique

- Vue de conception
- Décomposition orientée objet
- Exigences fonctionnelles: services que le système doit fournir aux utilisateurs
- Diagrammes UML impliqués
 - Diagramme de classes
 - Diagramme d'objet
 - Diagramme d'état
 - Diagramme de séquence
 - Diagramme de communication



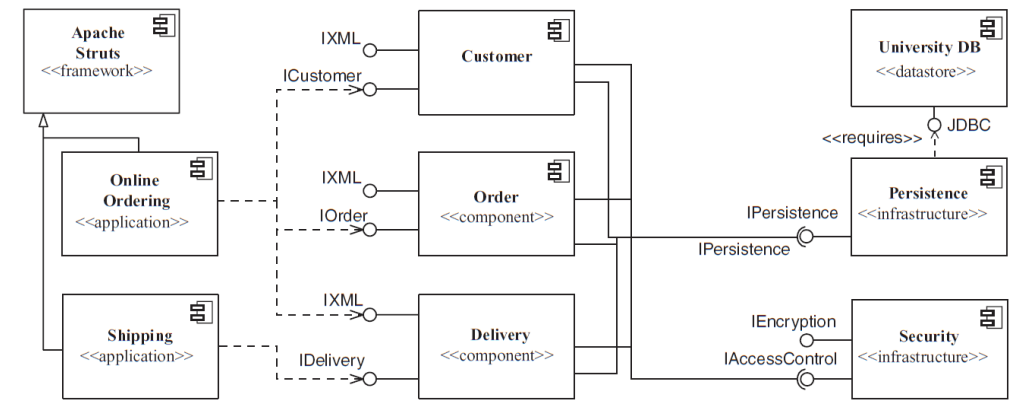
Vue du processus

- Processus et leurs communications
- Exigences non fonctionnelles
 - Performance, scalabilité, débit du système
- Diagramme UML impliqué
 - Diagramme d'activité



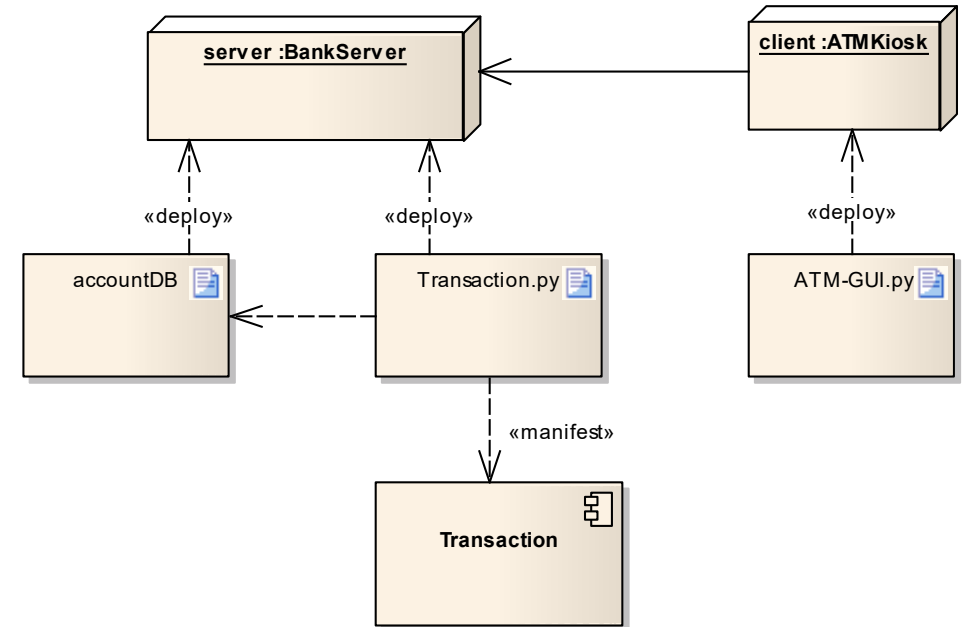
Vue de développement

- Décomposition en sous-systèmes
- Organisation des modules
- Couches hiérarchiques, réutilisation, contraintes d'outils
- Diagrammes UML impliqués
 - Diagramme de composants
 - Diagramme de paquets



Vue physique

- Lien entre logiciel et matériel hardware
- Exigences non fonctionnelles sur le matériel
 - Topologie, communication
- Diagramme UML impliqué
 - Diagramme de déploiement

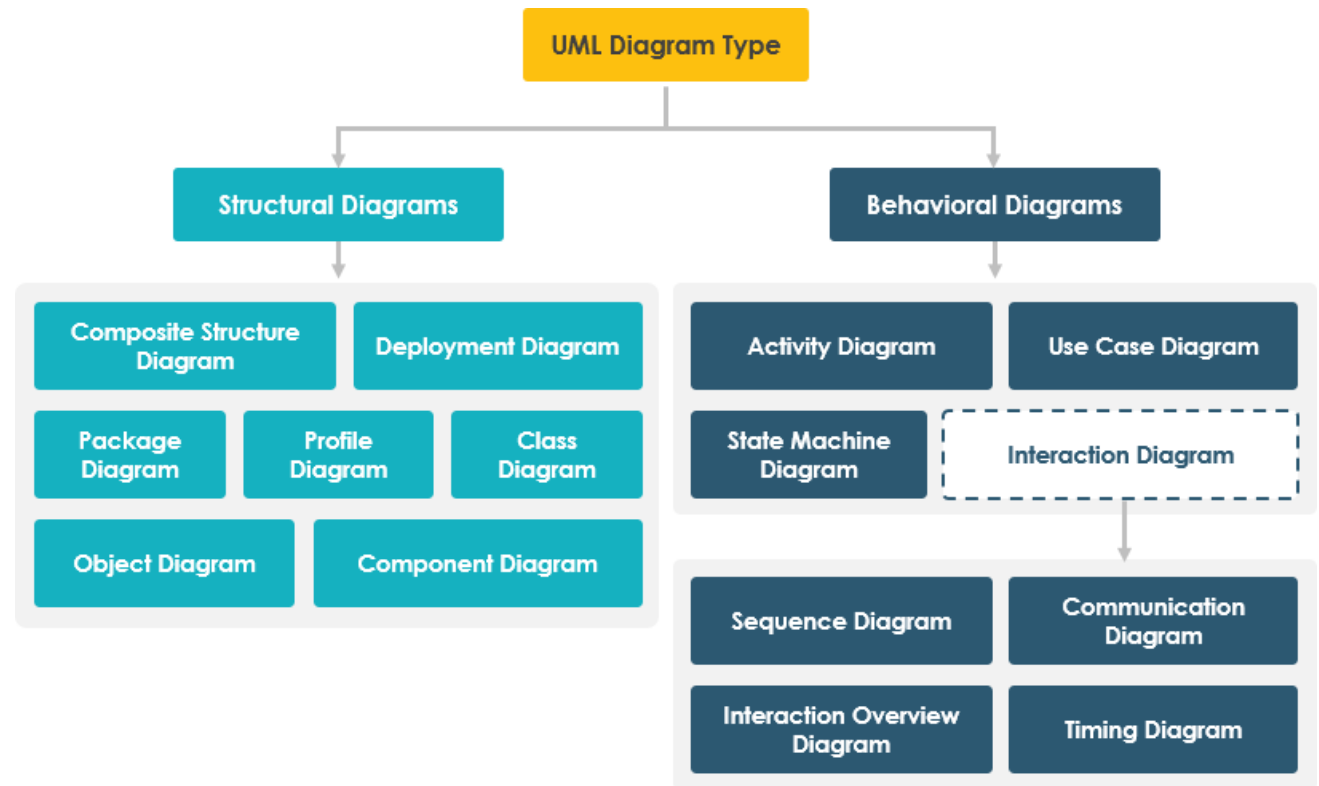




Diagrammes UML 2.X

14 diagrammes divisés en

- Structure
- Comportement





Diagrammes utilisés dans ce cours

- Diagramme de **cas d'utilisations**
Exigences du système du p.d.v. des acteurs qui jouent différents rôles en interagissant avec le système
- Diagramme d'**activités**
Comportement dynamique du processus d'affaire, logique des procédures
- Diagramme de **classes**
Modélisation de la structure statique des entités et leurs relations
- Diagramme de **séquence**
Comment les objets communiquent entre eux au fil du temps pour réaliser chaque CU