

Spécification et conception avec

UML

Partie 1 - Modélisation

Modélisation microscopique : mono/multi diagrammes

- Objectifs
 - Comprendre, élaborer et raisonner par le **modèle UML** (modélisation)
 - Assimilation notation et des règles de vérification des diagrammes
 - ⇒ le modèle = « un contrat » compris souscrits par tous les participants
- Déroulement
 - Vision objet et modélisation graphique
 - Diagramme par diagramme (classe, états, cas d'utilisation et activité)
 - Exemples, notation, exercices mono et multi diagrammes
 - Papier et outillage
 - Survol des diagrammes restants



(Partie 2 – Spécification)

Étude de cas *macroscopique et documenté*

- Objectifs
 - Comprendre, élaborer et raisonner par la spécification (documentation)
 - Assimilation de la méthodologie et des règles de vérification
 - ⇒ Les spécifications (SFG/SFD) = « un contrat » compris souscrits par tous : User ← CDP → Dev
- Entrée
 - Expression des besoins réaliste, variée, mais réduite en taille
 - Modèle Word SFG/SFD
 - Feuille de route méthodologique : notation + démarche / checklist
- Sortie
 - Projet UML/Outil
 - DocumentSpécification (SFG, sans redondance)

Plan (I)

- I. Introduction
 - Vision objet et modélisation graphique avec UML
 - UML - concepts clé et survol des diagrammes
- II. Modélisation objet, diagramme par diagramme
 - Classe, états, cas d'utilisation et activité
 - Exemples, notation, exercices
 - Mono et multi diagrammes
 - Papier et outillage
- III. Survol des diagrammes restants
 - Séquence & collaboration
 - Composantes & déploiement

Cycle de vie d'un projet

- Phases et activités (simplification)
 - Analyse, Conception, Développement, Test
 - Test : exécution
 - Exécutable + autres ressources (BD, fichiers)
 - Développement : programmation
 - Sources : PascalObjet, C++, Java, C#, ...
 - Analyse : compréhension du problème
 - Cahier de charges
 - Conception : ébauche de la solution
 - Modèle

© 2006 zam@lamsade.dauphine.fr

12

Ce que nous captons dans un modèle

- Décrire les objets / classes
 - Anatomie et physiologie des objets
 - Histoire individuelle
 - Interactions & histoire collective
 - Photos de famille

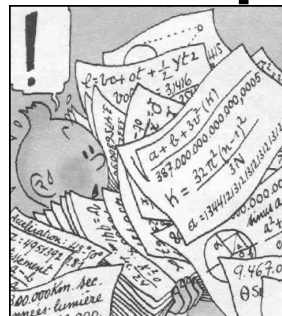


© 2006 zam@lamsade.dauphine.fr

16

UML par l'exemple

*La moitié des bugs
sont imputables
aux exigences*

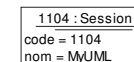
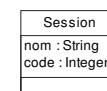


© 2006 zam@lamsade.dauphine.fr

27

Aperçu de la notation UML

- **Système**
 - Ce que l'on veut modéliser (application / organisation)
- **Modèle**
 - Description du système
 - Concepts
 - Acteur, cas d'utilisation
 - Objet, classe, attribut ...
- **Vue**
 - Diagrammes de ...
 - Cas d'utilisation, Classe, Objet
 - Séquence, Collaboration, État, Activité
 - Composants et Déploiement
 - Symboles
 - Icônes spécialisées : classe, objet



© 2006 zam@lamsade.dauphine.fr

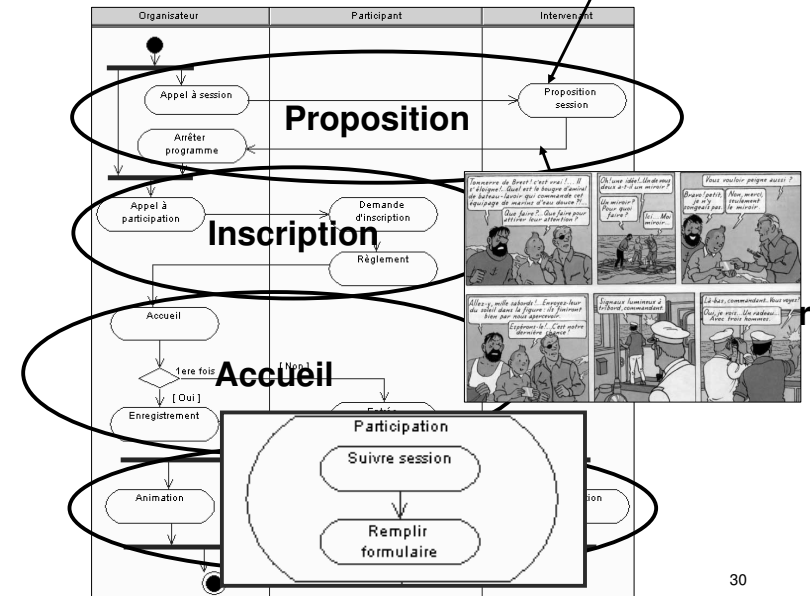
28

Diagramme de cas d'utilisation



29

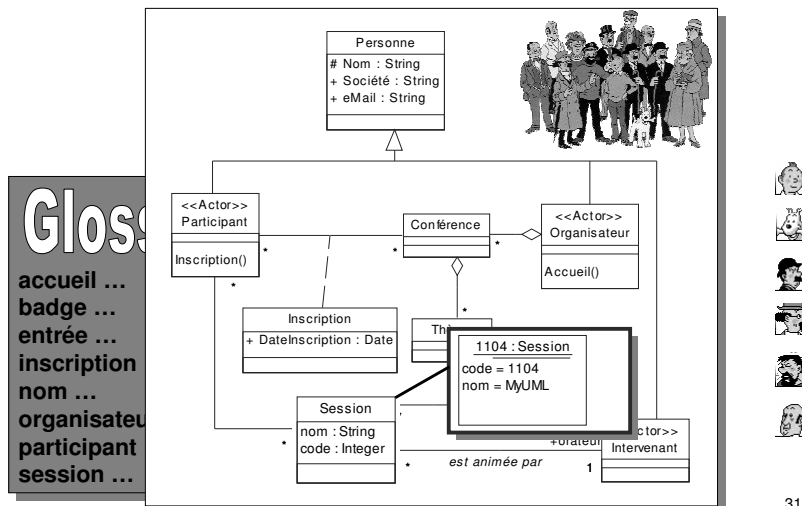
Diagramme d'activités



30

Diagramme de classes

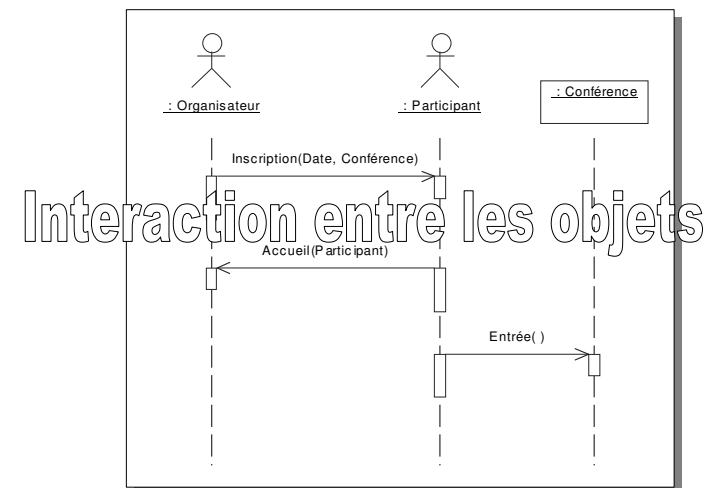
Classe, attribut, opération, association, agrégation, généralisation, multiplicité



31

Diagramme de séquence

- Objet, ligne de vie, message, activation



32

Diagramme de collaboration

- Objet, message, lien

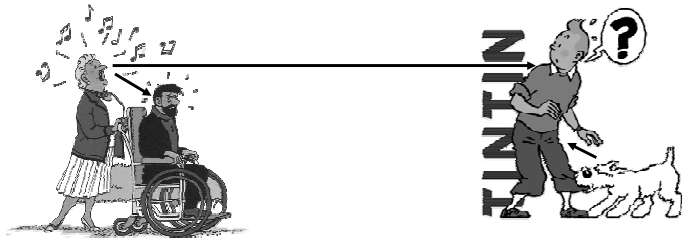
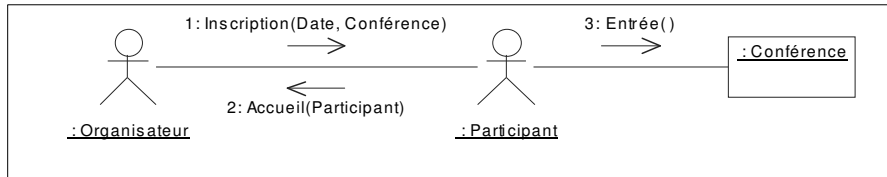
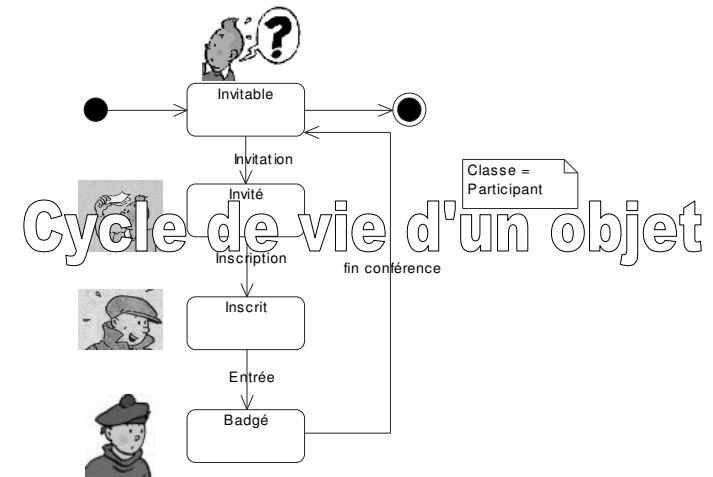


Diagramme états-transitions

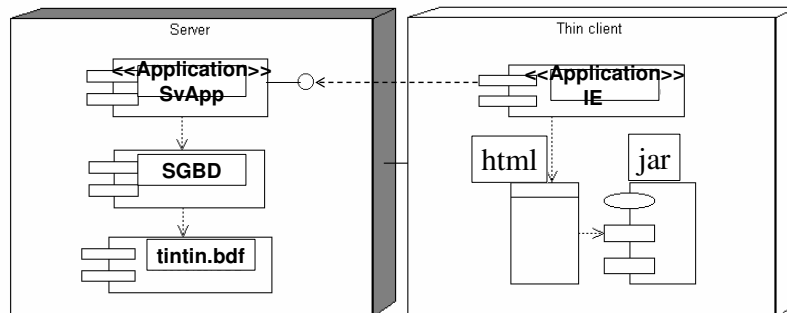
- État, transition



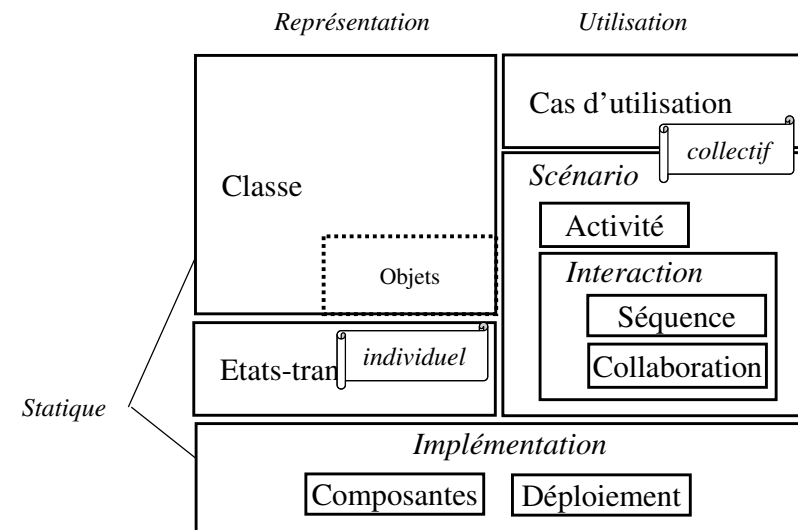
Cycle de vie d'un objet

Diagrammes d'implémentation

- Composants et déploiement



Diagrammes UML



II. Modélisation objet

Diagramme par diagramme
(Classes, Etats, UseCases & Activités)

Diagrammes statiques UML

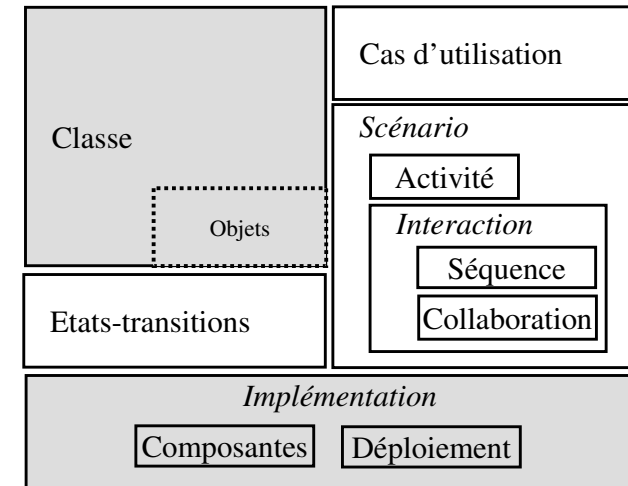


Diagramme d'objets

Diagramme d'objets

- Objectif
 - Illustrer des objets, les valeurs éventuelles de leurs attributs et leurs connections afin de pouvoir raisonner sur le fonctionnement de l'application

La classe

• Caractérise l'objet

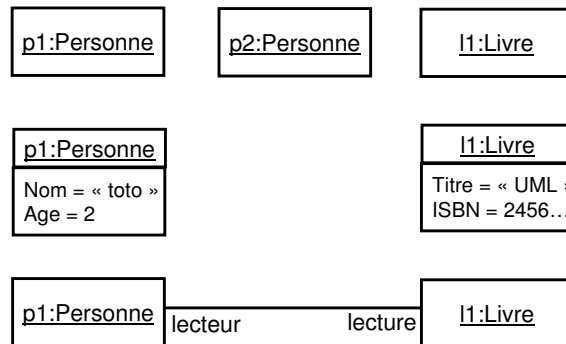


Diagramme de classes

Diagramme de classes

• Objectif

- Représenter la structure statique des objets qui composent un système à travers des classes et des relations entre les classes.
- Typer les objets
 - Structure
 - (Comportement)
 - Connectivité

Classe - notation UML 1.5 [p3-37]

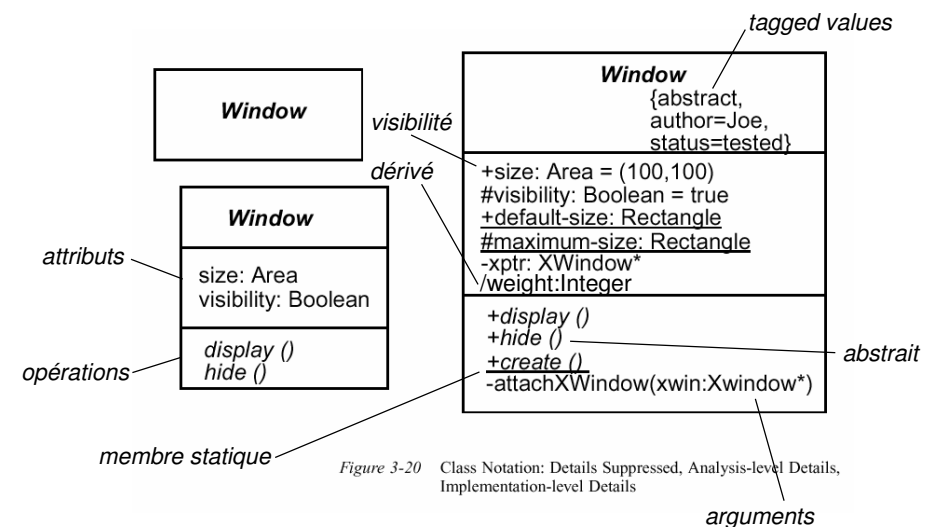
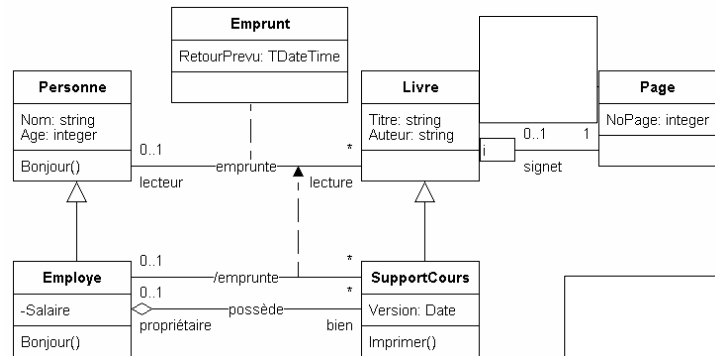


Figure 3-20 Class Notation: Details Suppressed, Analysis-level Details, Implementation-level Details

Exemple complet (8)

- Association dérivée



Objets et classes

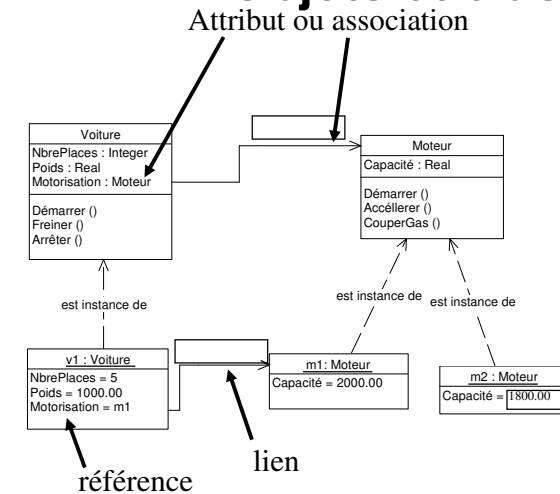
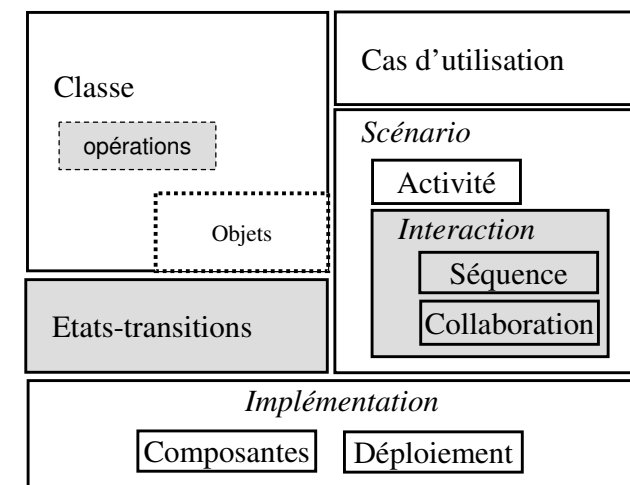


Diagramme d'états

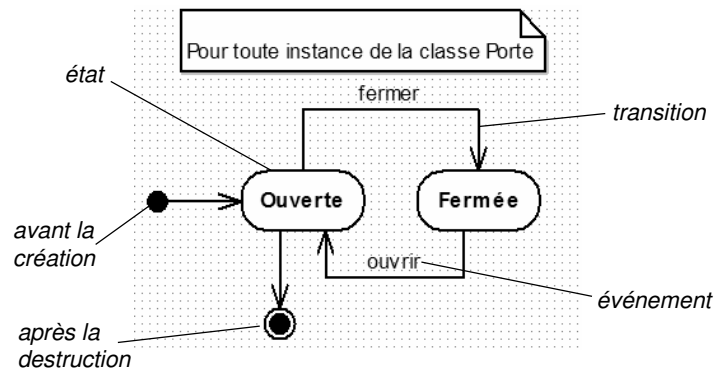
ou « de transitions d'états »
 ou « d'état-transitions »
 ou « machines à états (statecharts) »

Diagrammes dynamiques / objets

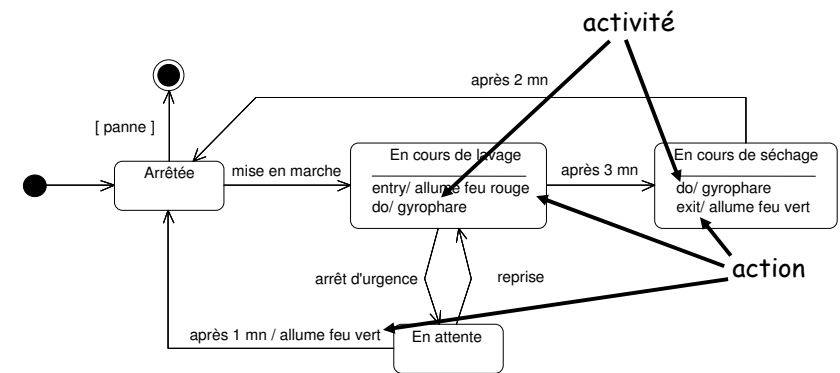


Objectif

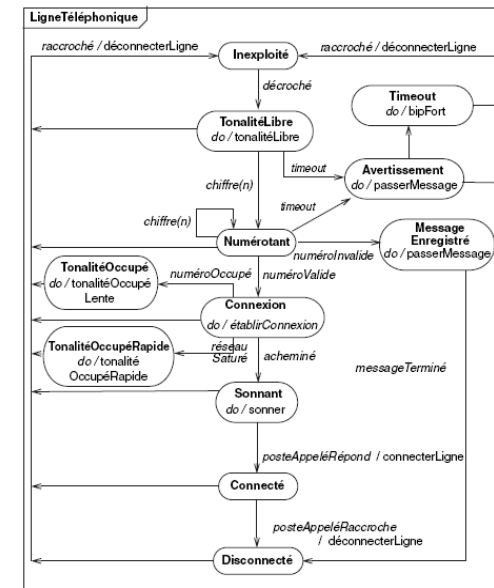
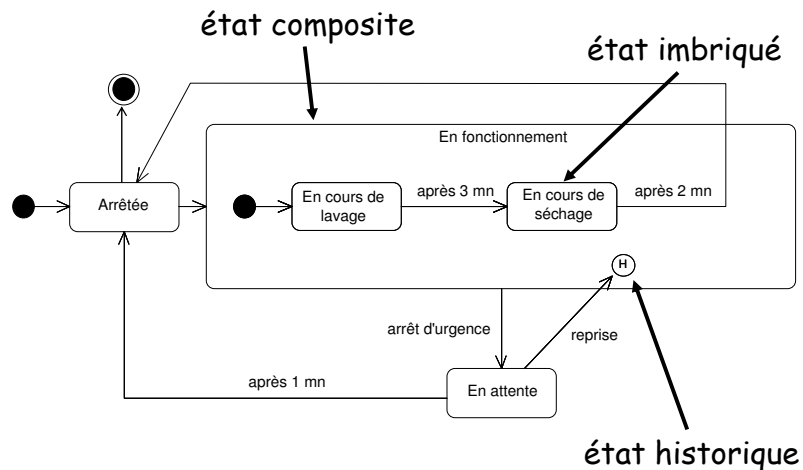
- Décrire les états remarquables d'un **seul** objet et les transitions possibles



Actions et activités



États imbriqués



[Blahe&Rumbaugh05]

Figure 5.17 • Diagramme d'états avec activités pour une ligne téléphonique. Les diagrammes d'états permettent d'exprimer ce que font les objets en réponse aux événements.

Conseils

- États d'un **seul** objet
 - Vue égoïste
- Choisir les classes adaptées
 - États remarquables qui conditionnent le comportement de l'objet : applicabilité des méthodes
- États mutuellement exclusifs
 - Éviter les états parallèles
- Déterminisme
 - état départ + événement + condition → état d'arrivée unique
- Attention aux objets complexes

1. Exercice de lecture

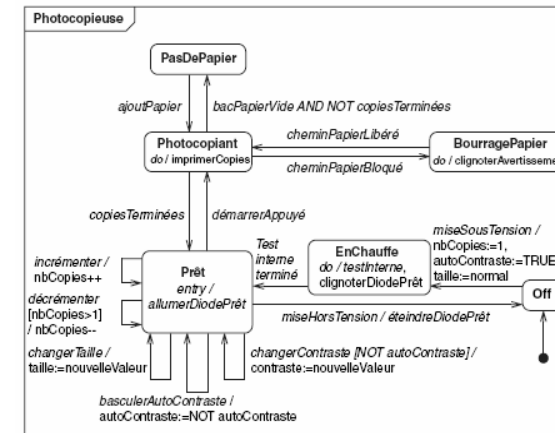


Figure E5.4 • Diagramme d'états d'une photocopieuse.

- Identifier les concepts UML
- Interpréter le diagramme
- Proposer un diagramme de classes associé
- Proposer une évolution