

POO

Programmation Orientée Objet Bases de conception



1

Introduction

- ❑ **Norme Object Management Group depuis 1997**
- ❑ **Langage pseudo-formel qui fournit :**
 - ❖ **des concepts**
 - sémantique des concepts
 - classement des concepts (niveau d'abstraction ou domaine)
 - ❖ **une notation**
 - représentation graphique des concepts
 - ❖ **des vues complémentaires**
 - dynamique, statique, fonctionnelle
 - ❖ **un support de communication**
 - indépendant des langages et domaines d'application
 - expression graphique
 - limite les ambiguïtés

2



2

Introduction

□ Avantages

❖ Standard, polyvalent et performant

- Expression des idées (relation client)
- Représentation de concepts abstraits
- Limite les ambiguïtés
- Facilite l'analyse

❖ Démarche de conception

- Description des principaux aspects d'une conception objet
- Définition des vues

□ Inconvénients

- ❖ Lourdeur de la mise en place
- ❖ Apprentissage assez long (si utilisation rigoureuse)

Doit rester une aide à la conception et non un obstacle ...

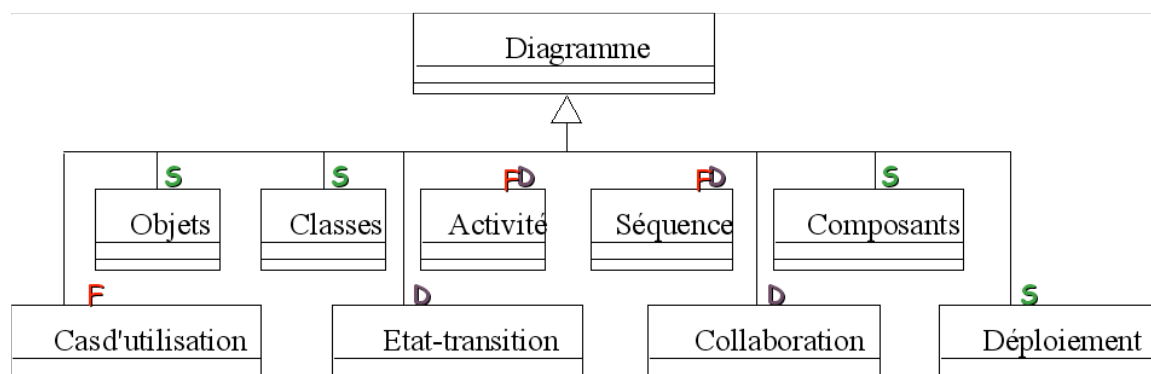
3



3

9 diagrammes (UML1)

- Use Case Diag. : représentation des fonctions du système du point de vue de l'utilisateur
- Diag. de classes : structure statique des classes et relations
- Diag. de séquence : représentation temporelle des interactions entre objets
- Diag. d'état : comportement interne d'une classe



4



4

Démarche de conception

□ Grandes lignes

❖ Spécification

- Analyser, déterminer, décrire les processus métier
- Identifier les besoins
- Déterminer les cas d'utilisation (UC)

❖ Conception (simultanément)

- Décrire les scénarios des UC
 - Diagrammes d'activité
 - Diagrammes de séquence détaillés
- Identifier les classes du système
 - Diagrammes de classes
 - Diagrammes d'état

5



5

Acteurs

□ Personne ou *Système externe*

□ Interagit DIRECTEMENT avec le système à développer

□ Acteur principal

❖ Déclenche l'UC

□ Acteur secondaire

❖ Sollicité par l'UC

□ Identifier les acteurs

❖ Rôle et non personne réelle

❖ Systèmes externes "boîtes noires" interagissent avec le système

- lecteur de badge, validation de carte bancaire, caisse enregistreuse, etc.

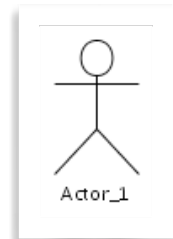
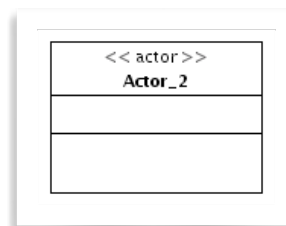
6



6

Acteurs

□ Représentation graphique

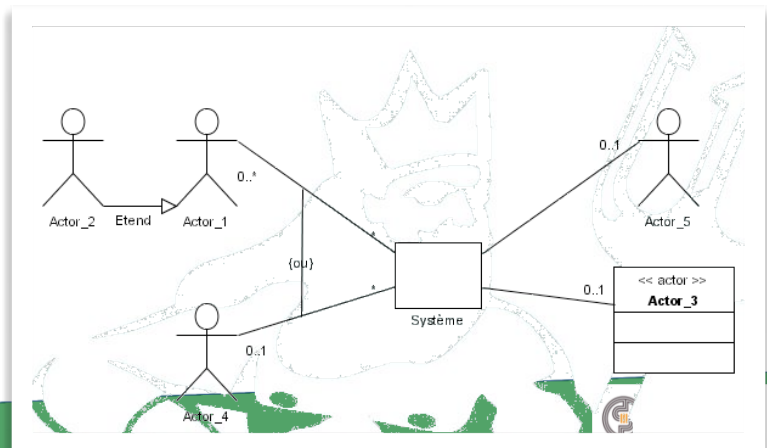


□ Représentation textuelle

- ❖ Description du rôle de l'acteur

□ Répertoireier les acteurs

- ❖ Diag. de contexte statique



7

Cas d'utilisation (UC)

□ Objectif

- ❖ Déterminer les besoins
- ❖ Décrire un système du point de vue de l'utilisateur
- ❖ Etablir le cahier des charges

Comportement du système, ce qu'il doit faire,
et **non** comment il le fait !!

□ Identifier les UC

- ❖ Pour chaque acteur

□ Analyser les UC

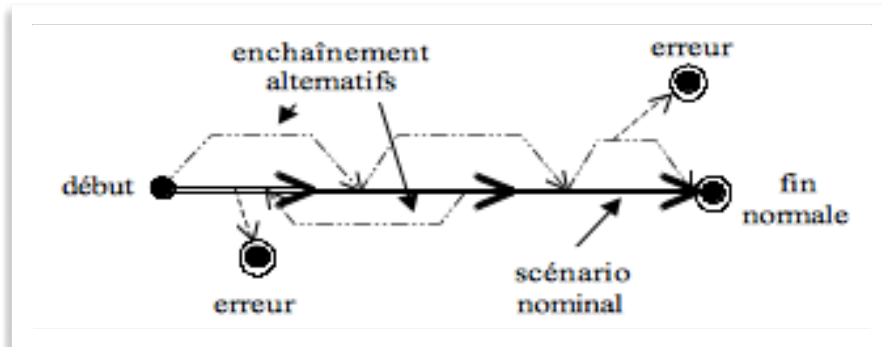
- ❖ Cas nominal
- ❖ Variantes
- ❖ Erreurs

8

Cas d'utilisation (UC)

□ Déroulement d'un UC

- ❖ Un scénario par variante
 - Cas particulier d'un UC
- ❖ Un scénario = séquence d'action

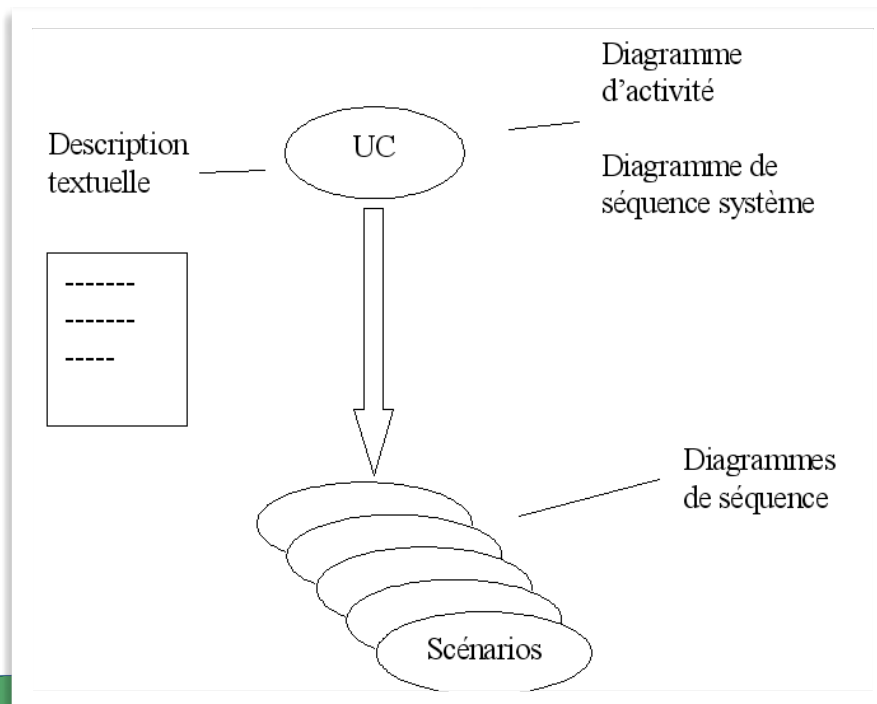


9

9

Cas d'utilisation (UC)

□ Représenter les UC



10

Représentation textuelle d'un UC	
Nom	Commence par un verbe
Cadre	Système en cours de conception
Brève description	
Pré-conditions	Hypothèses et conditions liées à l'environnement. Qu'est-ce qui doit être assuré ET qui vaut la peine d'être dit
Post-conditions	Si nécessaire
Création date	
Modification date	
Acteur primaire	
Acteurs secondaires	
Niveau	Primary (Directement accessible à l'utilisateur) or Secondary (Include or extend, sub fonction)
Source documents	
Besoins/prérequis	
Fréquence d'occurrence	
Micellaneous	
Scénario principal	(aka Main success scenario or Basic flow) Scénario typique et sans condition de succès
Extensions	Scénarios alternatifs amenant au succès ou en erreur (failure)

11

Représentation textuelle d'un UC

□ Numérotation des actions du scénario nominal

- ❖ 1. Action de l'acteur : ...
- ❖ 2. Action de l'acteur : ...
- ❖ 3. Réponse du système : ...
- ❖ Répétition des étapes 2 et 3 jusqu'à
- ❖ 4. Réponse du système : ...
- ❖ ...
- ❖ Fin normale du cas

12



12

Représentation textuelle d'un UC

❑ Scénarios alternatifs Flows (lettre = alternative, chiffre = flow)

- ❖ *a. A tout moment, changement d'état : ...
 - *a.1. Réponse du système : ...
 - *a.2. Action de l'acteur : ...
 - *a.3. Le système retourne dans le mode normal
- ❖ 2-3a : Variante 1 (Ex. : Erreur de saisie) break en 2 ou 3
 - 2-3a.1. Action de l'acteur : ...
 - 2-3a. 2. Réponse du système : ...
 - 2-3a. 3. Le système retourne à l'étape 2-3
- ❖ 3a : Variante 2 (Ex. Montant > max autorisé) break en 3
 - 3a.1. description des actions avec retour en 3
- ❖ 3b : Erreur 1 (Ex. Login invalide) break en 3
 - 3b : ... description des actions avec fin anormale du cas

13



13

Représentation graphique des UC

❑ Diagramme de Cas d'Utilisation

- ❖ "Include"
 - Factorisation de parties communes
 - Identification de sous-cas
 - Le cas inclus est TOUJOURS exécuté
- ❖ "Extend"
 - Séparation d'un sous-cas optionnel ou rare
 - Le cas étendu est exécuté de façon conditionnelle
- ❖ Généralisation
- ❖ Spécialisation
 - UC fils héritent de l'UC parente mais en modifiant les interactions du parent

14

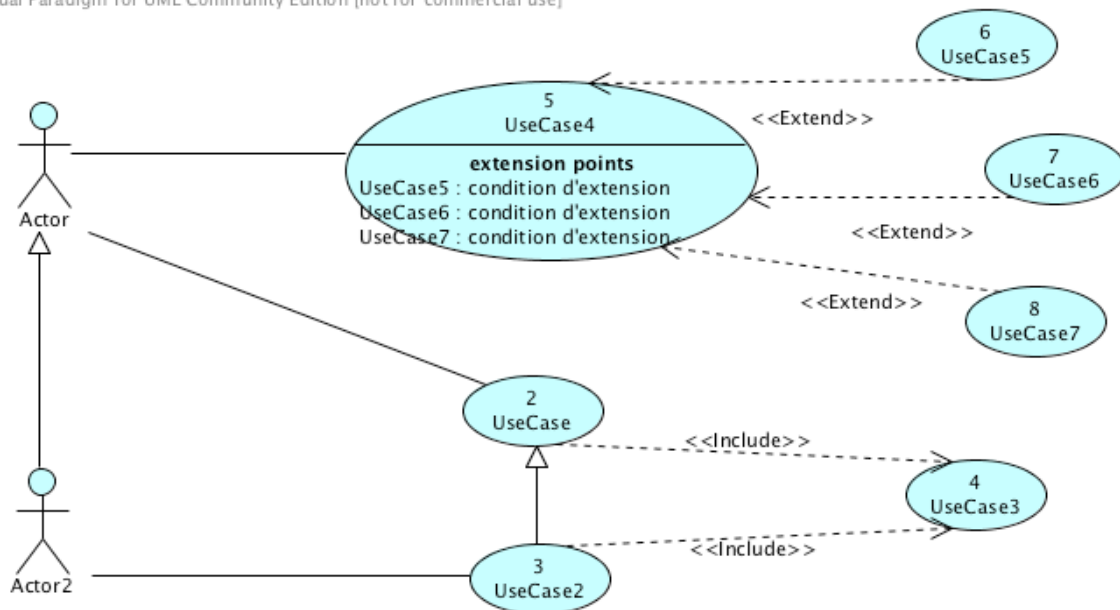


14

Représentation graphique des UC

Diagramme de Cas d'Utilisation

Visual Paradigm for UML Community Edition [not for commercial use]



15

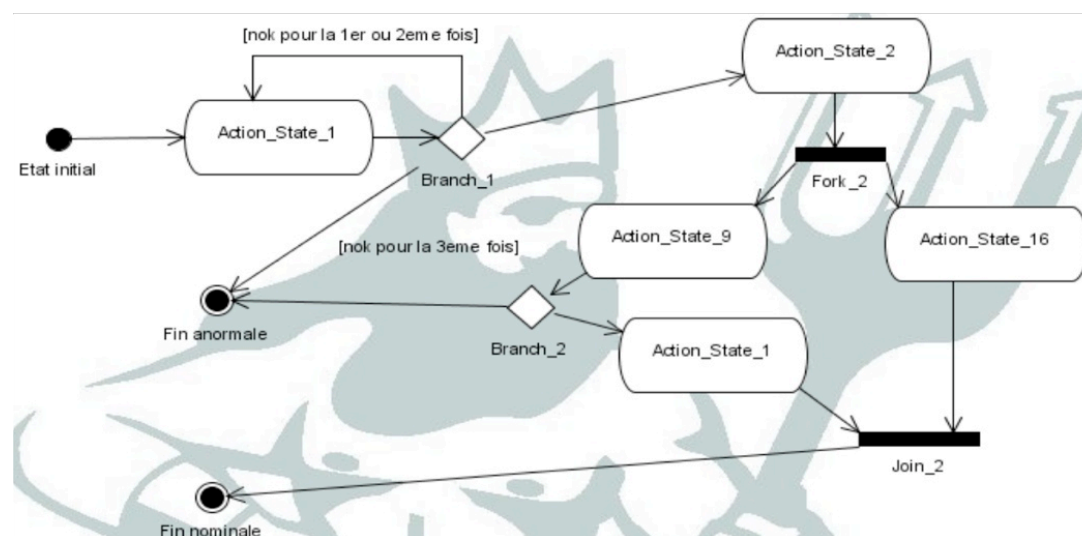
15

Représentation graphique des UC

Diagramme d'activité d'un UC

❖ Représente l'ensemble des activités avec

- Tous les branchements conditionnels
- Toutes les boucles possibles



16

Stéréotypes

□ Proposés par Jacobson

❖ Boundary

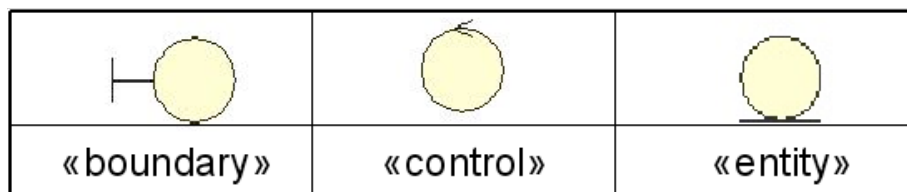
- Interface entre le contrôle et l'acteur principal

❖ Entity

- Classes du système à développer

❖ Control

- Contrôle et validation des données entre Boundary et Entity



Object Management Group, UML Superstructure Specification , v2.0, August 2005

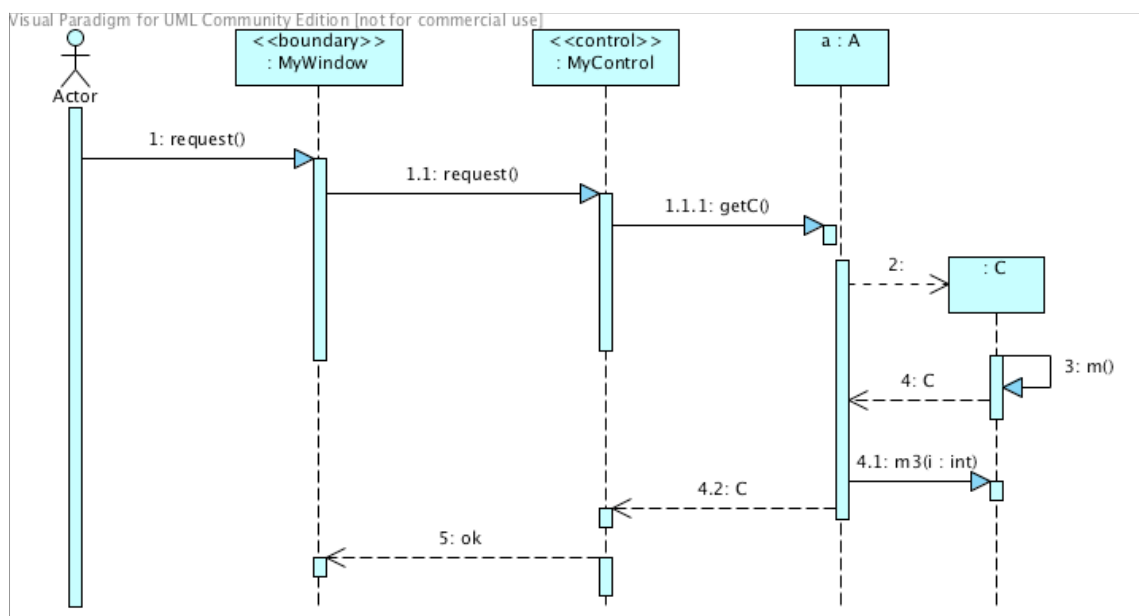


17

17

Diagrammes de séquence

□ Vue dynamique



18

18

Diagrammes de séquence

❑ Règles générales :

❖ Un acteur principal

- A l'origine du déclenchement de l'UC
- Plusieurs acteurs secondaires possibles

❖ Au moins une classe « Boundary »

- Un ou plusieurs éléments d'interface

❖ Au moins une classe « Control » par diagramme de séquence

- Contrôle les flux d'information entre « Boundary » et le système

❑ Par contre :

❖ Les classes «Entity » pas obligatoirement présentes

- En cas d'appel exclusif à des systèmes externes (ex : GAB)

19



19

Diagrammes de classes

❑ Multiplicité

❖ Nombre d'instances participant à une association

❖ Explicite

1	Un et un seul
0..1	Zéro ou un
M..N	De M à N (entiers naturels)
*	De zéro à plusieurs
0..*	De Zéro à plusieurs
1..*	D'un à plusieurs

20



20

Diagrammes de classes

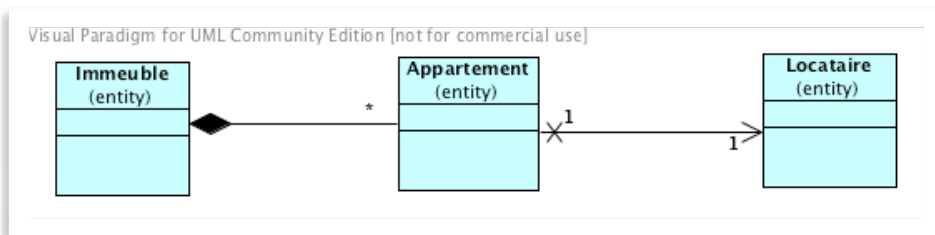
Relations

❖ Association simple ("avoir")

- Une classe possède un attribut instance de l'autre classe

❖ Composition

- Couplage fort
- Relation de composant composé



21

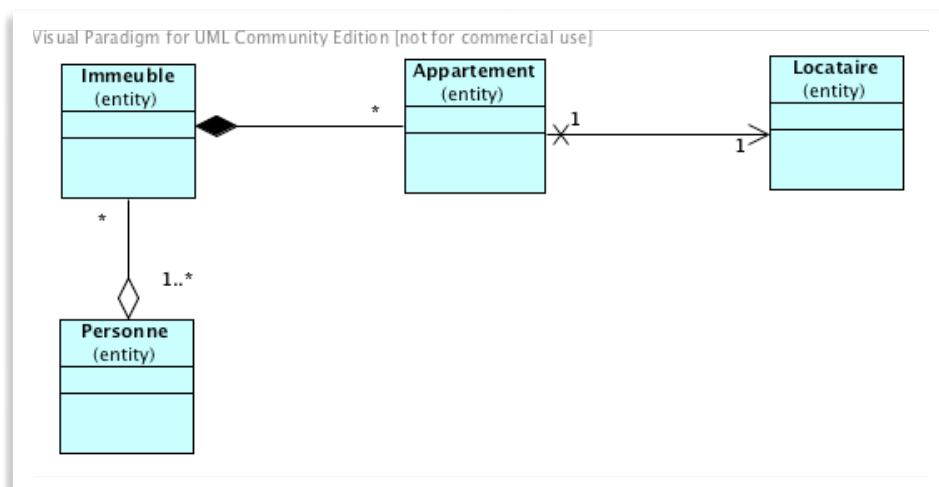
21

Diagrammes de classes

Relations

❖ Agrégation

- Couplage moins fort que composition
- Généralement, une classe contient une collection de références d'une autre classe



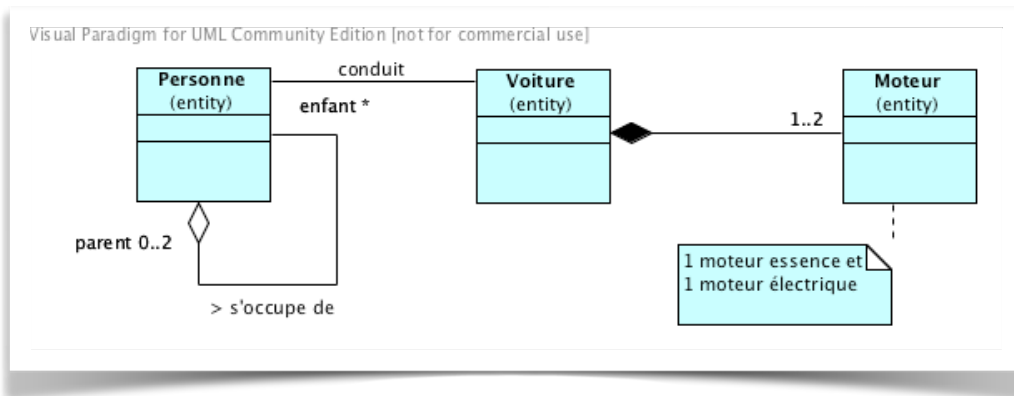
22

22

Diagrammes de classes

Relations

- ❖ Agrégation
 - Réflexive



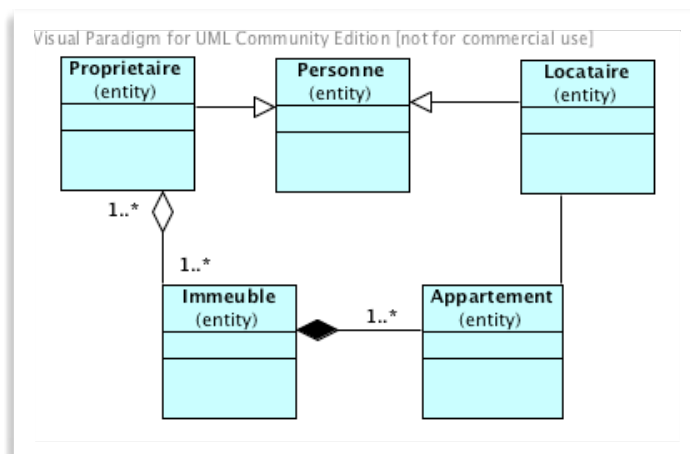
23

23

Diagrammes de classes

Relations

- ❖ Héritage ("être")
 - Cf. Classification

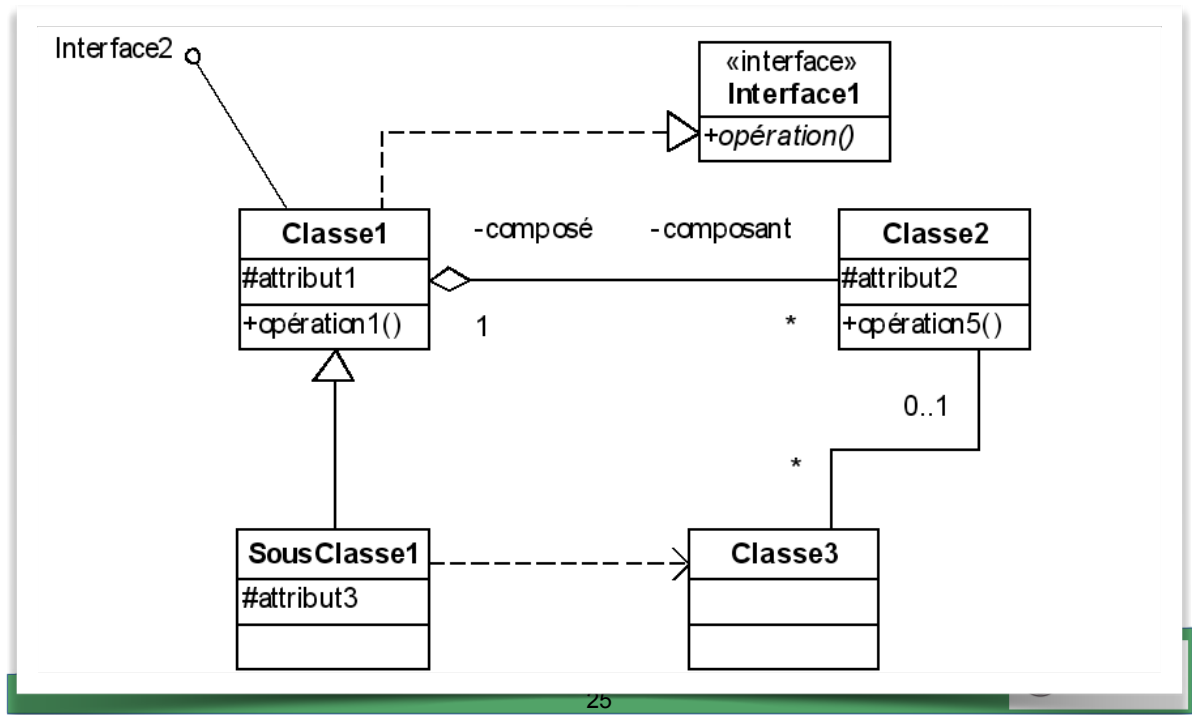


24

24

Diagrammes de classes

□ Récapitulatif



25

Diagrammes d'objets

□ Objets

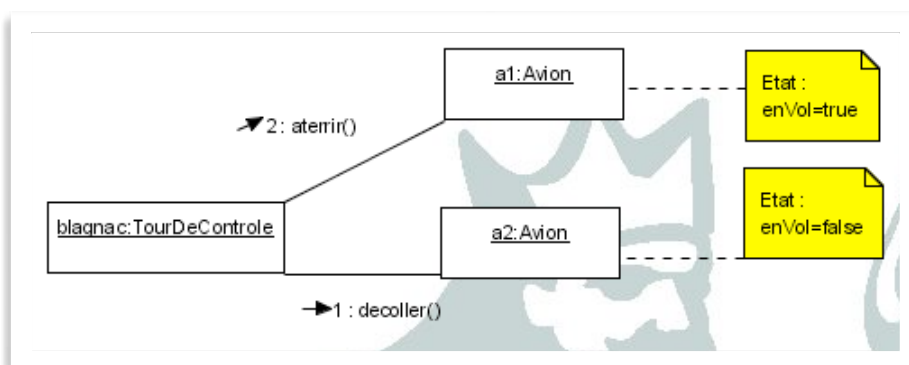
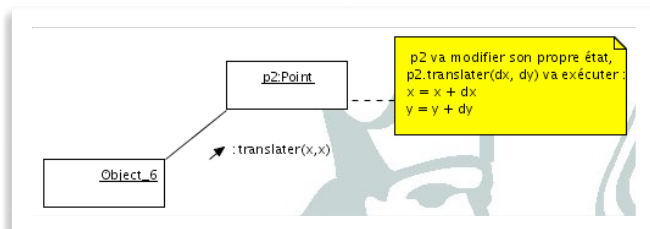
- ❖ instances de classes

□ Liens

- ❖ instances d'associations

□ Messages

- ❖ méthodes invoquées et "naviguant" le long des liens



26

26

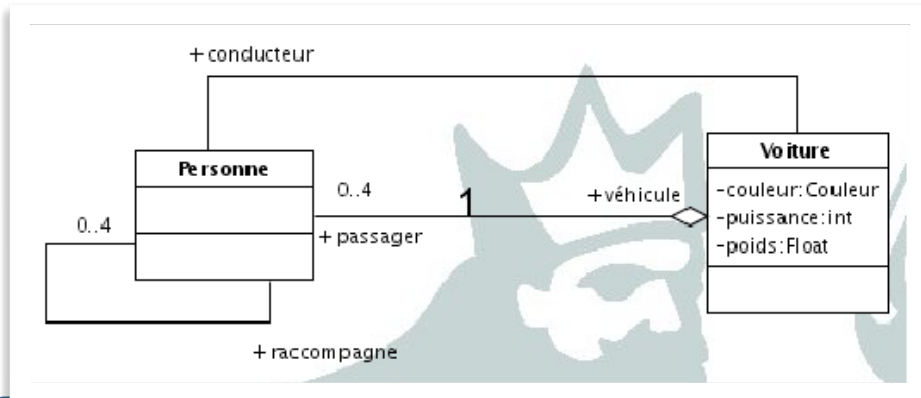
Exercices

□ On dispose des classes **Universite** et **Etudiant**

- ❖ Proposer un diagramme d'objets illustrant
 - Un étudiant est inscrit dans une université
 - Il peut s'inscrire au plus dans 2 universités

□ On dispose de la classe **Personne**

- ❖ Proposer un diagramme d'objets explicitant le diagramme de classes suivant



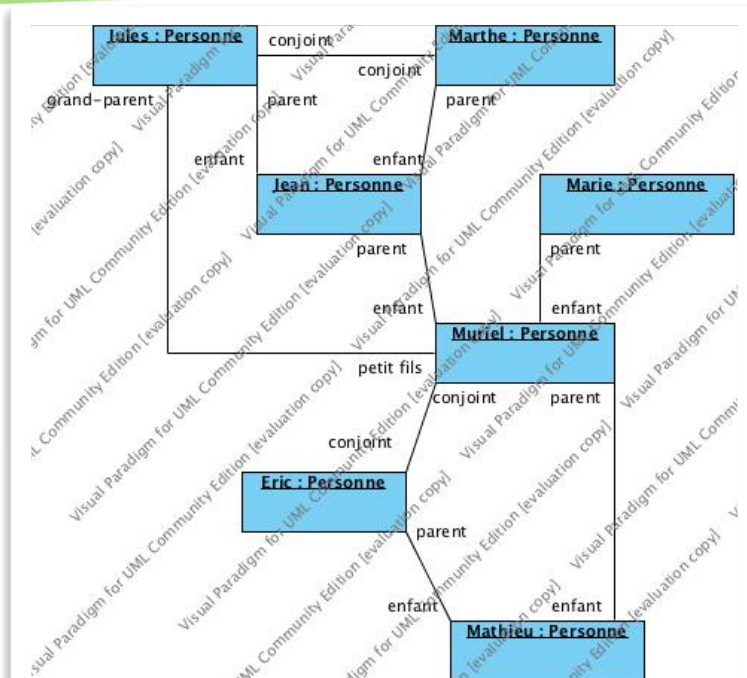
27

27

Exercices

□ On dispose de la classe **Personne**

- ❖ Proposer un diagramme de classes qui explique les relations données par les liens du diagramme d'objets suivant



28

28

Exercices

□ On dispose des classes **Ville et Musee**

- ❖ **Proposer un diagramme d'objets qui illustre :**
 - Un musée est dans une ville
 - Une ville accueille zéro, un ou plusieurs musées

□ On dispose des classes **Livre et Auteur**

- ❖ **Proposer un diagramme d'objets qui illustre :**
 - Un auteur a écrit au moins un livre
 - Un livre peut être écrit par plusieurs auteurs
- ❖ **Quelles classes pour les auteurs et les livres ?**

29



29

Exercice : Bibliothèque

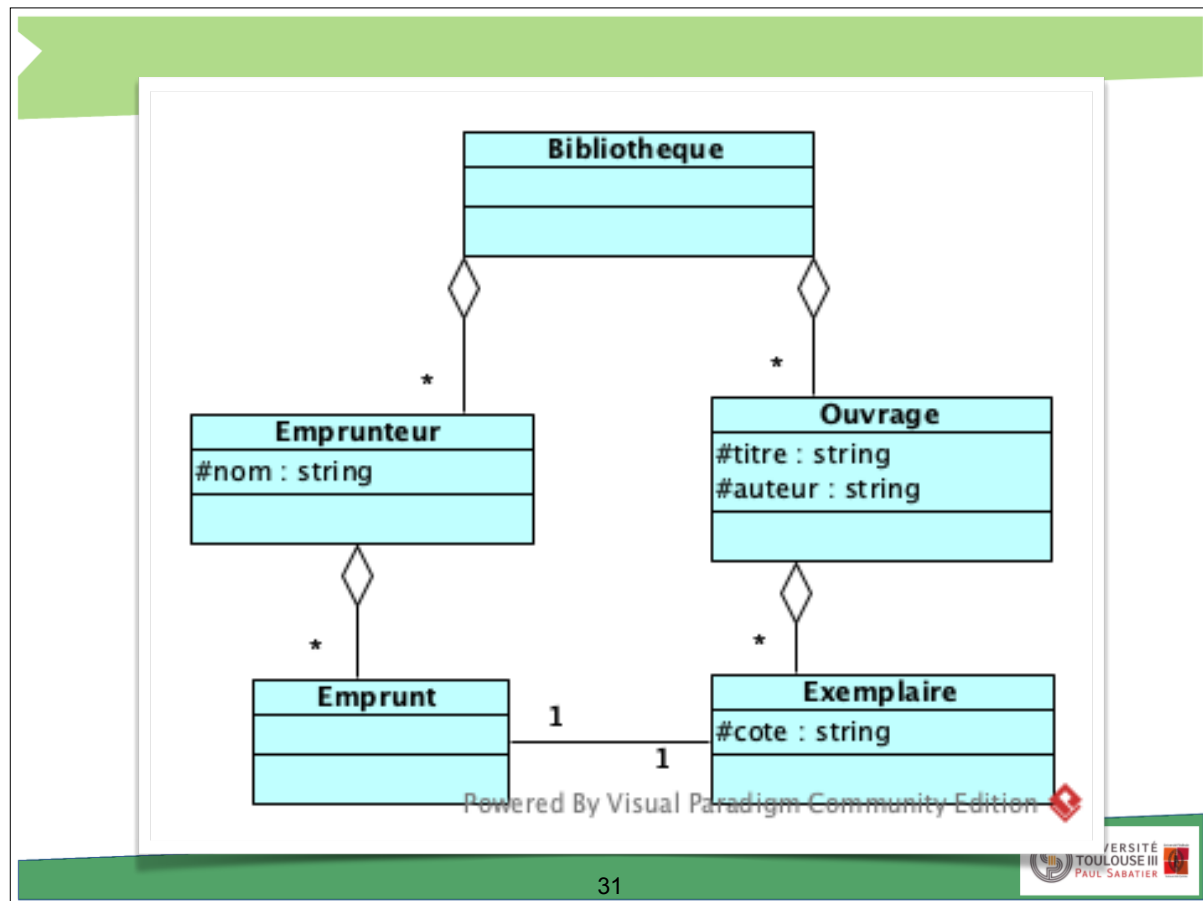
□ Diagramme de classes et codes :

- ❖ **La Bibliothèque permet de gérer Emprunteur, Ouvrage et Exemple**
- ❖ **Plusieurs exemplaires, identifiés par une cote, sont disponibles pour un ouvrage, identifié par le titre et l'auteur**
- ❖ **Les emprunteurs, identifiés par leur nom, empruntent des exemplaires**
- ❖ **On doit pouvoir :**
 - retrouver les exemplaires empruntés par un emprunteur
 - supprimer un emprunteur
 - supprimer un ouvrage et donc tous ses exemplaires et les emprunts liés

30

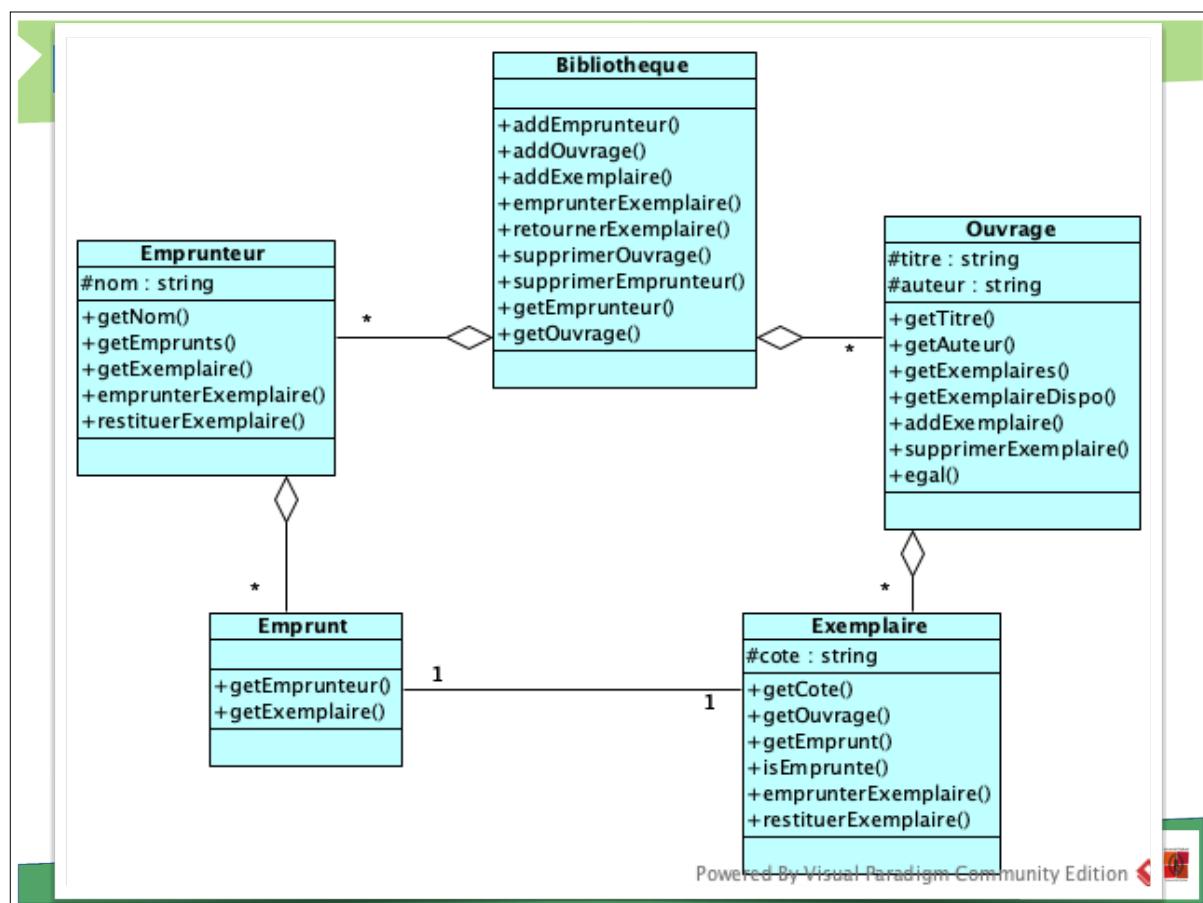


30

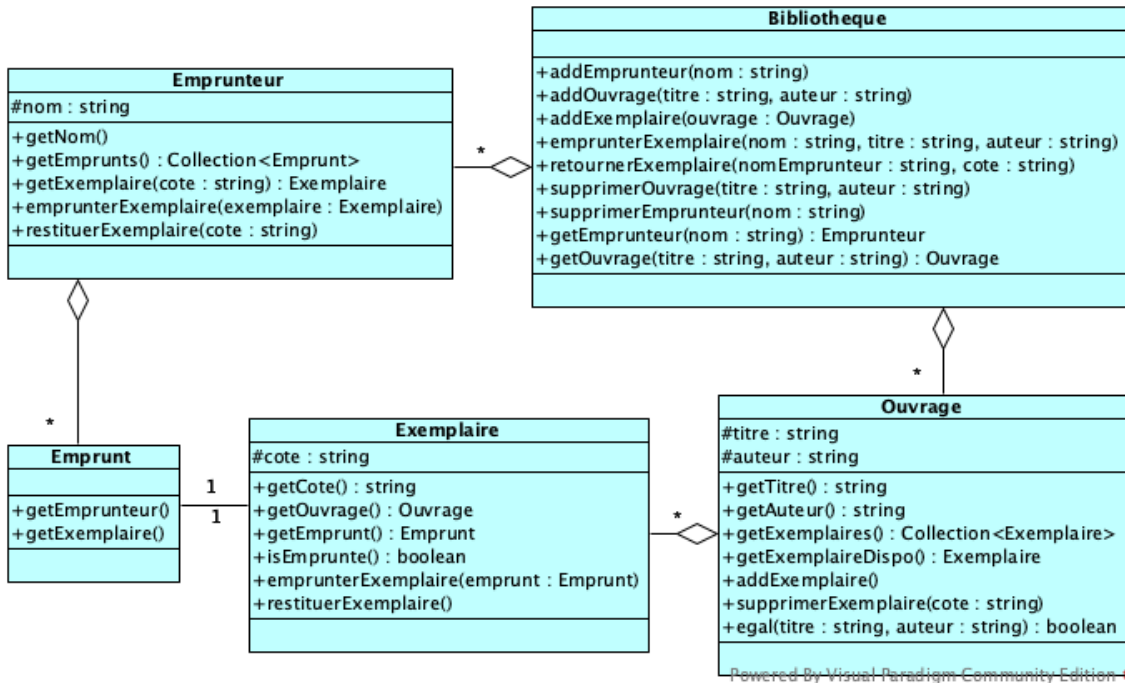


31

31



32



33

33

Diagrammes d'état

❑ Associés à *UNE* classe

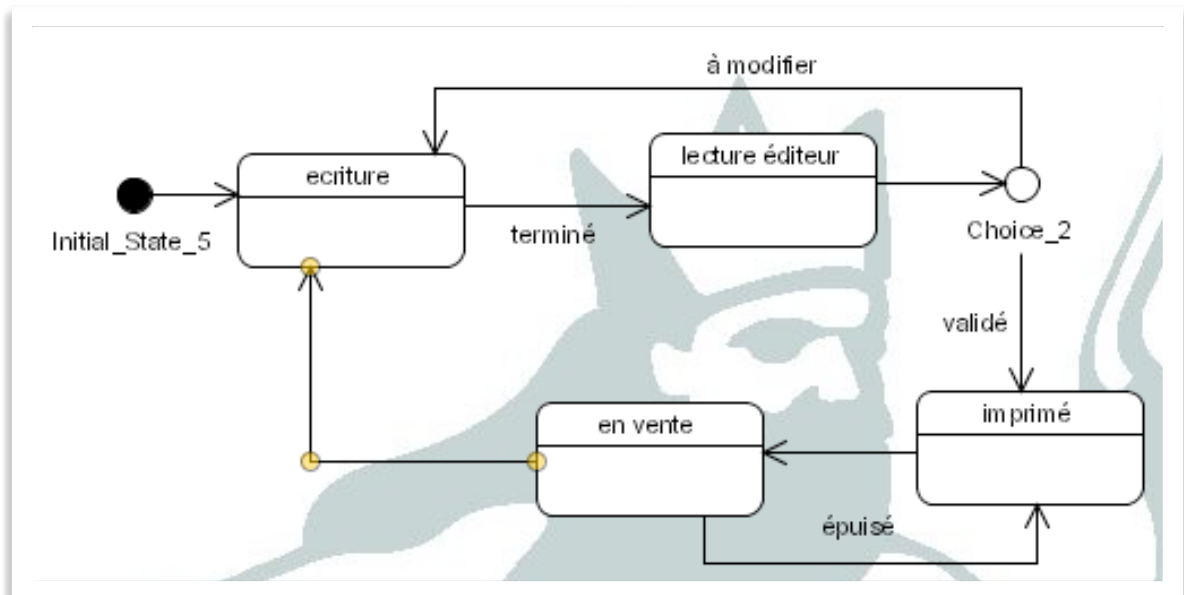
- ❖ **Etat** : valeurs des attributs de l'objet
- ❖ **Transition** : changement d'état déclenché par un événement (message)
- ❖ **Action** : associée à un événement (opération)
 - on event (il pleut) / action (ouvrir parapluie) : chaque fois que l'événement à lieu
 - entry/ action : à l'entrée dans l'état
 - exit/action : à la sortie de l'état
 - do/action : tant que l'objet est dans l'état

34

34

Diagrammes d'état

Exemple : rédaction d'un livre

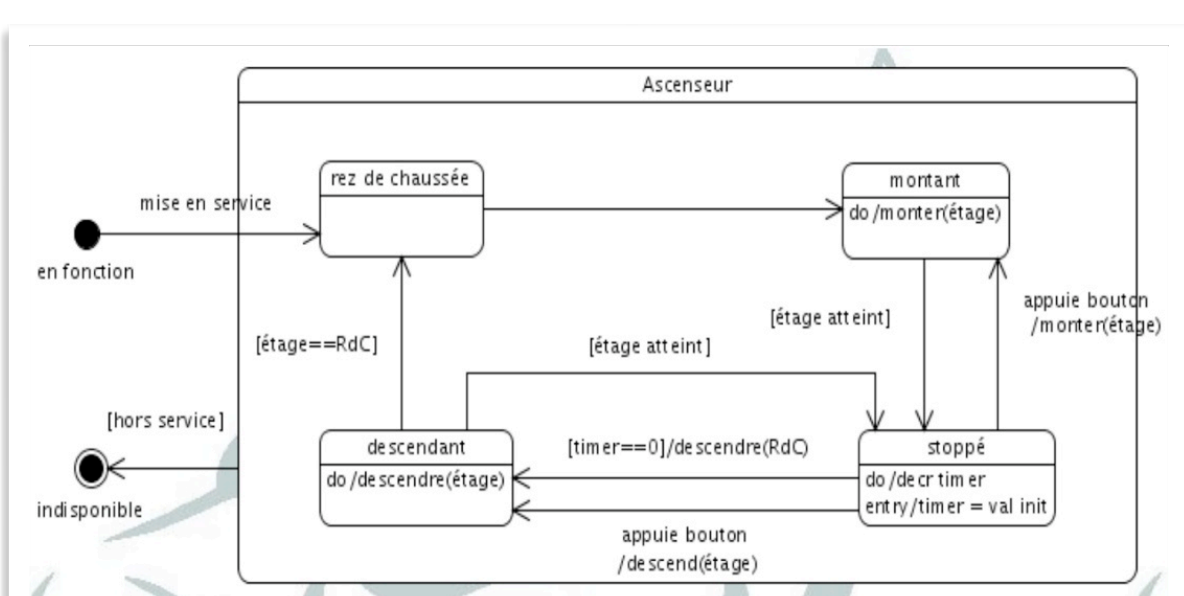


35

35

Diagrammes d'état

Exemple : ascenseur

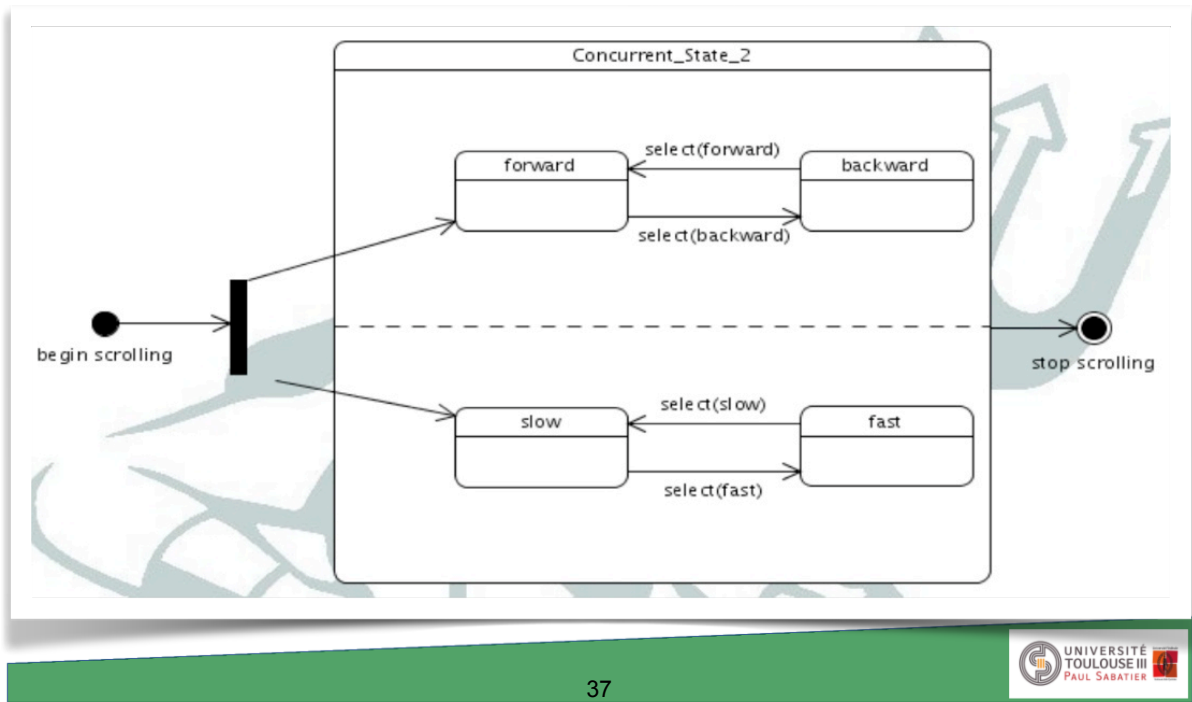


36

36

Diagrammes d'état

❑ Produits cartésiens d'état



37

37

Processus de création avec UML

❑ Point de vue fonctionnel

❖ Système conçu POUR les utilisateurs

- Déterminer les acteurs (primaires, secondaires, externes)
- Diagramme de Cas d'Utilisation
- Pour chaque UC :
 - Description textuelle
 - [Diagramme d'activité]
 - Déterminer quels sont les scénarios :
 - le nominal
 - les variantes (fin normale)
 - les échecs (fin anormale)

38

38

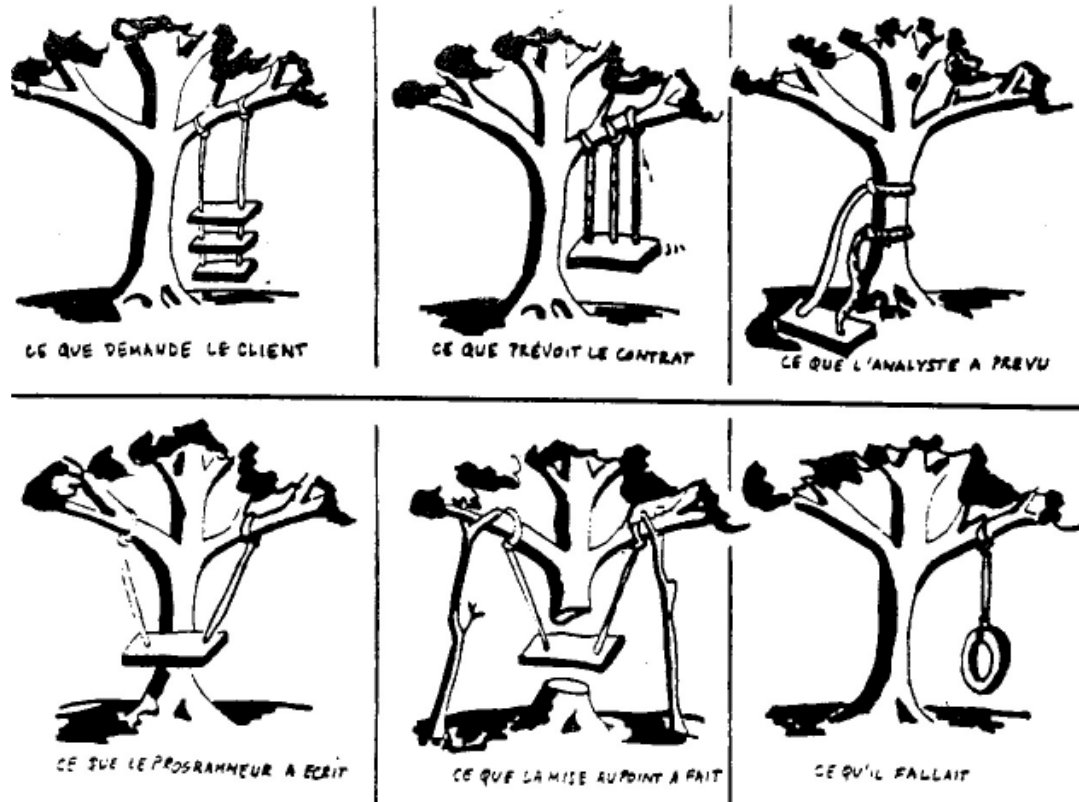
Processus de création avec UML

□ Point de vue dynamique ET statique

- ❖ Pour chaque UC :
- ❖ Pour chaque scénario :
 - Créer au fur et à mesure les classes avec attributs et méthodes
 - Diagramme de séquences détaillé
 - Diagramme de classe partiel
- ❖ Diagramme de classes
 - Complété à chaque scénario
- ❖ Eventuellement
 - Diagrammes d'état pour les classes qui le nécessitent
 - Diagrammes d'objets pour expliciter le D. de classes

39

39



40