ÉCOLE CENTRALELYON

CNAM Lyon – UE NFE 103
Méthodes Avancées d'Informatisation

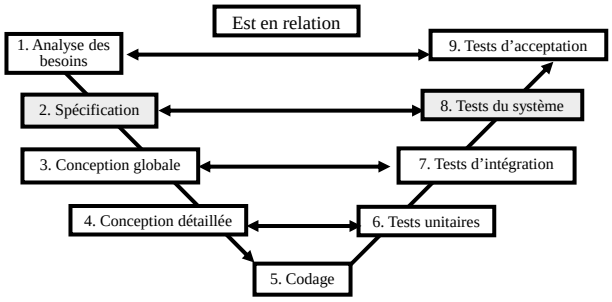
Spécifications avec UML

- **Spécifications avec UML**
- Conception d'architecture avec UML

Version du 5/10/2010

ÉCOLE CENTRALELYON

Cycle en V : Spécifications



ÉCOLE CENTRALELYON

Spécifications (principes)

- Traduire le Cahier des Charges dans un langage plus formalisé.
- Vérifier la cohérence des informations.
- Ce dossier est réalisé par l'équipe de développement

- Contenu :
  - Elaborer le dossier de spécifications
  - Rédiger le manuel provisoire d'utilisation et d'exploitation
  - Rédiger le dossier de validation

NFE 103 MAI Bertrand DAVID UML Concepts de base 3

ÉCOLE CENTRALELYON

Questionner

- Première étape critique pour l'équipe de développement

  L'étape doit répondre aux questions suivantes :

  - QUOI ? Identifier le besoin et produire les spécifications
  - POURQUOI ? Comprendre le contexte du problème
  - COMMENT ? Identifier des contraintes
  - QUAND ? Etablir le planning et comprendre les relations entre activités
  - OU ? Situer les flux d'information
  - QUI ? Identifier les parties concernées

NFE 103 MAI Bertrand DAVID UML Concepts de base 4

ÉCOLE CENTRALELYON

Analyser

- Borner le contexte du problème
- Identifier les éléments constitutifs, selon la méthode utilisée :
  - objets / entités
  - attributs / propriétés
  - fonctions / actions
  - liens / relations
- Construire un modèle
- Modélisation est une activité fondamentale du Génie Logiciel.
- La plupart de méthodes d'analyse - spécification consiste en construction de modèle.

NFE 103 MAI Bertrand DAVID UML Concepts de base 5

ÉCOLE CENTRALELYON

Modéliser

- modéliser quoi ?

  activités - données - dialogues

- modéliser comment ?

  formalismes - schémas - outils

  Objectif découvrir la structure et le comportement (signification, sémantique)

  Pour la structure : constitution d'un réseau de liens entre éléments identifiés (activités, données, utilisateurs, …)

NFE 103 MAI Bertrand DAVID UML Concepts de base 6

ÉCOLE CENTRALELYON

S'approprier

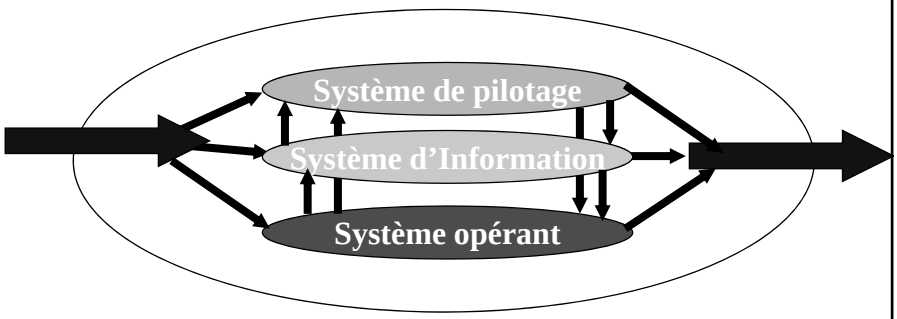
- **L'objectif principal est de comprendre pour pouvoir communiquer**
- **L'appropriation du problème par l'équipe de développement**

NFE 103 MAI Bertrand DAVID UML Concepts de base 7

ÉCOLE CENTRALELYON

Approche Système

- **Système de décision (pilotage)**
- **Système d'information**
- **Système opérant**



NFE 103 MAI Bertrand DAVID UML Concepts de base 8

Plan type du dossier de spécifications

Objectif : Réécrire le Cahier des Charges en le rendant plus complet et plus homogène.

1. Introduction
2. Objectifs et contraintes (de développement et d'exploitation)
3. Spécifications générales (délimiter le projet)
4. Spécifications détaillées
5. Spécifications de l'environnement

NFE 103 MAI Bertrand DAVID UML Concepts de base 9

Moyens d'expression formelle (plus ou moins) :

- **SADT**
- **DFD : Diagrammes de flux de données**
- **E-A : Modèle Entité Association**
- **UML**

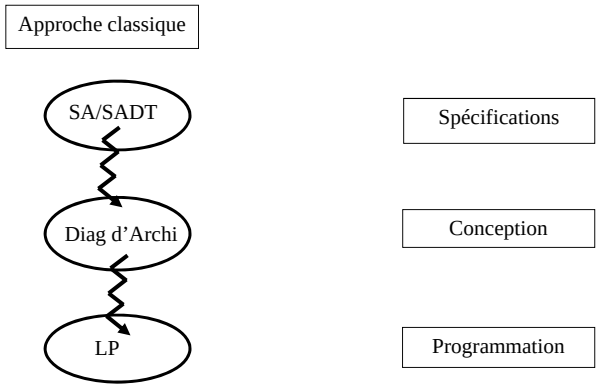
- **Notre choix : UML**

NFE 103 MAI Bertrand DAVID UML Concepts de base 10

ÉCOLE CENTRALELYON

Approches de développement (1/3)

- Approche classique

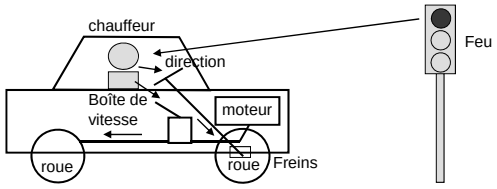


NFE 103 MAI Bertrand DAVID UML Concepts de base 11

ÉCOLE CENTRALELYON

Concept d'Objet

Le monde est composé d'entités qui « collaborent »



- L'approche objet consiste à résoudre un problème en termes d'objets qui collaborent.
- Ces objets sont des abstractions des objets réels

NFE 103 MAI Bertrand DAVID UML Concepts de base 12

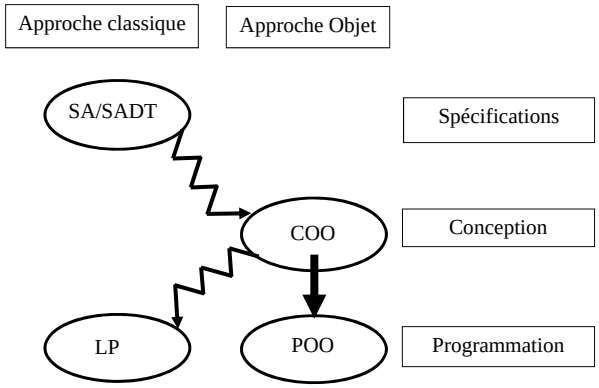
Penser objet

- Faire une analyse portant sur les « Etres » du domaine d'application (objets)
- Identifier à la fois leurs caractéristiques et leurs services
- Avoir une approche recherchant des modèles (représentants) donc classes
- Faire passer au second plan le fonctionnement global
- Identifier les liens statiques (relationnels) entre les objets (pour constituer un référentiel)
- Identifier des besoins d'utilisation de services proposés par d'autres classes

NFE 103 MAI Bertrand DAVID UML Concepts de base 13

Approches de développement (2/3)

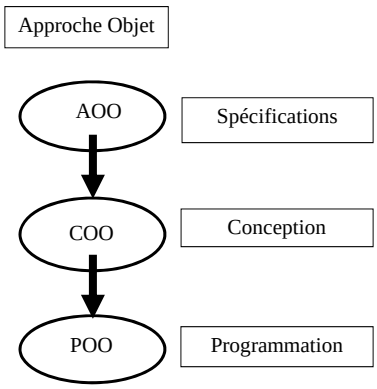
- Approches mixtes



NFE 103 MAI Bertrand DAVID UML Concepts de base 14

Approches de développement (3/3)

- Approche tout objet



NFE 103 MAI Bertrand DAVID UML Concepts de base 15

UML (Unified Modeling Language)

Un formalisme issu des méthodes :

- OMT - James Rumbaugh,
- Booch - Grady Booch,
- OOSE - Ivar Jacobson

Objectif global, proposer un support complet comportant :

- Processus et cycle de vie
- Méthodes
- Formalismes et outils

UML constitue le premier pas : formalisme unifié basé sur une présentation triaxiale d'un objet

- Les cas d'utilisation : le savoir-faire
- La description : classes, propriétés, associations
- La dynamique : états, événements

NFE 103 MAI Bertrand DAVID UML Concepts de base 16

UML (Unified Modeling Language)

- Normalisé en 1997 par l'OMG (Object Management Group) qui est aussi à l'origine de CORBA
- Version courante industriellement utilisée : 1.5
- Version 2.0 juste normalisée : bien plus complexe
- UML:
  - un langage graphique (supporté par un méta-modèle)
  - un ensemble de vues (=diagrammes) modélisant les aspects statiques et dynamiques
- Il ne propose pas UNE méthodologie d'analyse et de conception, mais peut en supporter plusieurs !

Méthodes : Processus Unifié (RUP – Rational Unified Process)

Model-Driven Architecture (MDA) – supportée par OMG

NFE 103 MAI Bertrand DAVID UML Concepts de base 17

UML et ses formalismes

- **Diagramme de classes**
- **Diagramme d'objets**
- **Diagramme de cas d'utilisation**
- **Diagramme d'états**
- **Diagramme de séquences**
- **Diagramme d'activités**
- **Diagramme de collaboration**

- **Diagramme de composants**
- **Diagramme de déploiement**

NFE 103 MAI Bertrand DAVID UML Concepts de base 18

Objet : définition

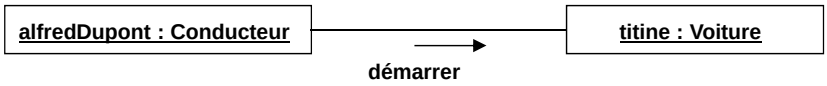
- Un objet est une entité qui possède :
  - une **identité** : nom qui permet de distinguer un objet d'un autre objet
  - un **état** : ensemble de valeurs associées à des propriétés et qui caractérisent l'objet à un instant t
  - un **comportement** : ensemble de services ou d'opérations que peut rendre un objet ou qui modifient son état

titine : Voiture
marque = "Ferrari" couleur = rouge proprio = "Alfred" vitesse = 250
demarrer () tourner () accelerer () ralentir () arreter ()

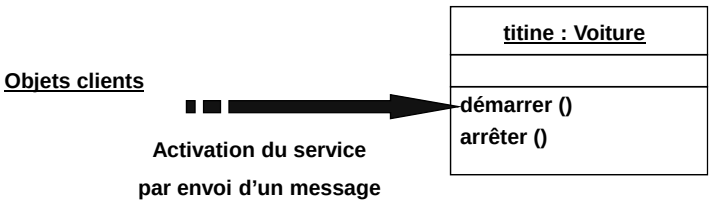
NFE 103 MAI Bertrand DAVID UML Concepts de base 19

Communication entre objets

- Les objets communiquent entre eux par l'envoi de messages
- Un message entraîne l'activation d'un service de l'objet



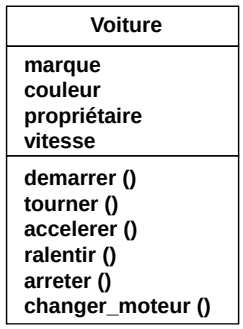
- message =
  - un sélecteur : identifie quel est le service de l'objet activé
  - des arguments : paramètres effectifs communiqués au service



NFE 103 MAI Bertrand DAVID UML Concepts de base 20

Notion de classe

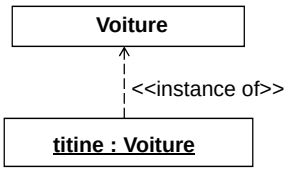
- Classe :
  - **type de donnée abstrait**
  - il regroupe des propriétés communes à des objets :
    - attributs (≈ variables)
    - méthodes (≈ fonctions)



Classe documentée

Classe non-documentée

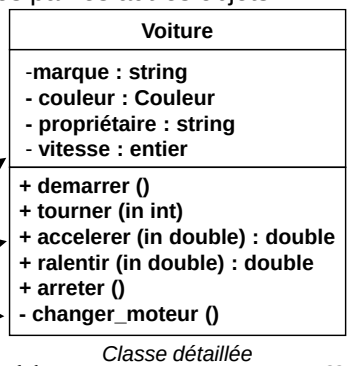
- Instances d'une classe :
  - ce sont les objets dérivés de cette classe



- on peut faire les équivalences :
  - ➔ classe : concept ≈ type
  - ➔ instance : représentant du concept ≈ variable

Concepts fondamentaux (1/3)

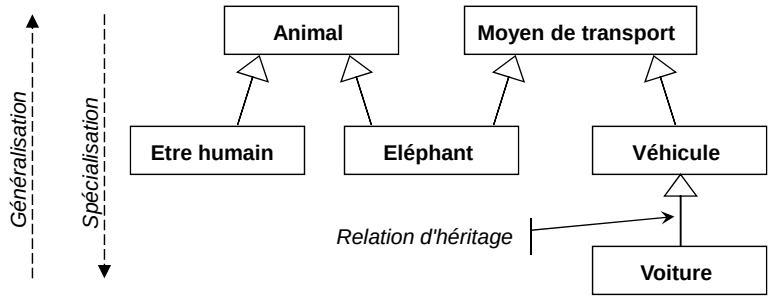
- Encapsulation :
  - regrouper en une seule entité (la classe) les données et les traitements sur ces données
  - masquer les détails de l'implémentation en définissant une interface
  - l'interface définit les services accessibles par les autres objets : méthodes publiques de l'objet
  - l'encapsulation facilite l'évolution car on peut changer l'implémentation de l'objet sans changer l'interface
  - les attributs et les méthodes privées sont protégées (accessibles seulement par les méthodes de la classe)



Classe détaillée

Concepts fondamentaux (2/3)

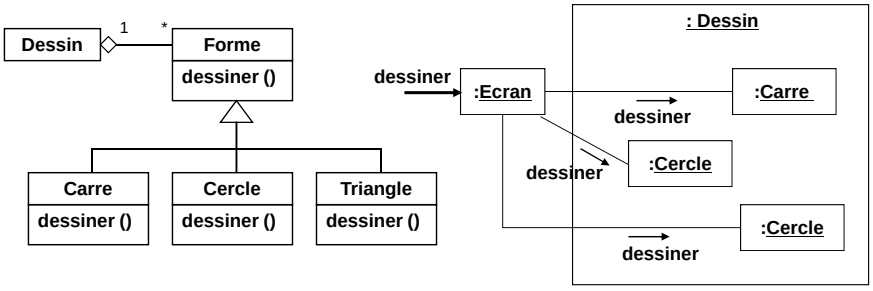
- Héritage (spécialisation et généralisation) :
  - transmission des propriétés d'une classe vers une sous-classe
  - la spécialisation permet d'ajouter des caractéristiques ou d'en adapter certaines
  - la généralisation permet de factoriser des classes en regroupant des propriétés qui leur sont communes
  - l'héritage évite la duplication et favorise la réutilisation



NFE 103 MAI Bertrand DAVID UML Concepts de base 23

Concepts fondamentaux (3/3)

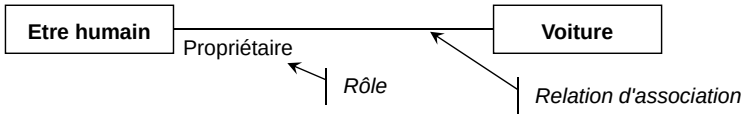
- Polymorphisme :
  - c'est un mécanisme qui permet à un objet client d'utiliser les services d'une interface « générique », sans savoir quel est l'objet qui va véritablement répondre à la requête
  - permet de créer des superclasses qui sont spécialisées par la suite



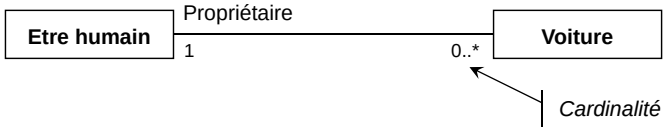
NFE 103 MAI Bertrand DAVID UML Concepts de base 24

Relations entre objets (1/2)

- Association :
  - relation signifiant que les instances de classes ont certaines liaisons entre elles (lien sémantique)



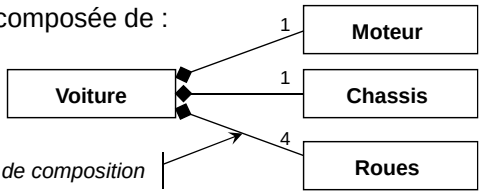
  - la cardinalité de la relation exprime le nombre d'instances pouvant être associées ; exemple :
    - un "être humain" peut ne pas avoir de voiture ou être propriétaire d'une ou plusieurs "voitures" (cardinalité 0..*)
    - une "voiture" est liée à un propriétaire et un seul (cardinalité 1)



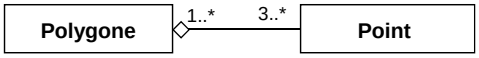
NFE 103 MAI Bertrand DAVID UML Concepts de base 25

Relations entre objets (2/2)

- Composition :
  - relation entre classes signifiant que les instances d'une classe sont les composants d'une autre classe
  - la composition permet d'assembler des objets pour en créer de plus complexes
  - exemple : une voiture est composée de :
    - 1 moteur,
    - 1 châssis
    - et 4 roues



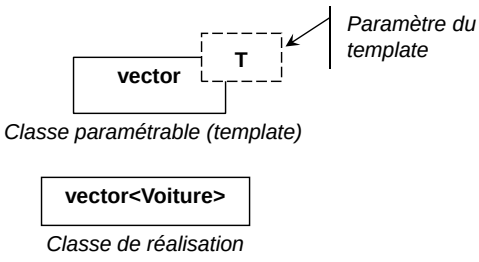
- Agrégation :
  - composition faible : les objets agrégés ont une durée de vie indépendante de l'agrégat
  - exemple :



NFE 103 MAI Bertrand DAVID UML Concepts de base 26

Autres concepts

- Classe abstraite :
  - classe très générale, non instanciable
  - utilisé pour la factorisation de propriétés dans des arbres d'héritage
- Généricité :
  - utilisation de classes paramétrables (templates en anglais, parfois appelées patrons) : classes génériques pouvant générer des familles de classes de réalisation concrètes (c-à-d instanciables)
  - exemple : classe paramétrable "vector <T>" permet de générer des :
    - vecteurs d'entier
    - vecteurs de réels
    - vecteurs de Voiture
    - etc...



Classe paramétrable (template)

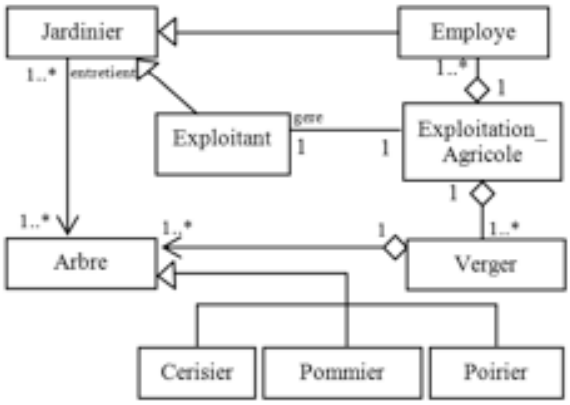
vector<Voiture>

Classe de réalisation

NFE 103 MAI Bertrand DAVID UML Concepts de base 27

Modélisation objet : un premier exemple

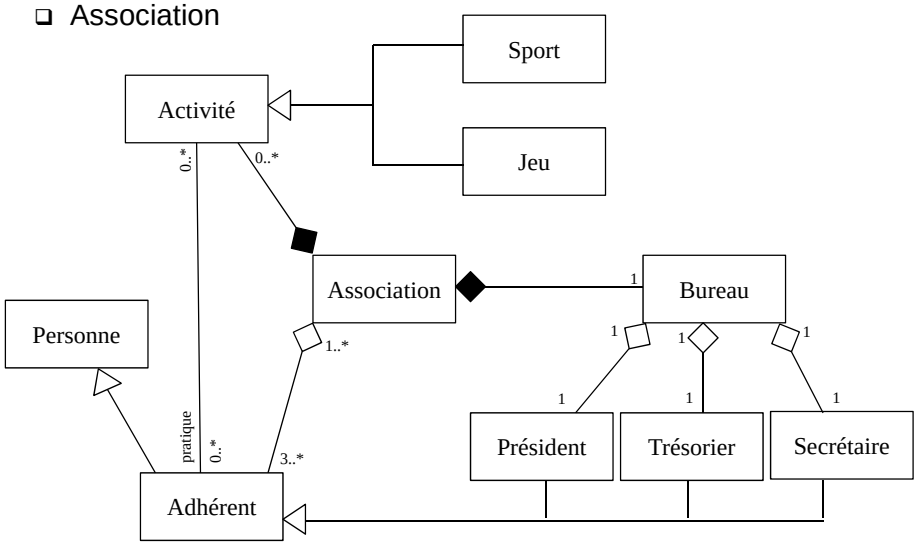
- Exploitation agricole



NFE 103 MAI Bertrand DAVID UML Concepts de base 28

Modélisation objet : un second exemple

- Association



NFE 103 MAI Bertrand DAVID UML Concepts de base 29

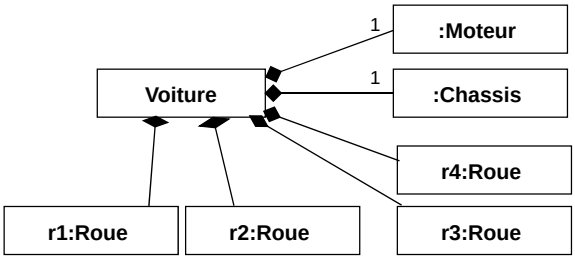
UML et ses formalismes

- **Diagramme de classes**
- **Diagramme d'objets**
- **Diagramme de cas d'utilisation**
- **Diagramme d'états**
- **Diagramme de séquences**
- **Diagramme d'activités**
- **Diagramme de collaboration**
- **Diagramme de composants**
- **Diagramme de déploiement**

NFE 103 MAI Bertrand DAVID UML Concepts de base 30

Diagramme d'objets

Buts :
1. Décrire schématiquement les relations entre objets
2. Support pour la recherche de diagramme de classes et les diagrammes de collaboration



NFE 103 MAI Bertrand DAVID UML Concepts de base 31

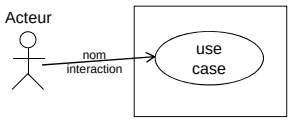
UML et ses formalismes

- **Diagramme de classes**
- **Diagramme d'objets**
- **Diagramme de cas d'utilisation**
- **Diagramme d'états**
- **Diagramme de séquences**
- **Diagramme d'activités**
- **Diagramme de collaboration**
- **Diagramme de composants**
- **Diagramme de déploiement**

NFE 103 MAI Bertrand DAVID UML Concepts de base 32

Les cas d'utilisation

But : **Spécifications fonctionnelles du système**



- un cas = un service (fonctionnalité)
- Acteur = utilisateur du service
- Il y a des acteurs principaux et secondaires

Description textuelle semi-structurée

Titre
But
Résumé
Acteurs
Date + version
Pré conditions
Enchaînements
Exceptions
Post conditions
{IHM}

Les cas d'utilisation sont de très bons moyens de communication entre le client et le développeur

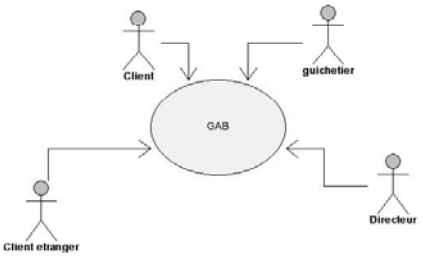
Acteurs dans les cas d'utilisation

Les acteurs peuvent être de trois types :

- humains, des utilisateurs du logiciel,
- logiciels qui manipulent le systèmes à l'aide d'une API,
- matériels, robots et automates qui exploitent les données du système ou qui sont pilotés par le système.

Typologie des acteurs :

- utilisateurs du système
- administrateurs du système



Diagrammes de cas d'utilisation

Les cas utilisation expriment

- les principales tâches de chaque acteur,
- les modifications des données du système,
- les cas d'anomalies

Pour voir de façon simple :

- les différents acteurs
- comment est délimité le système
- les fonctionnalités demandées au système
- les rôles des différents acteurs vis-à-vis du système

Descriptions :

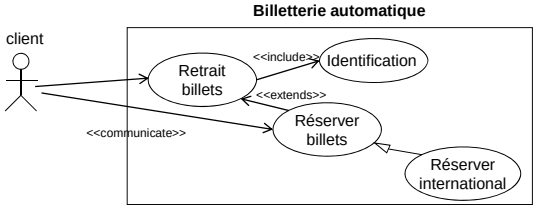
- Textuelle
- diagramme de séquence

NFE 103 MAI Bertrand DAVID UML Concepts de base 35

Les cas d'utilisation

Trois types de relations

a - utilisation
b – extension **« extends »** (Pour factoriser et réutiliser).
c - spécialisation / héritage **« uses »** (Pour hériter et affiner)



Attention au piège de la décomposition trop fine : fonctionnelle

NFE 103 MAI Bertrand DAVID UML Concepts de base 36

ÉCOLE CENTRALELYON

Les diagrammes dynamiques

- **Diagramme de séquences**
- **Diagramme d'états-transitions**
- **Diagramme d'activités**
- **Diagramme de collaboration**

ÉCOLE CENTRALELYON

UML et ses formalismes

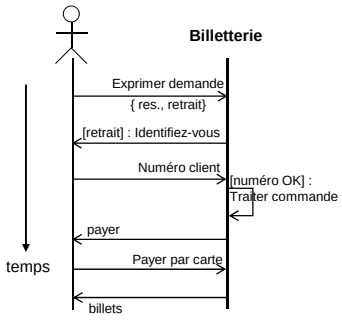
- **Diagramme de classes**
- **Diagramme d'objets**
- **Diagramme de cas d'utilisation**
- **Diagramme d'états**
- **Diagramme de séquences**
- **Diagramme d'activités**
- **Diagramme de collaboration**
- **Diagramme de composants**
- **Diagramme de déploiement**

ÉCOLE CENTRALELYON

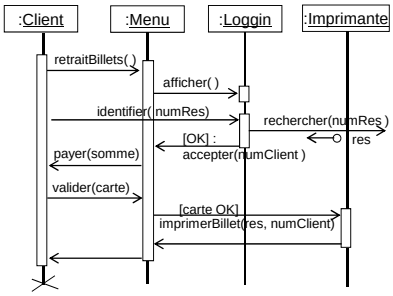
Les diagrammes de séquence

Buts :

1. décrire les cas d'utilisation (scénarios)
2. décrire les interactions entre les objets



(1)

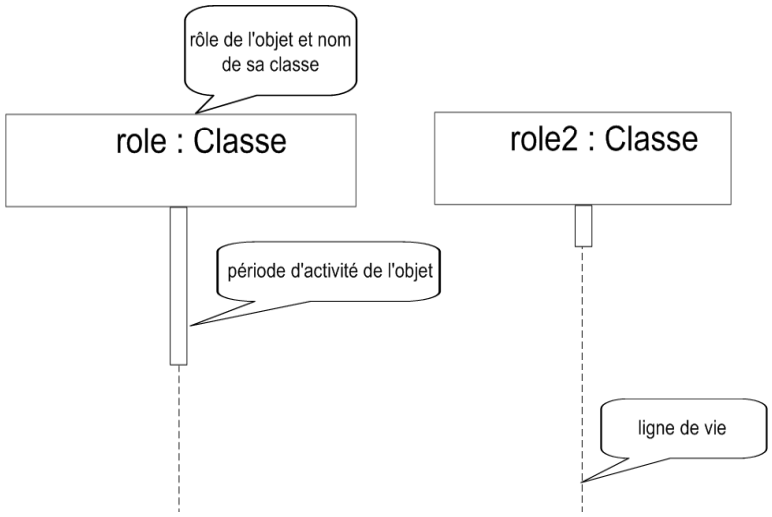


(2)

NFE 103 MAI Bertrand DAVID UML Concepts de base 39

ÉCOLE CENTRALELYON

Objets et leurs lignes de vie



NFE 103 MAI Bertrand DAVID UML Concepts de base 40

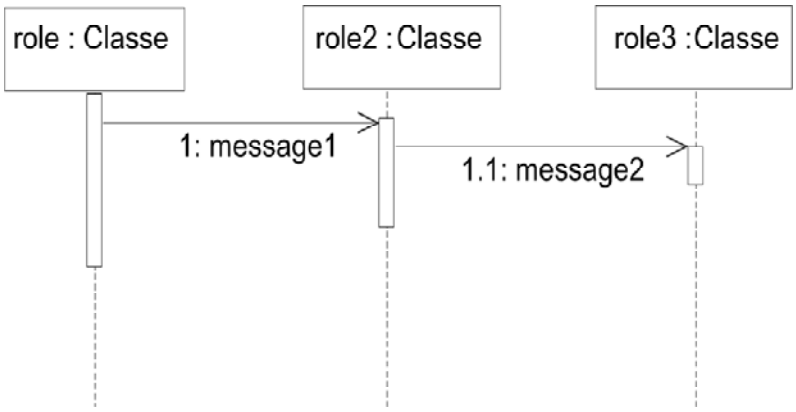
ÉCOLE CENTRALELYON

Envoi d'un message

NFE 103 MAI Bertrand DAVID UML Concepts de base 41

ÉCOLE CENTRALELYON

Numérotation séquentielle et composée de messages

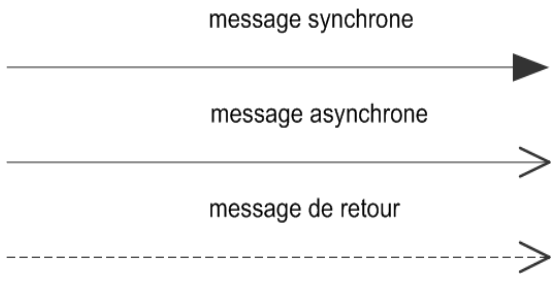


NFE 103 MAI Bertrand DAVID UML Concepts de base 42

Paramètres transmis avec le message

NFE 103 MAI Bertrand DAVID UML Concepts de base 43

Différents types de messages



NFE 103 MAI Bertrand DAVID UML Concepts de base 44

Un objet peut envoyer un message à lui-même

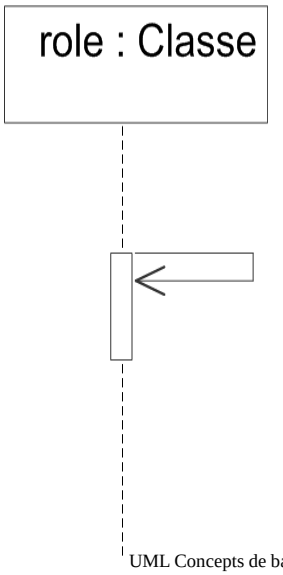
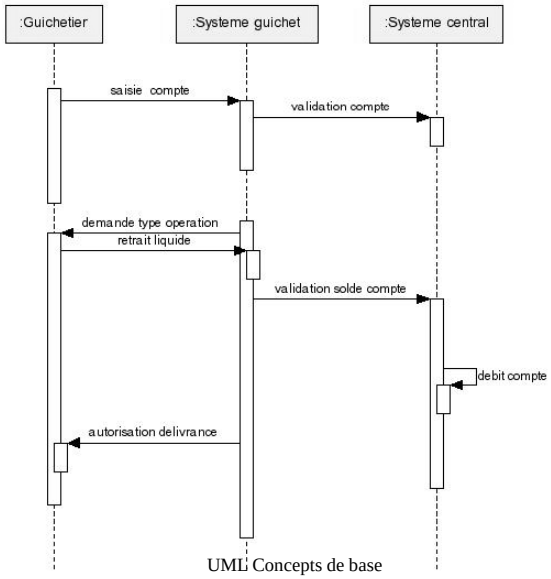
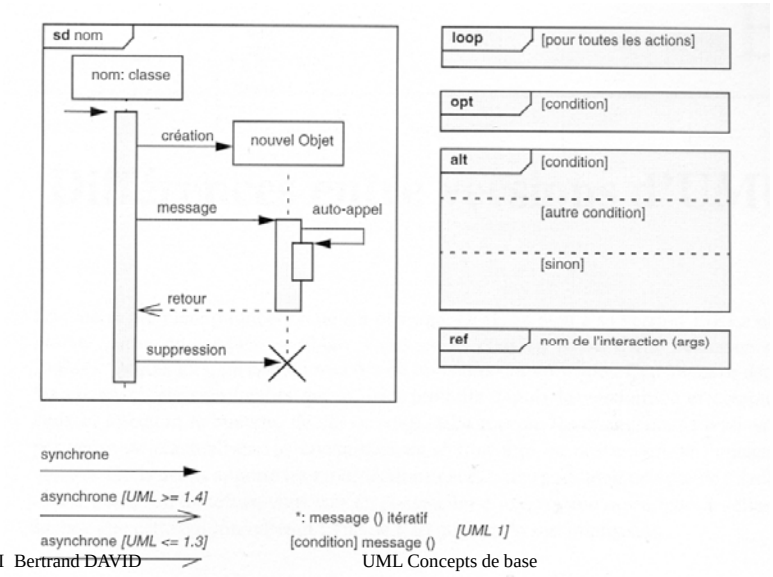


Diagramme de séquence



Diagrammes de séquence
Les cadres d'interaction



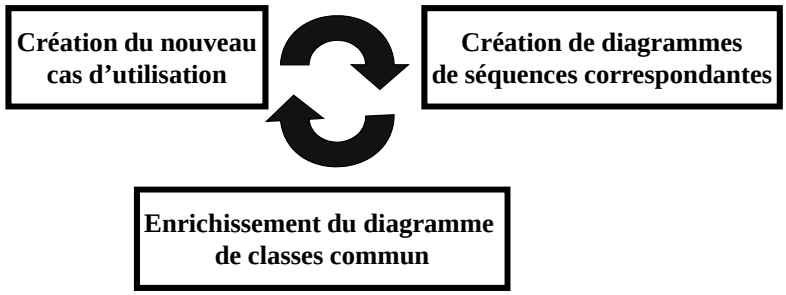
Scénario : Diagramme de séquence

Un **scénario** est une série d'événements ordonnés dans le temps, simulant une exécution particulière du système.

Les scénarios permettent d'expérimenter les exécutions du système, ils sont donc très utiles pour les phases de tests et de maintenance.

Démarche d'élaboration de spécifications

- Travail itératif entre :
  - Identification des cas d'utilisation
  - Elaboration de diagrammes de séquences
  - Enrichissement du diagramme de classes commun



NFE 103 MAI Bertrand DAVID UML Concepts de base 49

UML et ses formalismes

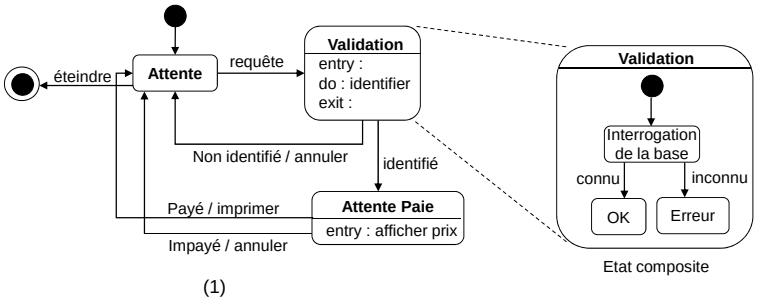
- **Diagramme de classes**
- **Diagramme d'objets**
- **Diagramme de cas d'utilisation**
- **Diagramme d'états-transitions**
- **Diagramme de séquences**
- **Diagramme d'activités**
- **Diagramme de collaboration**
- **Diagramme de composants**
- **Diagramme de déploiement**

NFE 103 MAI Bertrand DAVID UML Concepts de base 50

Diagrammes d'états / transitions

Buts :

1. Illustrer les cas d'utilisation
2. Décrire en détail le comportement d'une classe



(1)

NFE 103 MAI Bertrand DAVID UML Concepts de base 51

Le modèle dynamique

Pourquoi un modèle dynamique ?

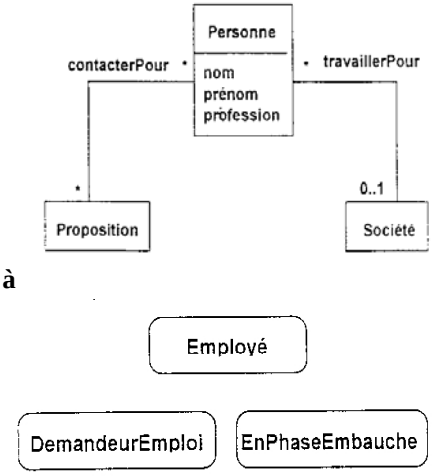
- **pour décrire les relations temporelles et événementielles,**
- **pour exprimer les états en fonction de modifications internes et d'options choisies par les utilisateurs,**
- **pour indiquer les actions possibles dans un contexte donné pour décrire les actions des systèmes extérieurs sur les objets du système étudié ainsi que les réactions de ces objets**

NFE 103 MAI Bertrand DAVID UML Concepts de base 52

ÉCOLE CENTRALELYON

Concepts et notations de base (1)
La notion d'état

- **L'état d'un objet est défini à la fois par les valeurs de ses variables d'instance et par les valeurs de ses liens avec d'autres objets. L'état d'un objet correspond à une durée, un intervalle de temps quantifiable à l'échelle du système.**



NFE 103 MAI Bertrand DAVID UML Concepts de base 53

ÉCOLE CENTRALELYON

Concepts et notations de base (2)
La notion d'événement

- **Un événement** est un stimulus pouvant transporter des informations (sous forme de paramètres), il se produit à un instant donné : un événement n'a pas de durée contrairement à un état.
- Un événement peut être émis par un objet du système ou par un objet externe au système, cela correspond à l'appel d'une méthode. Un événement peut également provenir de l'interface du système (clic souris, sélection d'un item dans un menu,…).
- La réponse d'un objet à un événement dépend de l'état dans lequel se trouve l'objet qui le reçoit.

NFE 103 MAI Bertrand DAVID UML Concepts de base 54

Concepts et notations de base (3)
La notion de message

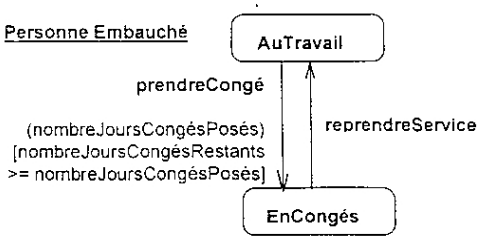
- **Un message est un événement particulier, issu de l'interaction entre deux objets, un objet appelle une méthode d'un autre objet.**
- **Tout message est un événement impliqué dans l'interaction entre deux objets.**
- **Tout événement n'est pas un message, car il n'est pas forcément émis par un objet.**

Diagramme d'états

Un **diagramme d'états** est un graphe constitué de noeuds représentant des états ainsi que des flèches représentant des transitions portant des paramètres et des noms d'événements.

Un diagramme d'états est propre à une classe donnée :
Un diagramme d'états ne permet pas de représenter les relations d'interaction entre les objets intervenant dans un système, mais le cycle de vie des objets d'une classe.

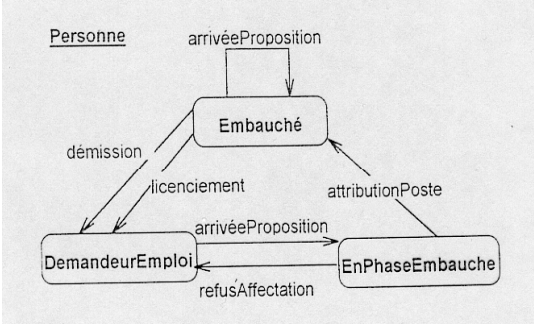
Un état peut avoir des sous-états



ÉCOLE CENTRALELYON

La notion de transition

Une transition est le changement d'état d'un objet causé par un événement. Les transitions sont représentées par des flèches portant le nom et les paramètres des événements associés.



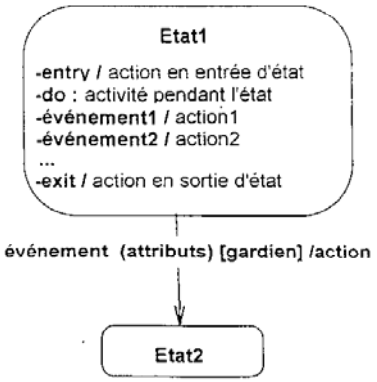
NFE 103 MAI Bertrand DAVID UML Concepts de base 57

ÉCOLE CENTRALELYON

Caractérisation d'une transition (1)

Les attributs **qui correspondent à des informations ou des paramètres portés par des événements.**
Les gardiens **ou conditions qui sont des fonctions booléennes.**
Deux types d'opérations peuvent être impliquées dans un diagramme dynamique : les activités et les actions

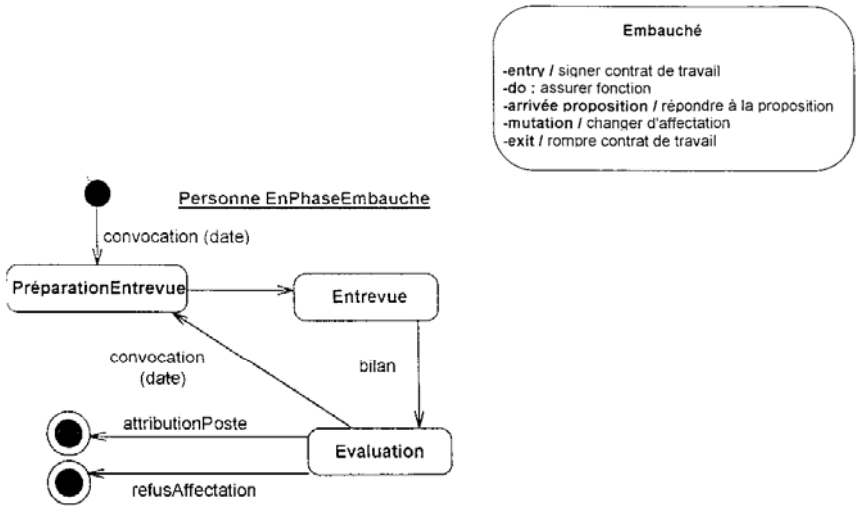
- **Une** activité **est une opération continue dans le temps, elle prend un certain temps pour se réaliser. Elle est forcément associée à un état.**
- **Une** action **est une opérations instantanée, elle est réalisée de façon immédiate, et peut être associée aussi bien à l'état d'un objet qu'à une transition. Elle peut intervenir soit en entrée d'état (préfixe entre/), soit en sortie d'état (préfixe exit/), soit en réponse à un événement déclencheur (préfixe NomEvenement/), soit enfin en cours d'une transition.**



NFE 103 MAI Bertrand DAVID UML Concepts de base 58

ÉCOLE CENTRALELYON

Caractérisation d'une transition (2)



ÉCOLE CENTRALELYON

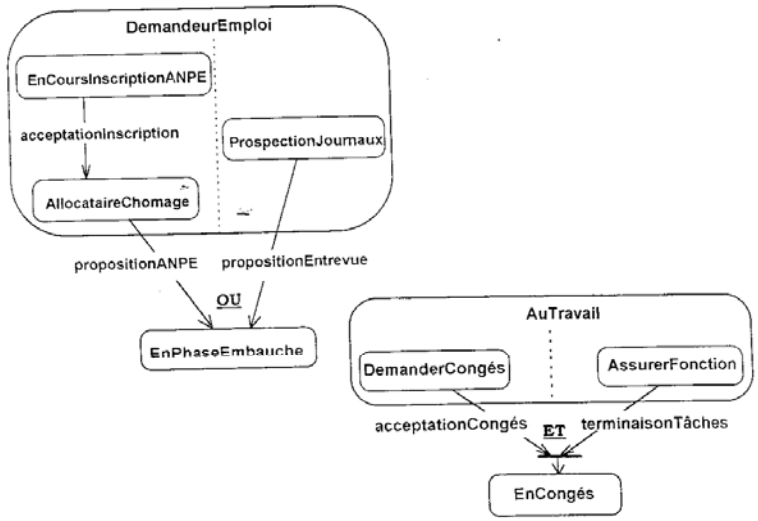
Concurrence et synchronisation d'états

Plusieurs sous-diagrammes d'états peuvent intervenir en parallèle dans un même état. Ils sont alors :

- **soit <u>en concurrence</u> : le premier sous-diagramme permettant de faire la transition de l'état englobant vers un autre état interrompt les autres sous-digrammes et fait quitter l'état englobant.**
- **soit <u>en synchronisation</u> : la transition de l'état englobant vers un autre état n'est effectuée que lorsque tous les sous-diagrammes le permettent, aucun sous-diagramme ne pouvant être interrompu.**

ÉCOLE CENTRALELYON

Concurrence et synchronisation d'états



ÉCOLE CENTRALELYON

UML et ses formalismes

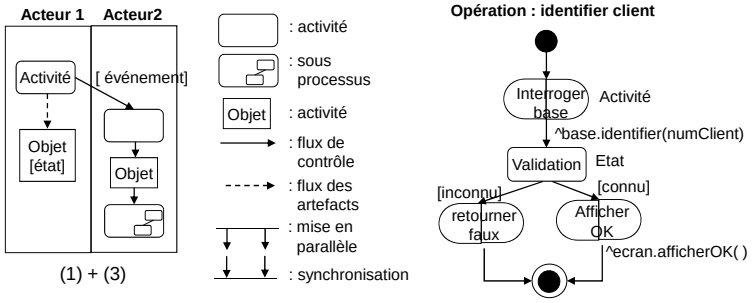
- **Diagramme de classes**
- **Diagramme d'objets**
- **Diagramme de cas d'utilisation**
- **Diagramme d'états-transitions**
- **Diagramme de séquences**
- **Diagramme d'activités**
- **Diagramme de collaboration**
- **Diagramme de composants**
- **Diagramme de déploiement**

Diagrammes d'activités

Buts :

1. Décrire le comportement générique d'un use case
2. Décrire en détail le comportement d'une opération
3. Modéliser les processus métiers

Dual des diagrammes d'états / transitions



UML et ses formalismes

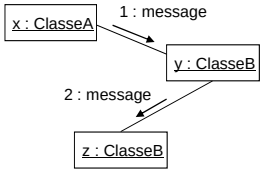
- **Diagramme de classes**
- **Diagramme d'objets**
- **Diagramme de cas d'utilisation**
- **Diagramme d'états**
- **Diagramme de séquences**
- **Diagramme d'activités**
- **Diagramme de collaboration**
- **Diagramme de composants**
- **Diagramme de déploiement**

Diagrammes de collaboration

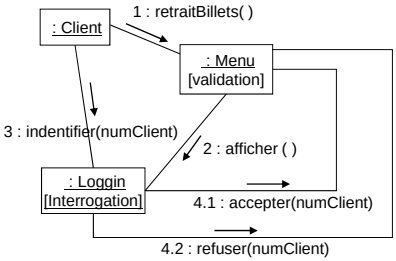
Buts :

1. Décrire les interactions entre objets
2. Aider à élaborer le diagramme d'architecture

- Diagramme partiel : exprime la dynamique
- Diagramme complet : exprime la vision statique de la dynamique



(1)



(1) + (3)

Diagramme de collaboration entre objets

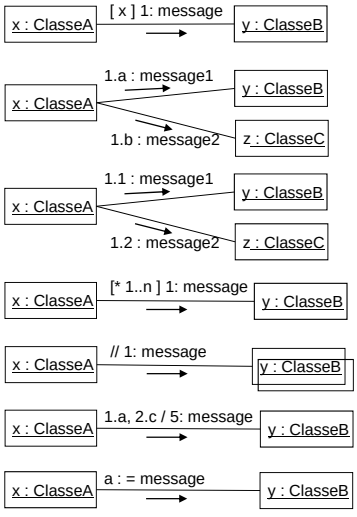
Un **diagramme de collaboration** entre objets vise à représenter du point de vue statique et dynamique les objets impliqués dans la mise en place d'une fonction applicative.

L'objectif est de construire un modèle expliquant la coopération entre les objets utilisés pour la réalisation d'une fonctionnalité.

Deux notions fondamentales :

- **Le contexte est une vue statique partielle des objets qui collaborent pour réaliser une fonction.**
- **Les interactions entre objets décrites dans un diagramme de collaboration sont des séquences de messages échangés par les objets dans le cadre de la réalisation d'une opération.**

Diagrammes de collaboration



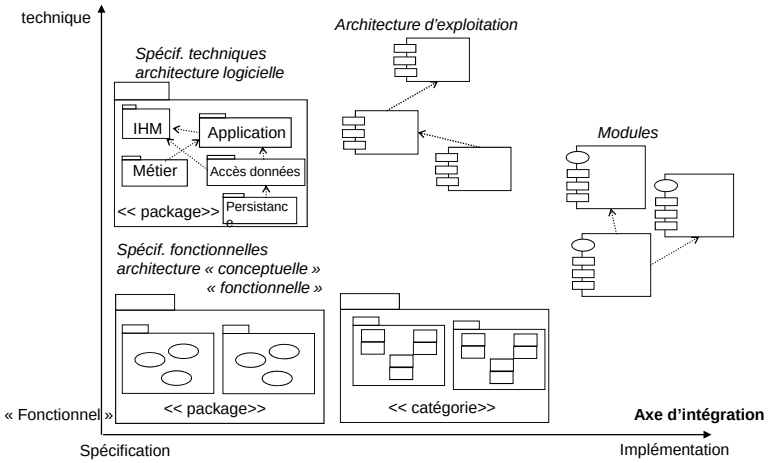
NFE 103 MAI Bertrand DAVID UML Concepts de base 67

UML et ses formalismes

- **Diagramme de classes**
- **Diagramme d'objets**
- **Diagramme de cas d'utilisation**
- **Diagramme d'états**
- **Diagramme de séquences**
- **Diagramme d'activités**
- **Diagramme de collaboration**
- **Diagramme de composants**
- **Diagramme de déploiement**

NFE 103 MAI Bertrand DAVID UML Concepts de base 68

Diagrammes de composants



NFE 103 MAI Bertrand DAVID UML Concepts de base 69

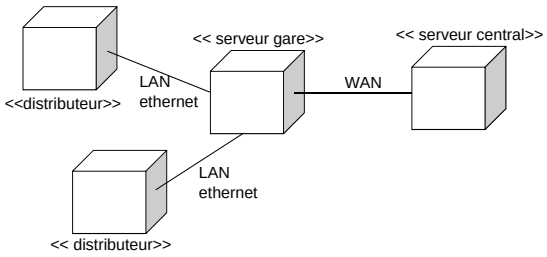
UML et ses formalismes

- **Diagramme de classes**
- **Diagramme d'objets**
- **Diagramme de cas d'utilisation**
- **Diagramme d'états**
- **Diagramme de séquences**
- **Diagramme d'activités**
- **Diagramme de collaboration**
- **Diagramme de composants**
- **Diagramme de déploiement**

NFE 103 MAI Bertrand DAVID UML Concepts de base 70

Diagramme de déploiement

But :
Décrire la projection sur une architecture matérielle



NFE 103 MAI Bertrand DAVID UML Concepts de base 71

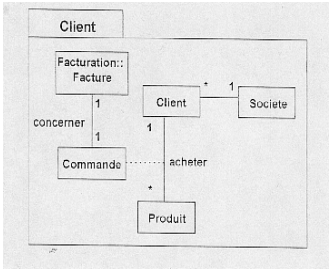
Les packages

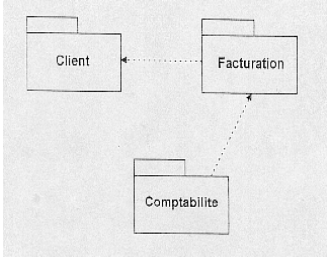
Un package regroupe en ensemble d'entités qui correspondent à une fonctionnalité bien définie.

Le package est un espace de nommage.
nomDuPackage :: NomDeLaClasse

Deux présentations :
- **explicite montrant le contenu du package**
- **limitée à la vision globale (sans montrer l'intérieur)**

Les interactions entre packages :
- **mécanisme de dépendance - toute modification du package source entraîne des modifications du package pointé.**





NFE 103 MAI Bertrand DAVID UML Concepts de base 72

ÉCOLE CENTRALELYON

CNAM Lyon – UE NFE 103
Méthodes Avancées d'Informatisation

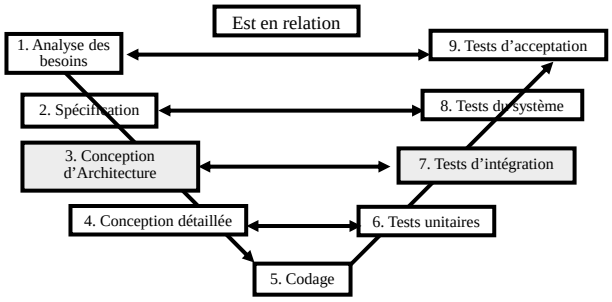
UML en conception

- Spécification avec UML
- **Conception d'architecture avec UML**

Version du 13/10/2009

ÉCOLE CENTRALELYON

Cycle en V - Conception d'Architecture : UML en phase de Conception



ÉCOLE CENTRALELYON

Conception d'Architecture

Objectif: Définir un découpage structurel de l'application pour permettre le travail coordonné et parallèle.

- **découper l'application pour répartir le travail de conception**
- **appliquer une méthode de conception pour définir les modules**
- **rédiger le dossier de conception d'architecture**
- **compléter le plan de tests de validation**

NFE 103 MAI Bertrand DAVID UML Concepts de base 75

ÉCOLE CENTRALELYON

UML en phase de conception

Objectif : passer du *quoi* au *comment*

Deux niveaux :

- Conception du système (conception d'architecture)
- Conception des objets (conception détaillée)

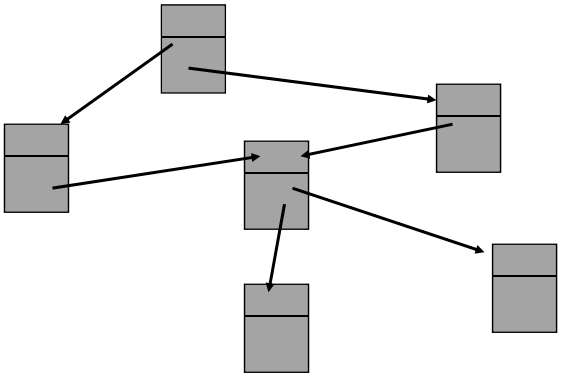
Deux approches :

- Architecture prédéfinie (framework)
- Architecture à concevoir

NFE 103 MAI Bertrand DAVID UML Concepts de base 76

Architectures objet

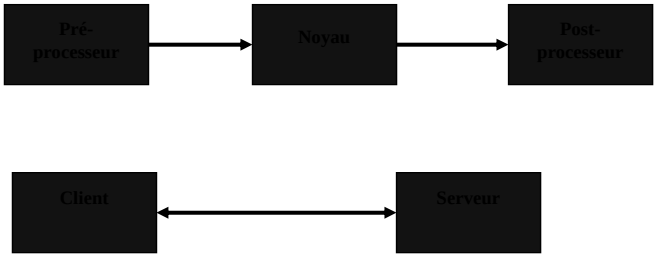
- **Les objets proposent et utilisent des « services »**
- **Le fonctionnement découle de l'enchaînement d'invocation des services**



NFE 103 MAI Bertrand DAVID UML Concepts de base 77

Architectures prédéfinies (frameworks)

- **Architecture pré-processeur - noyau - post-processeur**
- **Architecture Client-Serveur**

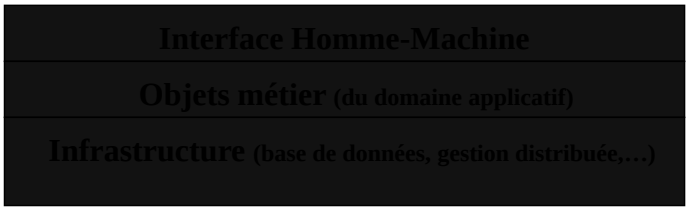


NFE 103 MAI Bertrand DAVID UML Concepts de base 78

La conception du système :

- **Décomposition en sous-systèmes**
- **Détermination des priorités**
- **Gestion des données**

L'architecture prédéfinie d'un système est organisée en trois couches représentant trois sous-systèmes de " haut niveau " :



NFE 103 MAI Bertrand DAVID UML Concepts de base 79

Exemple : GAB - Guichet Automatique de Banque (1/3)

- Le client pourra accéder grâce au GAB, à un compte courant et un compte épargne. Les transactions autorisées sont les dépôts et les retraits en liquide ; les retraits seront toujours des multiples de 20 € ; avec un maximum de 300 € par jour.
- L'identification des clients doit aller au delà de la simple possession de la carte d'accès au GAB. Le système vérifiera la validité d'accès physique, et de l'identification du client, de la validité des comptes accessibles à travers le GAB, et que le client n'a pas fait de déclaration signalant une mise en danger de la sécurité du système (perte ou vol dans le cas d'une carte d'accès).

NFE 103 MAI Bertrand DAVID UML Concepts de base 80

ÉCOLE CENTRALELYON

GAB (2/3)

- Le GAB doit s'assurer que le compte du client contient suffisamment de fonds pour couvrir la demande de retrait, et doit communiquer à l'ordinateur central les débits (retraits) et crédits (dépôts).
- Une transaction est un dépôt ou un retrait ; le client reçoit un reçu pour chaque transaction. Le système doit garantir la complétude des transactions : si une transaction en cours ne peut pas être terminée, le GAB affichera un message d'avertissement et annulera cette transaction.

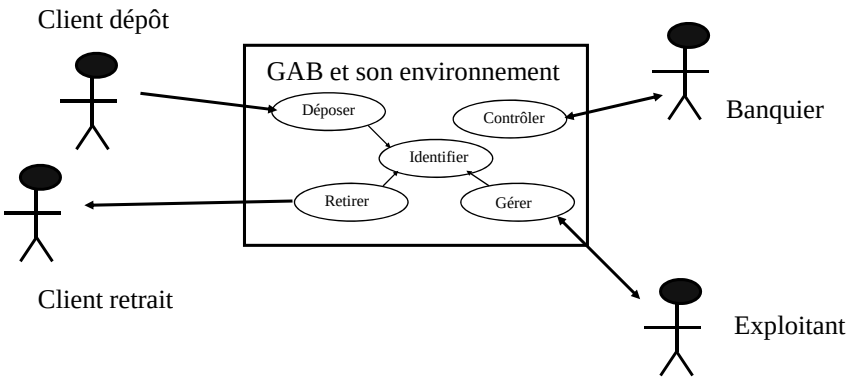
ÉCOLE CENTRALELYON

GAB (3/3)

- Le système doit annuler une transaction en cours et annuler l'autorisation d'accès si le client ne répond pas (correctement) à une demande du système dans un délai imparti raisonnable et/ou un nombre d'essais limite. Le système doit aussi répondre aux requêtes clients dans un temps raisonnable. La vérification d'erreurs et la sécurité physique font aussi partie du problème.
- Le GAB doit assurer un fonctionnement 24h sur 24, tous les jours, et doit être opérationnel 90% du temps. Pour éviter des mises hors services trop longues, les réparations entreprises sur le GAB ne doivent pas dépasser 2 heures (temps de réparation moyen).

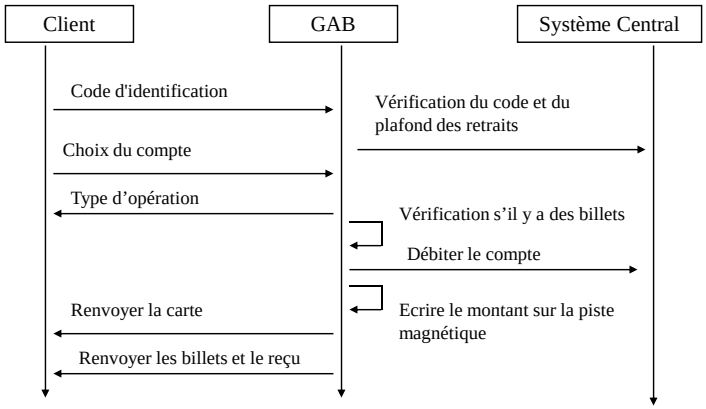
Diagramme de cas d'utilisation

- Contexte global du projet



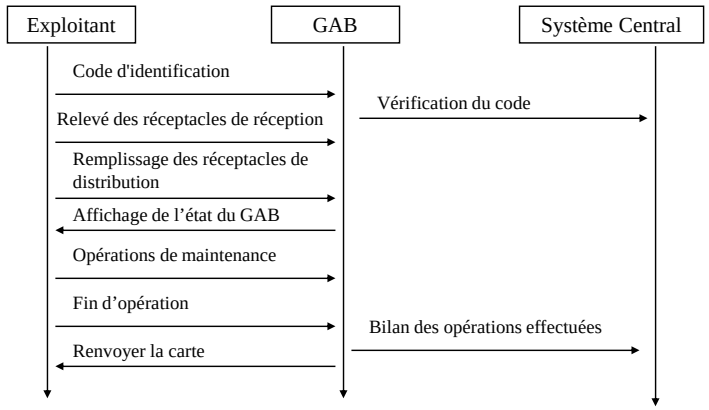
NFE 103 MAI Bertrand DAVID UML Concepts de base 83

Scénario de retrait en liquide par un client



NFE 103 MAI Bertrand DAVID UML Concepts de base 84

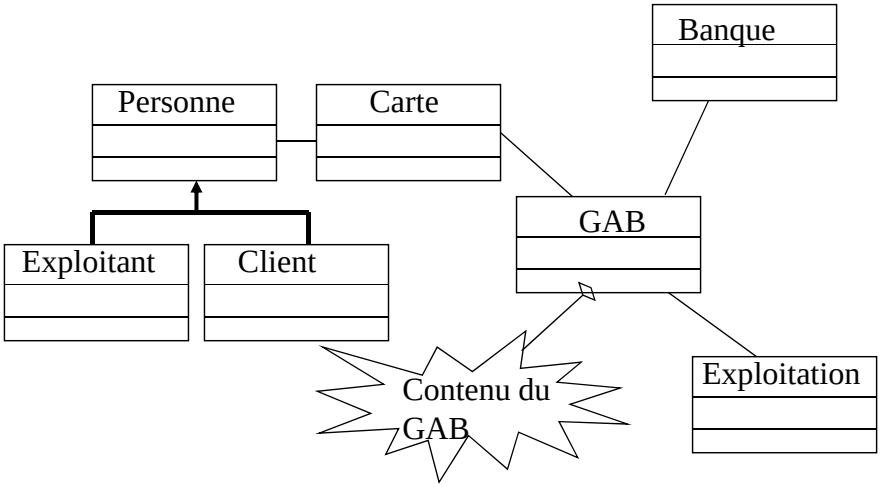
Scénario d'exploitation du GAB



NFE 103 MAI Bertrand DAVID UML Concepts de base 85

Diagramme de classes

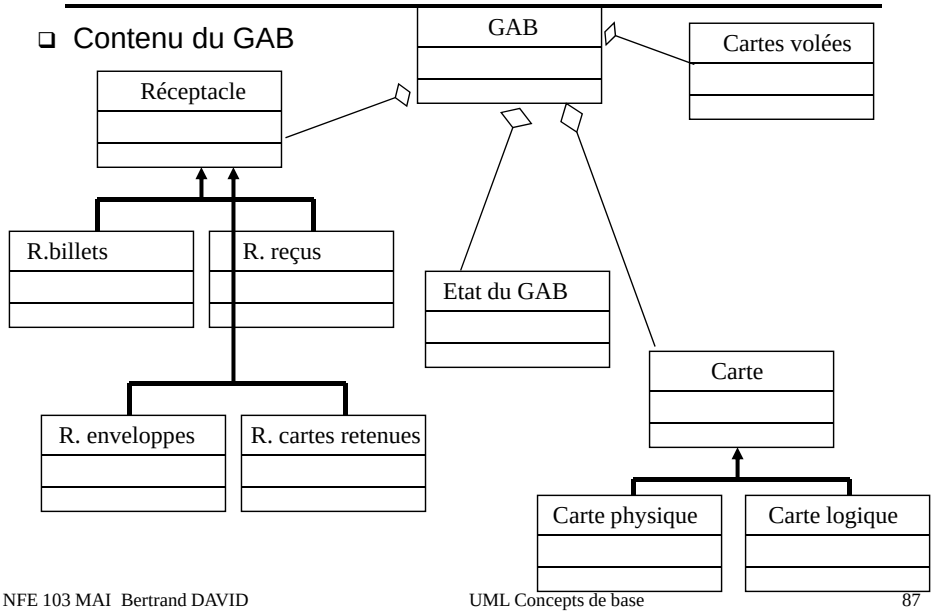
- Vue très partielle



NFE 103 MAI Bertrand DAVID UML Concepts de base 86

Diagramme de classes

- Contenu du GAB



NFE 103 MAI Bertrand DAVID UML Concepts de base 87

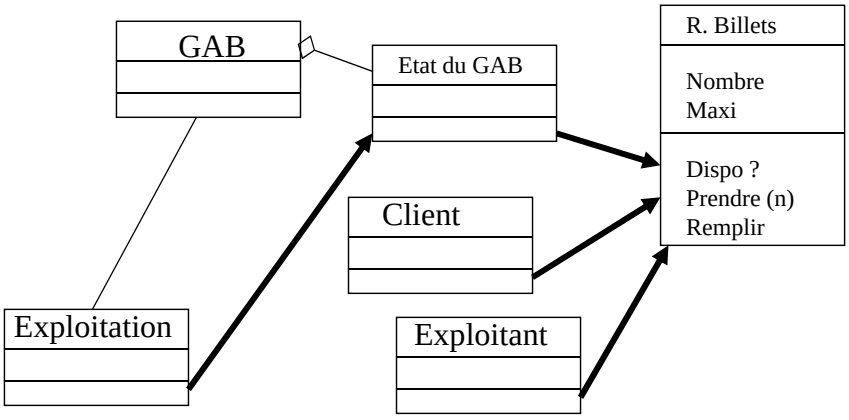
Définition précise de chaque classe

- Réceptacle des Billets
  - **Attributs**
    - **Nombre courant de billets**
    - **Nombre maximum de billets**
  - **Services**
    - **Nombre de billets disponibles**
    - **Nombre de billets à prendre**
    - **Nombre de billets (liasse) à mettre**
  - **Qui utilise les services :**
    - **Nombre de billets disponibles - EXPLOITATION**
    - **Nombre de billets à prendre - CLIENT**
    - **Nombre de billets (liasse) à mettre - EXPLOITANT**

NFE 103 MAI Bertrand DAVID UML Concepts de base 88

Diagramme de collaboration

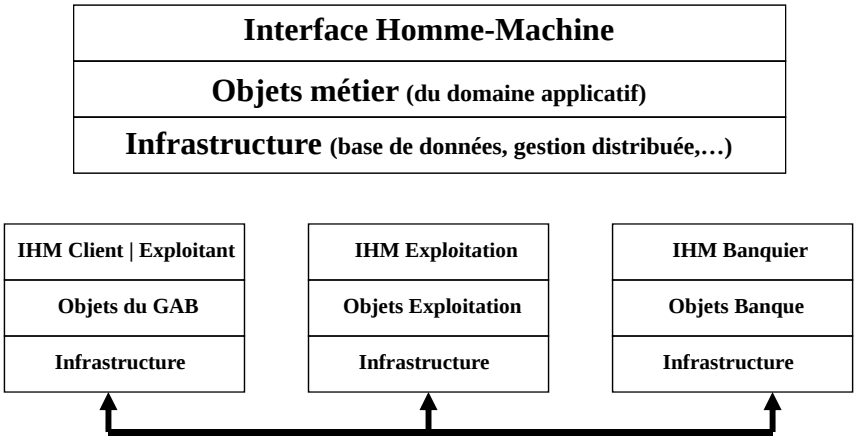
- Description de liens entre classes
  - liens statiques (associations)
  - liens dynamiques (utilisation de services)



NFE 103 MAI Bertrand DAVID UML Concepts de base 89

La conception du système

- L'architecture prédéfinie d'un système est organisée en trois couches représentant trois sous-systèmes de " haut niveau " :



NFE 103 MAI Bertrand DAVID UML Concepts de base 90