

# Bases de la conception orientée objet

À destination des étudiants de  
1<sup>e</sup> année IUT (S2)

Mireille Blay-Fornarino  
Université Nice Sophia Antipolis  
[blay@unice.fr](mailto:blay@unice.fr)

<http://mireilleblayfornarino.i3s.unice.fr/>

**Site web du module :**  
<https://mbf-iut.i3s.unice.fr/>

# Objectifs du cours



- ❧ **Savoir modéliser un problème pour ensuite pouvoir l'implémenter**
  - En répondant aux besoins des utilisateurs
  - En assurant la qualité du logiciel produit (performance, utilisabilité, sécurité, maintenabilité, ...)
- ❧ **Connaître la modélisation UML**
  - savoir lire des modèles ; savoir les construire
  - Faire le lien entre un modèle et le code qui pourrait correspondre.

**VOTRE OBJECTIF ?**

**Atteindre les objectifs du cours?**

**Avoir la meilleure note possible à cette UE?**

**Comment?**

**écouter, essayer, demander,**

**RESPECTER LES REGLES DONNEES**



# Notation

- Des notes de TD
- Un rendu final type carnet de bord
- Un examen final portant sur une étude de cas
- Des contrôles surprises

**Soyez Rigoureux :**

- Respectez les dates;
- Respectez les notations;
- Respectez les consignes.



# Introduction

# Où se trouve le logiciel ?

Finances



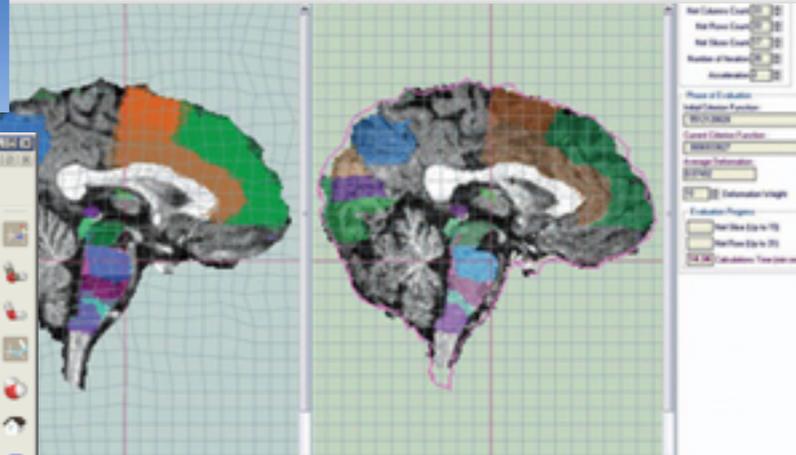
Arts, ..



Télécommunication



Médecine



Industrie



Transports



Domotique



Web



Fouille de données



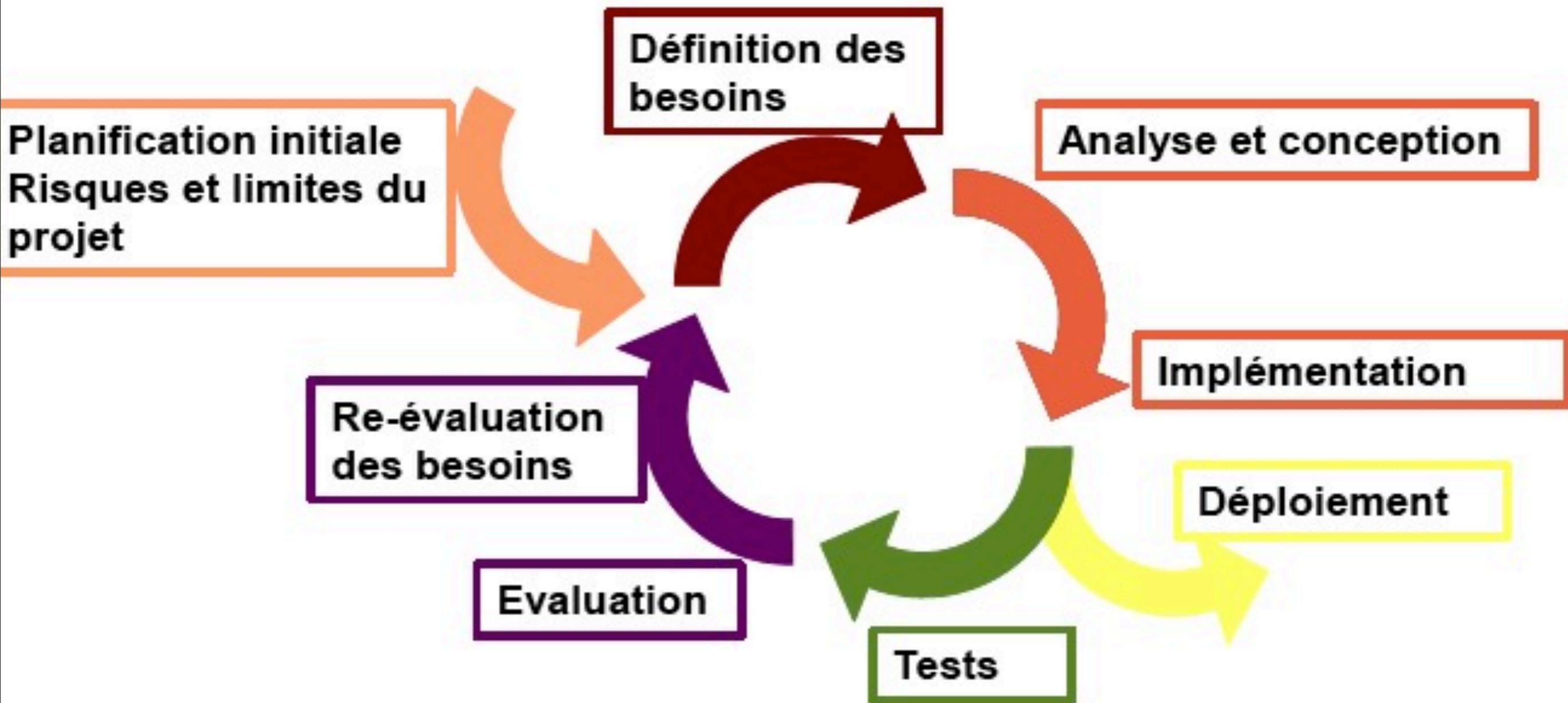
Nouvelles interactions



En musique, sport, ...

Vous devez réaliser une application logicielle pour visualiser les données météo dans un cockpit, que faites-vous?

# Activités du développement logiciel



La programmation n'est qu'une étape du développement !!

# Quelle qualité pour le logiciel ?

Monteriez-vous dans l'avion pour lequel vous avez écrit le système de visualisation des données météo?

Celui dont vous avez écrit le logiciel support aux commandes automatiques?

Vous avez râlé sur certains logiciels, ...

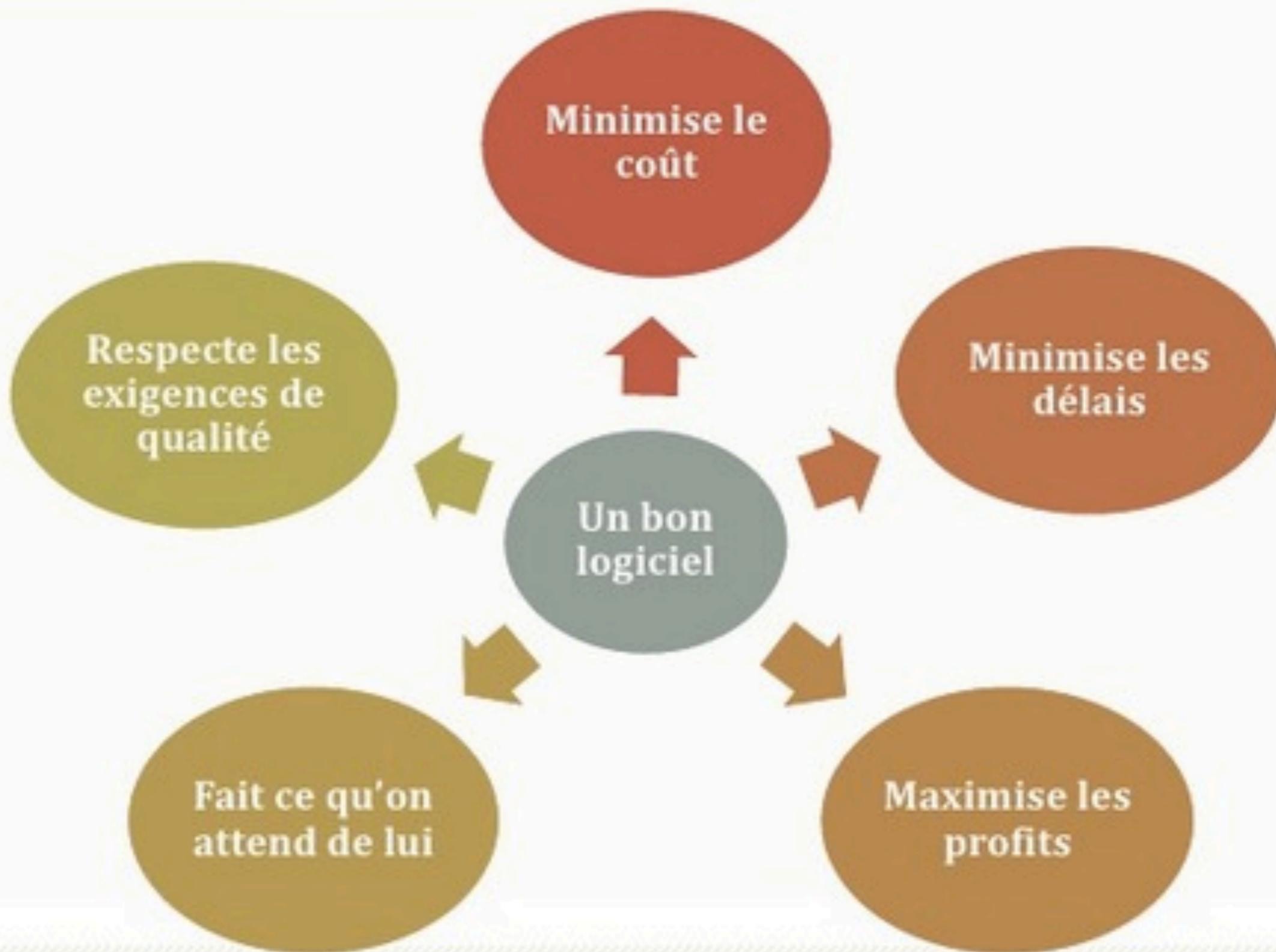
Pourquoi?

Etes-vous certains de savoir faire mieux?

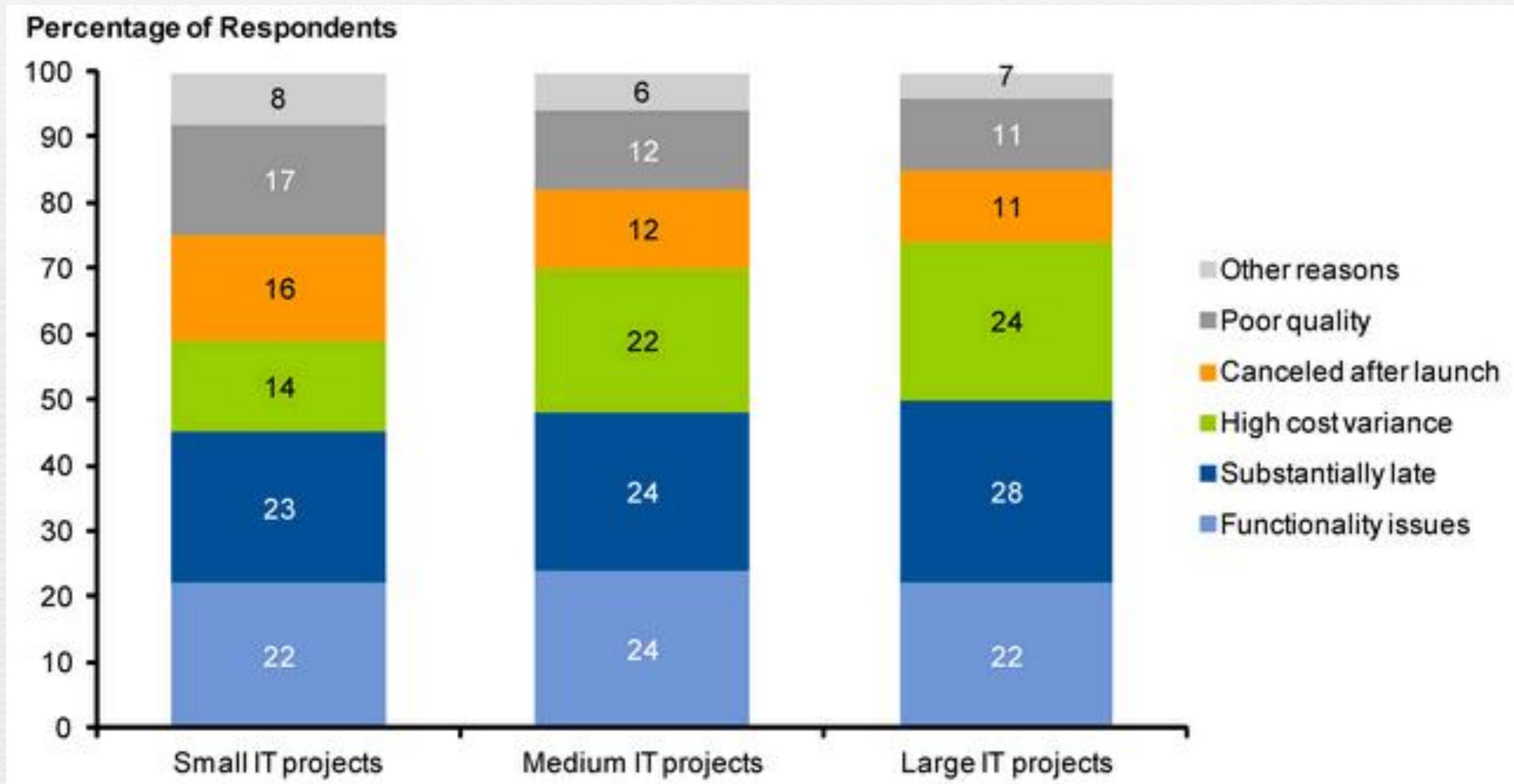
Vos pages web... sont-elles de «qualité»?

## Un Bon Logiciel du Point de Vue d'un Client





# Raisons de l'échec des projets (Gartner 2012)



UNIFIED  
MODELING  
LANGUAGE



# III. Introduction à UML

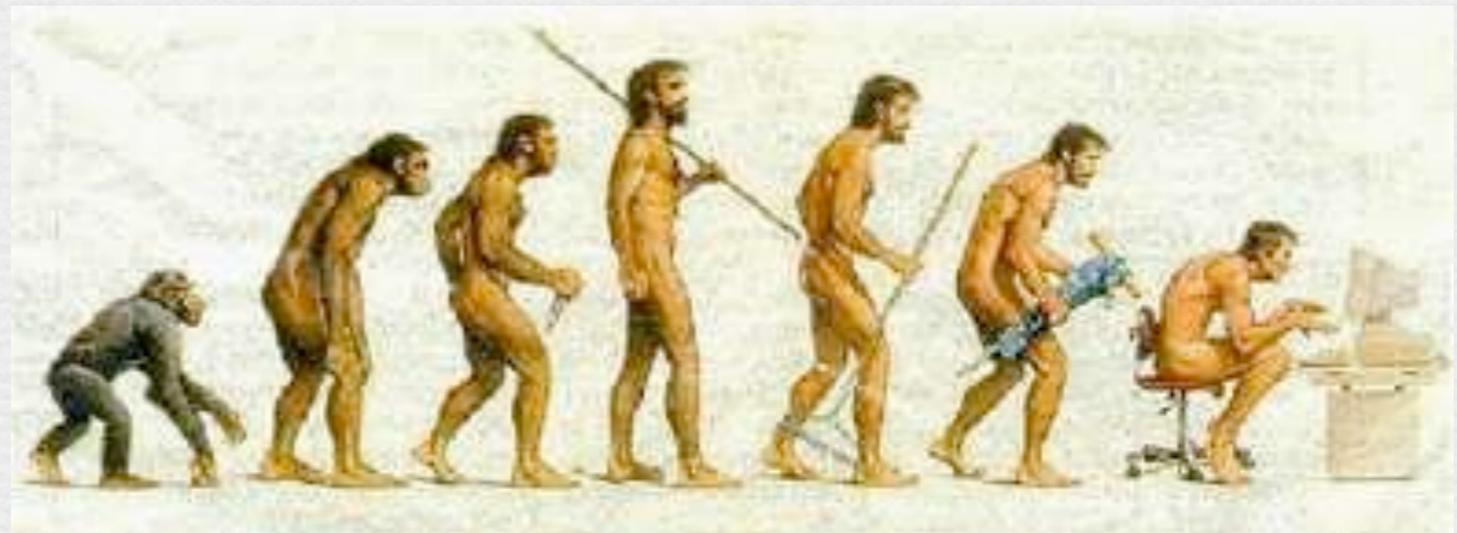
# III. Introduction à UML

UML, histoire, généralité

# Un peu d'histoire : La guerre des méthodes

Fin des années 90 , des méthodes : Booch, OMT, Coad/Yourdon, Fusion, SADT, OOSE, Schlaer/Mellor, HOOD...

On a au moins besoin d'un «langage» de modélisation standard !



# Et un langage unique, un !



*Un cocktail de notations éprouvées.*

*(...mais pas toutes, p. ex. RdP,  
SADT/IDEF0, DFD, etc.)*

- **Auteurs** : Grady Booch, Ivar Jacobson, James Rumbaugh.
- **Standardisation **OMG**** (Object Management Group) en 1997
- **Promoteurs** :
  - Rational Software, Oracle
  - HP, Microsoft, IBM

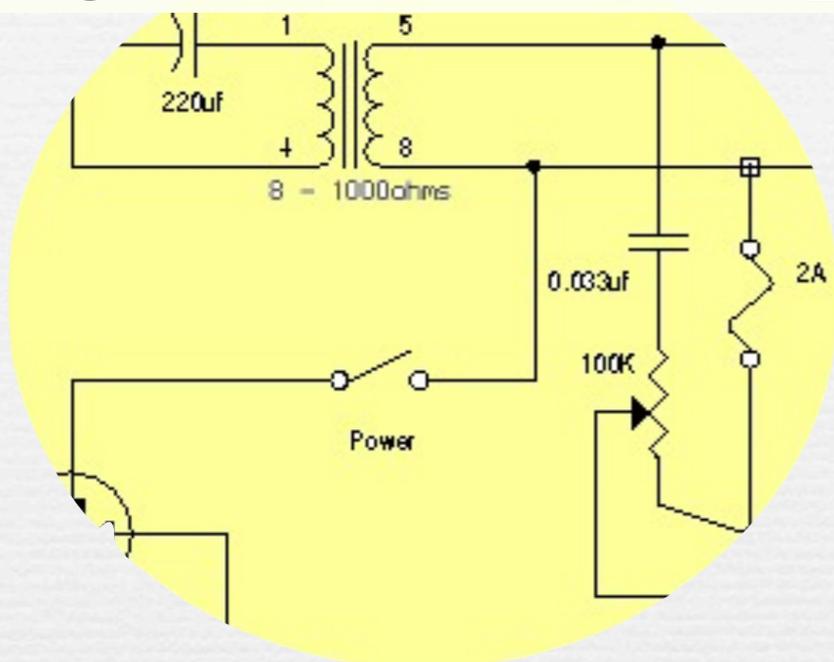
*"Lingua franca"*



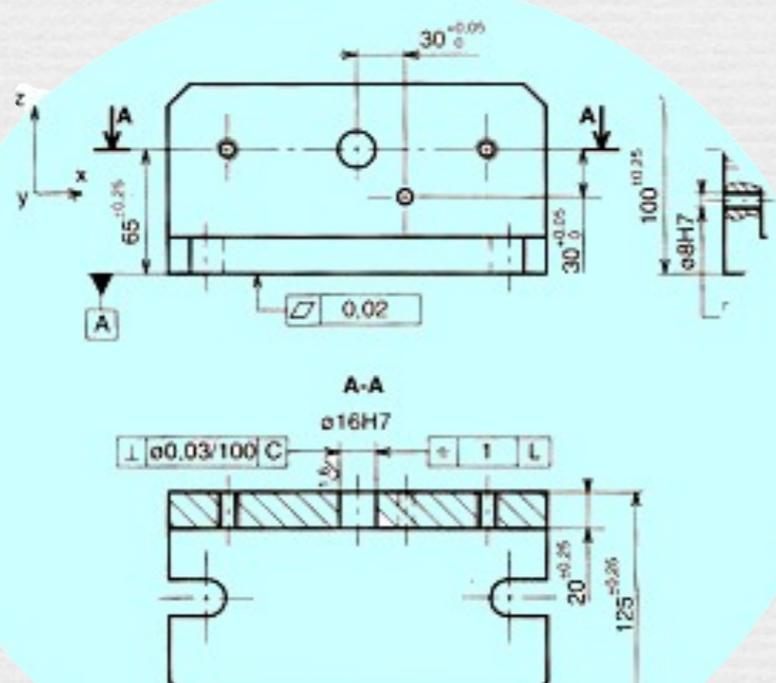
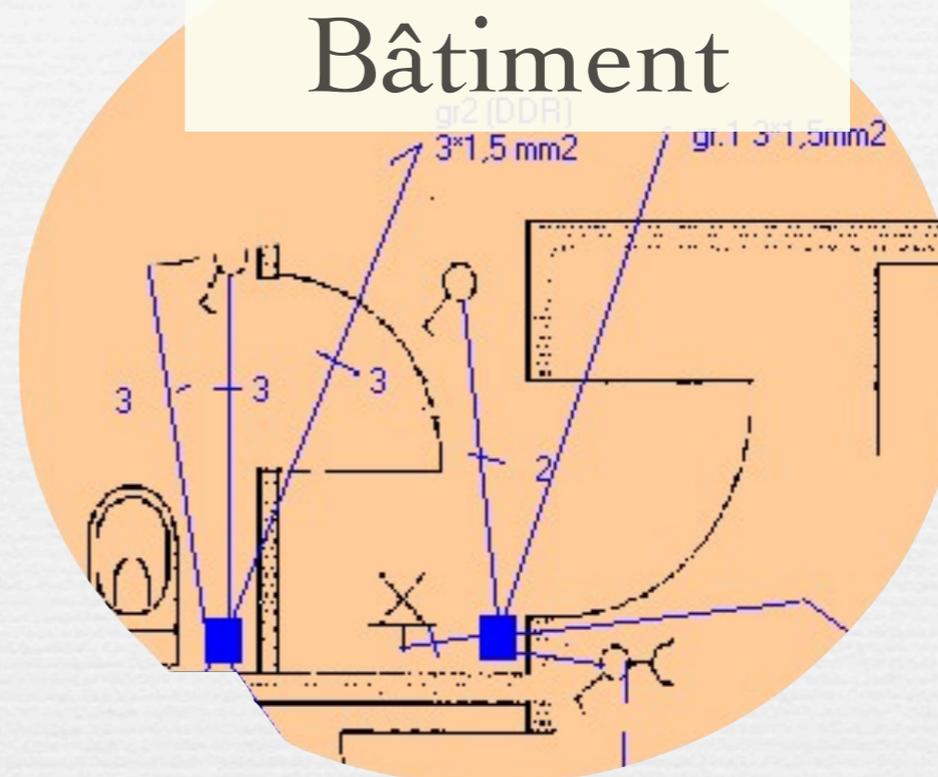
# UML dans l'industrie

- Sur 500 développeurs : 97% connaissent UML, 56% l'utilisent dans leurs projets de développement, (en 2005)
- <http://www.prweb.com/releases/2005/04/prweb231386.htm>

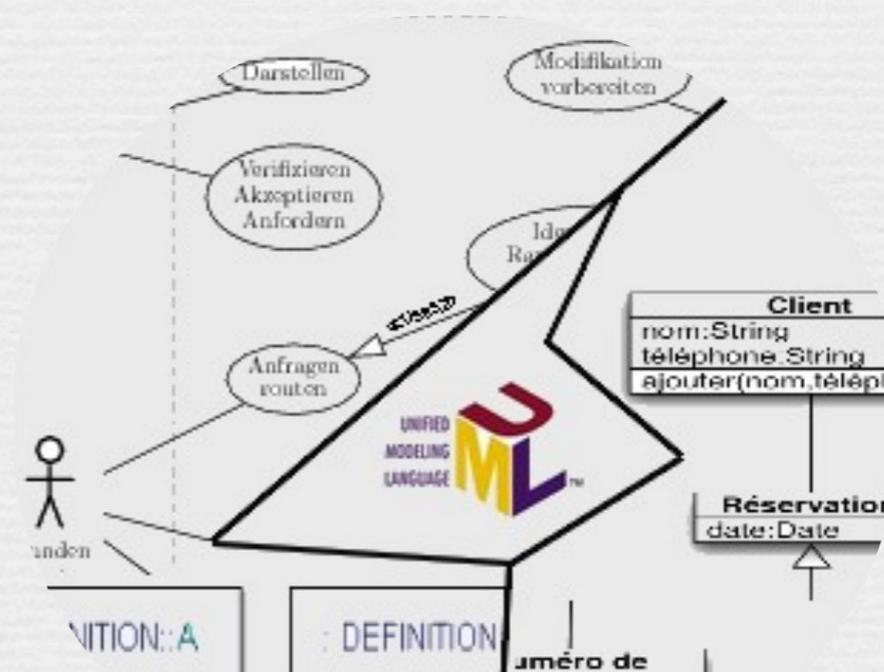
# Ingénierie Électrique



# Ingénierie du Bâtiment



# Ingénierie Mécanique



# Ingénierie Logicielle

# Qu'est-ce qu'UML ?

## un support à la modélisation

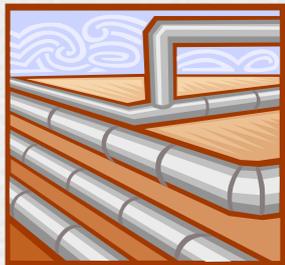
➔ **Modèle** : simplification de la réalité dont les buts sont



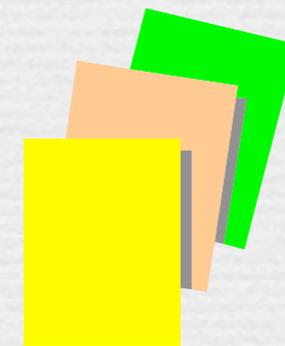
**Visualiser**  
le système



**Spécifier** la  
structure et le  
comportement  
du système



**Aider à la  
construction**  
du système



**Documenter**  
les décisions

# Qu'est-ce qu'UML ?

→ UML est un langage «visuel»

→ Il supporte

- La visualisation
- La spécification
- La construction
- La documentation

- *Descriptions graphiques et textuelles*
- *Syntaxe et sémantique*
- *Architecture et comportement*
- *Génération de code*

■ UML n'est pas une méthodologie de développement.

# Les points forts d'UML

- UML est un langage normalisé
  - gain de précision
  - gage de stabilité
  - encourage l'utilisation d'outils
- UML est un support de communication performant
  - Il cadre l'analyse.
  - Il facilite la compréhension de représentations abstraites complexes.
  - Son caractère polyvalent et sa souplesse en font un langage universel.

Il doit vous aider à comprendre les concepts de la programmation par objets!

# Les points faibles d'UML

- La mise en pratique d'UML nécessite un apprentissage et passe par une période d'adaptation.
- Le processus de développement (non imposé par UML) est une autre clé de la réussite d'un projet.



# Questions

# III. Introduction à UML

## survol

# Vue fonctionnelle

la vue fonctionnelle cherche à appréhender  
les interactions entre les  
acteurs/utilisateurs

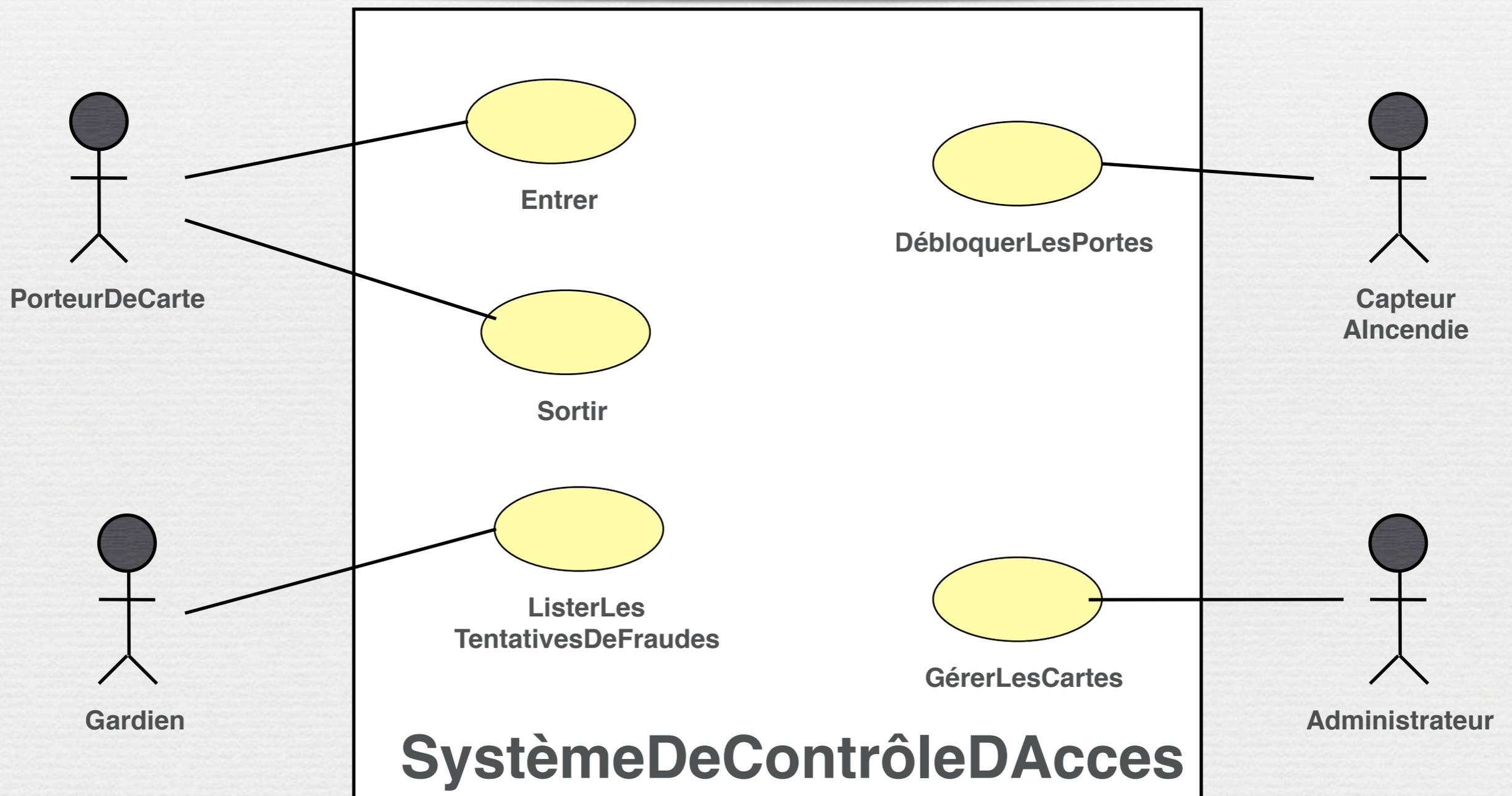
**et le système,**

sous forme d'objectifs à atteindre (**cas d'utilisation**) et  
sous forme chronologique de scénarios d'interaction  
typiques (**diagrammes de séquences**)

# Qu'est-ce qu'UML ?

## Diagrammes des cas d'utilisation

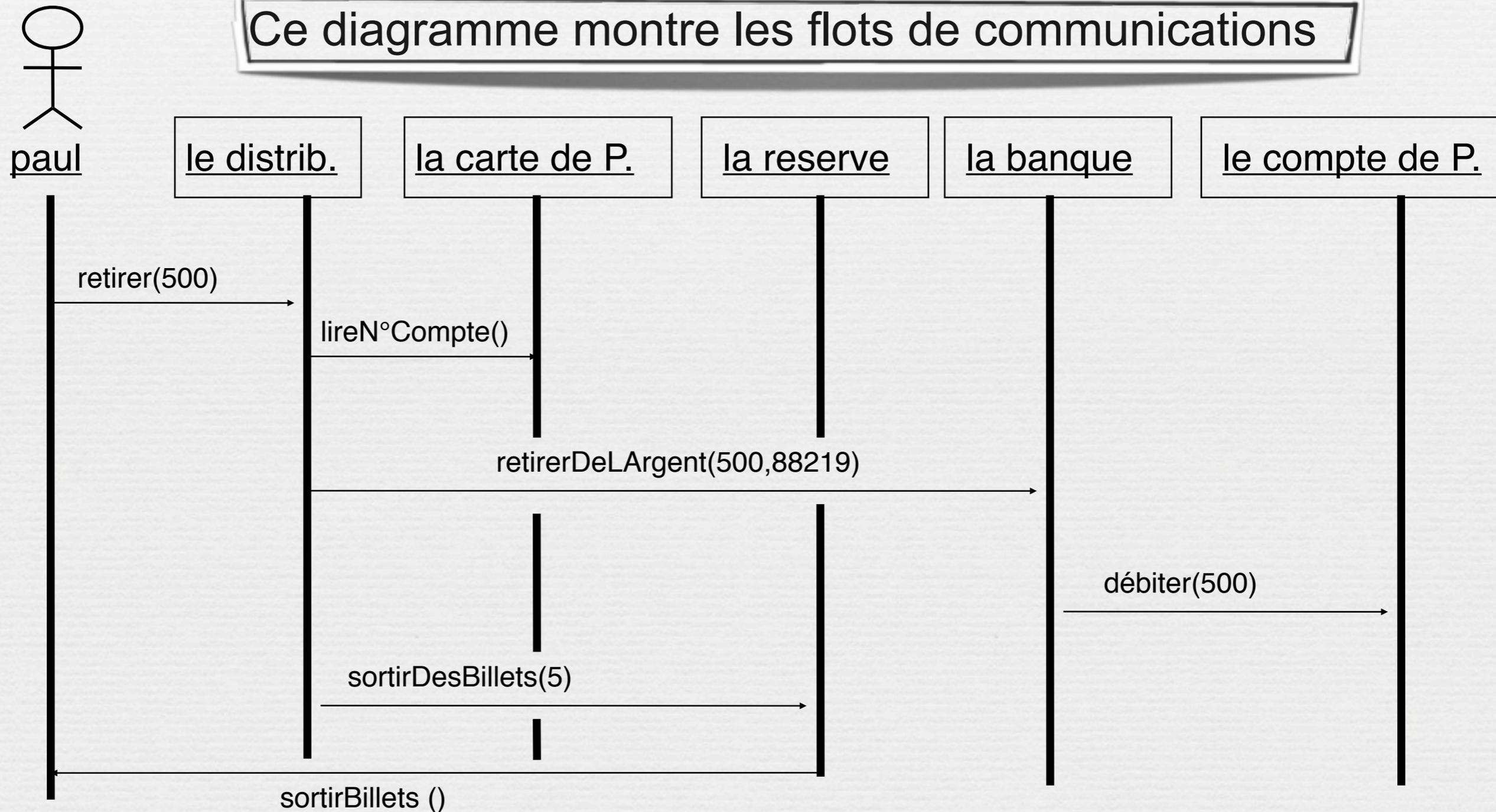
Ce diagramme montre ce que fait le système et qui l'utilise



# Qu'est-ce qu'UML ?

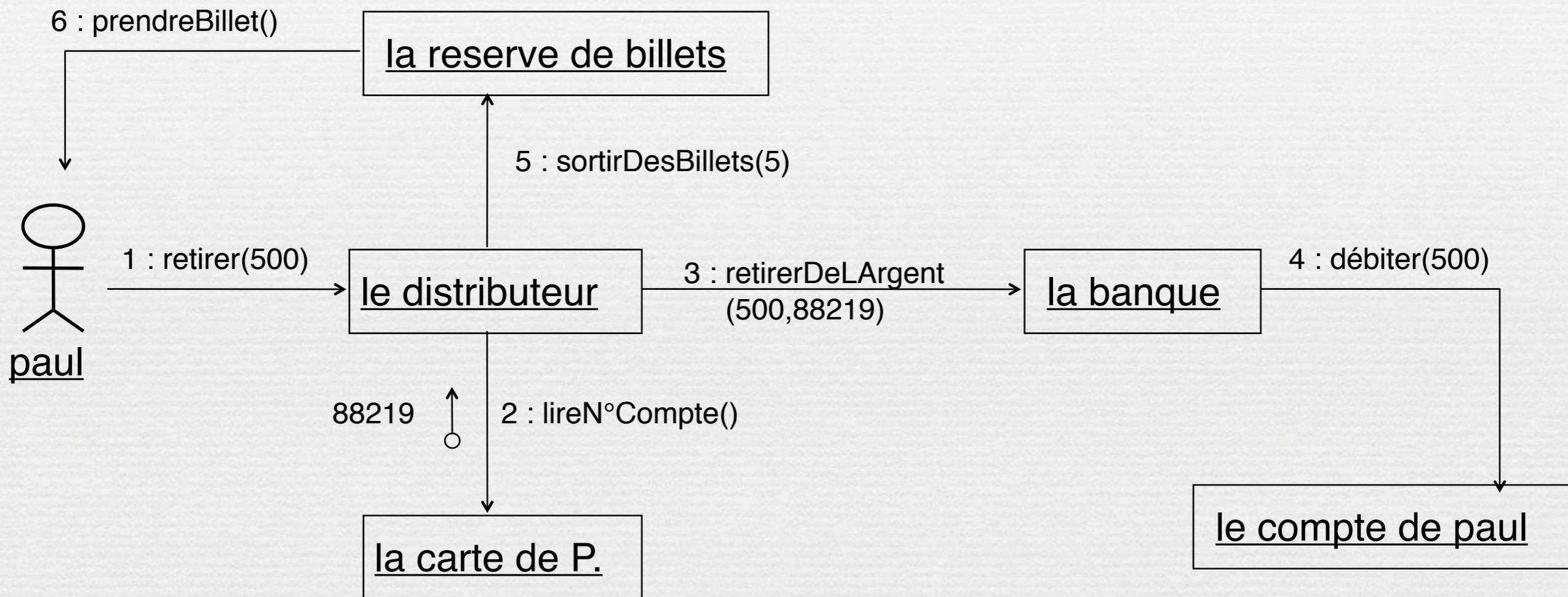
## Diagrammes de séquence

Ce diagramme montre les flots de communications



# Qu'est-ce qu'UML ?

## Diagrammes de collaboration



Non étudié cette  
année

# Vue Structurelle

la vue structurelle, ou statique, vise à

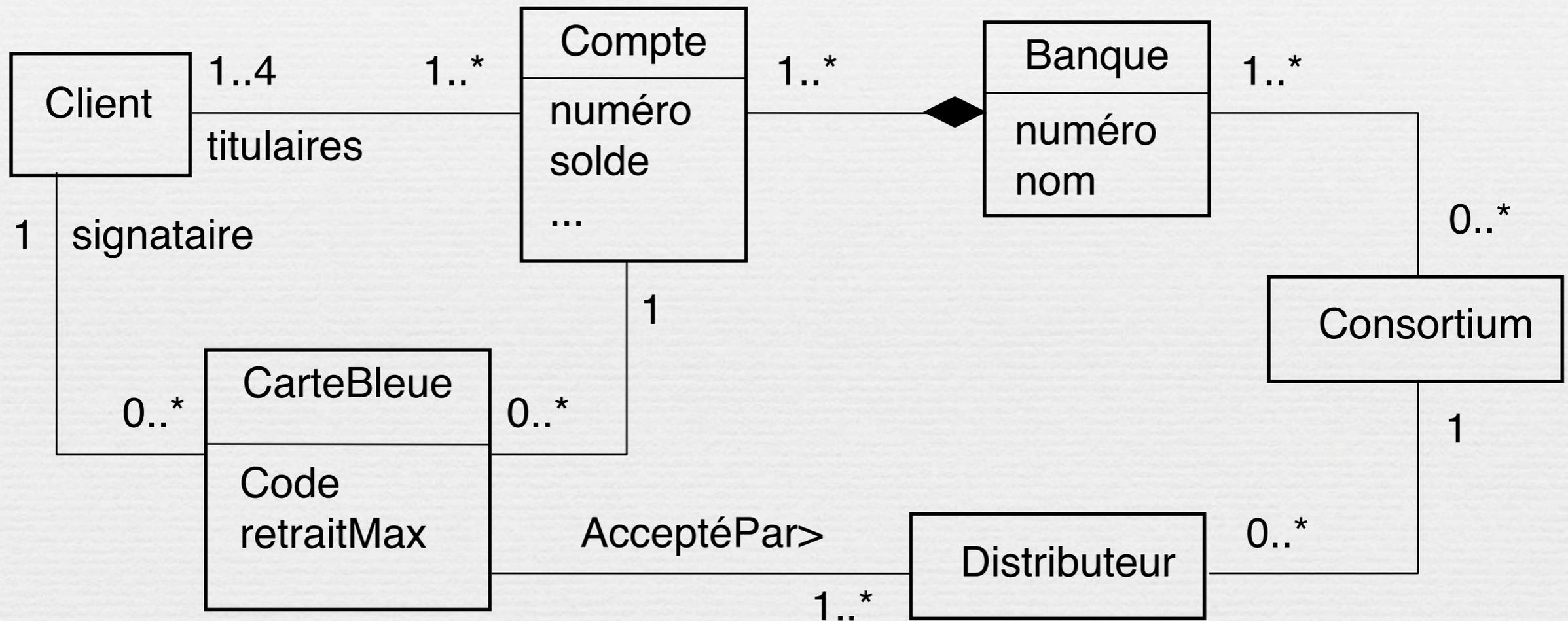
**identifier les objets/composants**

constituant le programme, leurs attributs, opérations et méthodes, ainsi que les liens ou associations qui les unissent (**diagramme de classes**). Elle permet aussi de regrouper les classes fortement liées entre elles en des composants les plus autonomes possibles (**diagramme de packages**).

A l'intérieur de chaque package, on trouve un diagramme de classes.

# Qu'est-ce qu'UML ?

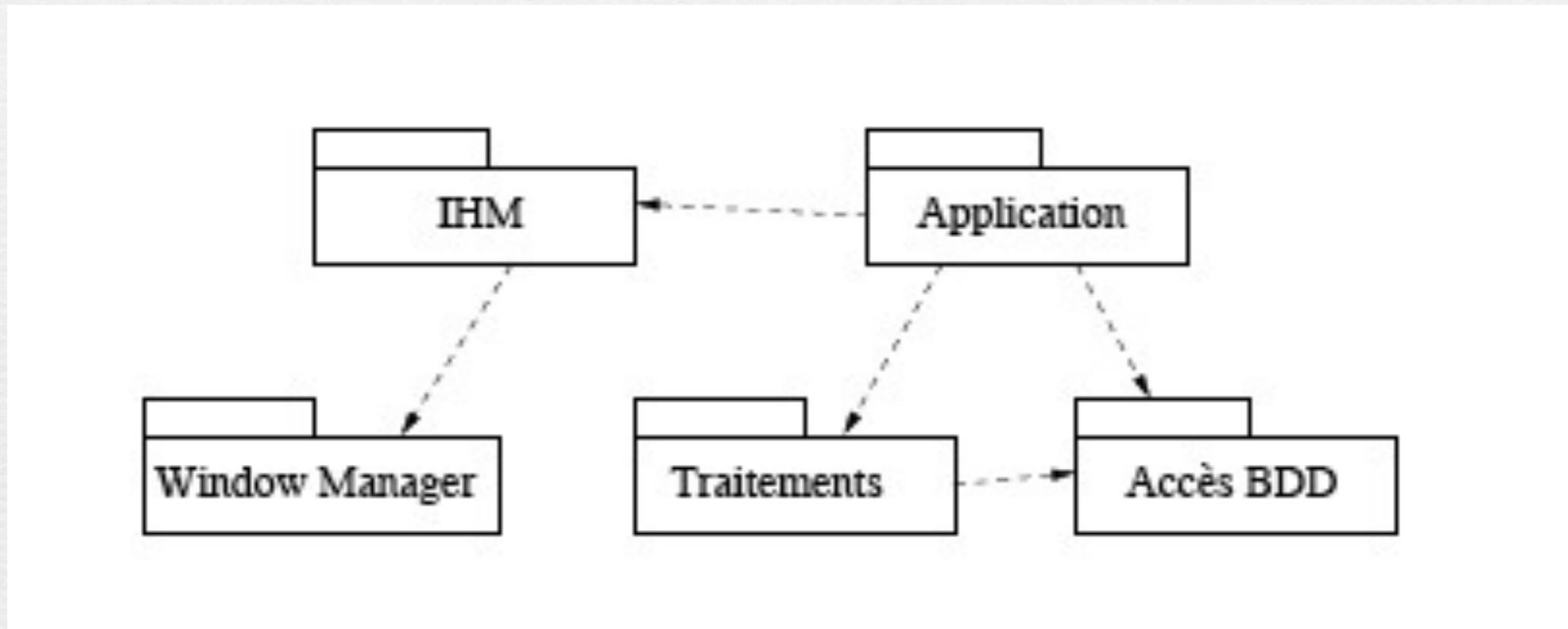
## Diagrammes de classes



Ce diagramme montre les classes et les relations entre elles

# Qu'est-ce qu'UML ?

## Diagrammes de packages



Regrouper entre elles des classes liées les unes aux autres de manière à faciliter la maintenance ou l'évolution du projet et de rendre aussi indépendantes que possible les différentes parties d'un logiciel.

# Vue Dynamique

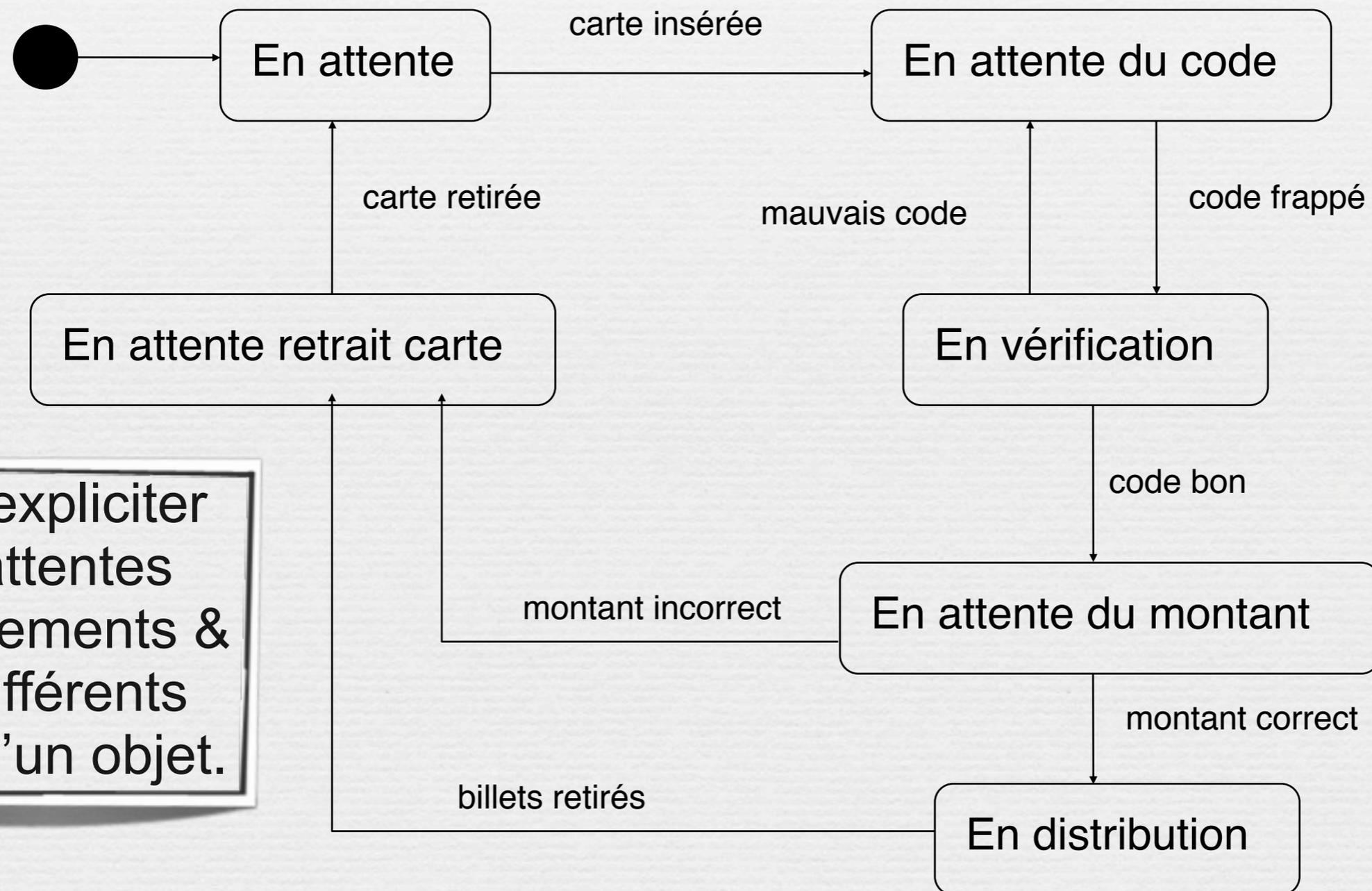
la vue dynamique vise à décrire l'évolution (la dynamique) des objets complexes du programme tout au long de leur cycle de vie.

De leur naissance à leur mort, les objets voient leurs changement d'états guidés par les interactions avec les autres objets (les **diagrammes d'états**).

Le **diagramme d'activité** est une sorte d'organigramme correspondant à une version simplifiée du diagramme d'états. Il permet de modéliser des activités qui se déroulent en parallèle les unes des autres, quand ce parallélisme peut poser problème.

# Qu'est-ce qu'UML ?

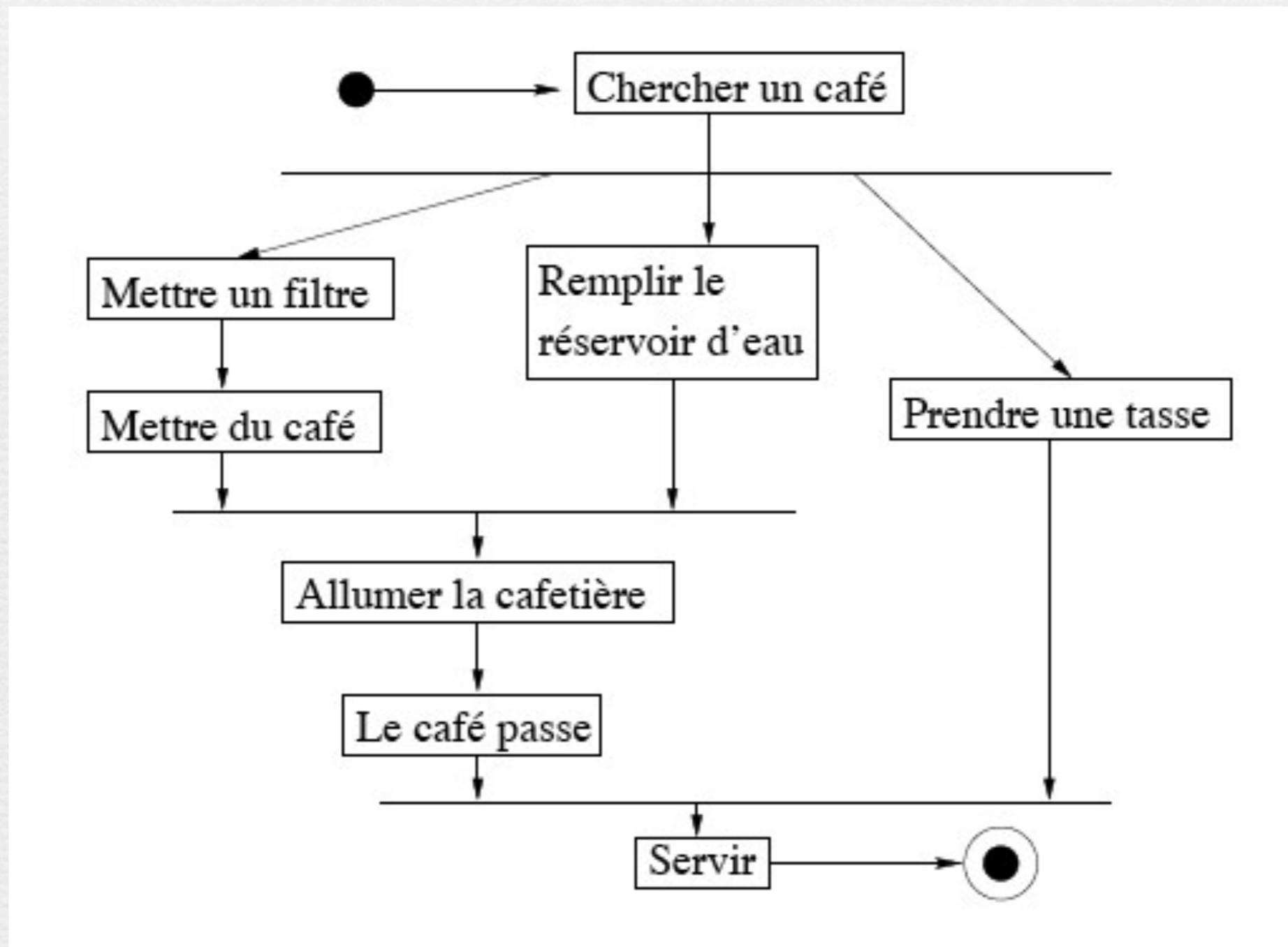
## Diagrammes d'états



Pour expliciter  
les attentes  
d'évènements &  
les différents  
états d'un objet.

# Qu'est-ce qu'UML ?

## Diagrammes d'activités



# Utilisation de UML dans l'industrie

Modèle	Niveau d'utilisation
Classe et package	<b>5</b> Le plus utilisé, le plus essentiel
Use case	<b>3</b> Dépend des cultures. Il y a des domaines où cela a peu d'intérêt
Séquence	<b>3</b> Assez employé, bien compris
Collaboration	<b>2</b> Les diagrammes d'objet sont utilisés, mais la modélisation des collaboration et des rôles est encore marginale
Etat/transition	<b>2</b> Très utilisé dans le temps réel, peu utilisé dans le tertiaire.
Activité	<b>1</b> Arrivée tardive dans les ateliers, habitude culturelle locale à certaines parties du tertiaire
Composant/Déploiement	<b>1</b> Peu de personnes l'utilisent, sur peu de parties de leurs applications

# Qu'est-ce qu'UML ?

## Divers modes d'utilisation selon [Fowler 2003]

### ➔ Mode esquisse (*sketch*)

- Informelle, incomplète

- Souvent manuelle (tableau)

- ➔ *Support de communication pour concevoir les parties critiques*

### ➔ Mode plan (*blue print*)

- Diagrammes détaillés

- ➔ *Génération d'un squelette de code à partir des diagrammes*

- ➔ *Nécessité de compléter le code pour obtenir un exécutable*

### ➔ Mode langage de programmation

- Spécification complète, formelle et **exécutable**

- ➔ *Pas vraiment disponible actuellement !*

# Qu'est-ce qu'UML ?

## Divers modes d'utilisation selon [Fowler 2003]

### → Modèles descriptifs vs prescriptifs

- Descriptifs ; Décrire l'existant (domaine, métier)
- Prescriptifs ; Décrire le futur système à réaliser

### → Modèles destinés à différents acteurs

- Pour l'utilisateur ; Décrire le quoi
- Pour les concepteurs/développeurs ; Décrire le comment

# Différents niveaux de description

Selon l'activité de l'ingénieur, qu'il s'agisse d'analyse, de conception ou d'implémentation, le niveau de détail avec lequel est représenté le diagramme des classes change énormément.

– le point de vue de l'analyse, qui en général se doit d'oublier tout aspect de mise en oeuvre et, en ce sens, est complètement indépendant du logiciel (on n'y parlera pas de structuration des données : tableaux, pointeurs, listes, ...)

– le point de vue de la conception, qui cherche à identifier les interfaces, les types des objets, leur comportement externe et la façon interne de les mettre en oeuvre, sans être encore fixé sur un langage ;

– le point de vue de l'implémentation, qui cherche à décrire une classe, ses attributs et ses méthodes en pensant déjà au code qui les implémentera et prend en compte les contraintes matérielles de temps d'exécution, d'architecture, etc.

Dans le cadre d'une analyse, seuls les noms des attributs et les principales méthodes publiques de la classe ont à être mentionnées. Dans le cadre d'une conception et, à plus forte raison, d'une implémentation, la description des classes devra être exhaustive. Mais les différences ne se limitent pas au seul niveau de description. De nombreuses classes spécifiques seront ajoutées lorsque l'on passe de l'analyse à la conception, l'organisation des diagrammes peut évoluer, etc.

# Bibliographie

Ce cours a été monté en utilisant de nombreux supports dont je remercie chaleureusement ici les auteurs  
D'autres références se trouvent sur le site du module.

- Merise: 5ème Partie Dossier "SAM l'Informaticien" du 5 Mars au 18 Mars 2001 par Stéphane Lambert <http://www.vediovis.fr/index.php?page=merise5>
- Introduction au langage UML, SUPINFO
- De Merise à UML, Nasser Kettani, Dominique Mignet, Eyrolles
- [http://www.compucycles.com/nouveausite/articles/Merise/Article\\_07.htm](http://www.compucycles.com/nouveausite/articles/Merise/Article_07.htm)
- UML-MERISE Etude Comparative, OSITEC-Consultants, 2004-2005
- Modélisation Orientée objet, M.Grimaldi – janvier 2010

