

Université de Corse

2024-2025

L3 Informatique

UE Conception et Programmation Objet Modélisation UML

CH1 – UML et ACOO



Objectifs de ce chapitre



- Découvrir et situer les activités d'analyse et conception orientée objet
- Découvrir le formalisme UML et comprendre son rôle
- Introduire une démarche pratique d'ACOO



Modélisation UML

Plan du Cours



CH1 – UML et ACOO

CH2 – MODELE DU DOMAINE

CH3 – MODELE DES CAS D'UTILISATION

CH4 – MODELE D'ANALYSE

CH 1 – UML et ACOO

1.1 – Notions d'Analyse et de Conception

1. Rôle central des modèles
2. Notion de Méthode
3. Etapes d'Analyse et Conception
4. Historique de l'ACOO : vers UML



1.2 – Le Formalisme UML (survol)

1.3 – Démarche d'ACOO avec UML



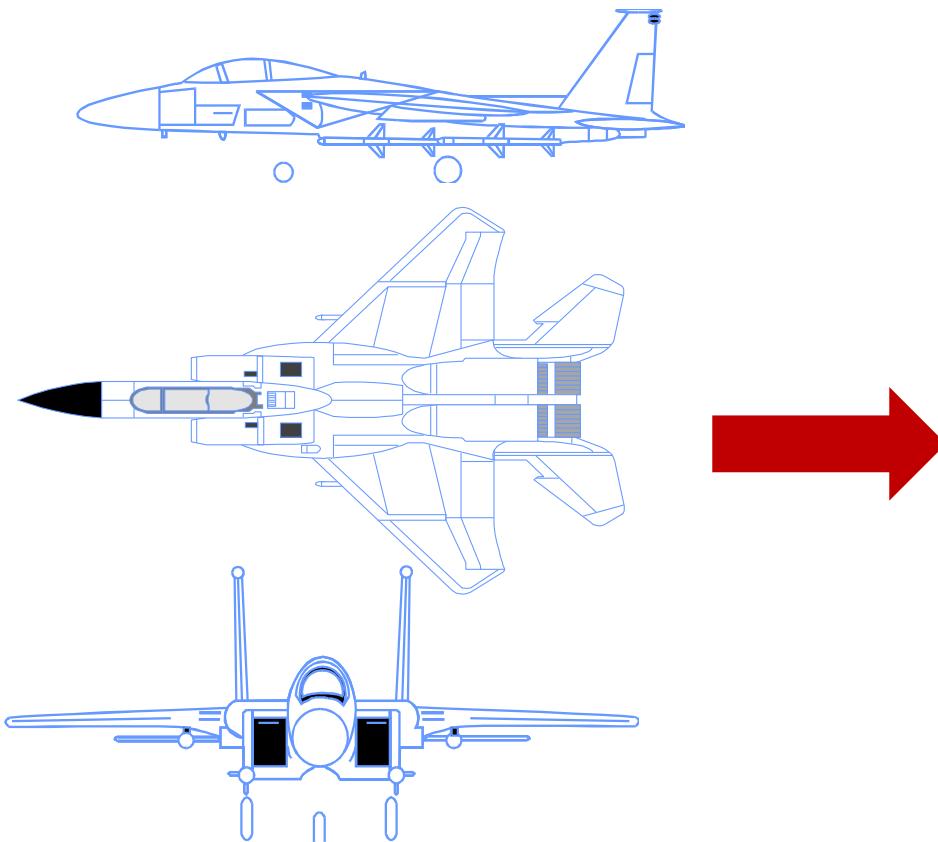
Qu'est ce qu'un
modèle?





1 – Rôle central des modèles

Un modèle est la simplification de la réalité.





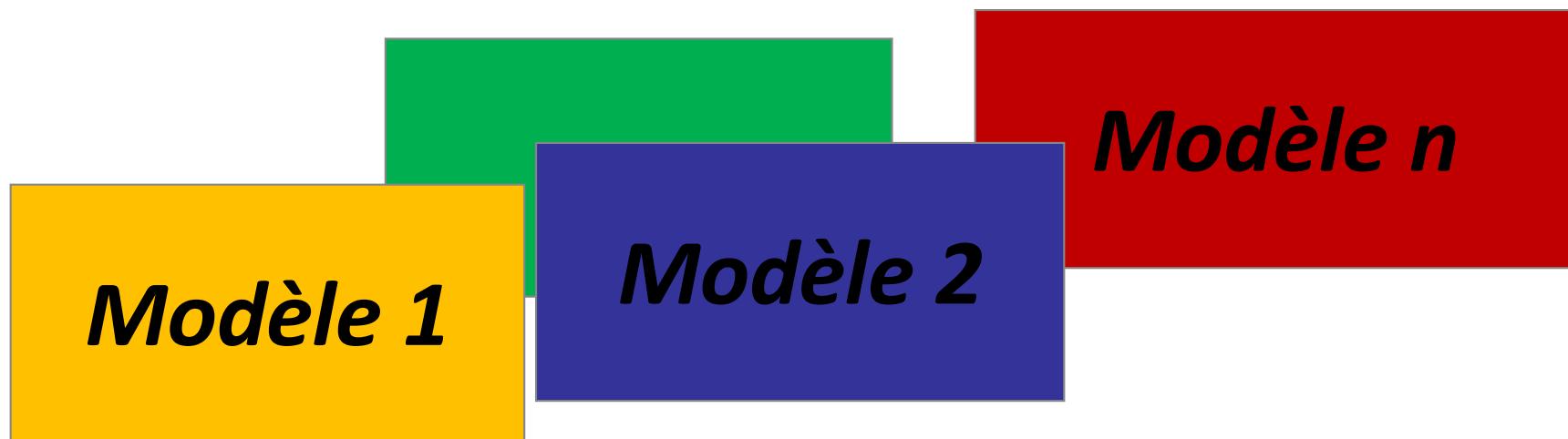
Qu'est ce que la modélisation?





1 – Rôle central des modèles

Modélisation= technique d'ingénierie consistant à définir une ou plusieurs représentation(s) sémantiques «simplifiée(s)» du système à construire (*modèles*)

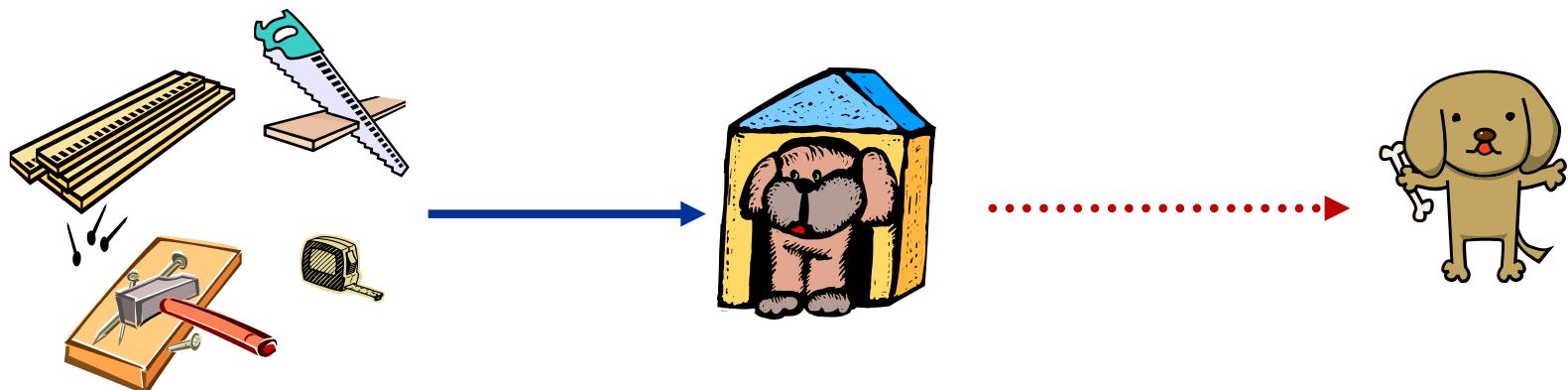


Système logiciel

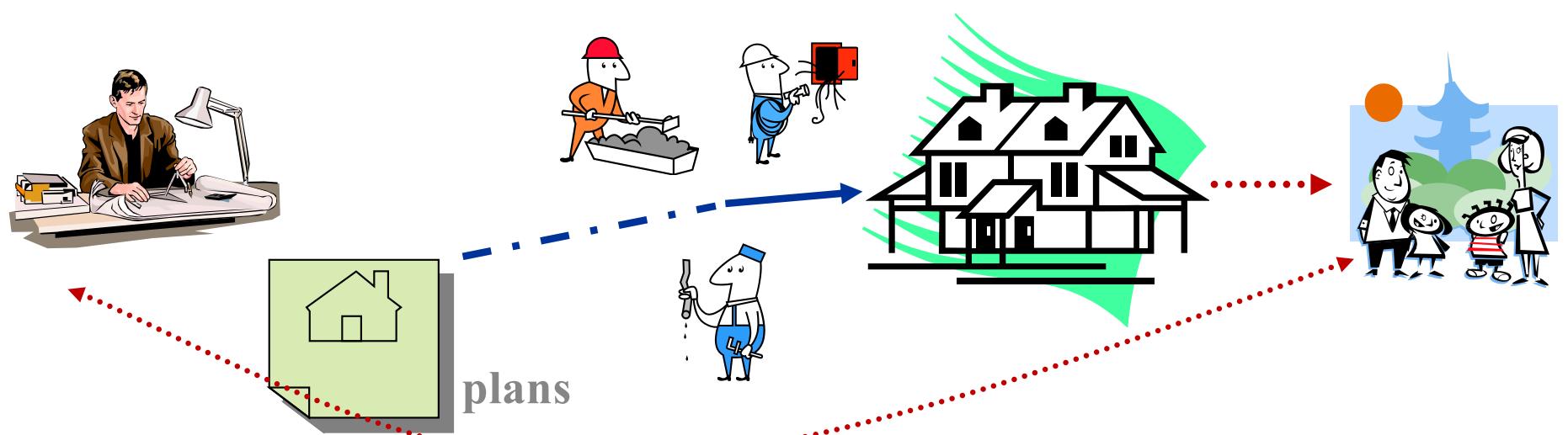


1 – Rôle central des modèles

Construction d'une niche



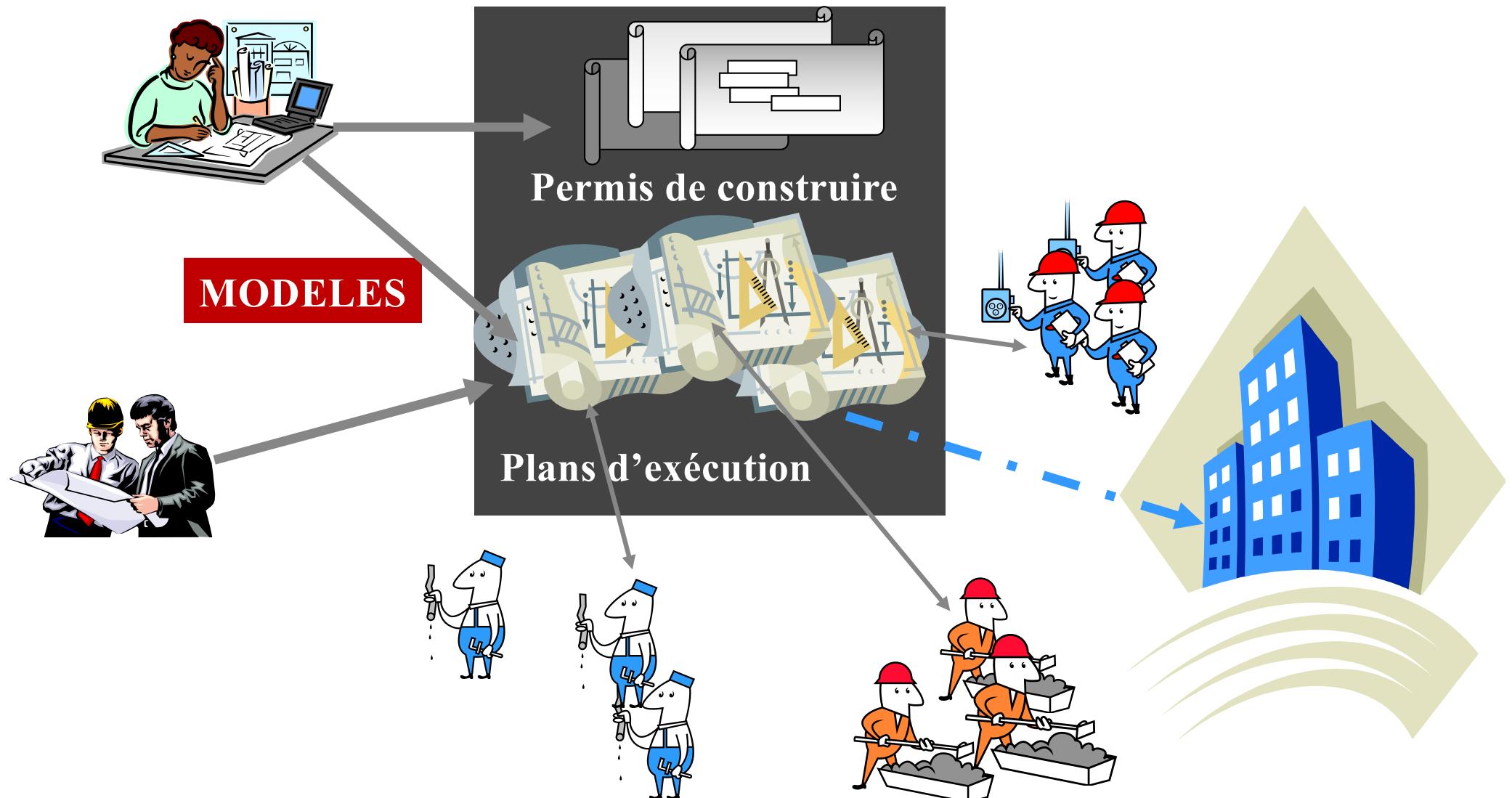
Construction d'une maison





1 – Rôle central des modèles

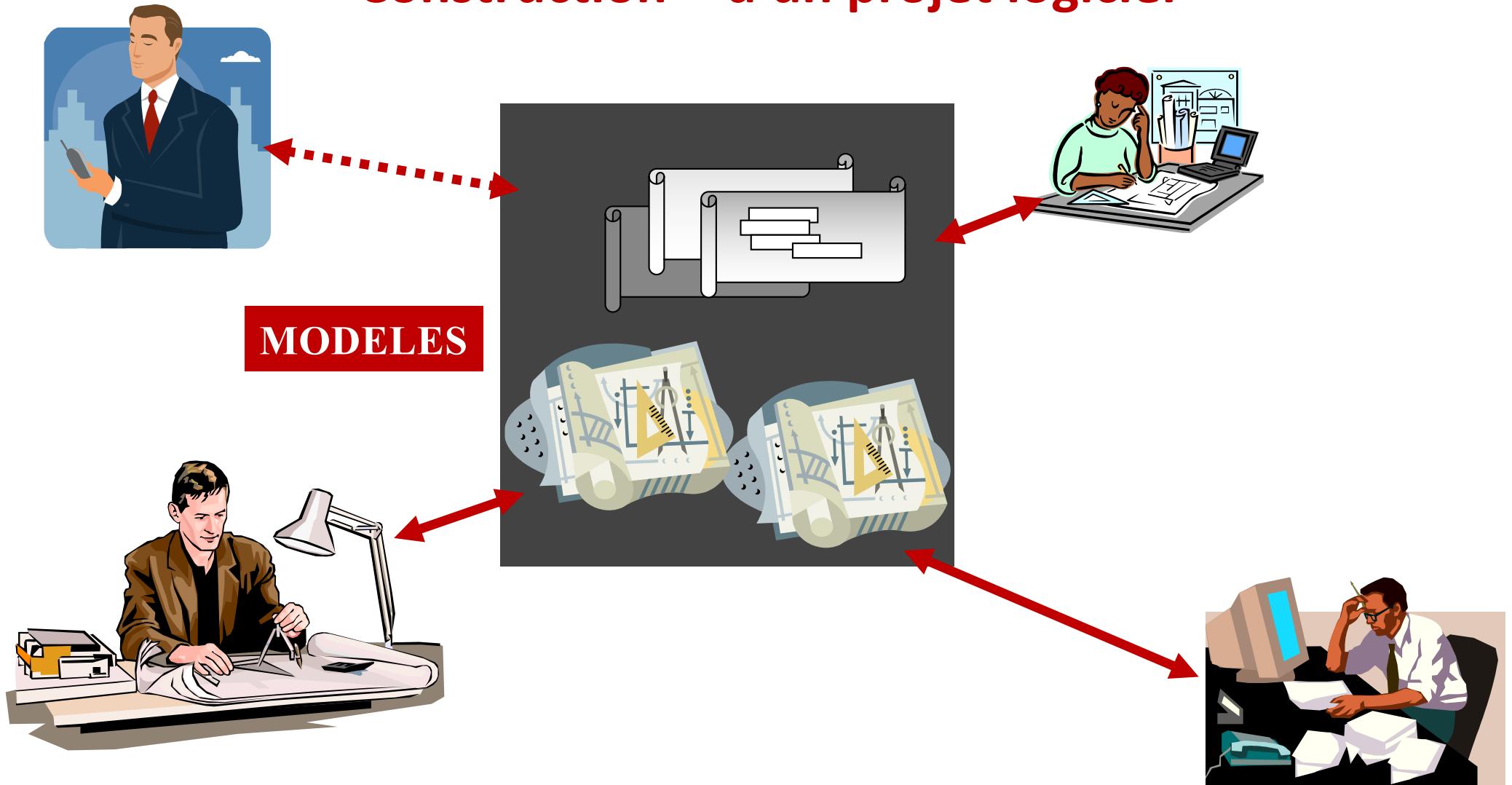
Construction d'un immeuble





1 – Rôle central des modèles

« Construction » d'un projet logiciel





1 – Rôle central des modèles

Les modèles sont au centre du processus de développement.

- Communication
 - avec les clients
 - entre les intervenants
- Réalisation
 - Gestion des risques et contraintes techniques
 - Planification
- Optimisation
- Maintenance et Evolutivité

niveaux d'abstraction
points de vue



Qu'est ce qu'une
méthode d'analyse
et conception?





2 - Notion de Méthode d'Analyse et Conception

Objectif : Construire un (ou plusieurs) modèles facilitant l'implémentation, la maintenance et l'évolutivité d'un système

Concepts de modélisation

- *Notation*
- *Langage*
- *Représentation graphique*

→ *Définition formelle*

+ Démarche + Outils

- *Etapes*
- *Processus de développement*

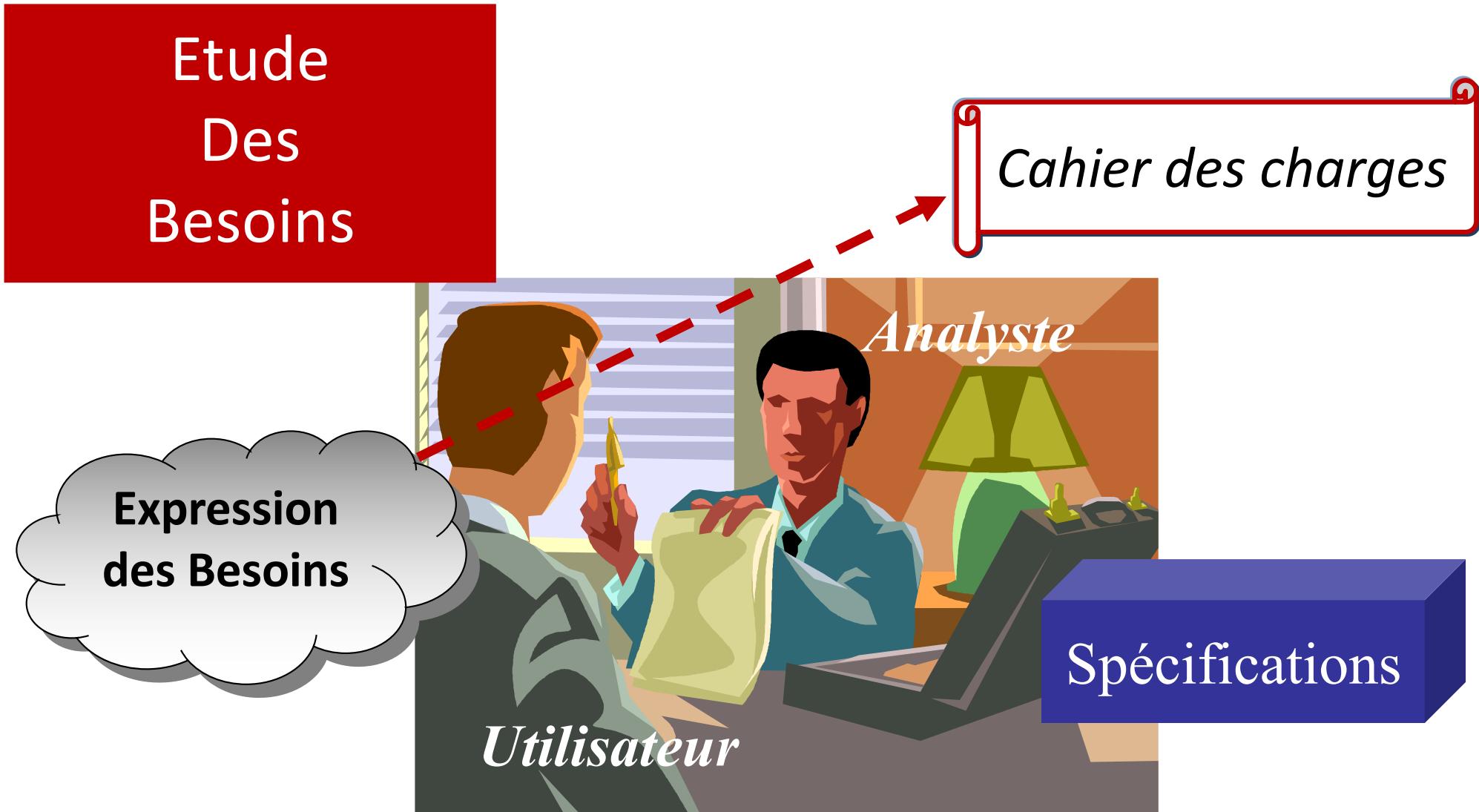


Quelles sont les
étapes d'un
processus de
développement?





3 - Etapes d'analyse et de Conception



3 - Etapes d'analyse et de Conception

ANALYSE



Première phase de construction
de la solution informatique

- Solution “idéale” **indépendante** de la plateforme technique choisie
- Besoins **fonctionnels** uniquement
- Modèle **minimal**



3 - Etapes d'analyse et de Conception

CONCEPTION

Modélisation détaillée de la solution informatique

- Proche du **code**
- Prise en compte des **contraintes** techniques liées à la plateforme
- Prise en charge des besoins **non fonctionnels**
- Modèle **complet**



3 - Etapes d'analyse et de Conception

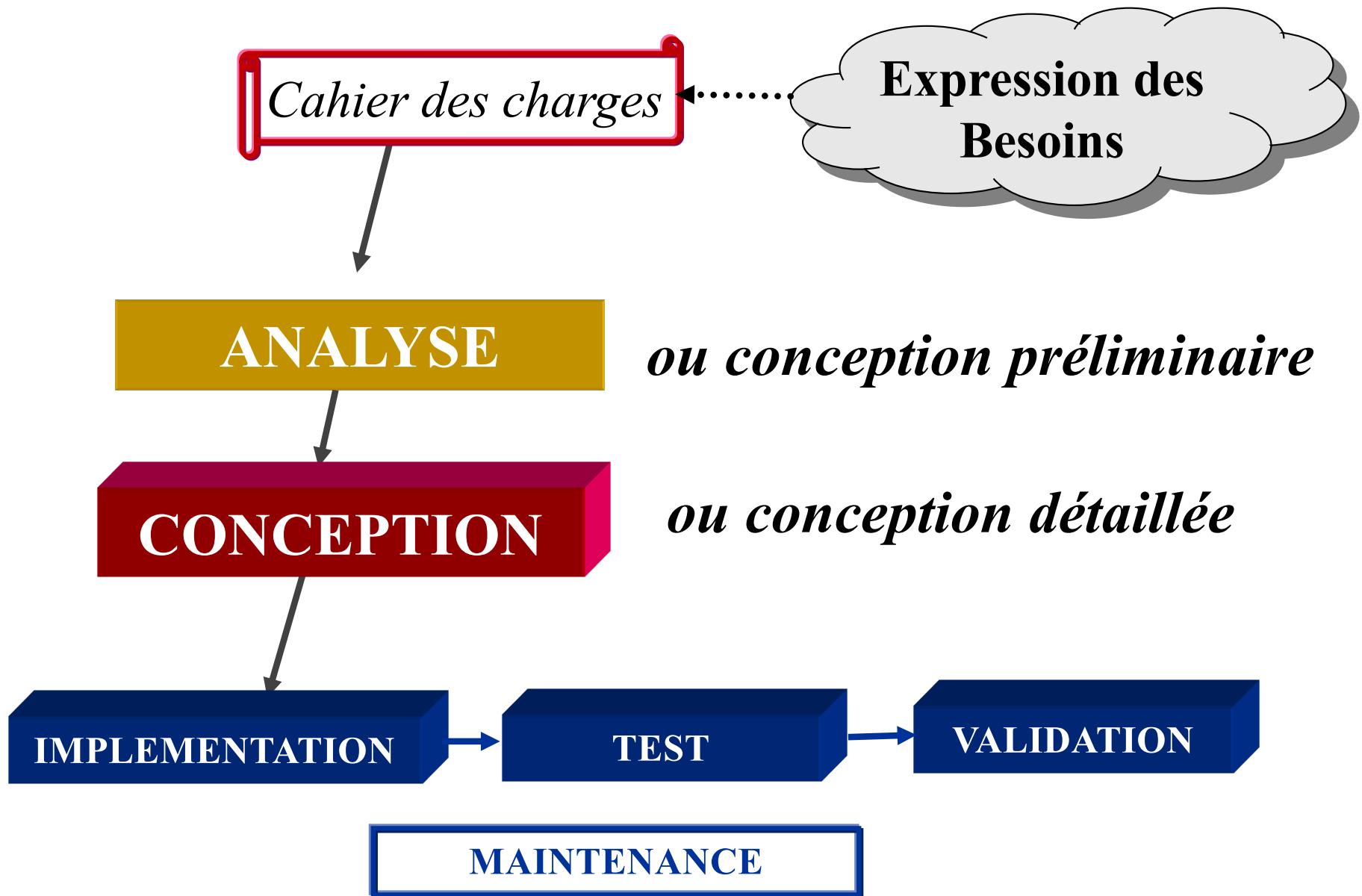
IMPLEMENTATION

Phase de réalisation
→ **programmation**





3 - Etapes d'analyse et de Conception





3 - Etapes d'analyse et de Conception

Etapes de développement et Modèles

Méthode MERISE

ETAPES

ANALYSE

MODELES

MCD Modèle Conceptuel des Données

MCT Modèle Conceptuel des Traitements

CONCEPTION

MLD Modèle Logique des Données

MOT Modèle Organisationnel des Traitements



3 - Etapes d'analyse et de Conception

Etapes de développement et Modèles

Méthodes basées sur le formalisme UML

ETAPES

ANALYSE

CONCEPTION

MODELES

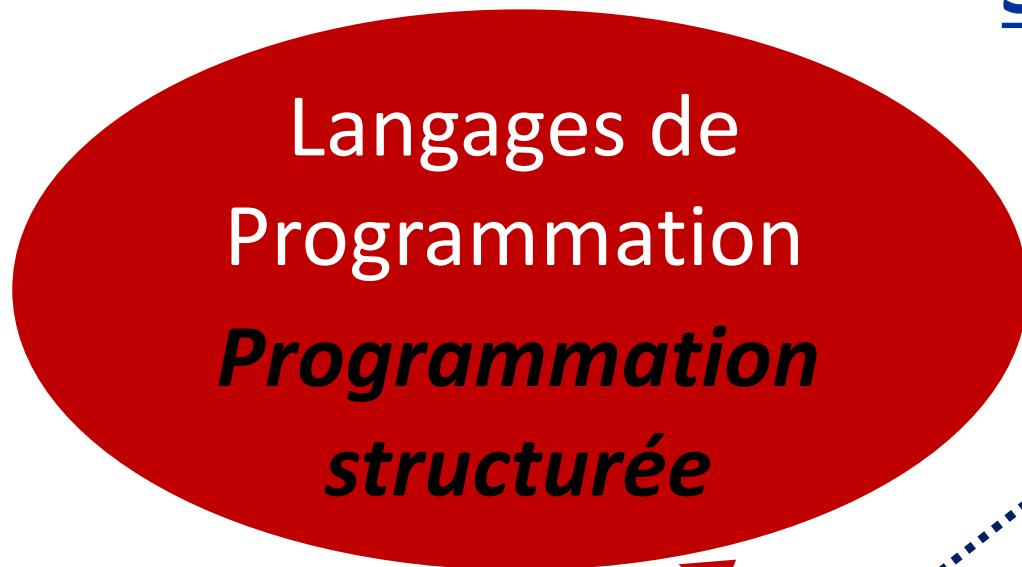
Les mêmes outils
(diagrammes) sont utilisés
avec des niveaux de détails
différents



4 - Historique des méthodes d'A & C

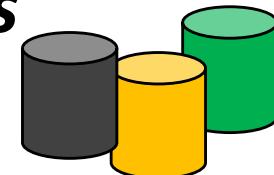
Avant l'objet

Années 60 : L'approche structurée



Stockage des données

Fichiers



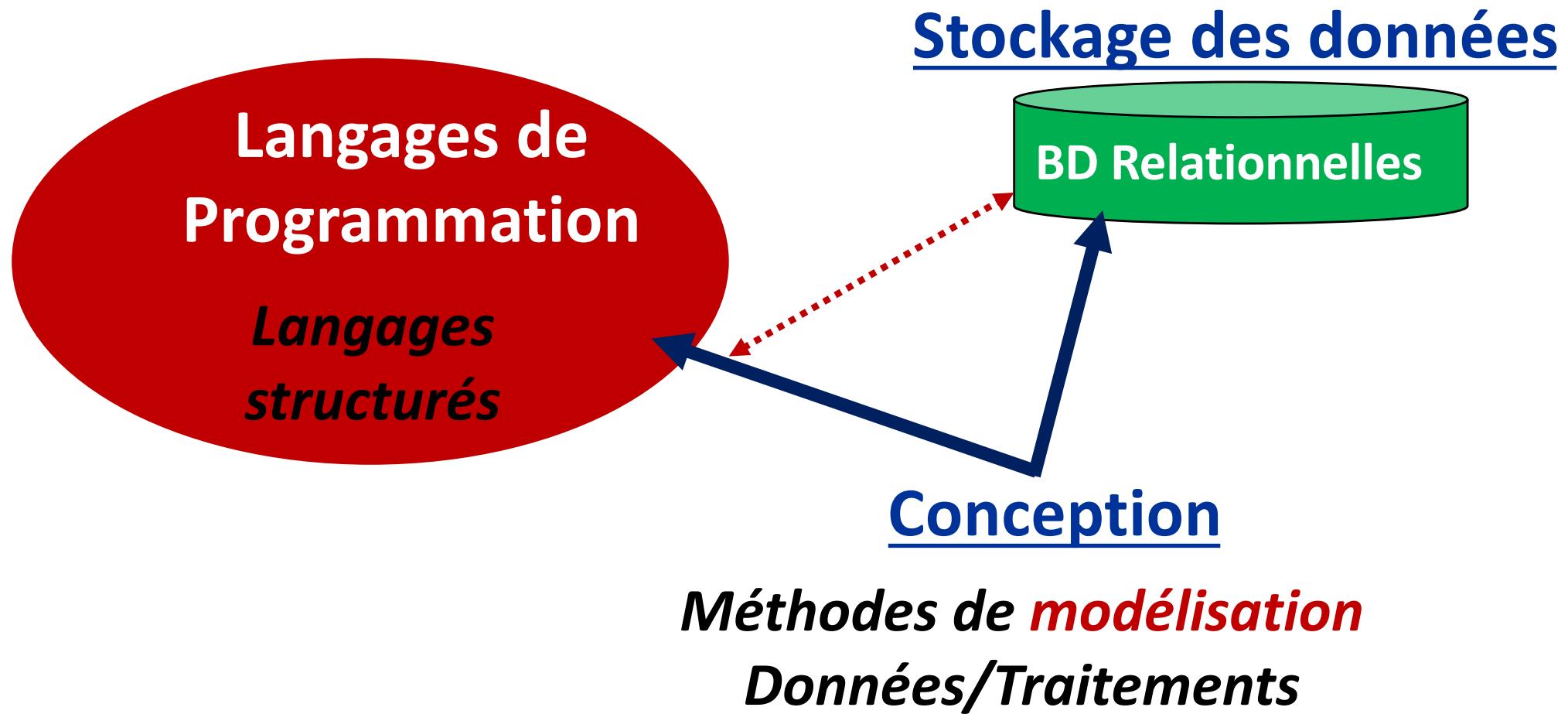
Conception

Méthodes d'Analyse structurée

4 - Historique des méthodes d'A & C

Avant l'objet

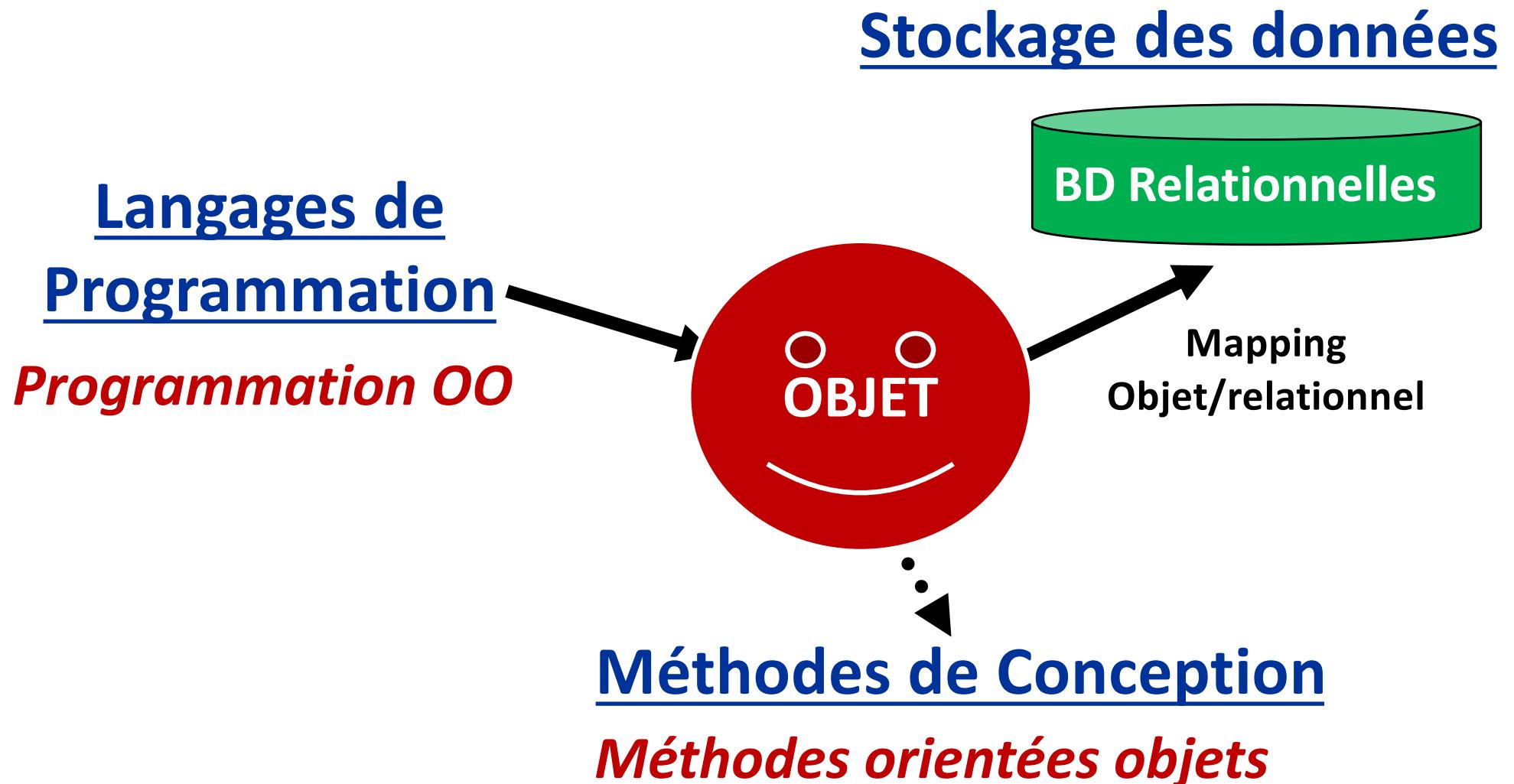
Années 70 :
L'approche Bases de Données





4 - Historique des méthodes d'A & C

Depuis 1980 : l'ère de l'Objet





4 - Historique des méthodes d'A & C

Apparition des méthodes OO

1980-91
OOD
Object Oriented Design
G. Booch

1987
HOOD
Hierarchical Object Oriented Design

1988-92
OOA
Object Oriented Analysis
S. Shlaer – S.J. Mellors

1991
OMT
Object Modeling Technique
J. Rumbaugh

1991
OOA/OOD
Object Oriented Analysis/Design
Coad - Yourdon

1992
OOSE
Object Oriented Software Engineering
I. Jacobson

... . . .



4 - Historique des méthodes d'A & C

1988 – 1992 Nombreuses méthodes de modélisation 00

- très proches sur le fond : Concepts orientés objets
- très différentes sur la forme
 - Terminologie
 - Représentations graphiques
- très spécifiques au niveau des démarches

Problèmes d'intéropérabilité!



4 - Historique de l'A/COO

Recherche d'un formalisme standard



*Nous ne nous comprenons pas!!
Pourtant, nous devons collaborer!!*

vers UML ...



4 - Historique des méthodes d'A & C

Recherche d'un formalisme standard



Object Management Group

Organisme à but non lucratif créé en 1989

- **Initiative:** HP, Sun, Unisys, American Airlines, Philips...
- **Mission :**
 - **Essor des technologies objet**
 - **Promotion de standards**
 - **Garantie de l'interopérabilité**



4 - Historique des méthodes d'A & C

Recherche d'un formalisme standard

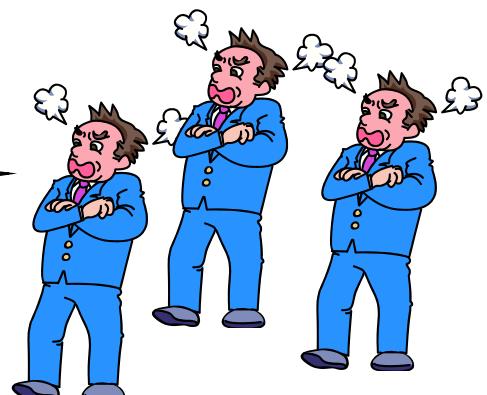


Nous allons proposer un standard

Lettre ouverte de protestation



NON !!!!!!
Jamais !!!!



Méthodologistes



4 - Historique des méthodes d'A & C

Recherche d'un formalisme standard

Rational. software



Grady BOOCH
OOD



James RUMBAUGH
OMT

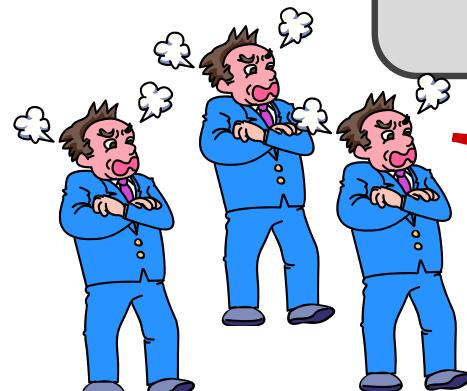
General Electric



James RUMBAUGH
OMT



La guerre des méthodes est terminée.
Nous l'avons gagnée!!



Méthodologistes

*Il faut résister!!
Formons une coalition
anti-Booch!!*



2 - Historique d'UML

Recherche d'un formalisme standard

Rational. software



Grady BOOCH



James RUMBAUGH

Conférence OOPSLA'95

Version 0.8 de la
« METHODE UNIFIEE »



Ivar JACOBSON
OOSE





4 - Historique des méthodes d'A &C versions d'UML

VERSION	ADOPTION DATE
2.5.1	December 2017
2.4.1	July 2011
2.3	May 2010
2.2	January 2009
2.1.2	October 2007
2.0	July 2005
1.5	March 2003
1.4	September 2001
1.3	February 2000
1.2	July 1999
1.1	December 1997

UML 2.5.1 : Version actuelle

Prise en compte de toutes les étapes du cycle de vie du logiciel jusqu'à la spécification de tests

- Nouveaux diagrammes adaptés aux architectures communicantes distribuées
- Amélioration des mécanismes de profils (extensions d'UML)
- Support de l'approche MDA
- Mécanisme d'échange de diagrammes XMI (*XMI = XML Metadata Interchange*)



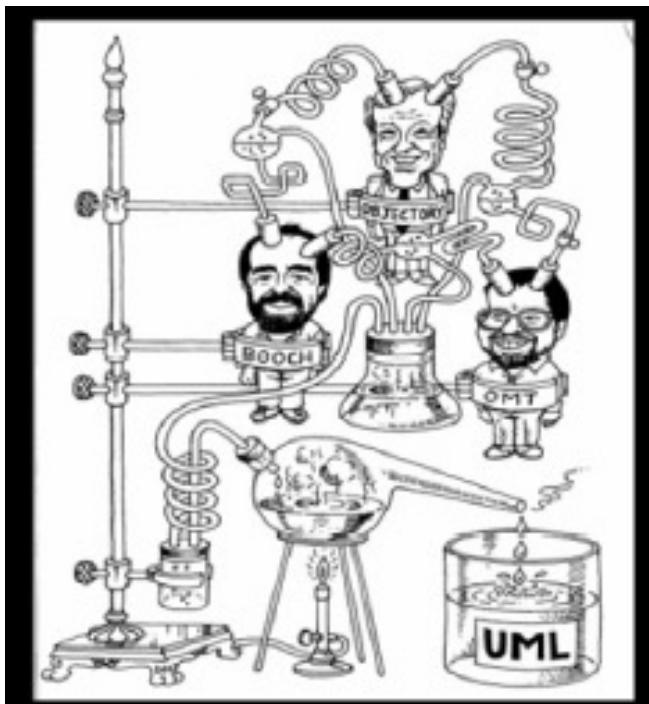
1^{ère} version Standard Officiel OMG 1.1



4 - Historique des méthodes d'A & C

Historique d'UML : En résumé

Les « pères » fondateurs :



*Les 3 amigos : BOOCH - RUMBAUGH
- JACOBSON*

Les « bâtisseurs » :





CH 1 – UML et ACOO

1.1 – Notions d'Analyse et de Conception



1.2 – Le Formalisme UML (survol)

1. Objectifs
2. Modes d'utilisation
3. Définition formelle
4. La « Boîte à outils » UML
5. Outils et références



1.3 – Démarche d'ACOO avec UML



Qu'est ce qu'UML?



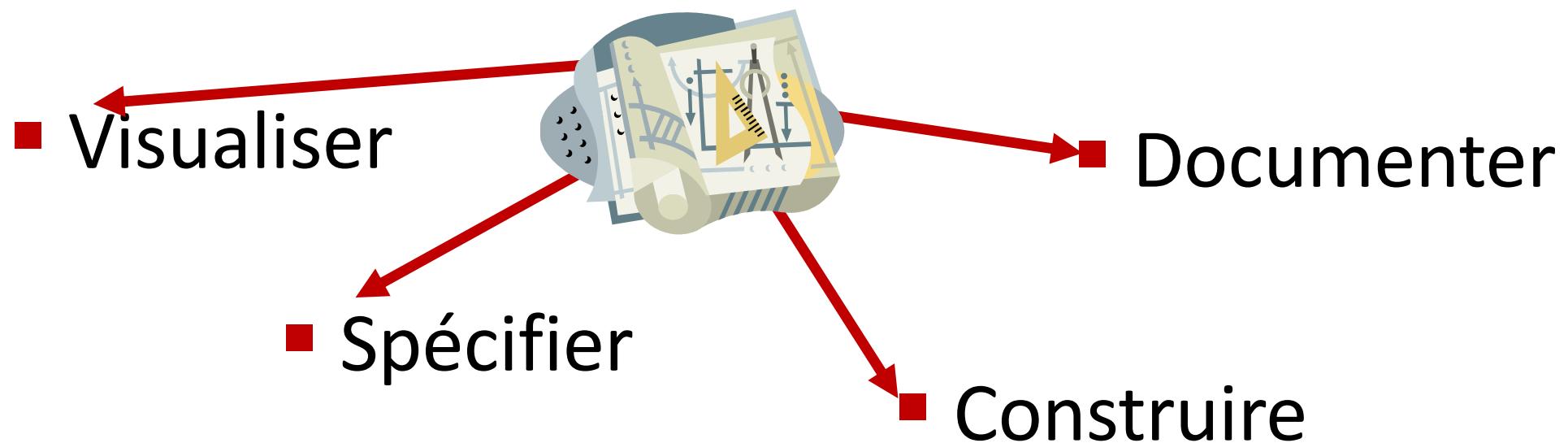


Le Formalisme UML

1- Objectifs

Unified Modeling Language
Langage unifié pour la modélisation

Langage **standard** pour l'écriture
de plans d'élaboration de logiciels.





1 - Objectifs du formalisme UML

- Outil de Communication
 - **Indépendant** de la méthode
 - **Indépendant** des langages de programmation
 - Adapté à **toutes les phases** du développement
 - Compatible avec **toutes les techniques** de réalisation
- Notation **graphique** Simple, Générale et Flexible



Pour en savoir plus: <http://www.uml.org/>



WATCH VIDEO



BPM+
Business Process Management
Plus



CORBA®
Common Object Request Broker
Architecture™



DDS™
Data-Distribution Service for
Real-Time Systems™



FIGI®
Financial Instrument Global
Identifier®



IEF™
The Information Exchange
Framework™



MOF™
MetaObject Facility
Specification™



SYSML®
Systems Modeling Language™



UAF®
Unified Architecture
Framework®



UML®
Unified Modeling Language™

*UML helps People thinking
in different way!*



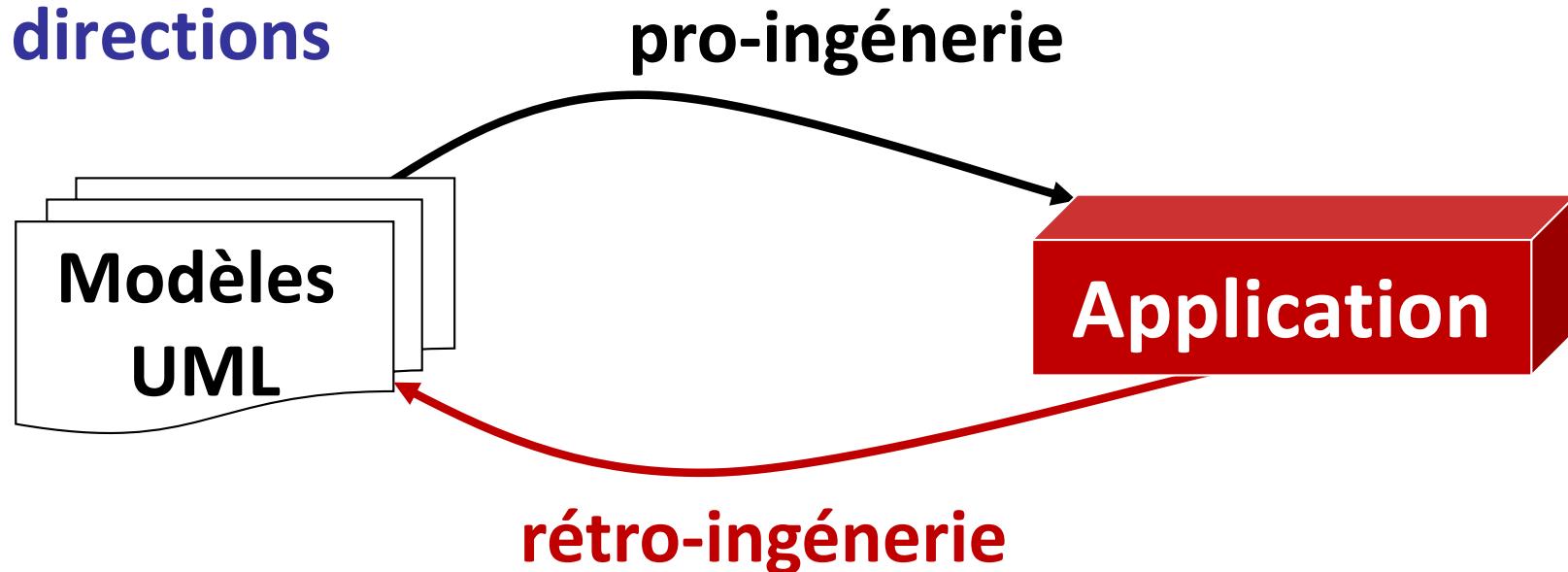
Quand et comment utiliser UML?





2 - Modes d'utilisation d'UML

Deux directions



Trois modes d'utilisation



- Esquisse
- Plan
- Langage de programmation





3 – Définition formelle

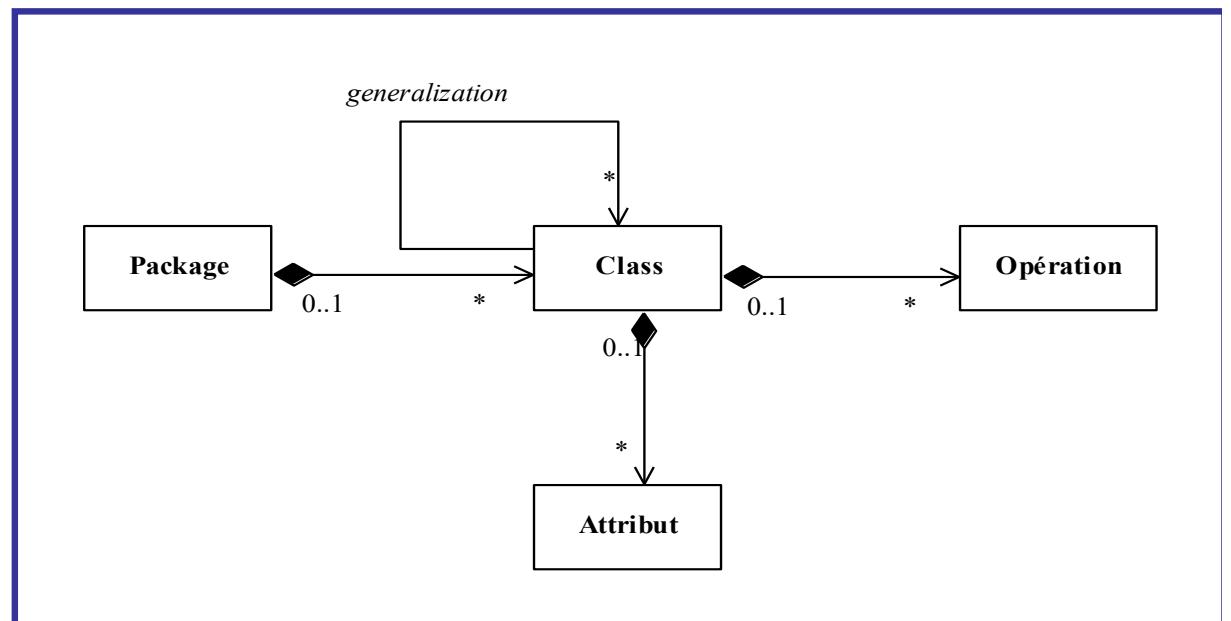
Meta-modèle d'UML

Identification des concepts fondamentaux

- Définition de la sémantique de ces concepts
- Choix d'une représentation graphique

Métamodélisation d'UML avec UML

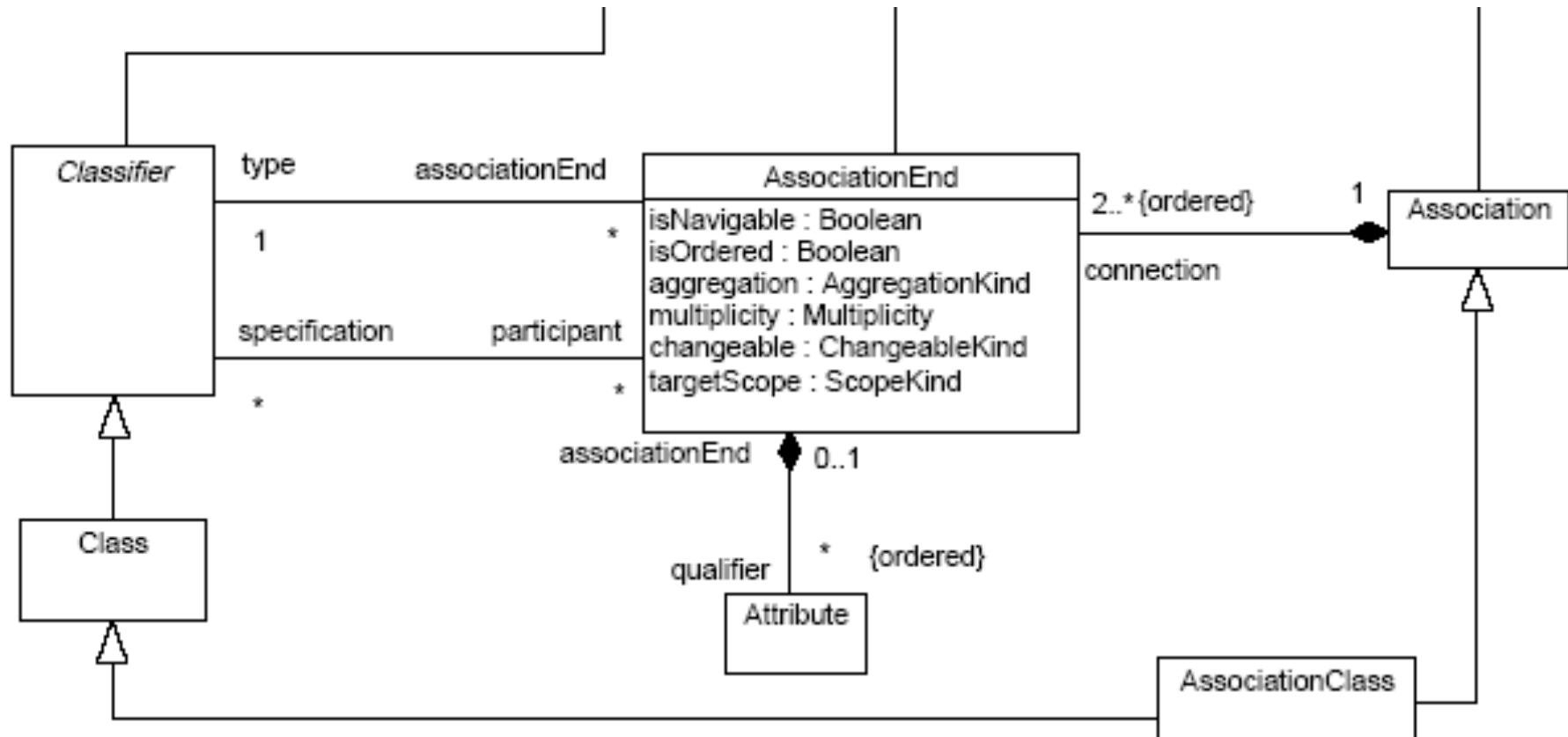
- Description formelle des éléments de modélisation





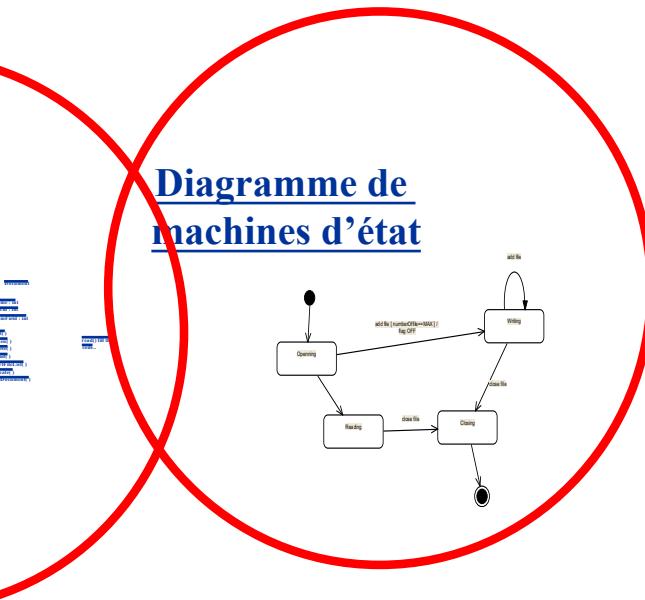
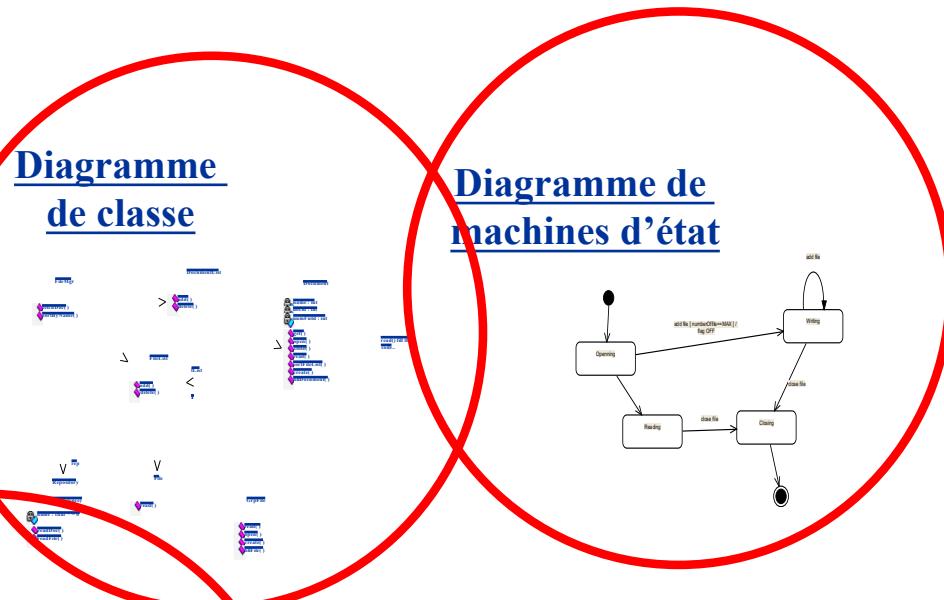
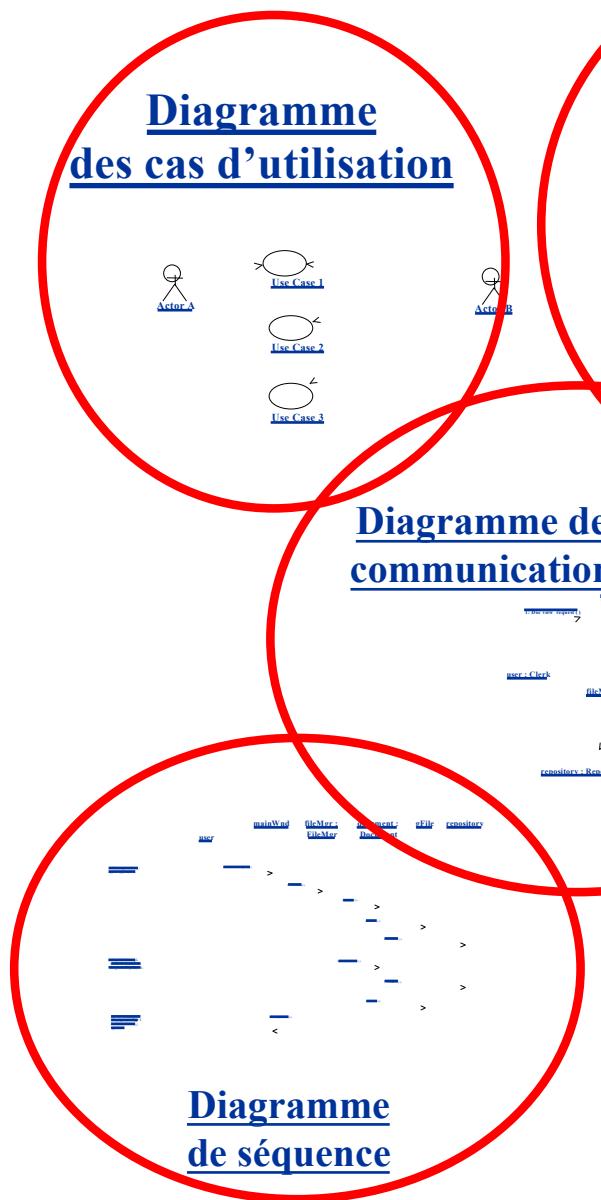
3 – Définition formelle

Une « Portion » du métamodèle UML





4 – La « boîte à outils » UML2



Une boîte à outils de
13 Diagrammes



4 – La « boite à outils » UML: les Diagrammes

Un diagramme est :

- une projection visuelle d'un aspect du système.
- un graphe composé de :
 - *Sommets* : **éléments de modélisation**
Classes , états, composants, ...
 - *Arcs* : relations décrivant les **liens entre éléments**
Dépendance, association, généralisation, réalisation

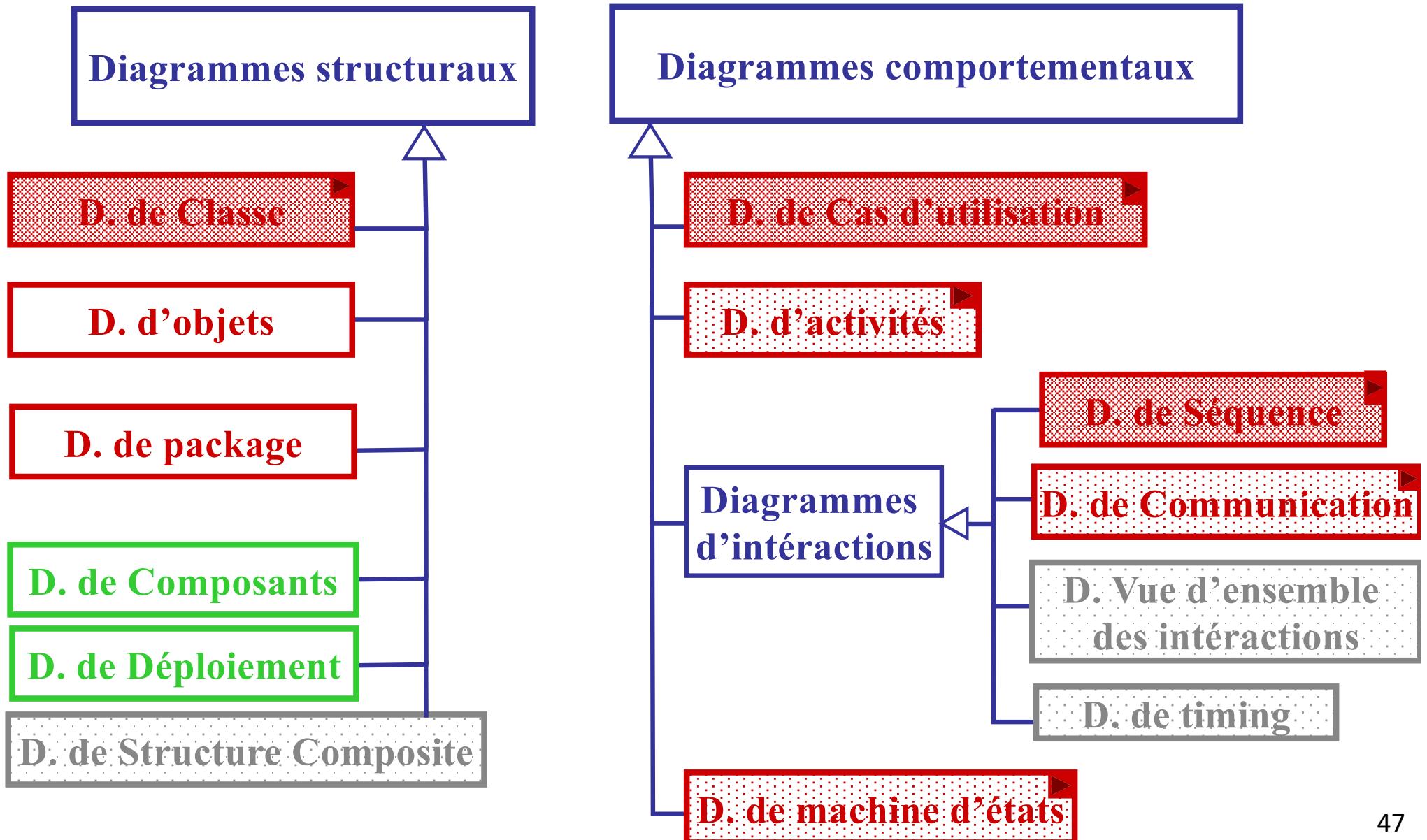


Quels sont les principaux diagrammes UML?





4 – La « boîte à outils » UML: les Diagrammes UML2





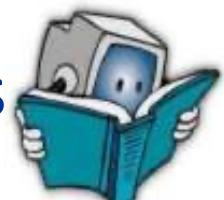
CH 1 – UML et ACOO

1.1 – Notions d'Analyse et de Conception

1.2 – Le Formalisme UML (survol)

1.3 – Démarche d'ACOO avec UML

1. Formalisme UML et démarche (processus)
2. Caractéristiques des Processus unifiés
3. Une démarche pratique





1 – Formalisme UML et démarche

Depuis 1997 – Le langage UML s'impose comme le formalisme standard de la modélisation orientée objet.



Mais UML n'est qu'un langage de modélisation pas une méthode!!



Une boîte à outils
sans mode d'emploi!!!

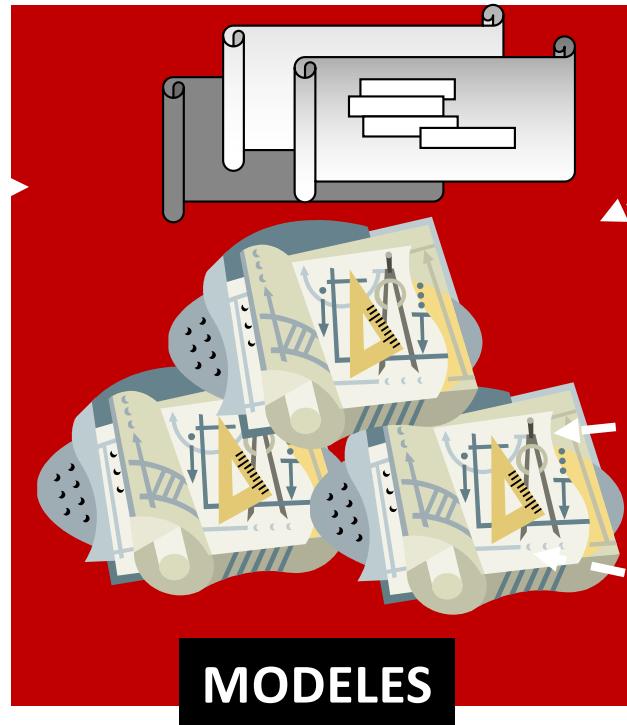


1 – Formalisme UML et démarche

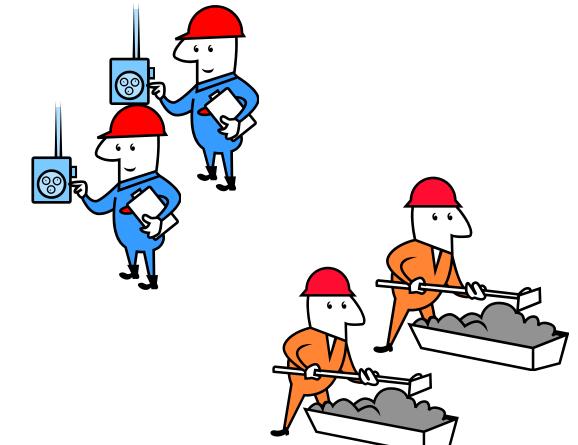
Le formalisme ne suffit pas



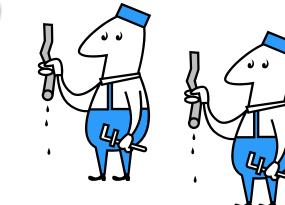
??



??



??



Problèmes à résoudre:

- Comment procéder pour construire les plans (modèles)?
- Comment utiliser les plans pour construire?
- Qui doit intervenir, Quand, Comment dans quel ordre??





Existe-t-il une
méthode d'ACOO
standard?

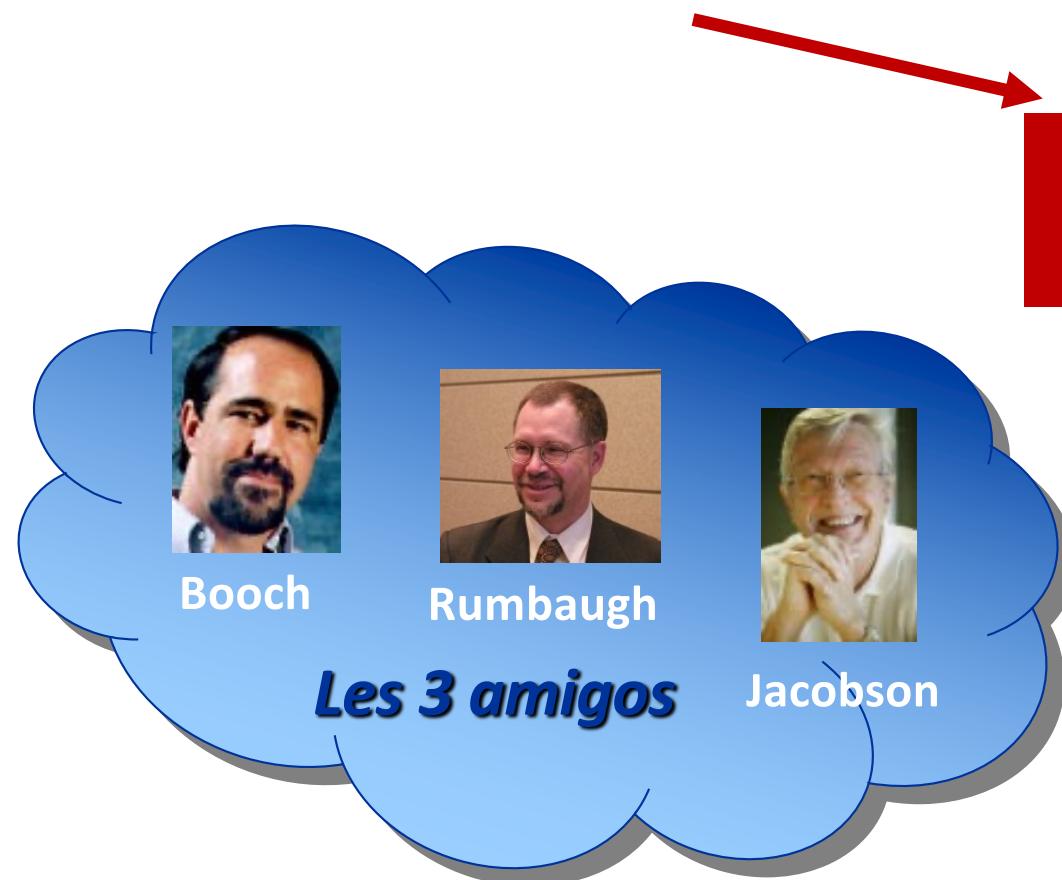




1 – Formalisme UML et démarche

Définir une seule et unique démarche est irréaliste!

Solution: un processus générique



**le processus Unifié
(Unified Process)**

« trame commune »
des processus associés
à UML et à l'approche
orientée objet.



1 – Formalisme UML et démarche

La famille du Processus Unifié

UP n'est pas un processus mais un cadre général

Garantie des meilleures pratiques (best practices) de développement orienté objet avec UML.

Versions industrielles

RUP : Rational Unified Process
(IBM Rational)

2TUP: 2 Track Unified Process
(Valtech)



2 – Caractéristiques des Processus Unifiés

Caractéristiques des Processus (ou démarches) de la famille UP

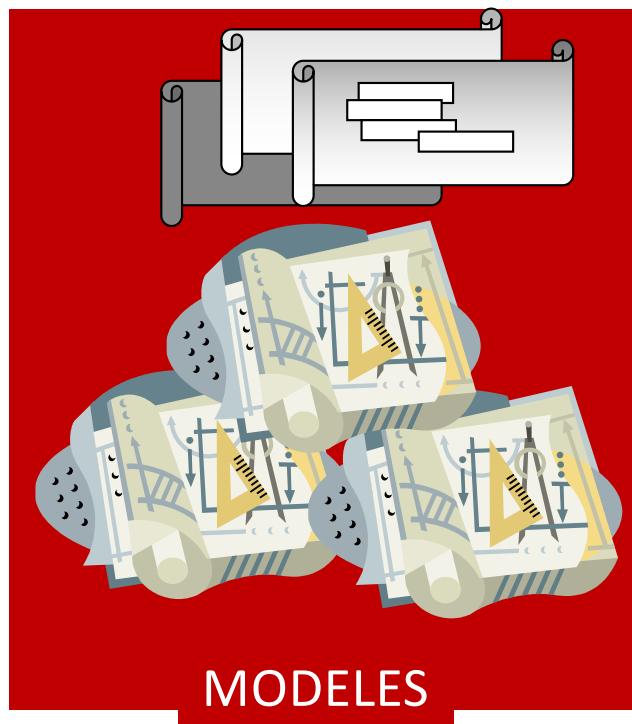
- Rôle central des **modèles**
- **Vocabulaire** commun
- Des **principes** fondamentaux: les Piliers du UP



2 – Caractéristiques des Processus Unifiés

Rôle central des modèles

Les processus de la famille UP sont construits autour de la création et la maintenance de modèles.



Formalisme standard
de modélisation:
UML

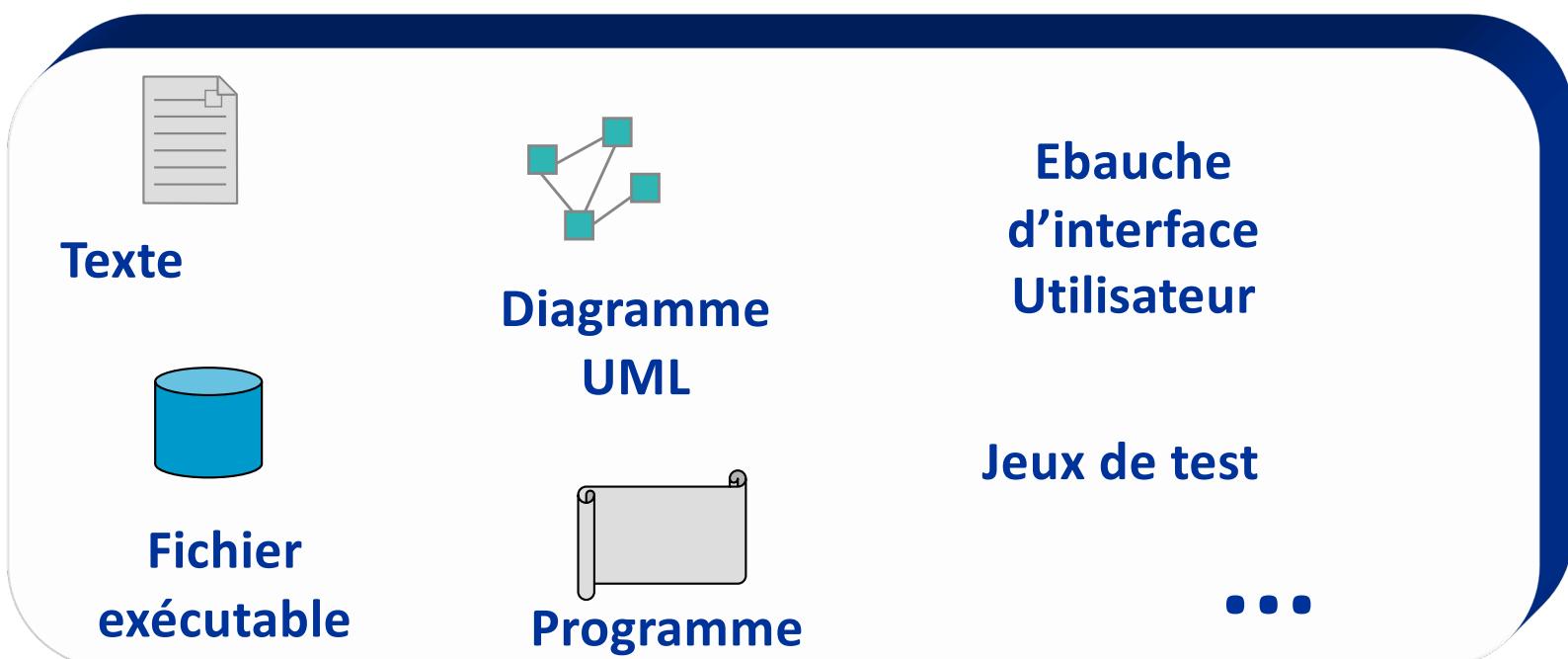




2 – Caractéristiques des Processus Unifiés

Vocabulaire

Artefact = toute sorte d'information créée, modifiée ou utilisée au cours de la mise au point du système





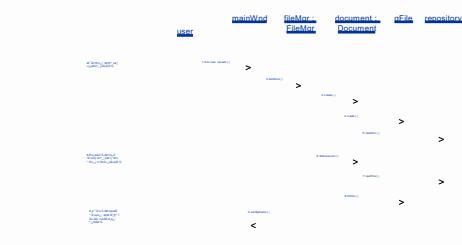
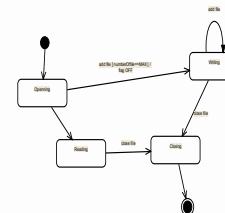
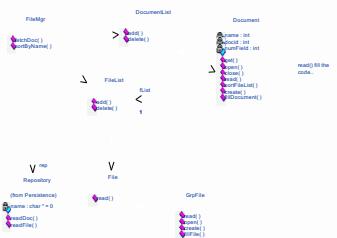
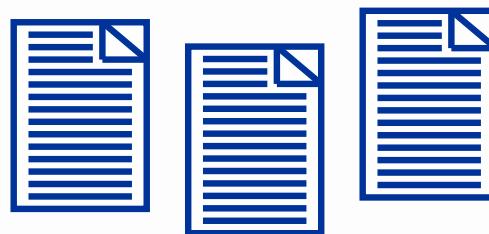
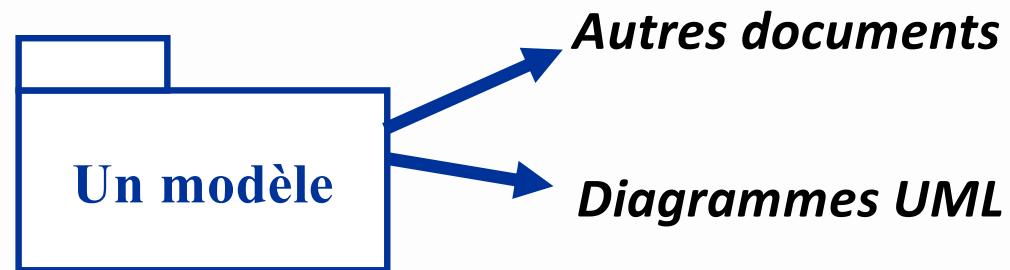
2 – Caractéristiques des Processus Unifiés

Vocabulaire

Modèles =

Les modèles sont les artefacts principaux du processus.

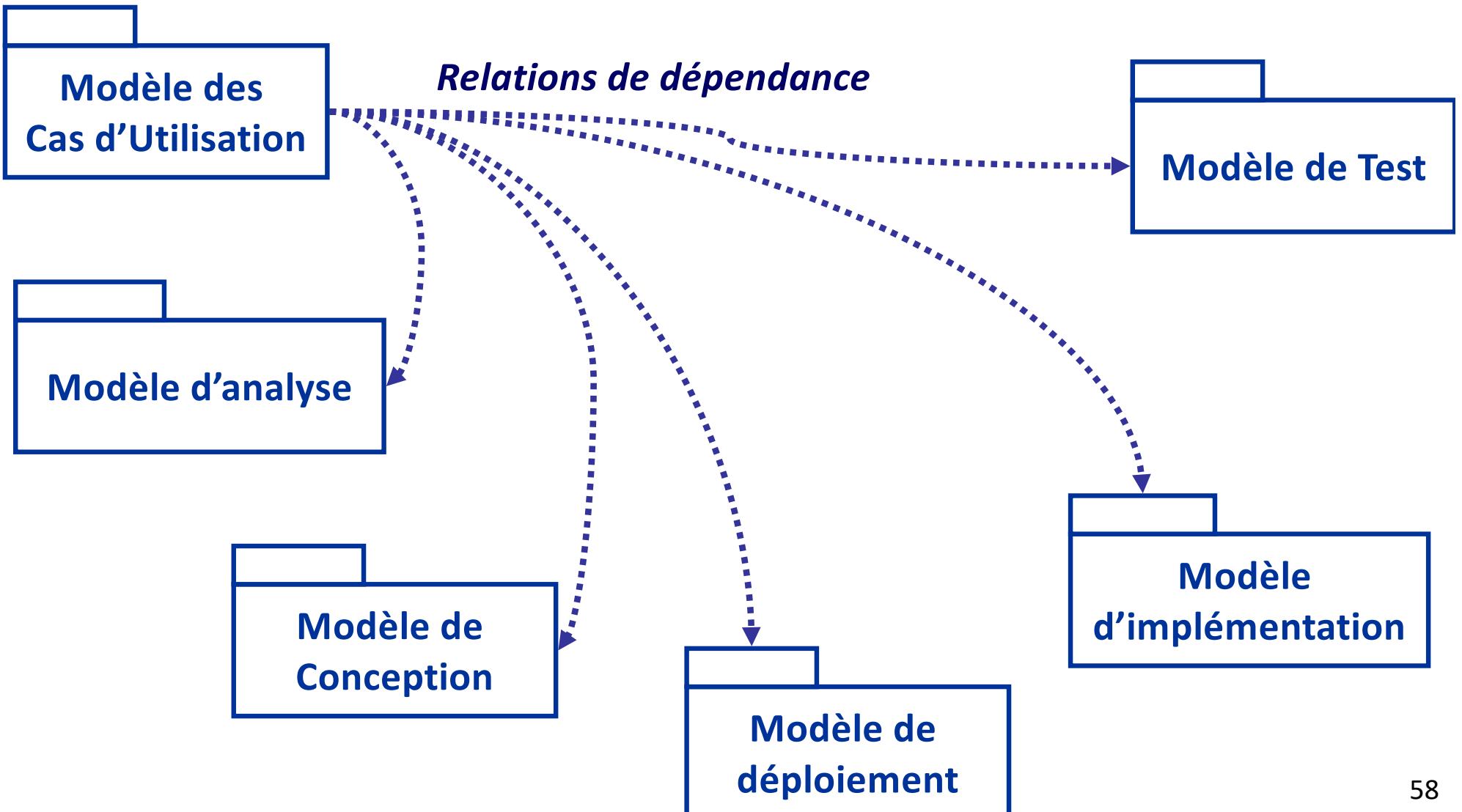
Ils sont eux mêmes décrits par un **ensemble d'artefacts**.





2 – Caractéristiques des Processus Unifiés

Exemples de Modèles





2 – Caractéristiques des Processus Unifiés

Les Processus unifiés proposent une démarche:

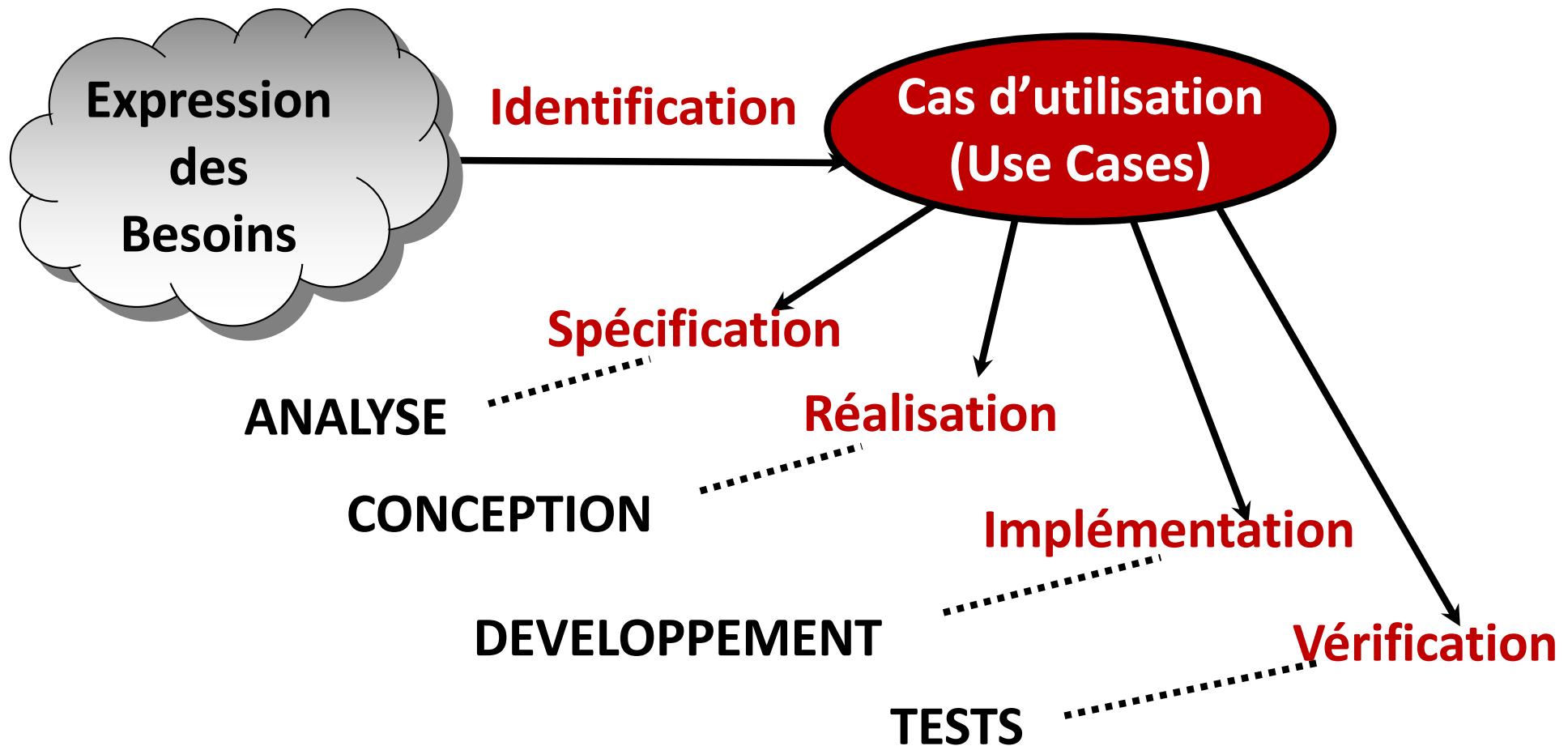
guidée par les **besoins des
utilisateurs**

itérative et incrémentale



2 – Caractéristiques des Processus Unifiés

Processus guidé par les cas d'utilisation



Les besoins des utilisateurs sont au centre du processus.



2 – Caractéristiques des Processus Unifiés

Processus Itératif et incrémental

Le projet est **découpé** en mini-projets réalisés l'un après l'autre.



Chaque mini-projet conduit à une **version** opérationnelle qui vient s'ajouter au système global.



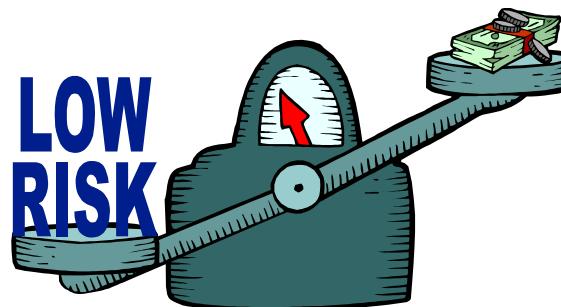
Maison témoin



2 – Caractéristiques des Processus Unifiés

Processus Itératif et incrémental

Objectif Principal : Réduire les Risques



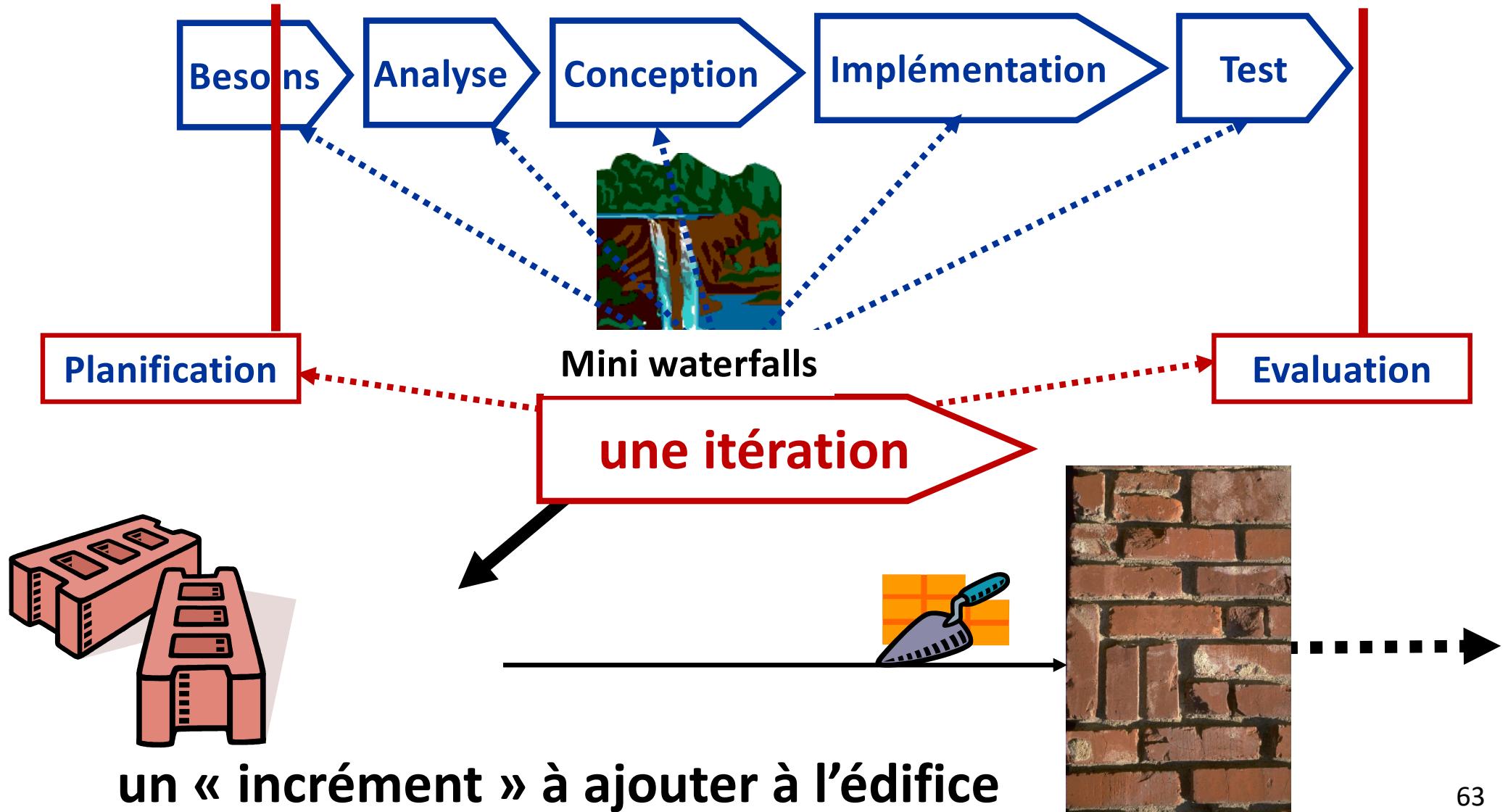
Eviter le «cauchemar » du schéma de développement WaterFalls!!

Cf : http://dchaffiol.free.fr/info/chefprj/art_chefprjprincNul_t.htm



2 – Caractéristiques des Processus Unifiés

Déroulement d'une itération





2 – Caractéristiques des Processus Unifiés

Une itération :

- correspond à la réalisation **d'un ou plusieurs cas d'utilisation**
- produit **une version exécutable** (« *release* ») qui peut être montrée au client.

Les modèles sont construits de manière itérative et incrémentale par enrichissement progressif.



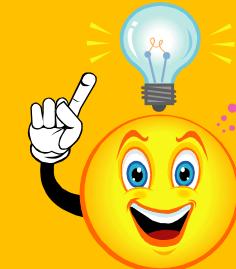
Vers les méthodes agiles

- Limites des UP « classiques »
 - Cadre rigide (voire bureaucratique!!)
 - Documentation très riche (mais lourde!!)
 - Adapté aux projets de très grande envergure mais souvent peu efficace

La solution: L'AGILITE

- ✓ Respecter les piliers du UP
- ✓ Mais alléger, alléger ...

Se débarrasser des contraintes pour se focaliser sur l'essentiel

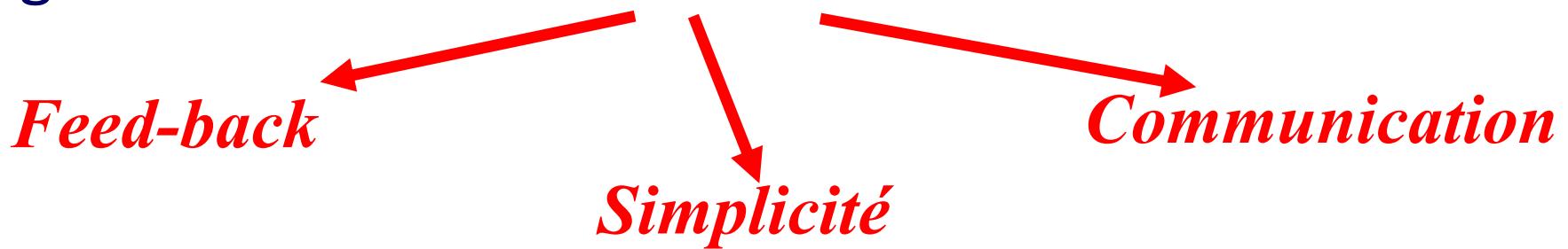




Vers les méthodes agiles

Principe général des méthodes agiles

Processus itératifs et incrémentaux qui constituent des versions « allégées » des UP centrées sur l'activité de programmation.



Ensemble de règles pratiques concrètes

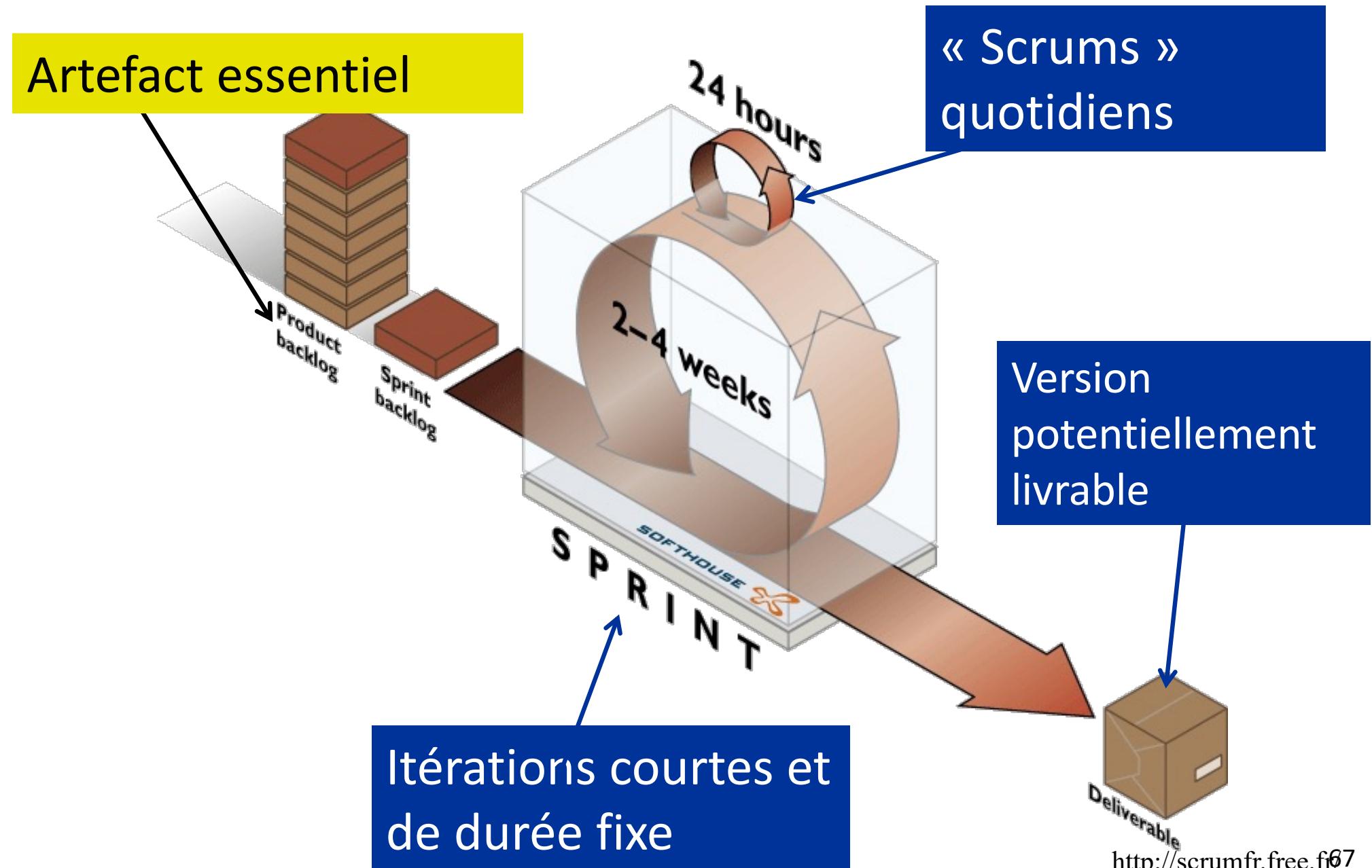
- Organisation du travail (binomes, ...)
- Transferts de compétence
- Élaboration des tests avant le codage
- ...

Exemples

- XP: eXtreme programming
- SCRUM
- Open UP



Un exemple SCRUM: Principe général





Vers les méthodes agiles

- La modélisation joue néanmoins un rôle fondamental dans l'agilité
 - Approche pragmatique
 - Efficacité avant tout!!
 - Ne pas modéliser pour modéliser mais modéliser lorsque cela est utile
 - Limiter la documentation au maximum
 - Utiliser UML en mode esquisse



Dans le cadre d'une cours, nous allons étudier une démarche pratique. Quelles sont ses étapes?

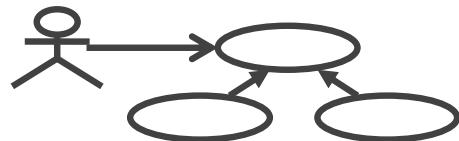




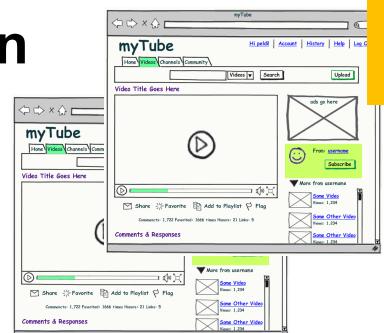
3 – Une démarche pratique

Spécifications
des exigences

Quoi?



Modèle des Cas
d'Utilisation

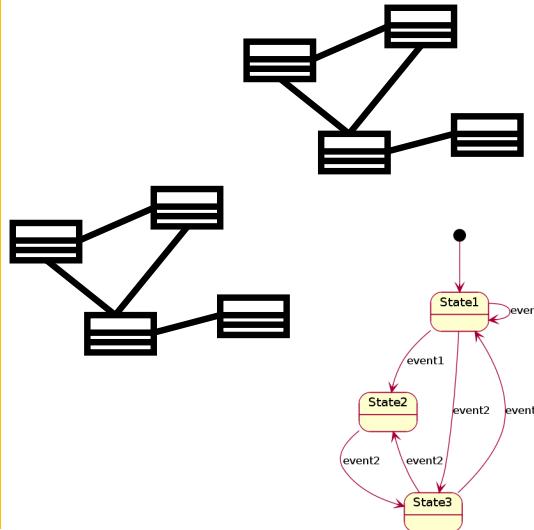


Maquette d'IHM

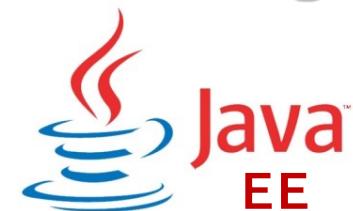
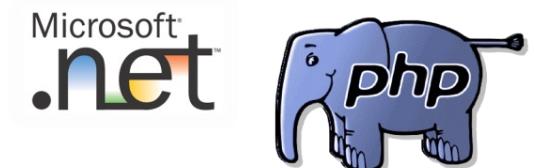
Analyse

Conception

Comment?

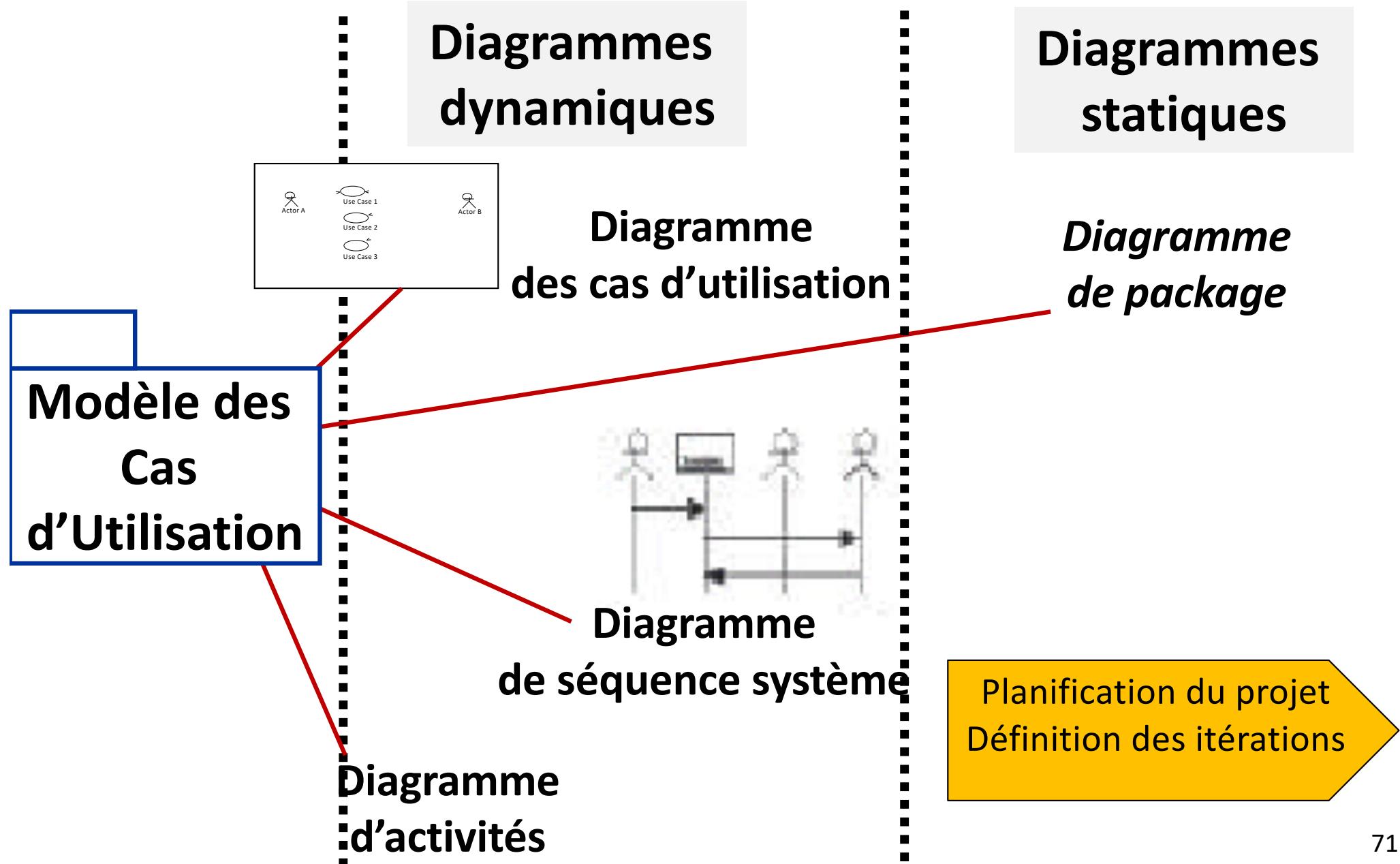


Modèle
d'analyse





3 – Une démarche pratique

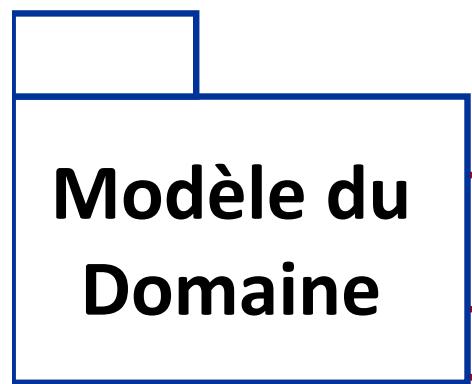




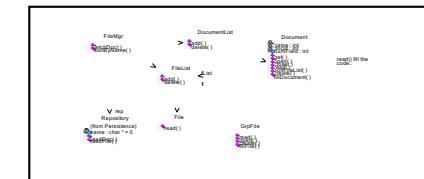
3 – Une démarche pratique

Diagrammes
dynamiques

Diagrammes
statiques



*Diagramme
de package*



*Diagramme
de classes du domaine*

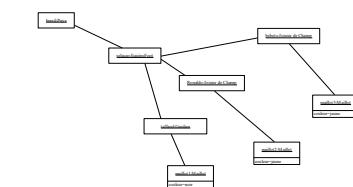


Diagramme d'objets



3 – Une démarche pratique

Analyse

Diagrammes
dynamiques

Diagrammes
statiques

Diagramme
d'activités

*Diagramme
de package*

Diagramme
de machine d'état

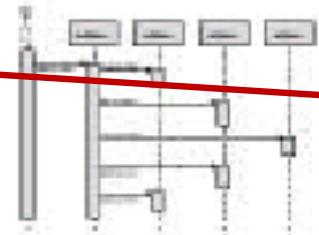
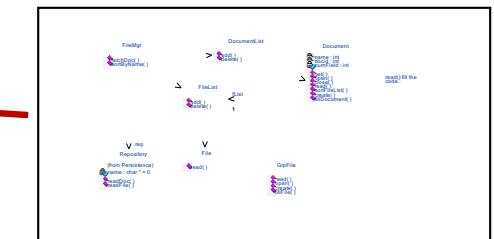


Diagramme
de séquence d'analyse



Modèle
d'Analyse

Diagramme
de communication

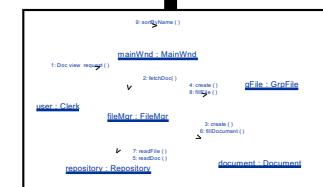


Diagramme
de classes d'analyse



3 – Une démarche pratique

Conception

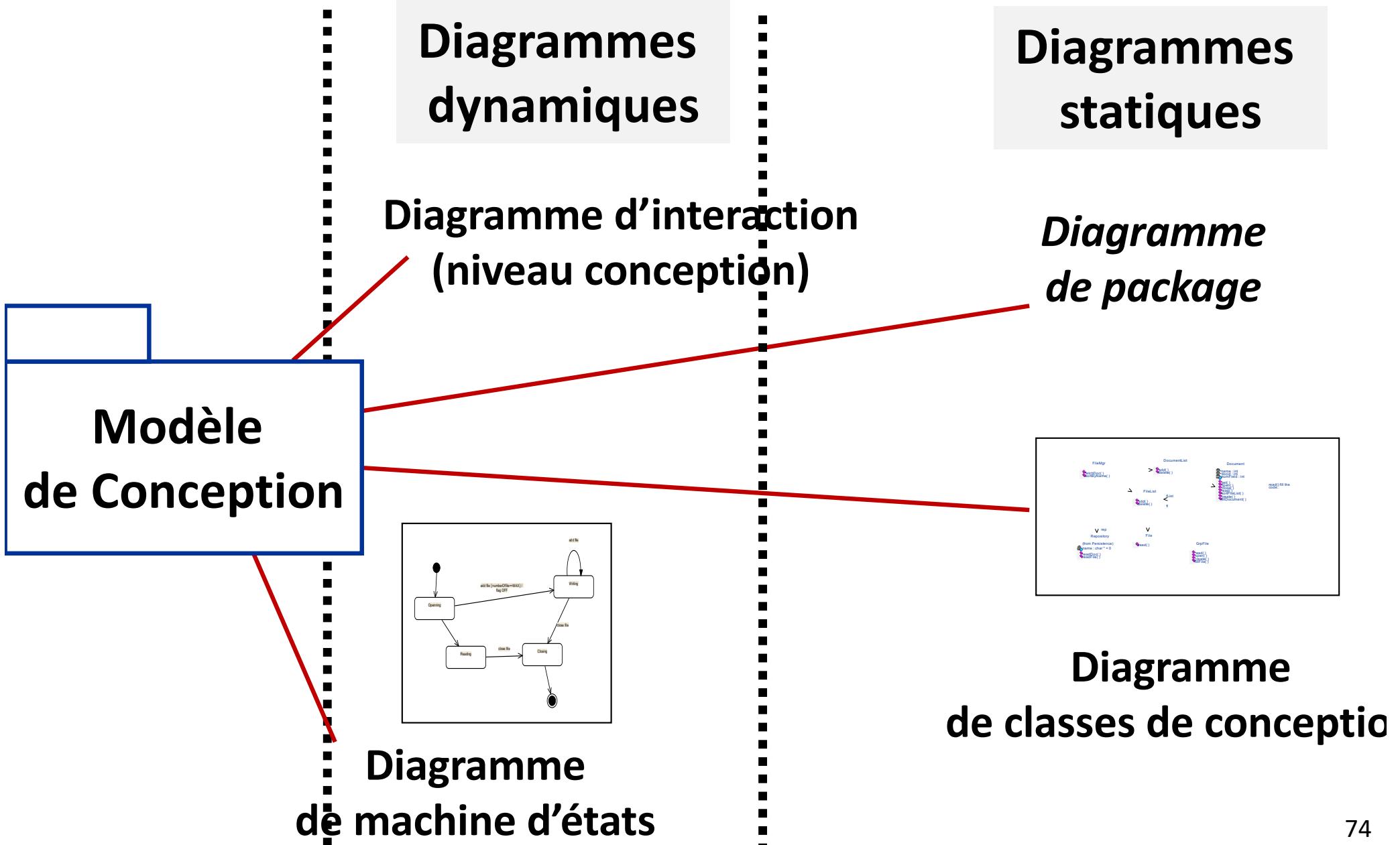




Diagramme de classes

Pour représenter
la structure
statique d'un
système

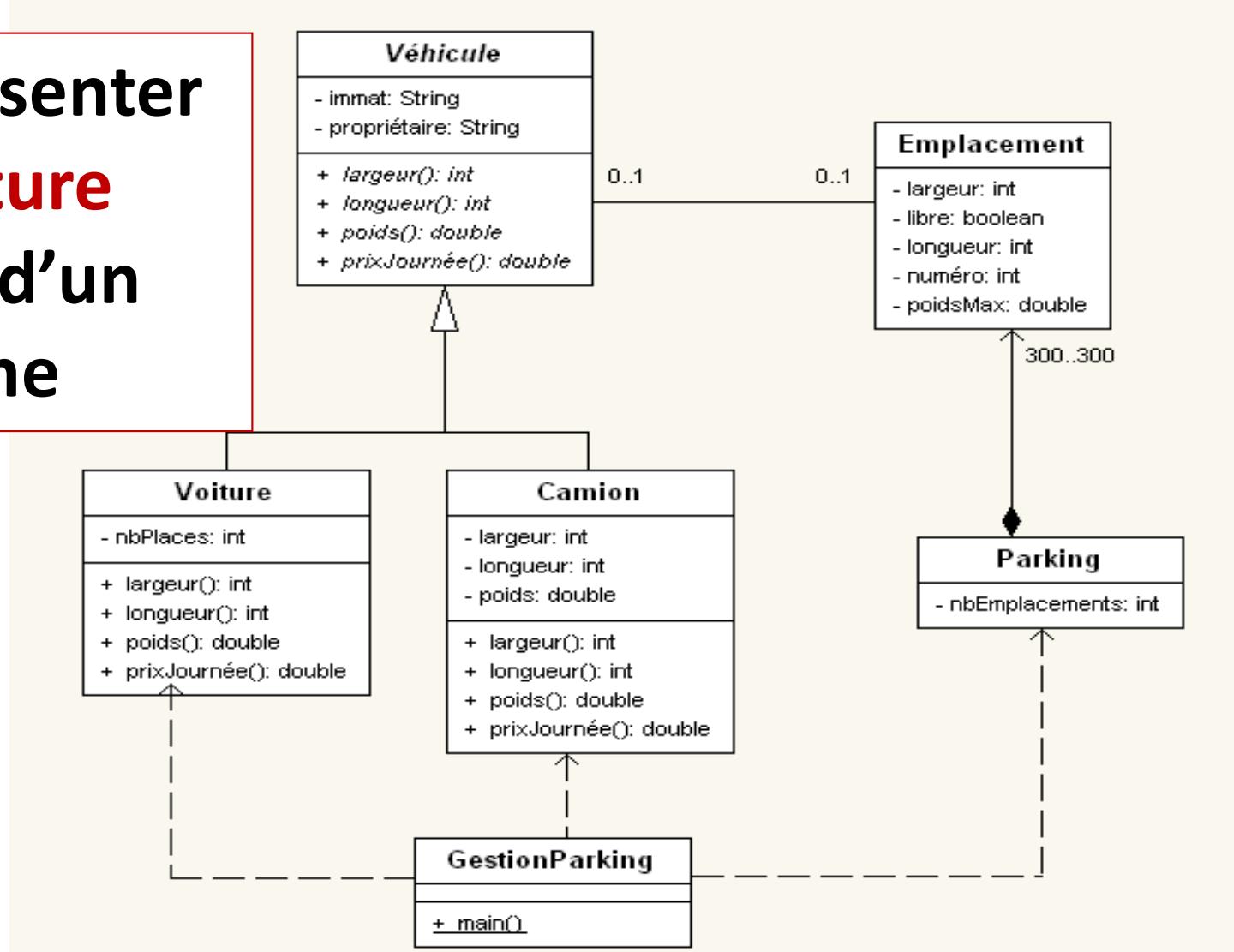
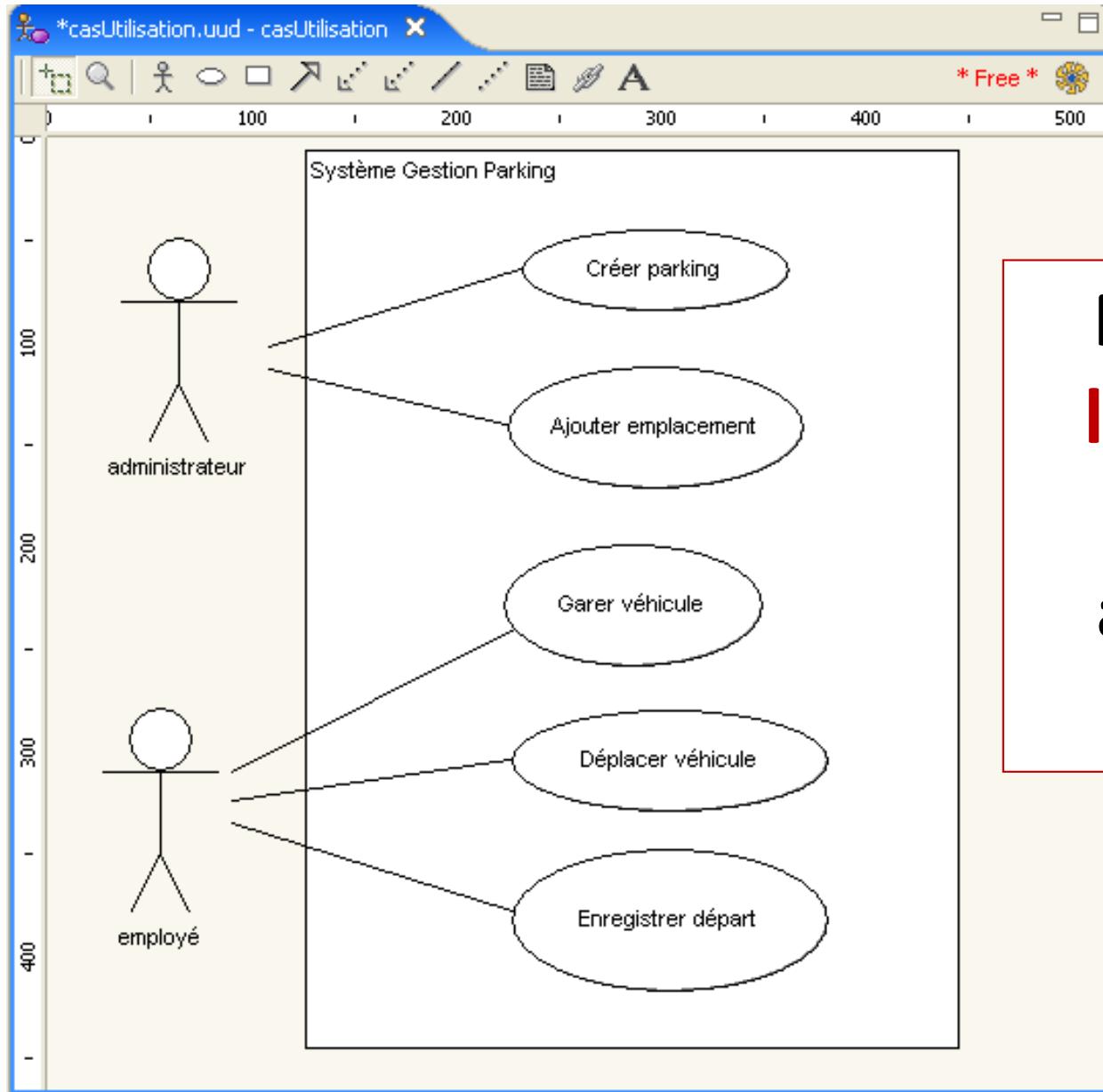




Diagramme de cas d'utilisation



**Pour visualiser
l'interaction de
votre système
avec le monde
extérieur**

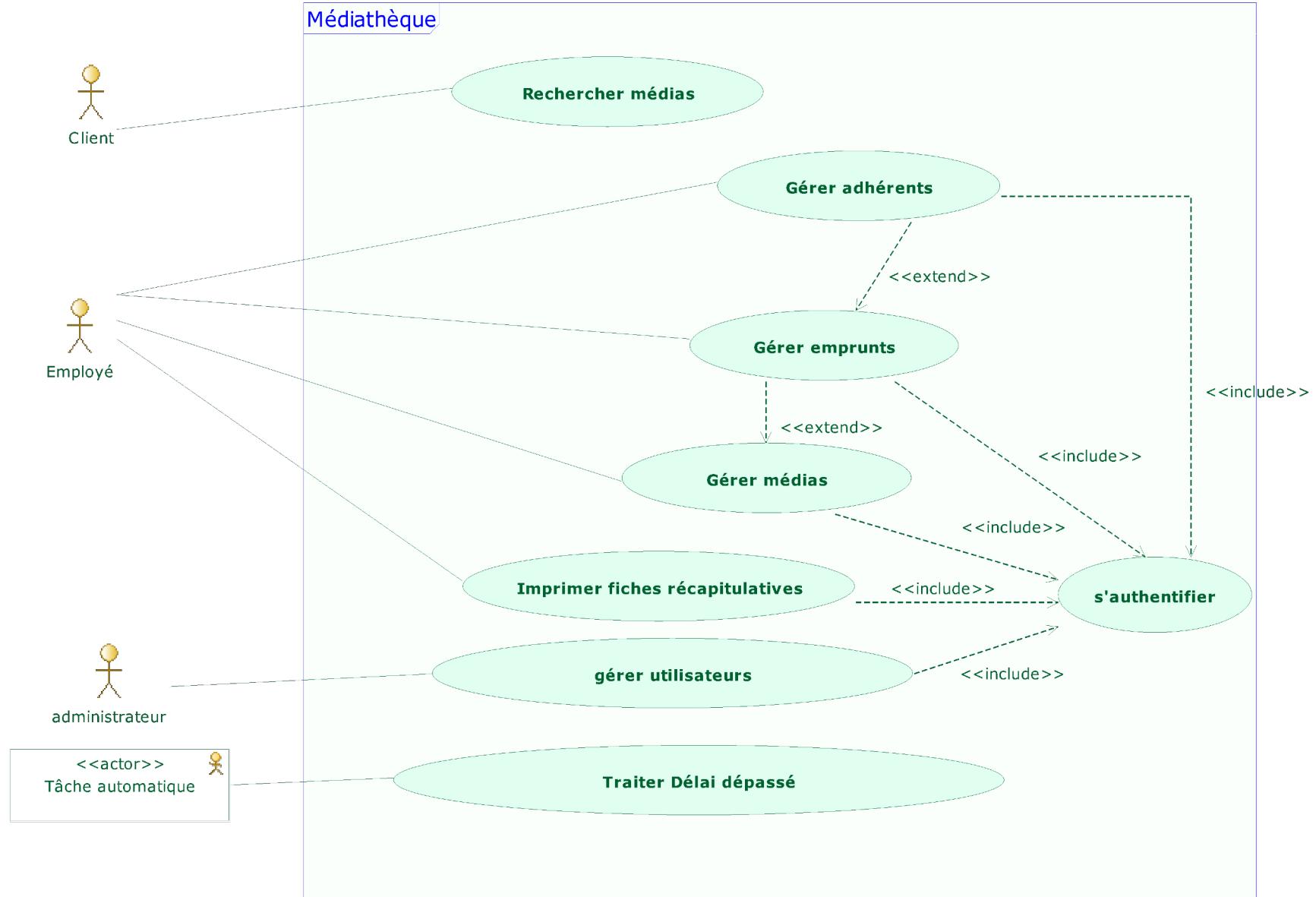




Diagramme d'activités

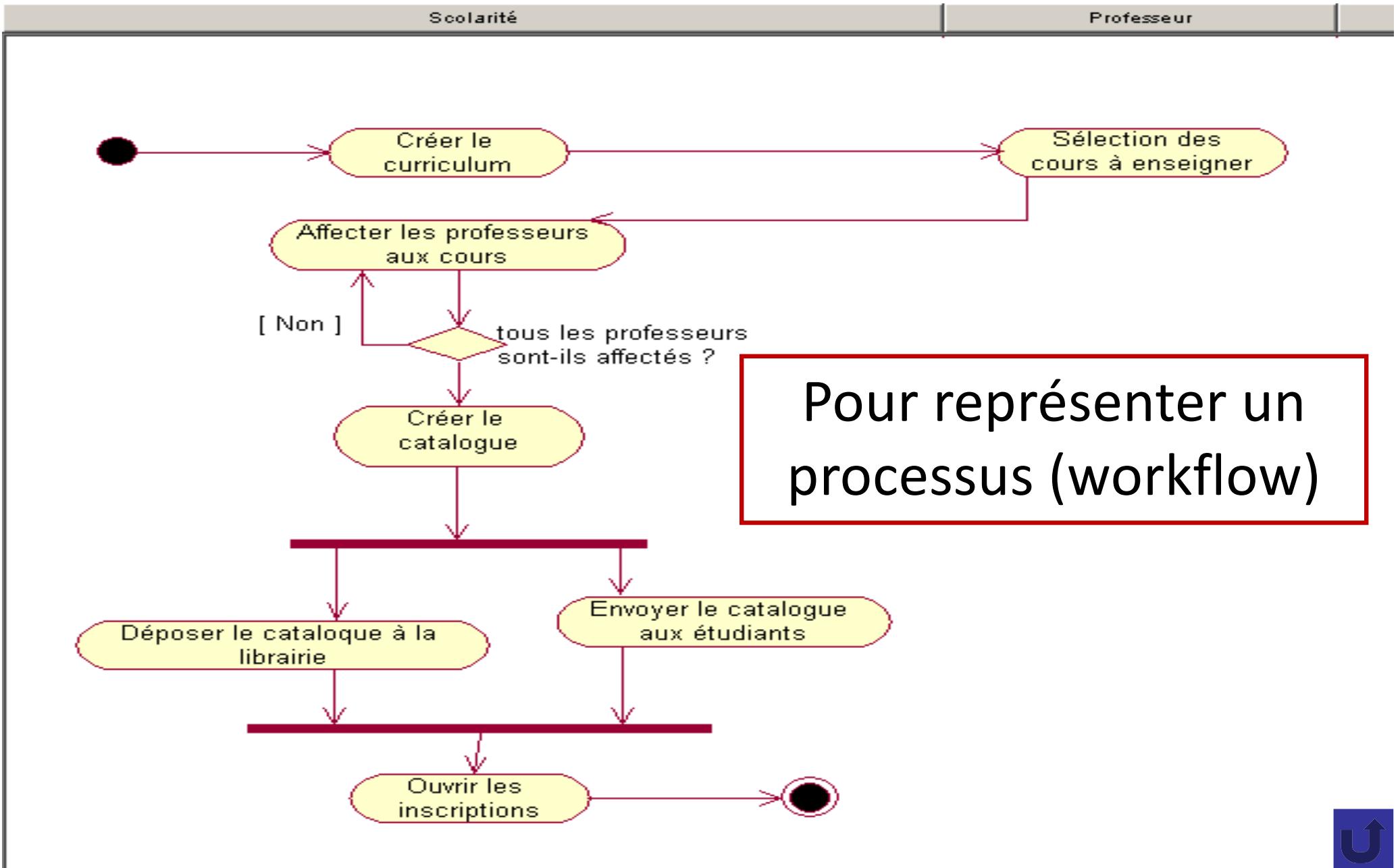




Diagramme de séquence

Pour montrer comment est réalisée une fonctionnalité du système d'un point de vue temporel

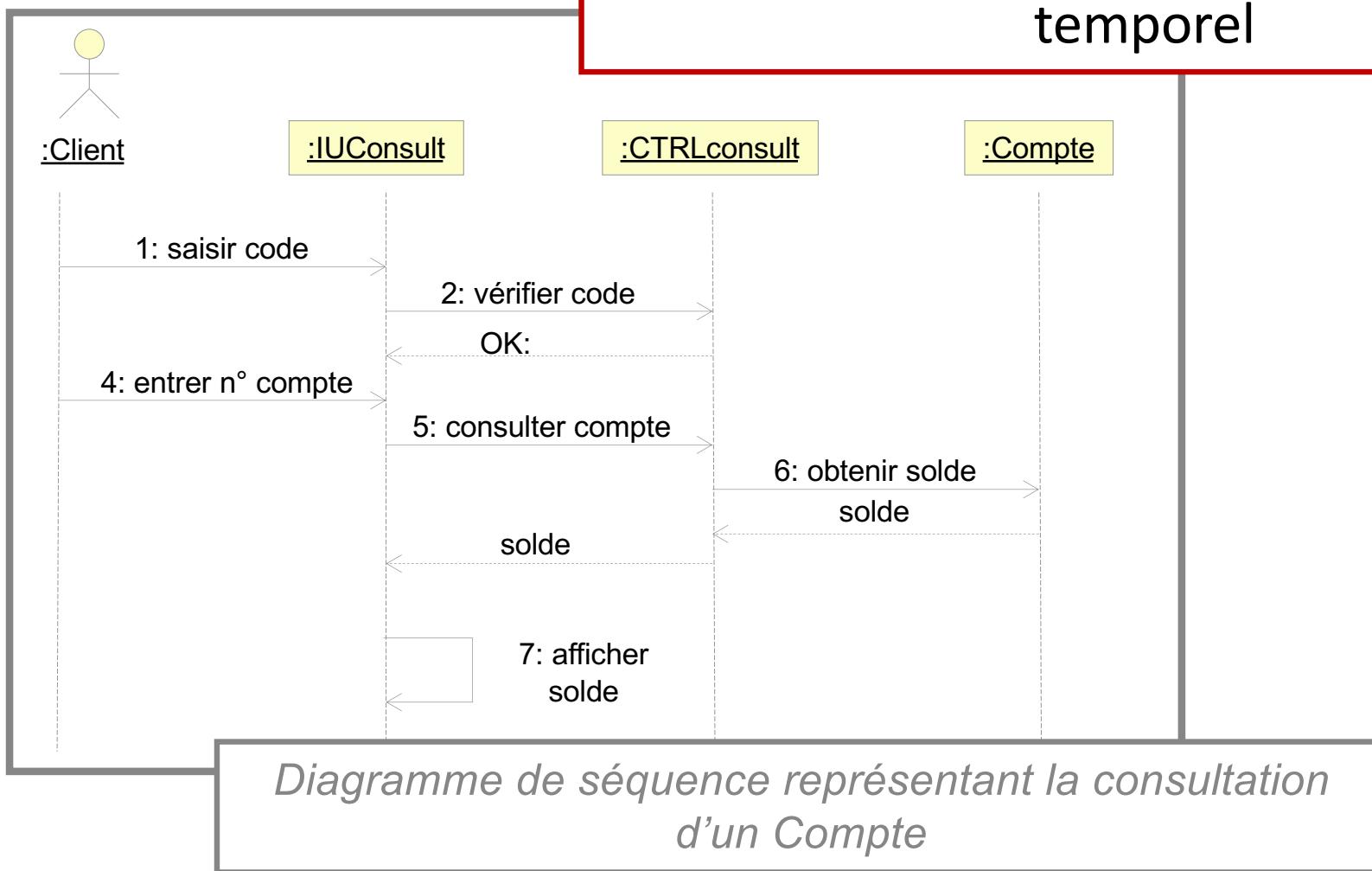




Diagramme de Communication

L'équivalent d'un diagramme de séquence d'un point de vue temporel

Collaboration en
UML1.xx

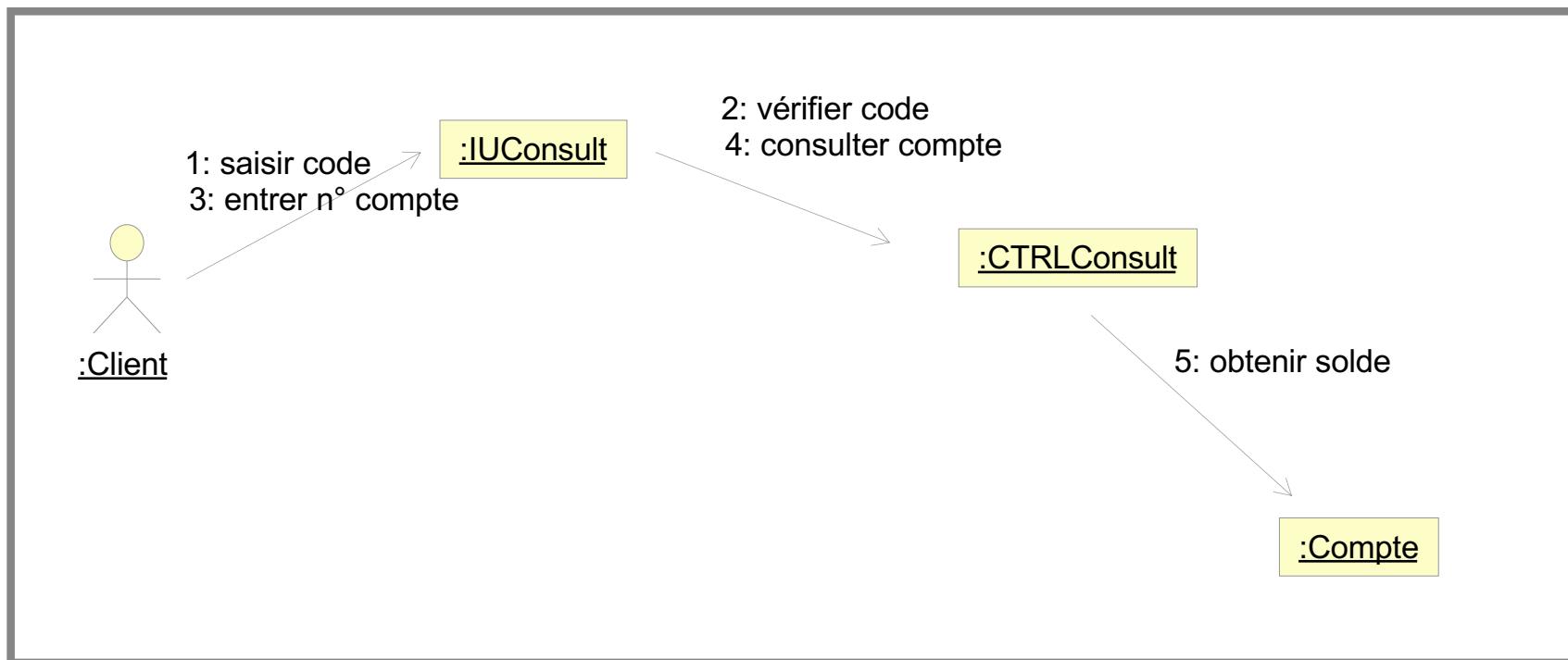




Diagramme de machine d'état

Pour montrer le cycle de vie
d'une classe

Etat-Transition
en UML1.xx

Exemple: classe Personne

