

Introduction

à la Modélisation par Objets :

UML à la rescousse !



Isabelle BLASQUEZ
@iblasquez

Mars 2019



Isabelle BLASQUEZ



[@iblasquez](https://twitter.com/iblasquez)

Enseignement : Génie Logiciel

Recherche : Développement logiciel agile



ICSTUG #IUTAgile



CodeWeek. 

**#Software
Craftsmanship**

 **IUSEOMIX** LIMOUSIN

Sur une feuille blanche,

Expliquer à l'aide d'un dessin
comment vous faites (feriez)
griller votre pain le matin

(n'oubliez pas vos nom, prénom et groupe au verso de la feuille)



<http://e.ggtimer.com>

Vous avez 3 minutes



Bravo !

vous venez de réaliser votre premier modèle !



Quelques extraits de :
<http://www.drawtoast.com/gallery.html>

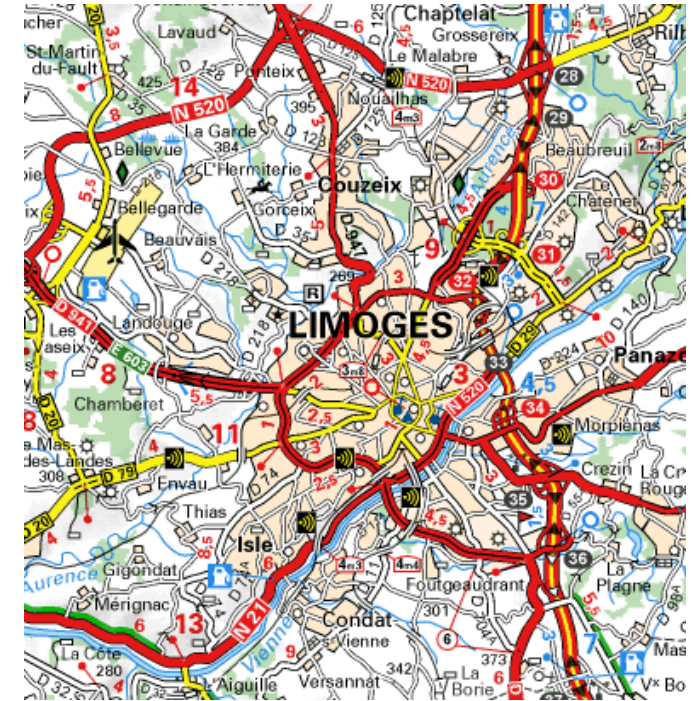
D'après une idée originale de Tom Wujec : <http://www.tomwujec.com/design-projects/draw-toast>

Modélisation

Qu'est-ce qu'un modèle ?



$$E=mc^2$$



<http://www.viamichelin.fr>

Un modèle est une **abstraction (représentation abstraite)** de la réalité

(Image **simplifiée** du monde réel **selon un point de vue**
suffisante pour **comprendre** le système modélisé
et **répondre** aux questions que l'on se pose sur lui)

Un modèle permet ...

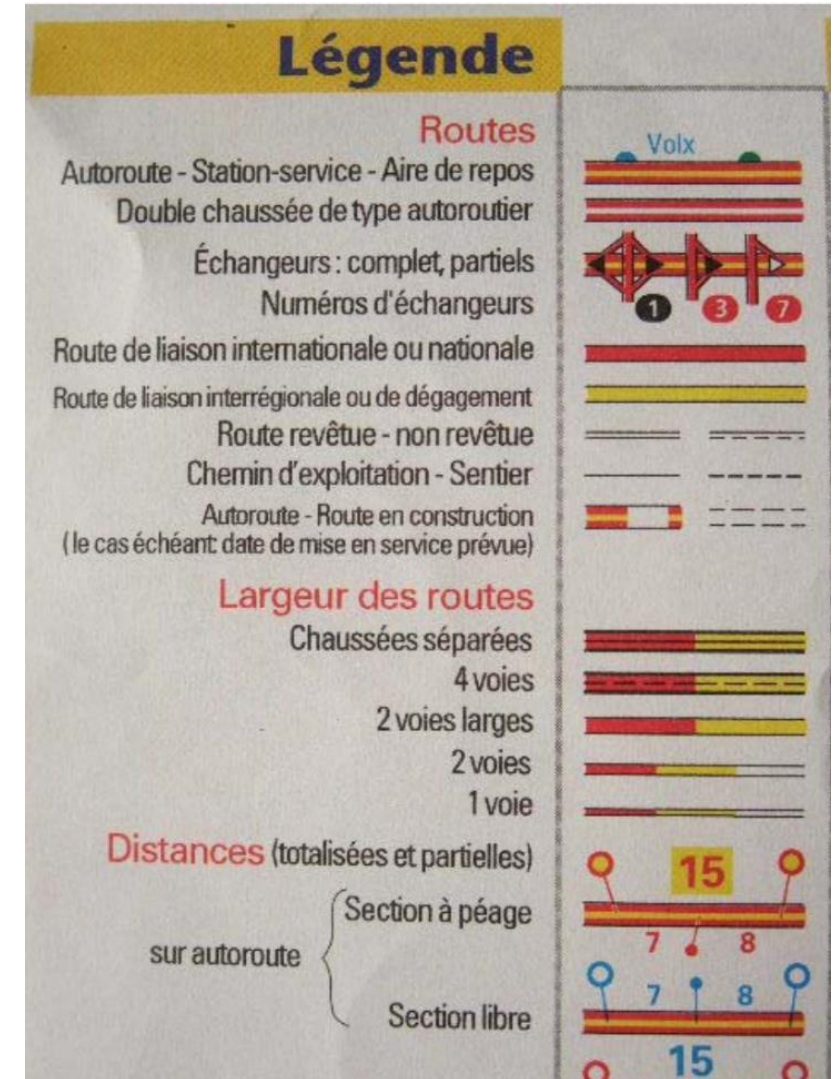
- de **comprendre** et **décrire la réalité** (en réduisant la complexité)
- de **simuler le monde réel**
- de **trouver** et **d'exprimer une solution** à un problème du monde réel
 - ... en **communiquant** à l'aide d'un **langage commun**
composé d'un nombre restreint de **concepts**

Le métamodèle : une légende est nécessaire pour la bonne compréhension d'un modèle

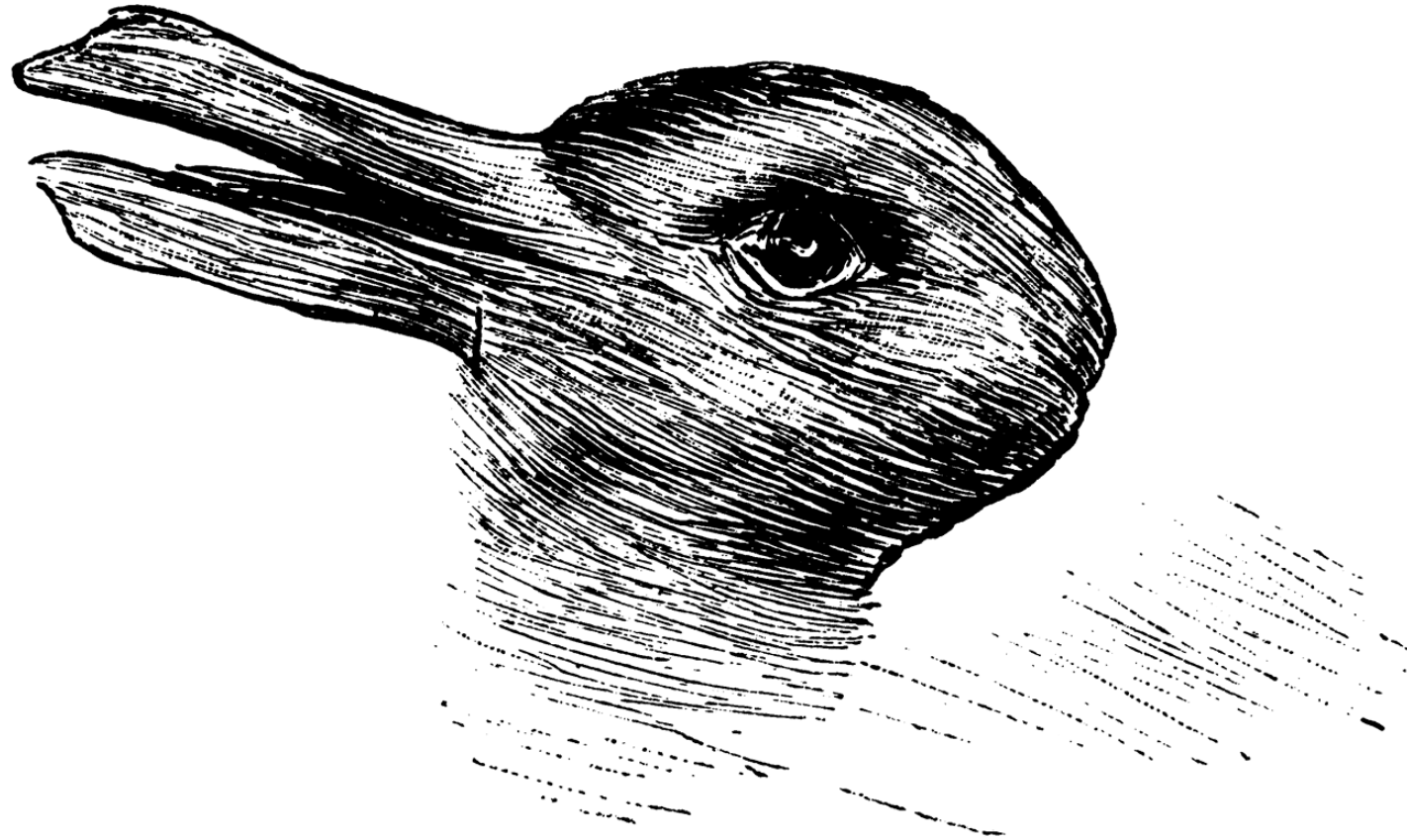
La légende, c'est le **langage** du modèle
A partir d'une **grammaire** précise et documentée,
elle permet d'interpréter les **concepts**/dessins
composant le modèle

La légende est, elle-même, un modèle !
⇒ On l'appelle un **métamodèle**

Un metamodelle est un **modèle** qui définit
le langage d'expression d'un modèle,
c.-à-d. le **langage de modélisation**.



Quizz : Que représente ce modèle ?



Extrait : <http://www.laboiteverte.fr/>

Un modèle dépend forcément d'un point de vue

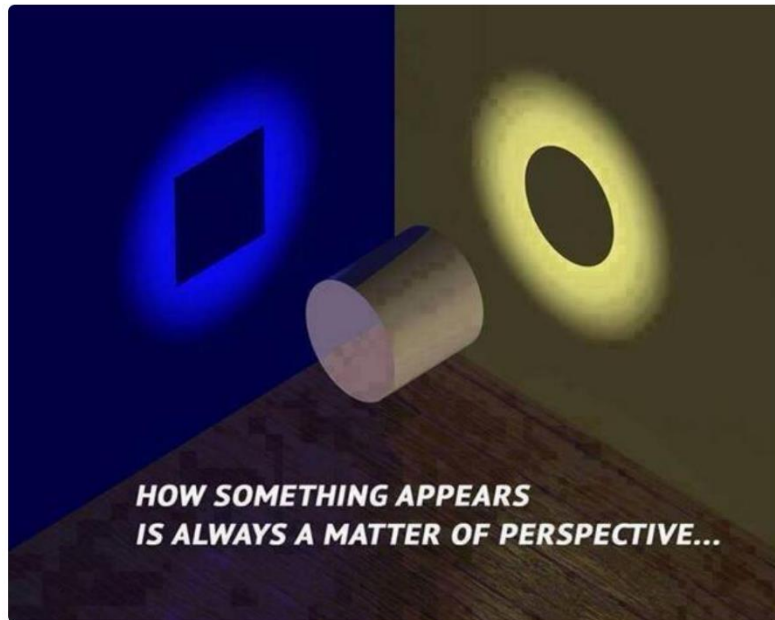


Un modèle dépend forcément d'un point de vue



Un modèle dépend forcément d'un point de vue

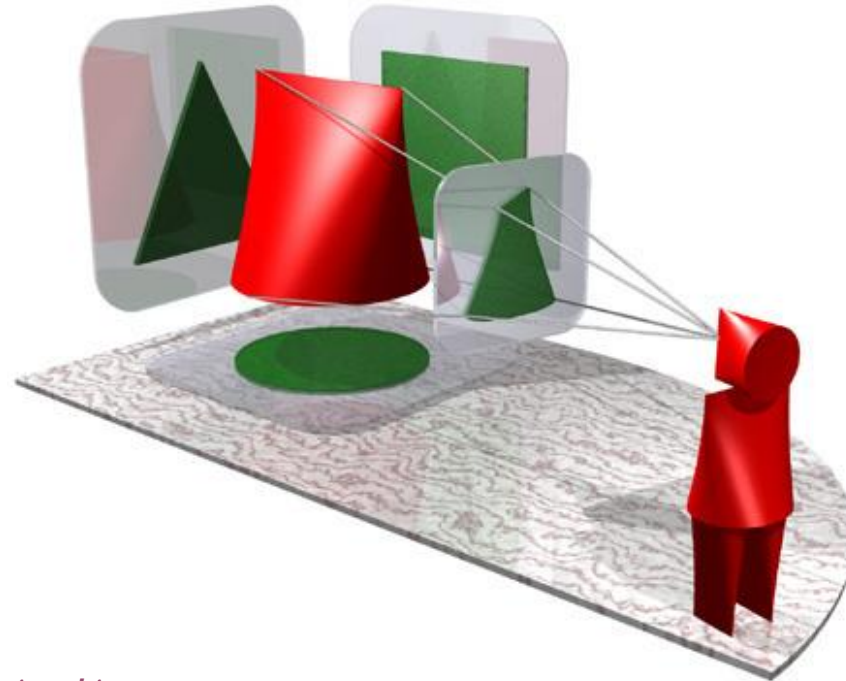
Place perspective on everything you look at.



Extrait :

<https://twitter.com/SciencePorn/status/424992760530481155>

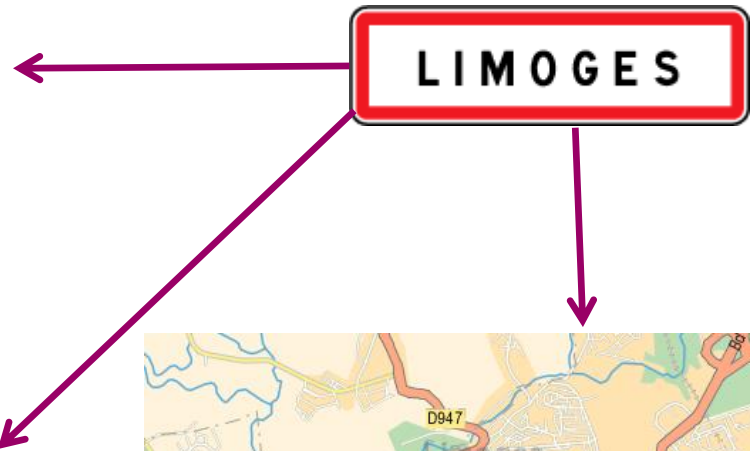
Tout est question de points de vue ...



Extrait :

<http://lucacardelli.name/Topics/TheoryOfObjects/ObjectSubject.html>

A map of the region around Limoges, France. The Vézère river is shown flowing through the area. Key towns labeled include Magnac-Laval, La Souterraine, Guéret, Bellac, Bessines-sur-Gartempe, Ambazac, Bourgneuf, Aubusson, Saint-Junien, Ri, Rancon, Limoges (marked with a green 'A' in a circle), Saint-Yrieix-la-Perche, Thiviers, Uzerche, Meymac, Ussel, Eglevins, and Neuville. Road markers for A20, E62, N21, and E09 are visible. A small inset map shows the location of this region within France.



Avantages d'un modèle (récapitulatif)

- **Abstrait** : Il fait ressortir les points importants tout en enlevant les détails non nécessaires
- **Compréhensible** : Il permet d'exprimer une chose complexe dans une forme plus facilement compréhensible par l'observateur
- **Précis** : Il représente fidèlement le système modélisé
- **Prédictif** : Il permet de faire des prévisions sur le système modélisé
- **Peu coûteux** : Il est bien moins coûteux à construire et étudier que le système lui-même

UML

(Unified Modeling Language)

Un langage pour modéliser ...

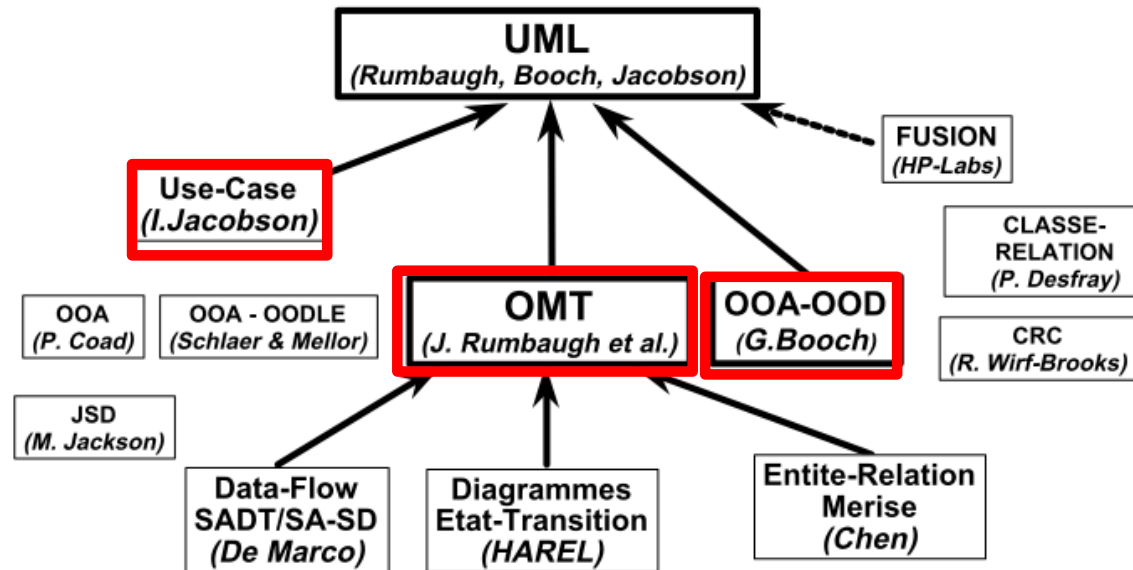
... des démarches par objet ...

Unifié par OMG

Historique d'UML

1995 : les « 3 Amigos » commencent un travail **d'unification des 3 démarches par objet**

- **OMT** : **O**bject **M**odeling **T**echniques (Rumbaugh de Rational Software)
- **OOD** : **O**riented **O**bject **D**esign (Booch de Général Electric)
- **OOSE** : **O**riented **O**bject **S**oftware **E**ngineering (Jacobson d'Ericsson)



Objectifs :

- Créer un langage de modélisation utilisable à la fois par **les hommes et les machines**,
- permettant de représenter des **systèmes** par des concepts objets.

Sites de référence pour UML



Unified Modeling Language® (UML®) Resource Page

[Introduction to UML](#) | [UML Success Stories](#) | [UML Certification Program](#) | [Vendor Directory](#) |

<http://www.uml.org>

Getting Started With UML:



The Unified Modeling Language - UML - is [OMG's](#) most-used specification, and the way the world models not only application structure, behavior, and architecture, but also business process and data structure.

UML, along with the [Meta Object Facility \(MOF™\)](#), also provides a key foundation for [OMG's Model-Driven Architecture®](#), which unifies every step of development and integration from business modeling, through architectural and application modeling, to development, deployment, maintenance, and evolution.

™OMG is a [not-for-profit technology standards consortium](#); our members define and maintain the UML specification which we publish in the series of documents linked on this page for your free download. Software providers of every kind build tools that conform to these specifications. To model in UML, you'll have to obtain a compliant modeling tool from one of these providers and learn how to use it. The [links at the bottom of this page](#) will help you do that.

If you're new to modeling and UML, start with our own [Introduction to UML, here](#), and possibly this piece on the [benefits of modeling to your application development cycle](#).

Unified Modeling Language® (UML®)

NOTE: There are no XSD files associated with Version 2.0

The current version is found at <http://www.omg.org/spec/UML/Current>

OMG Formal Versions Of UML®

Version	Release Date	URL
2.5	June 2015	http://www.omg.org/spec/UML/2.5
2.4.1	August 2011	http://www.omg.org/spec/UML/2.4.1
2.4	March 2011	http://www.omg.org/spec/UML/2.4
2.3	May 2010	http://www.omg.org/spec/UML/2.3
2.2	February 2009	http://www.omg.org/spec/UML/2.2
2.1.2	November 2007	http://www.omg.org/spec/UML/2.1.2
2.1.1	August 2007	http://www.omg.org/spec/UML/2.1.1

Spécifications officielles

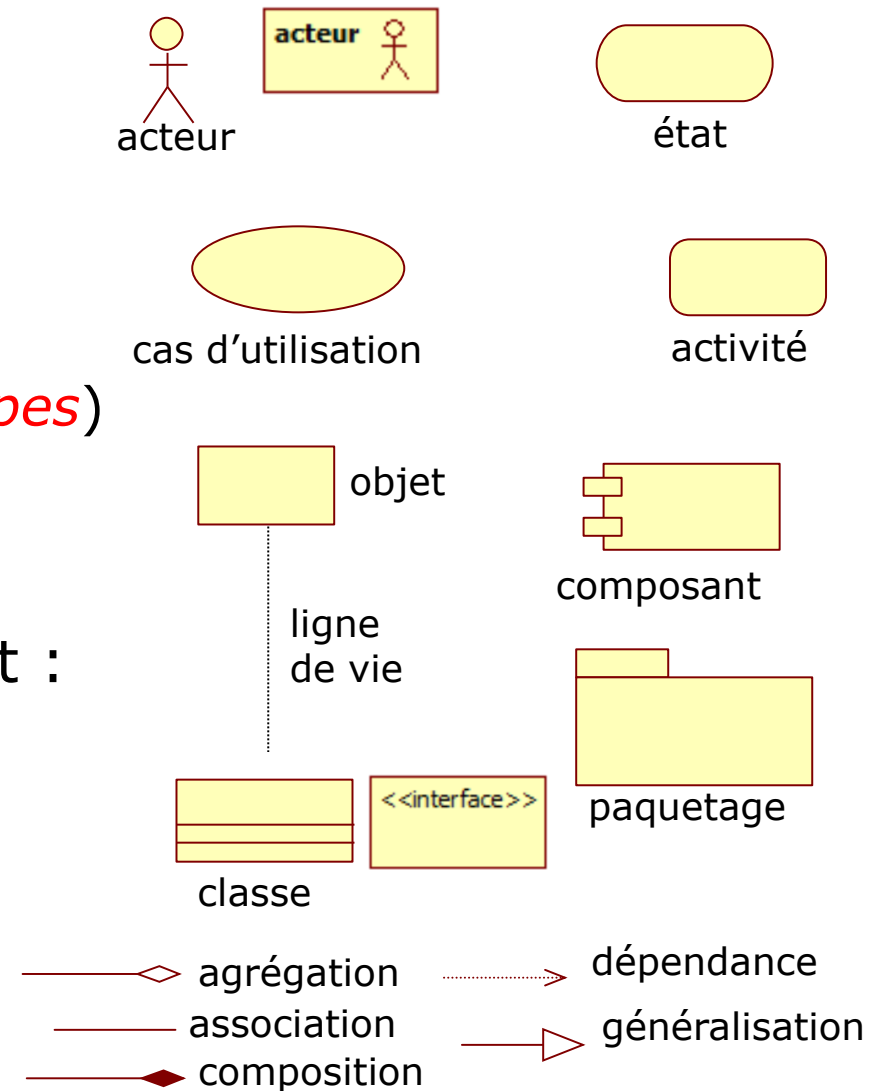
<http://www.omg.org/spec/UML>

Quelques éléments du langage UML (méta-modèle)

Le langage UML propose une **notation** :
→ composée d'une **Syntaxe graphique**
→ et respectant une certaine **Sémantique**
(avec des points de variation sémantique et des *stéréotypes*)

Cette notation graphique, support du langage UML, est :

- Normalisée
- Semi-formelle
- Universelle
- Indépendante du langage de programmation
- Supportée par de nombreux outils



... et bien d'autres ...

Pyramide de modélisation de l'OMG

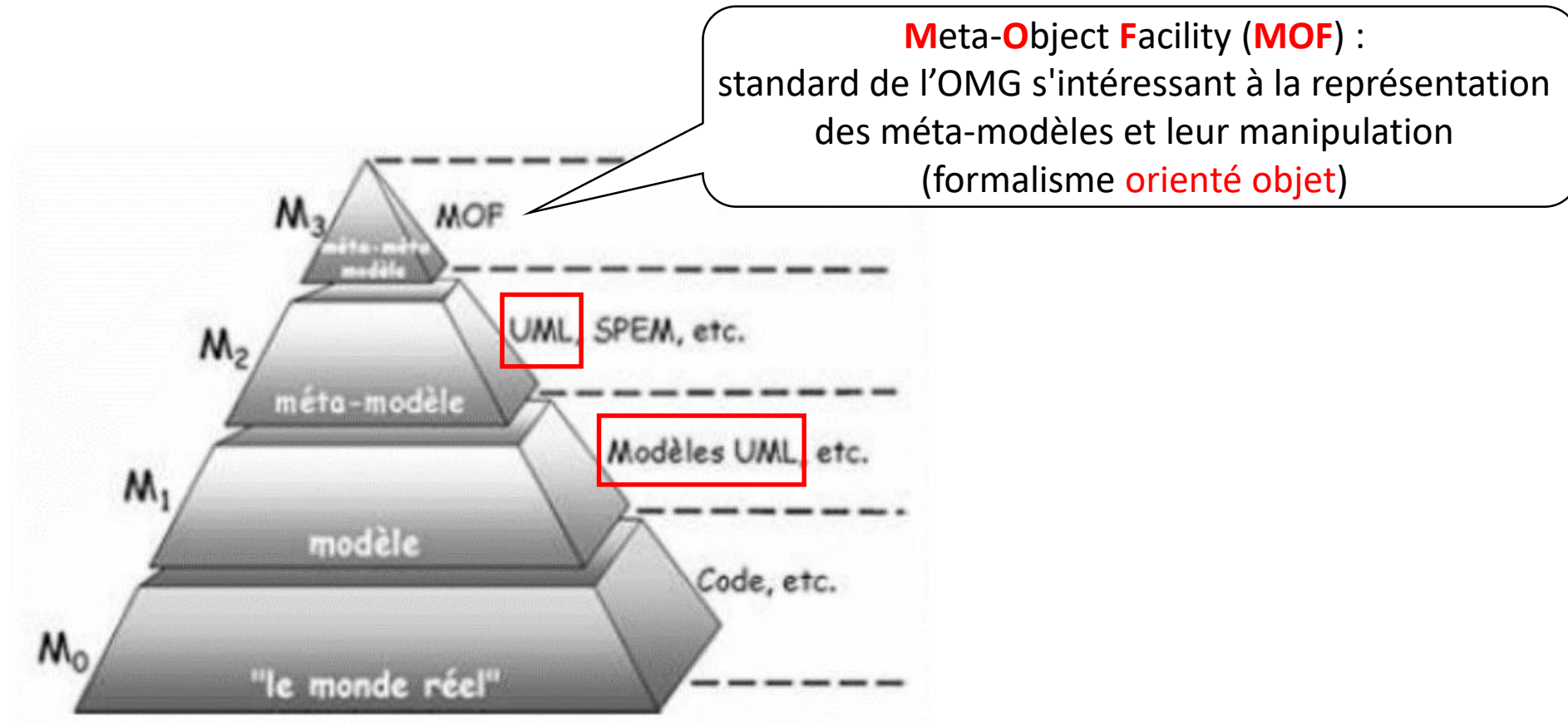


Figure 3. *Pyramide de modélisation de l'OMG (Bézivin, 2003)*

Figure 3 extraite de : *Etat de l'art sur le développement logiciel dirigé par les modèles.*

disponible sur : ftp://ftp.irit.fr/IRIT/MACAO/Article_TSI-IDM-final-coulette.pdf

Originalement proposé par Jean Bézivin. La transformation de modèles. Ecole d'Été d'Informatique CEA EDF INRIA 2003, cours #6.

UML propose une notation qui permet de représenter des modèles

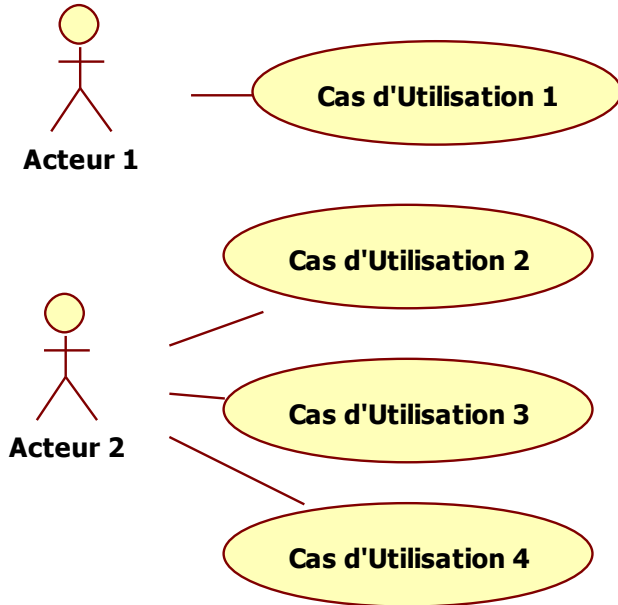


Diagramme des cas d'utilisation

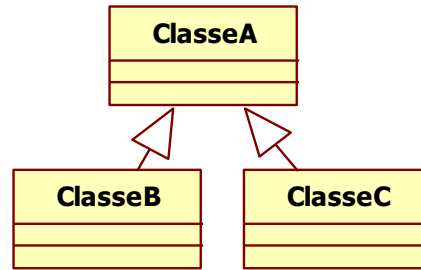


Diagramme
de classes

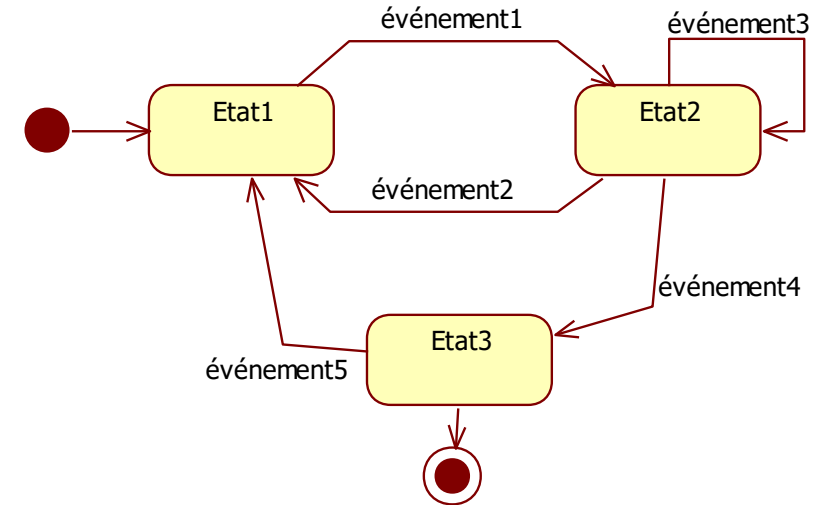
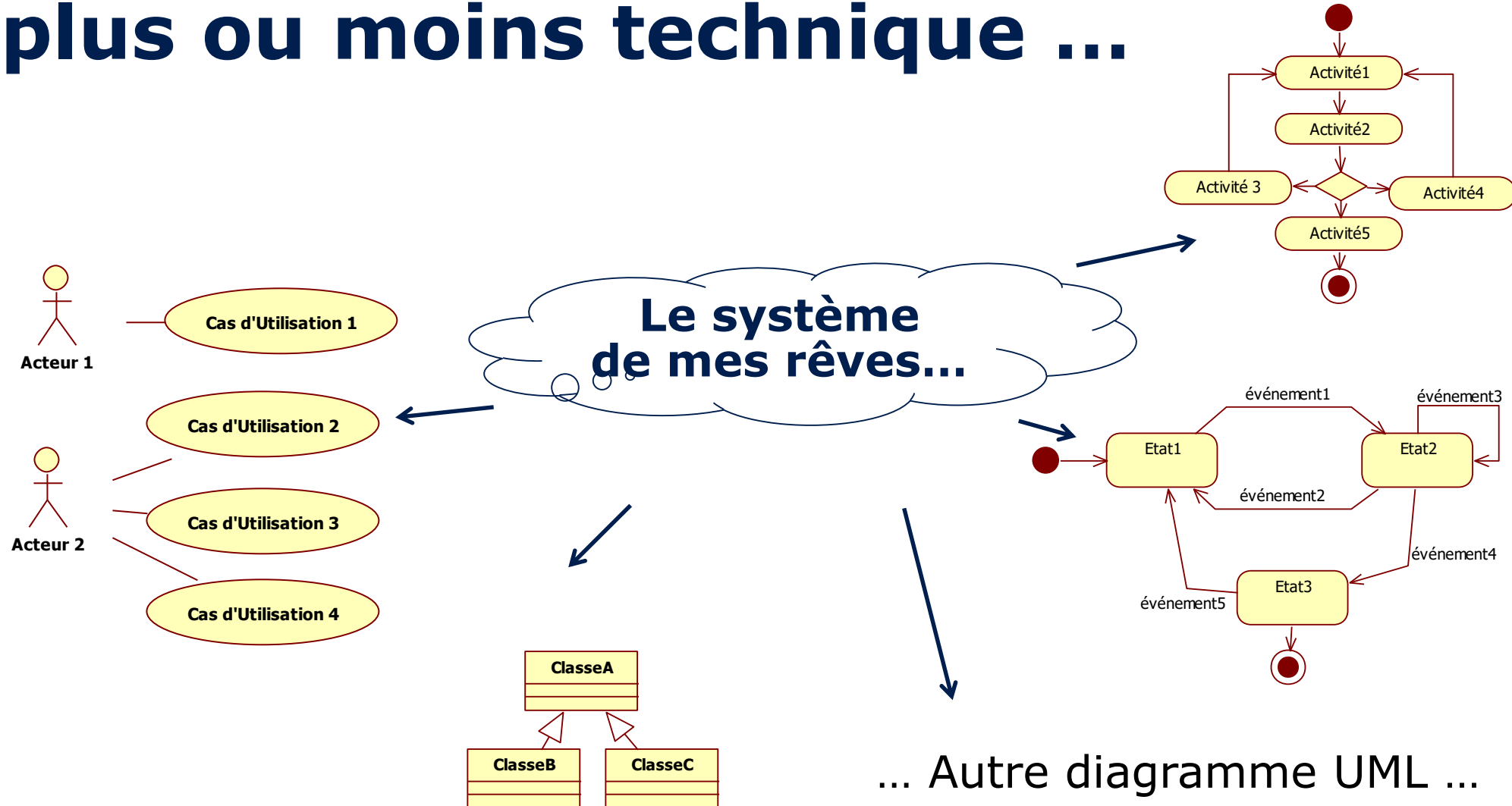
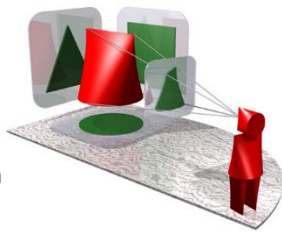


Diagramme
d'états-transitions

Exemples de modèles respectant le méta-modèle UML
(**représentation graphique** sous forme de **diagrammes**)

Chaque diagramme UML représente une VUE partielle du système, différente et plus ou moins technique ...

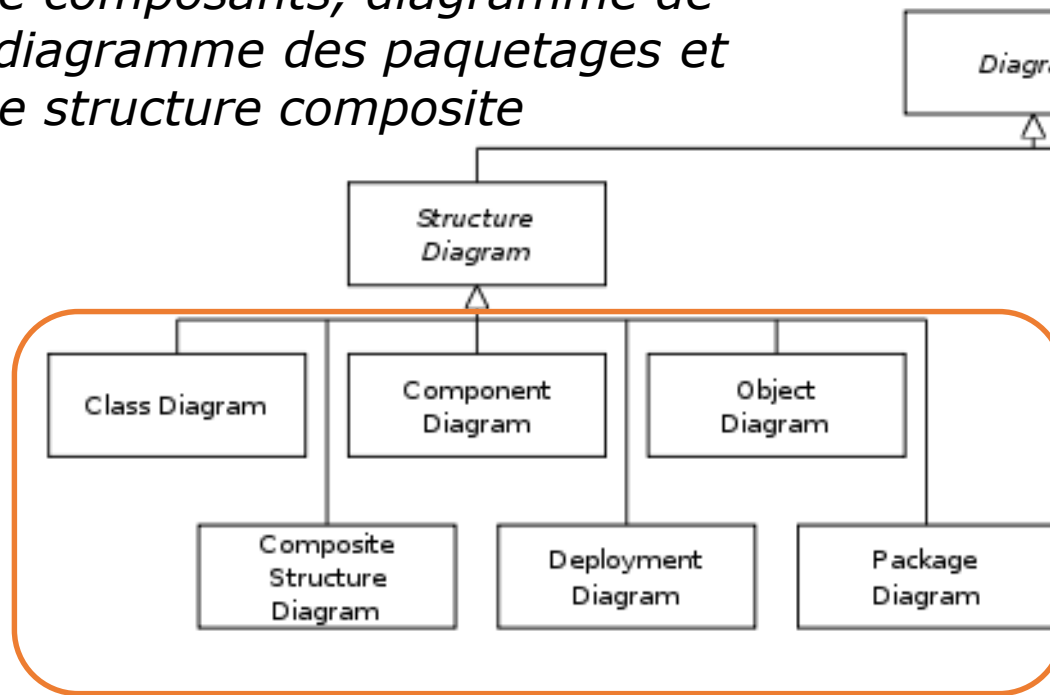


... Autre diagramme UML ...

Les 13 diagrammes d'UML 2

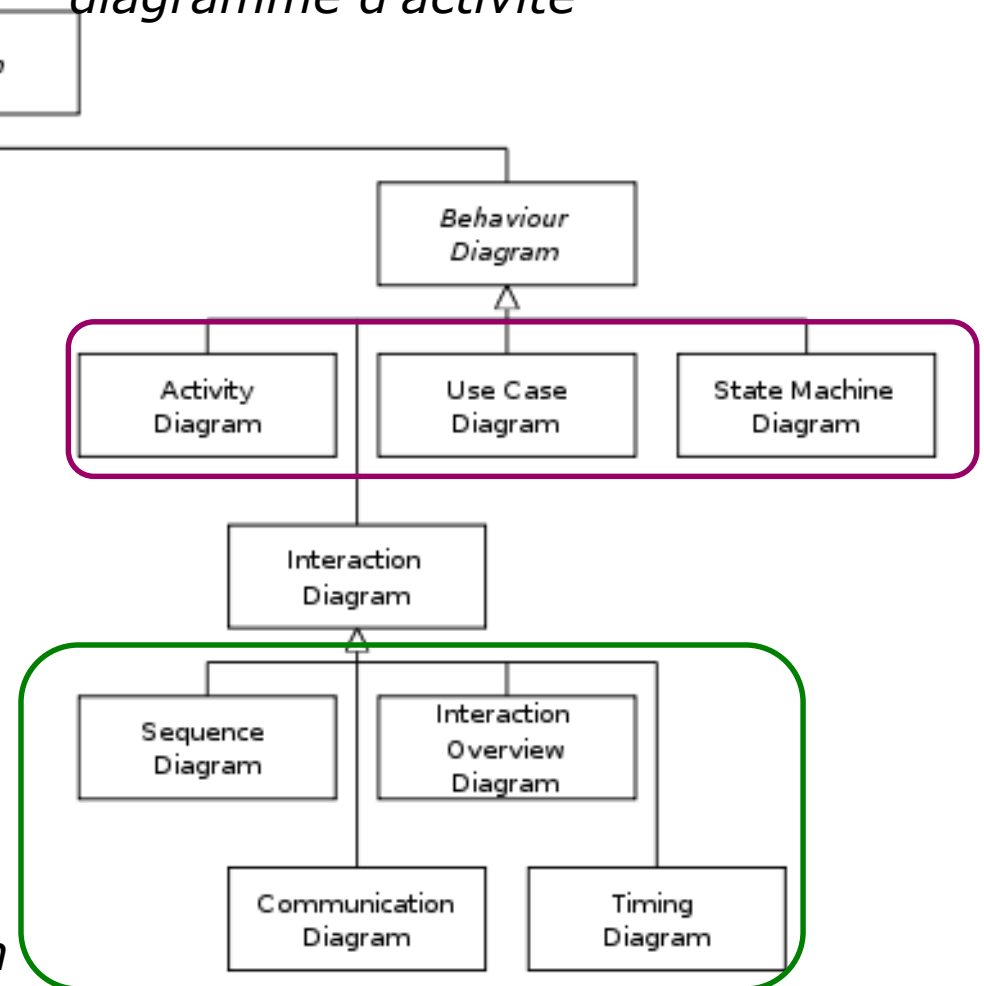
➤ Les diagrammes structurels ou statiques (Structure Diagram)

diagramme de classes, diagramme d'objets, diagramme de composants, diagramme de déploiement, diagramme des paquetages et diagramme de structure composite



➤ Les diagrammes comportementaux ou dynamiques (Behavior Diagram) :

diagramme des cas d'utilisation, diagramme états-transitions, et diagramme d'activité



➤ Les diagrammes d'interaction ou dynamiques (Interaction Diagram)

diagramme de séquence, diagramme de communication et diagramme global d'interaction

Modéliser avec UML permet de construire un **modèle quadridimensionnel** selon 4 points de vue

services rendus, architecture statique, comportement dynamique, et déploiement (installation)

QUOI ?

Aspect fonctionnel

Services rendus par le système
cas d'utilisation

QUI ?

Aspect statique

(organisation des éléments)

description des objets et de leurs relations
structuration en paquetages

OU ?

Installation du système

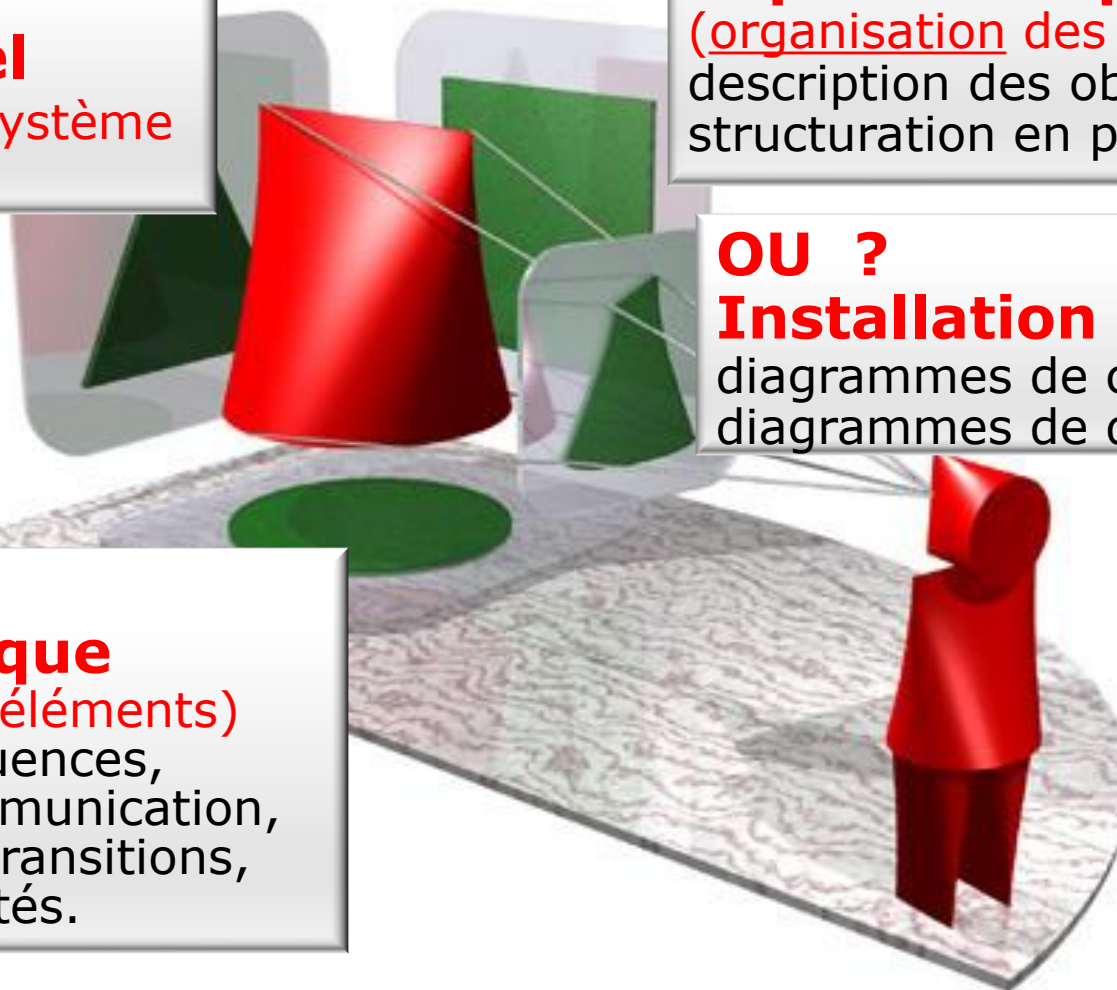
diagrammes de composants et
diagrammes de déploiement.

QUAND ?

Aspect dynamique

(comportement des éléments)

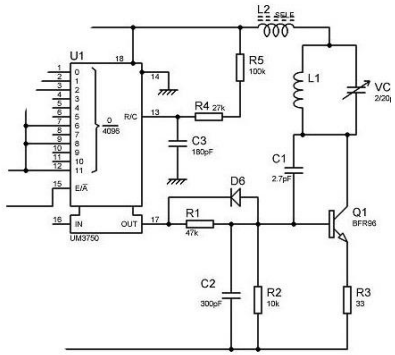
diagrammes de séquences,
diagrammes de communication,
diagramme d'états-transitions,
diagrammes d'activités.



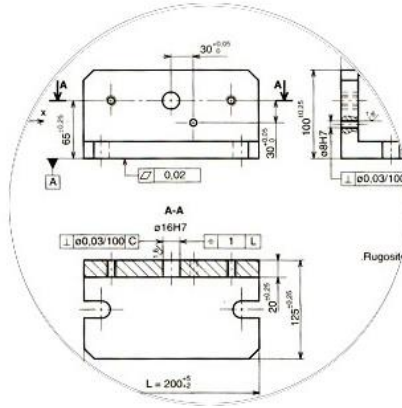
UML en pratique

UML : un support de communication ...

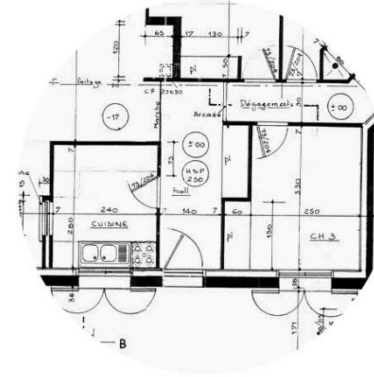
Ingénierie Electrique



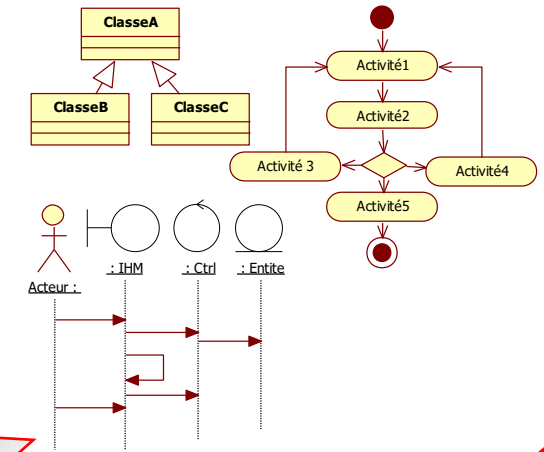
Ingénierie mécanique



Ingénierie du bâtiment



Ingénierie logicielle

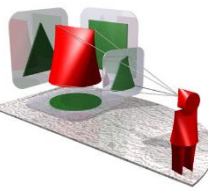


UML est un langage de modélisation graphique
avec un formalisme **orienté objet** (*méta-méta-modèle MOF*)

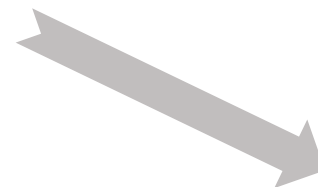
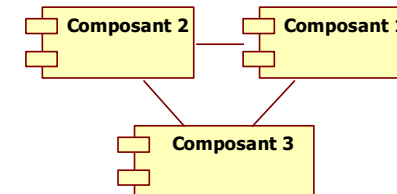
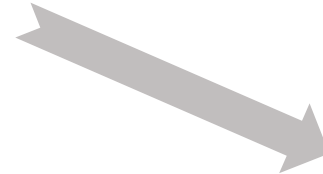
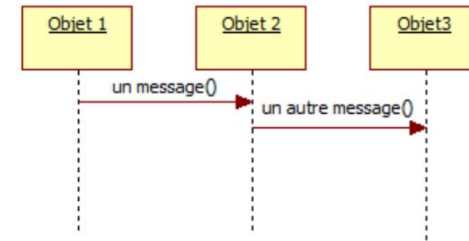
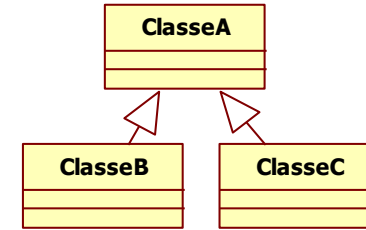
UML n'est *pas une méthode*

... et donc ne définit aucun processus de développement de produit !

Utiliser en reverse-engineering, UML permet de proposer une autre vision sur du code existant



```
function use_array(a, b) {  
  var c = -1;  
  for (var i = 0; i < a.length; i++) {  
    c.push(a[i]);  
  }  
  return c;  
}
```



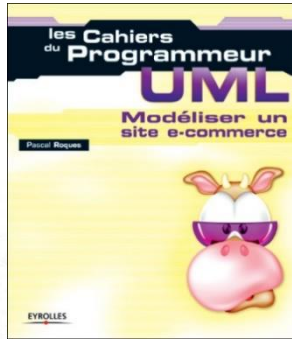
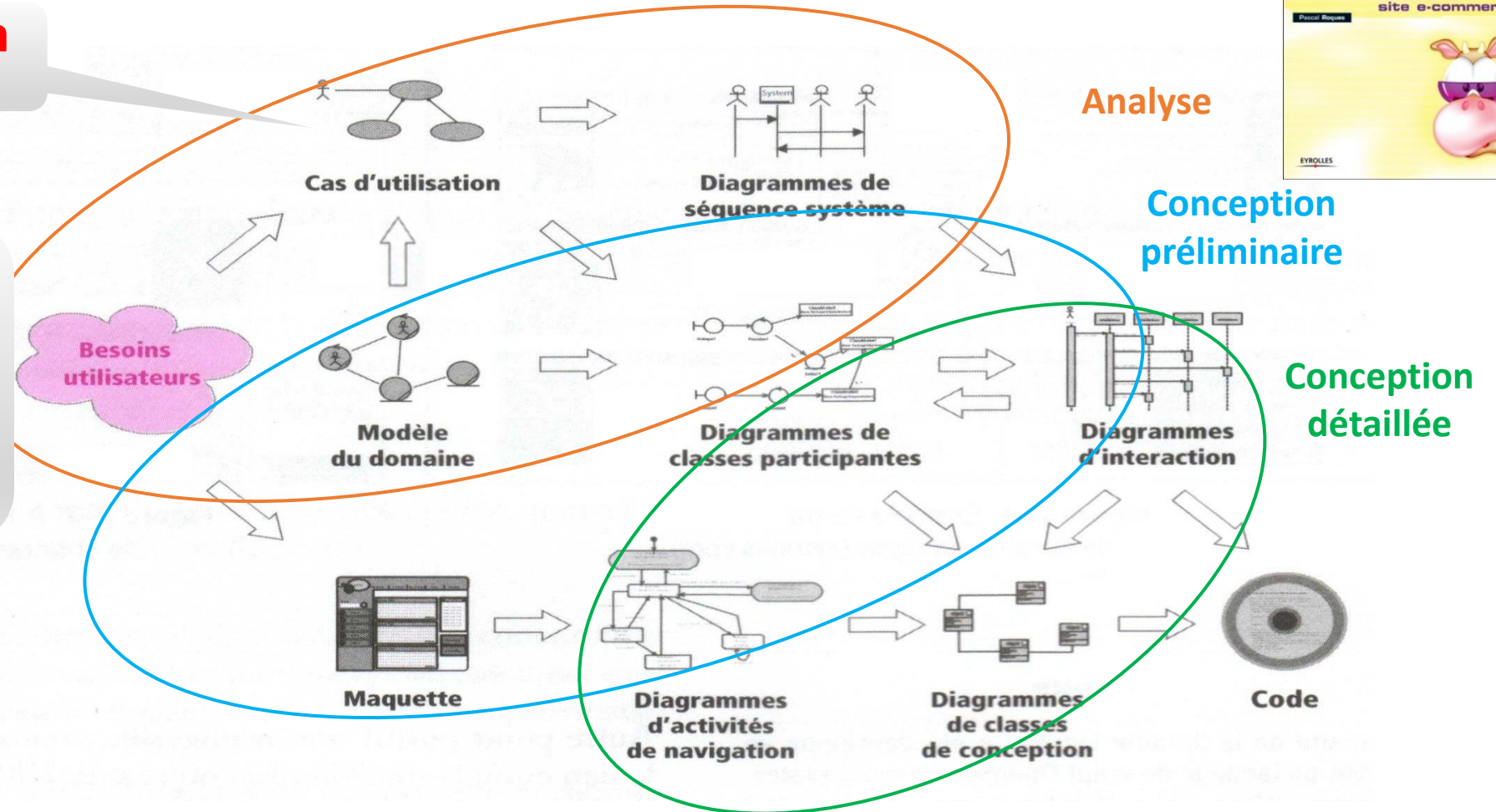
Autre diagramme ...

Une démarche possible de développement logiciel dirigée par les modèles basée sur une succession de diagrammes UML depuis l'expression des besoins (**Analyse**) jusqu'au code (**Implémentation**)

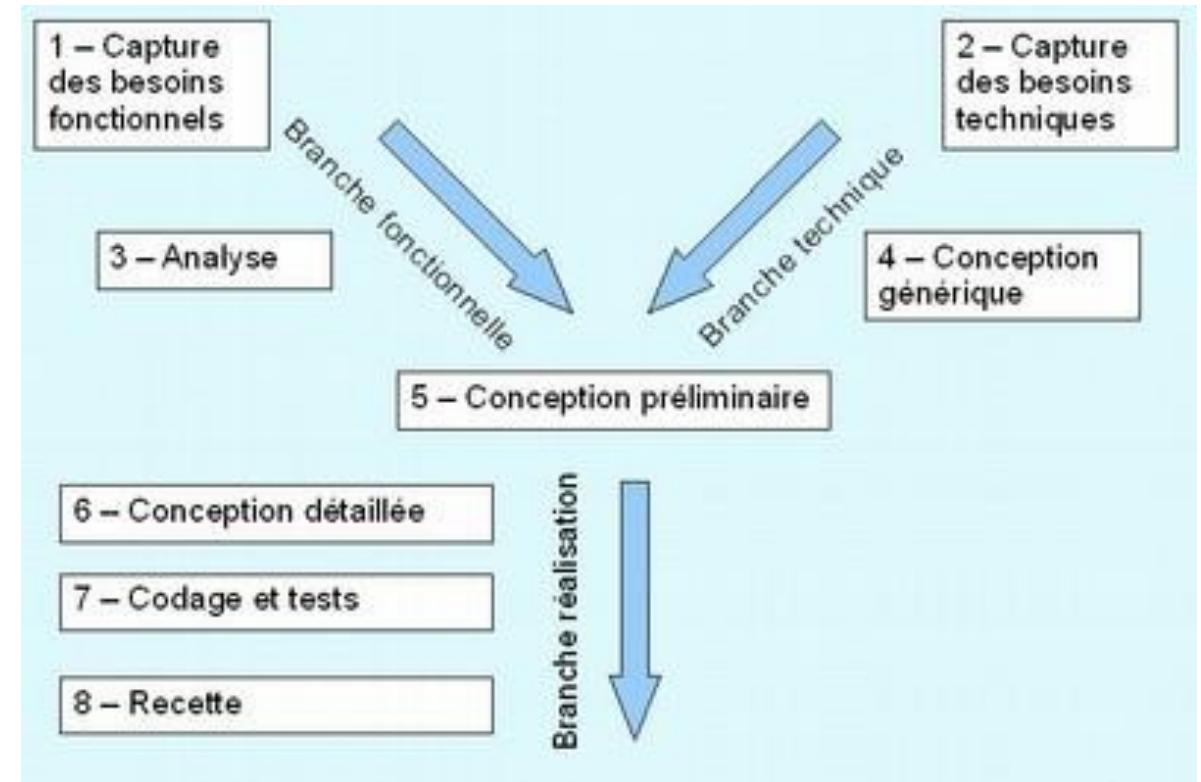
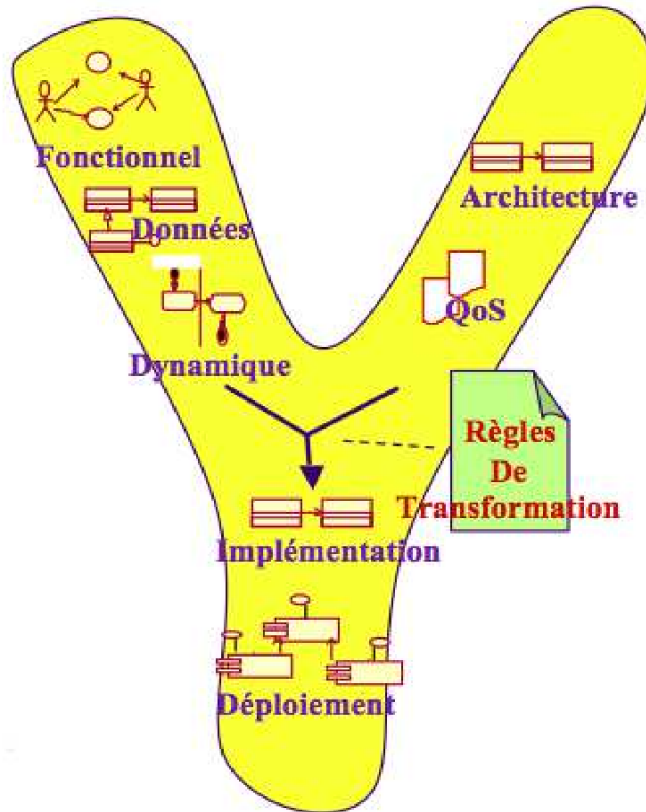
Les cas d'utilisation
pilotent la démarche

Démarche
centrée sur l'architecture
(autour de la Conception :
diagrammes de classes &
de séquences
au cœur de la démarche)

Démarche
**Itérative et
incrémentale**



... une illustration de la branche fonctionnelle d'un cycle en Y (ou processus 2TUP)



Et pour finir ...



What do their creators think about UML now?

By **Jordi Cabot** 5/08/2016 | 5:09

Posted in [opinion](#), [UML](#) and [OCL](#)

15



Everybody has its own opinion about the [Unified Modeling Language](#) but I think it's interesting to collect some UML opinions expressed by the people that created the language in the first place some twenty years ago.

Grady Booch's views on UML

"

The UML should be used to reason about alternatives. Put up some diagrams. Throw some use cases against it. Throw away those diagrams then write some code against you best decision. Repeat ★

"

When we began with the UML, we never intended it to become a programming language

"

We never got the notation for collaborations right. Component and deployment diagrams needed more maturing. The UML metamodel became grossly bloated, because of the drive to model driven development. I think that complexity was an unnecessary mistake.

"

I rather still like the UML 😊 Seriously, you need about 20% of the UML to do 80% of the kind of design you might want to do in a project – agile or not – but use the UML with a very light touch: use the notation to reason about a system, to communicate your intent to others...and then throw away most of your diagrams.


A lire sur : <http://modeling-languages.com/uml-opinions-creators/>

Annexe

Draw Toast Challenge : une idée de Tom Wujec

DrawToast workshops provide an effective introduction to **systems thinking** and **design collaboration**.

DrawToast INTRO TEMPLATES GALLERY WPS™ QUESTIONS




DRAW HOW TO MAKE TOAST

A simple and fun introduction to Systems Thinking

An Introduction to Systems Thinking and Wicked Problem Solving™

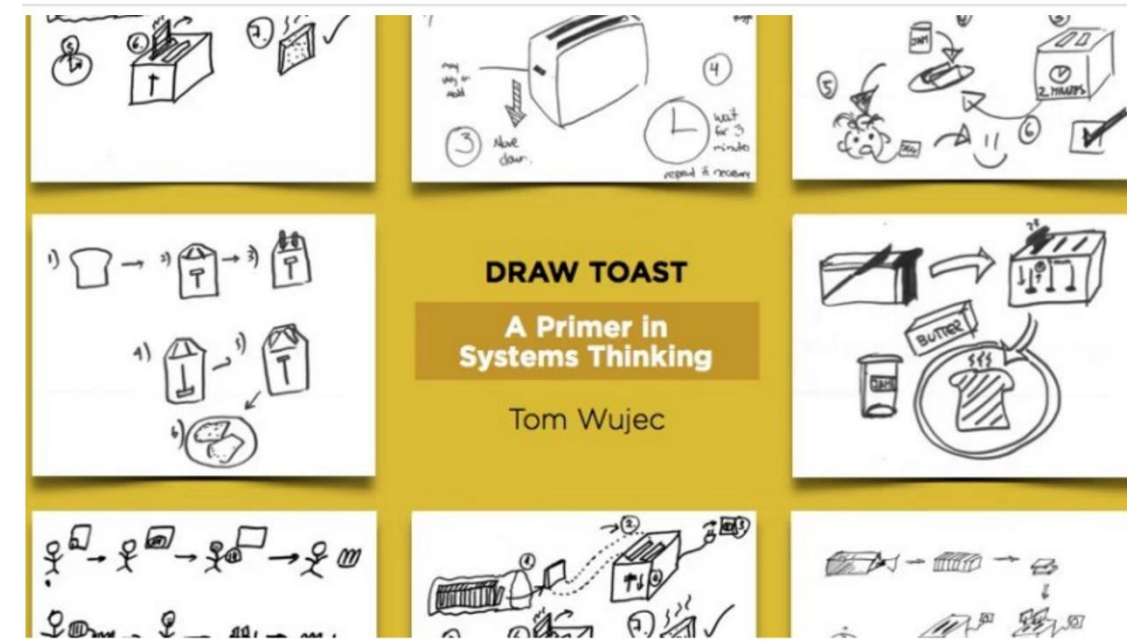
HOW TO RUN DRAW TOAST WORKSHOPS

DrawToast workshops are a great way to get groups to think freshly about mental models. In just 3 minutes, each person sketches a diagram of how to make toast. When comparing diagrams, people are shocked at how diverse the diagrams are, revealing a wide range of models of what's important in making toast. It's a great launch pad for drawing out what's really important to the group.



TOM WUJEC
GOT A WICKED PROBLEM? FIRST, TELL ME HOW YOU MAKE TOAST

TED



En savoir plus sur Draw Toast : <http://www.drawtoast.com/>
<http://www.tomwujec.com/design-projects/draw-toast/>