



Novembre 2023

Cours de: COO & UML

Chargé du cours:

Dr Ing. (MA) Houndji V. Ratheil

Chargé de cours



Houndji Vinasétan Ratheil,
<https://ratheil.info>

Enseignant-Chercheur à l'UAC

- Intelligence Artificielle
- Optimisation/RO
- Modélisation avec UML
- Fiabilité des SI
- etc.

Coordonnateur du cycle de Licence à IFRI, UAC
Chef du Département Génie Logiciel

Vos attentes ?



Vos attentes ?

UML



Il est temps de démarrer ;-)

Ordre du jour



Plan du cours

Discussion d'ordre général

Généralités sur la Conception Orientée Objet

Diagramme de cas d'utilisation

Plan du cours



- Voir planDeCours.pdf

Discussion d'ordre général



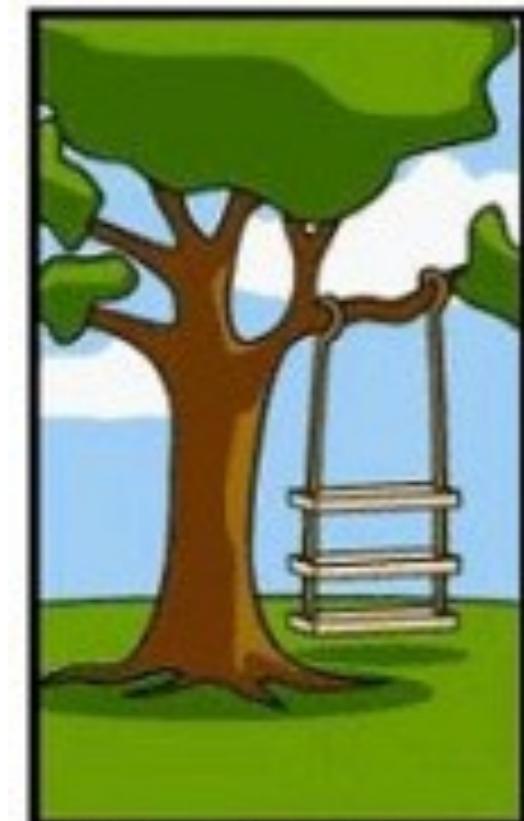
Classe inversée

Ponctualité. <= 10 minutes après l'heure du début.

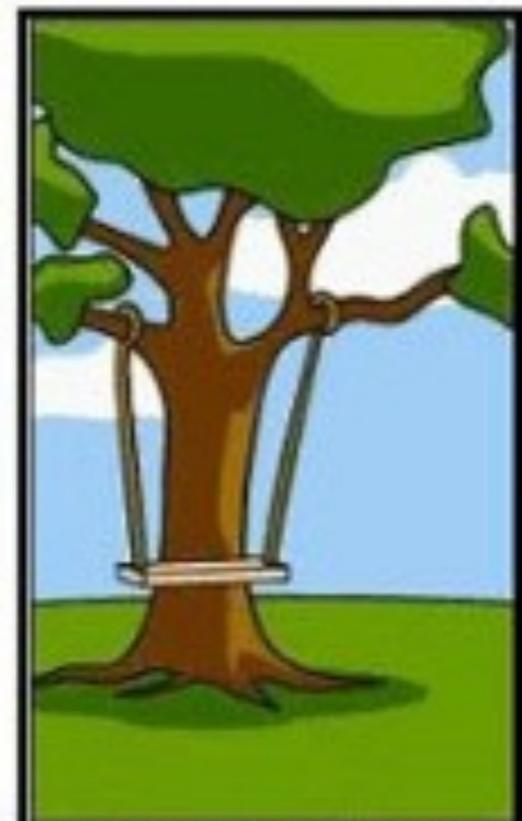
Attention pendant l'enseignement.

Echanges en dehors des cours.

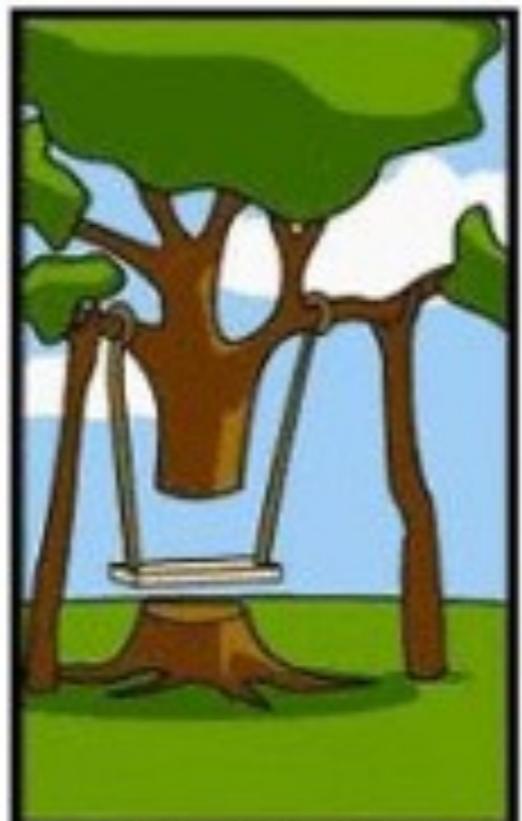
Deadlines pour rendre les travaux:
-2/20 chaque 12h de retard.



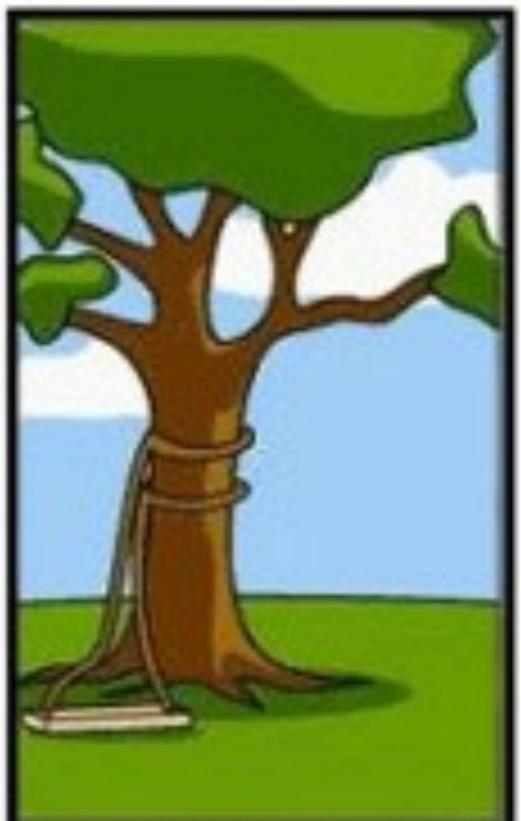
Comment le client
a exprimé son besoin



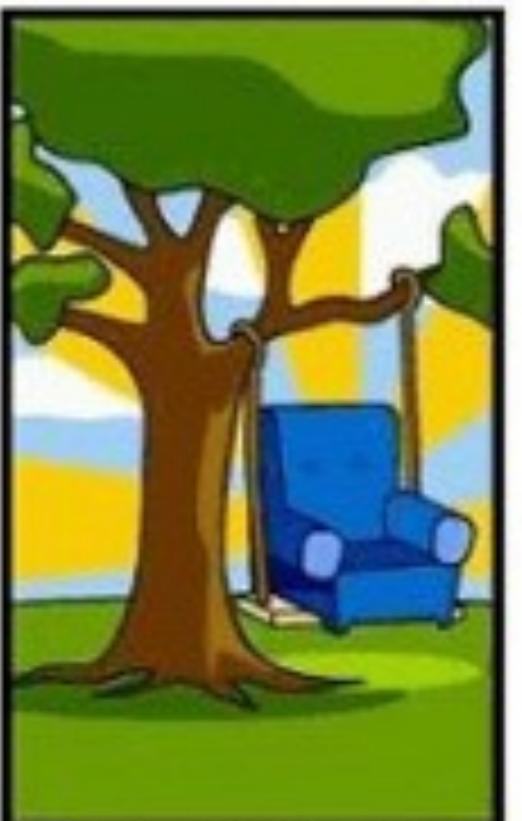
Comment le chef de
projet l'a compris



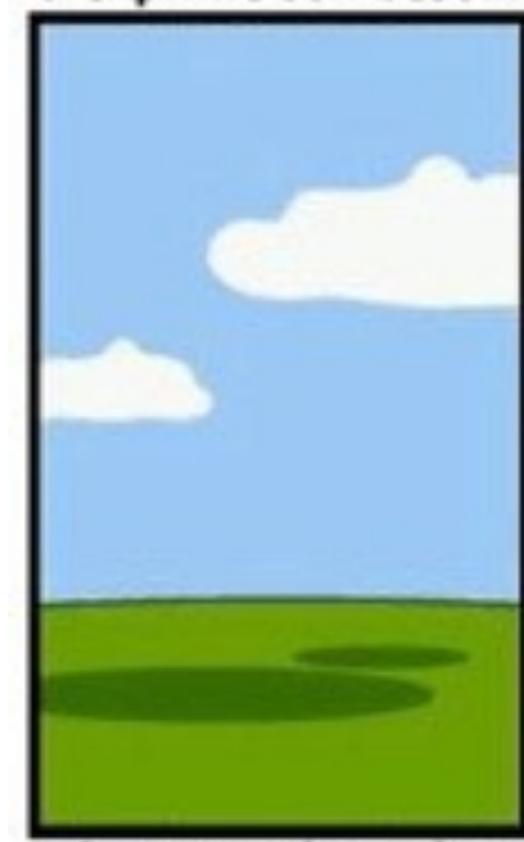
Comment l'ingénieur
l'a conçu



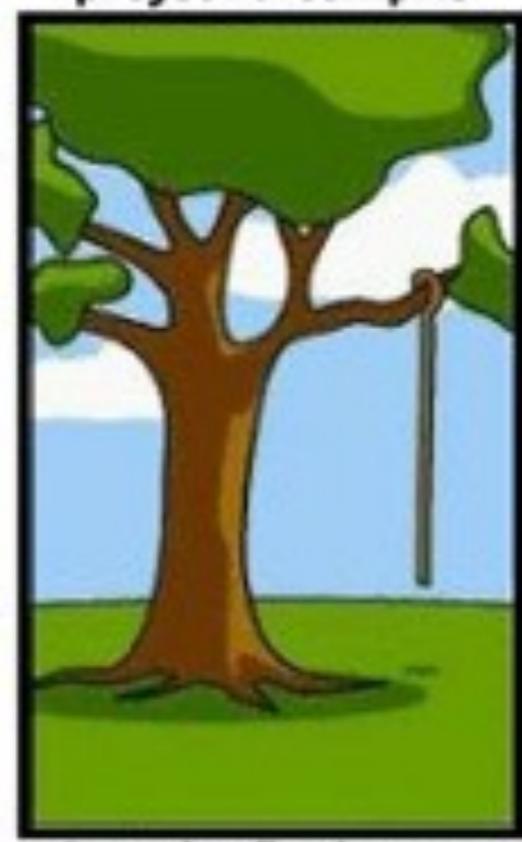
Comment le
programmeur l'a écrit



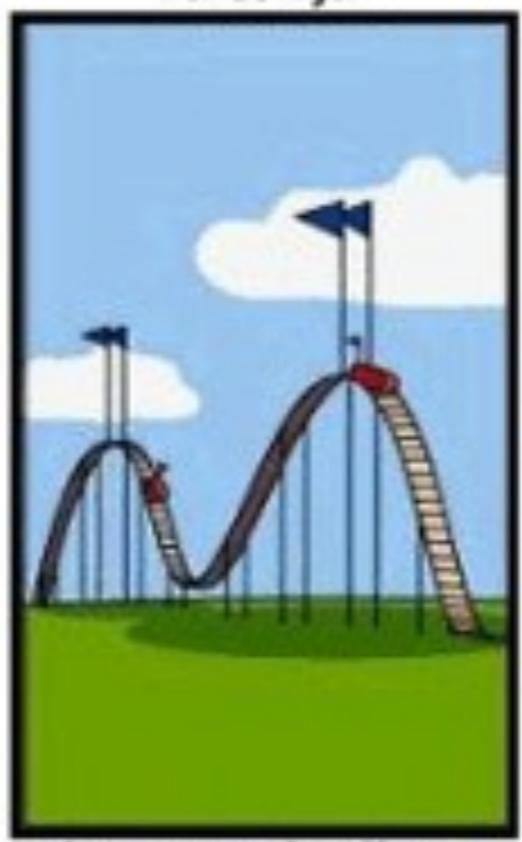
Comment le responsable
des ventes l'a décrit



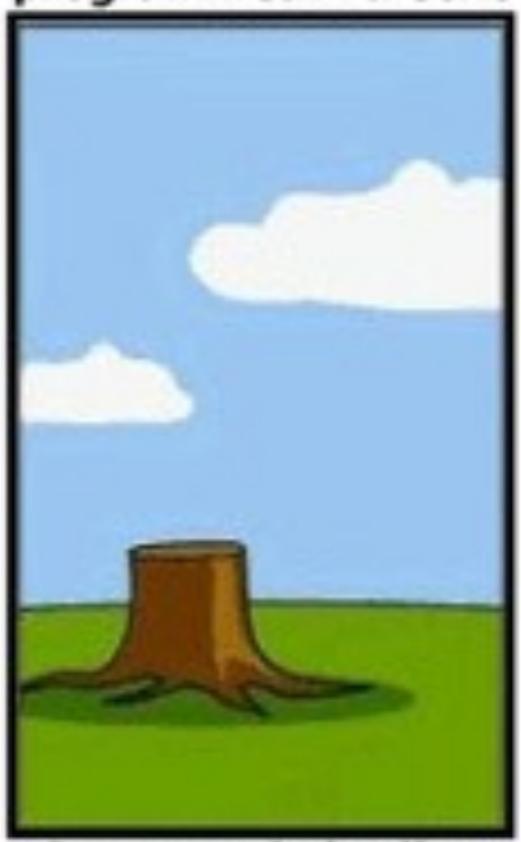
Comment le projet
a été documenté



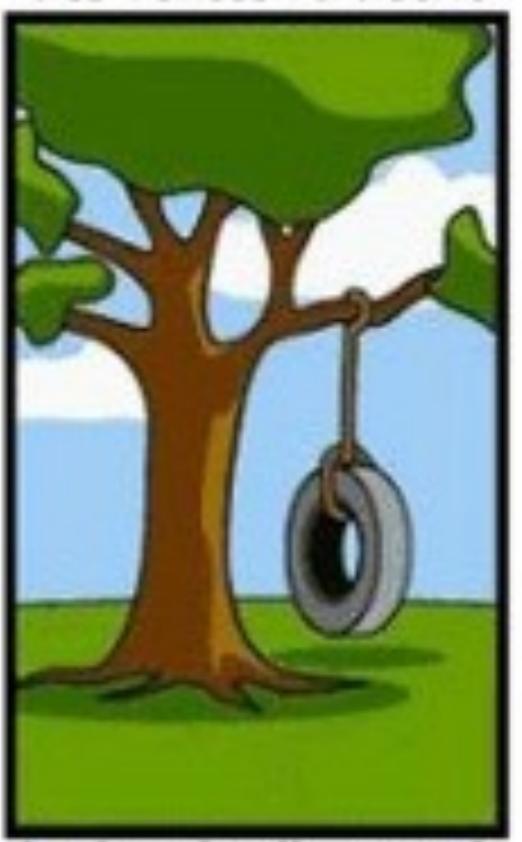
Ce qui a finalement
été installé



Comment le client
a été facturé



Comment la hotline
répond aux demandes



Ce dont le client avait
réellement besoin

Matériel et logiciel

- Systèmes informatiques :
 - **80 % de logiciel** ;
 - 20 % de matériel.
- Depuis quelques années, la fabrication du matériel est assurée par quelques fabricants seulement.
 - Le matériel est relativement fiable.
 - Le marché est standardisé.

Les problèmes liés à l'informatique sont essentiellement des problèmes de logiciel.

Matériel et logiciel

- Systèmes informatiques :
 - **80 % de logiciel** ;
 - 20 % de matériel.
- Depuis quelques années, la fabrication du matériel est assurée par quelques fabricants seulement.
 - Le matériel est relativement fiable.
 - Le marché est standardisé.

Les problèmes liés à l'informatique sont essentiellement des problèmes de logiciel.

Poids de la maintenance

	Répartition effort dév.	Origine des erreurs	Coût de la maintenance
Spécification	6%	56%	82%
Conception	5%	27%	13%
Implantation	7%	7%	1%
Intégration - Tests	15%	10%	4%
Maintenance	67%		

(Zeltovitz, De Marco)

La réduction du coup de maintenance se joue en amont.

Conception Orientée Objet



Conception/Modélisation ?



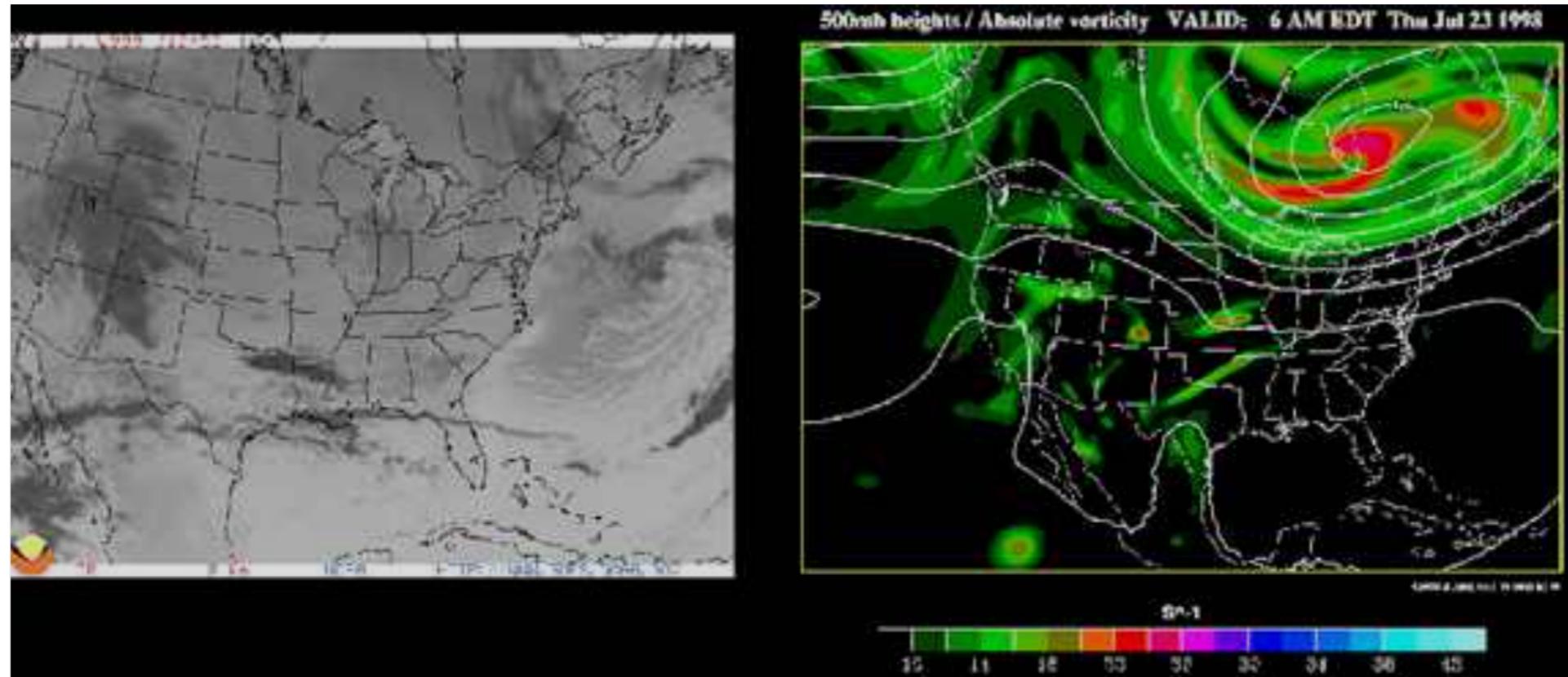
Conception Orientée Objet

Conception - Modélisation





COO: Modélisation

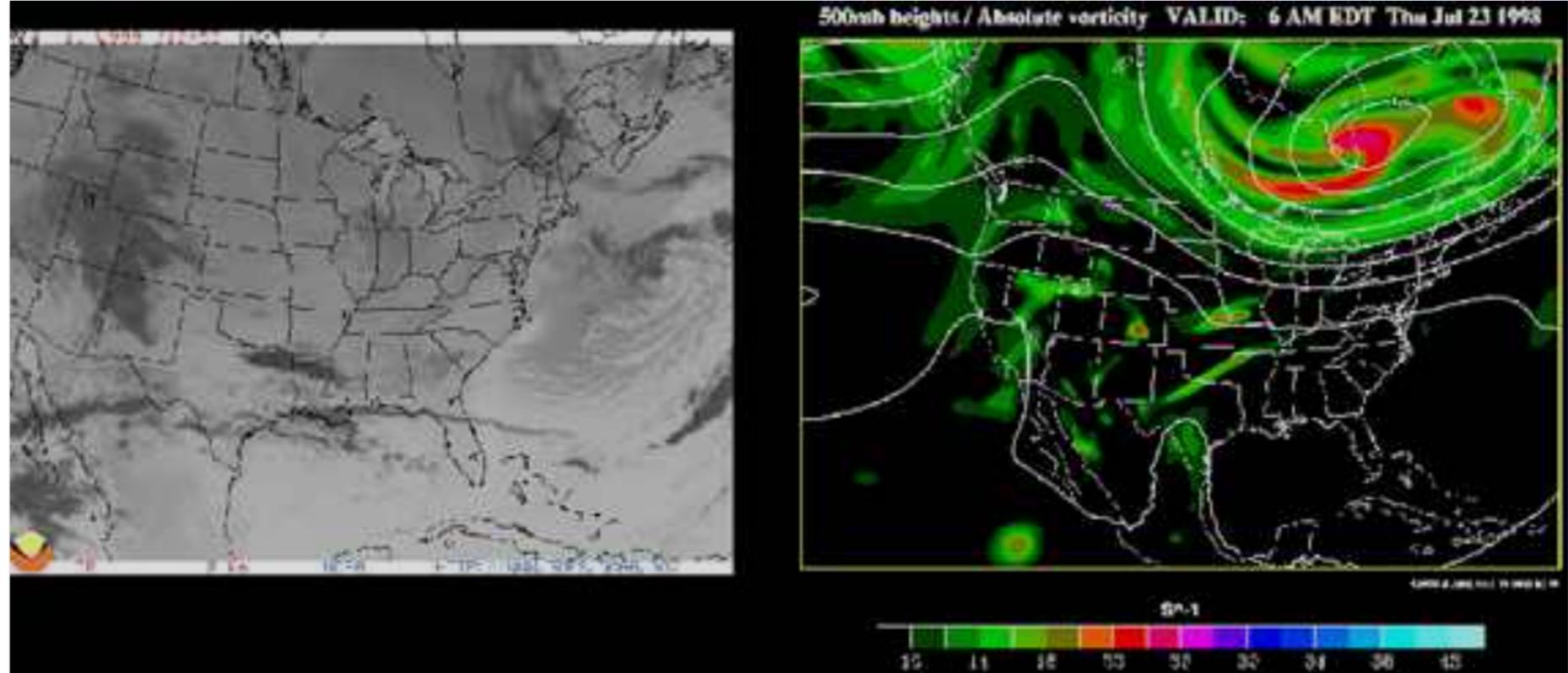


Modéliser = abstraire la réalité pour mieux comprendre le système à réaliser / réalisé.

Un modèle est une vue subjective mais pertinente de la réalité.



COO: Modélisation



Il s'agit d'un processus qui consiste à identifier les caractéristiques intéressantes d'une entité, en vue d'une utilisation précise.

COO: Modélisation



Caractéristiques essentielles d'un modèle

Un modèle représente le système étudié et reproduit ses comportements.

Un modèle permet de faciliter la compréhension du système étudié.

Un modèle réduit la complexité du système étudié.

Un modèle permet de simuler le système étudié.

COO: Modélisation



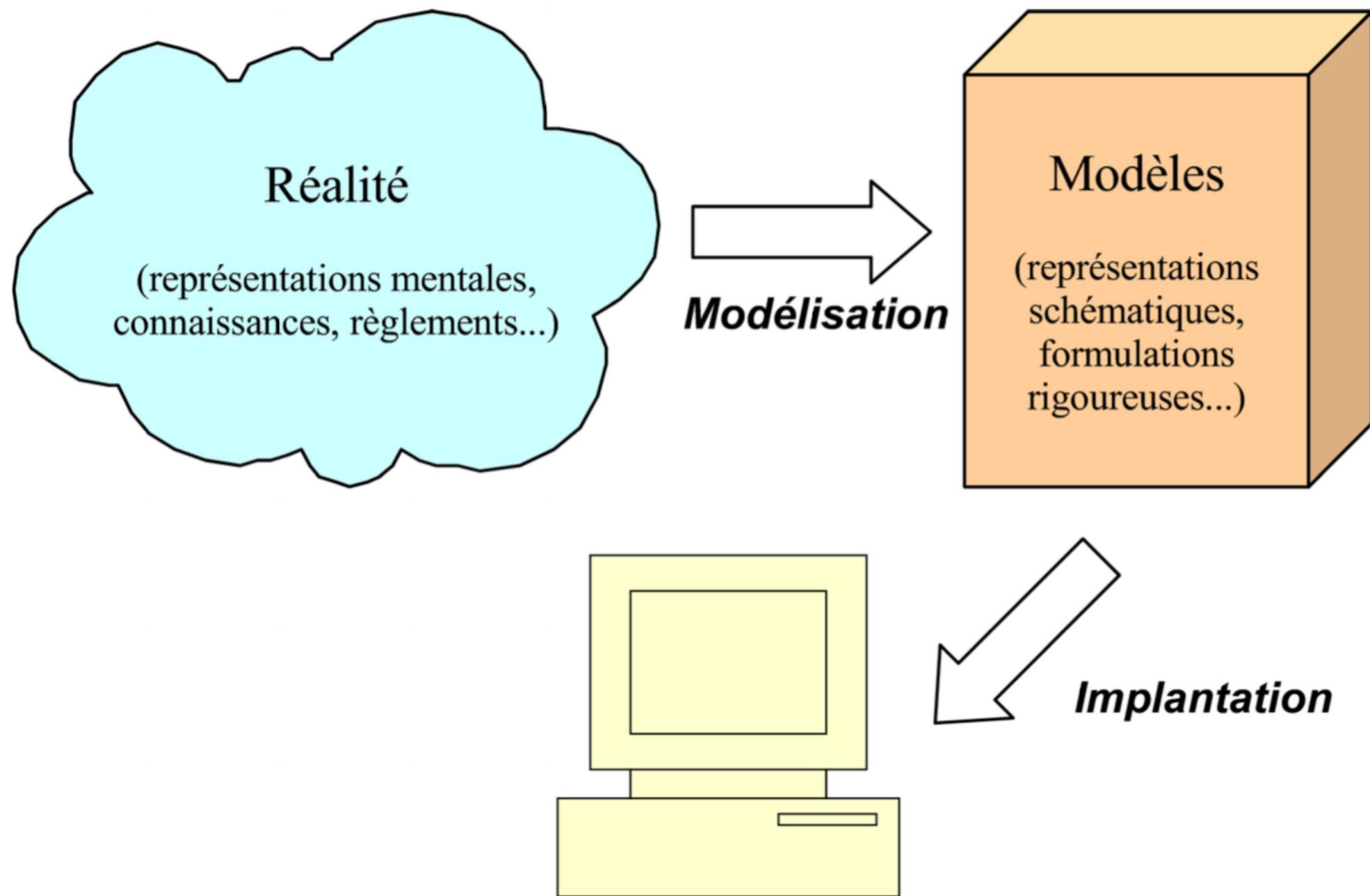
Modèle météorologique

Modèle économique

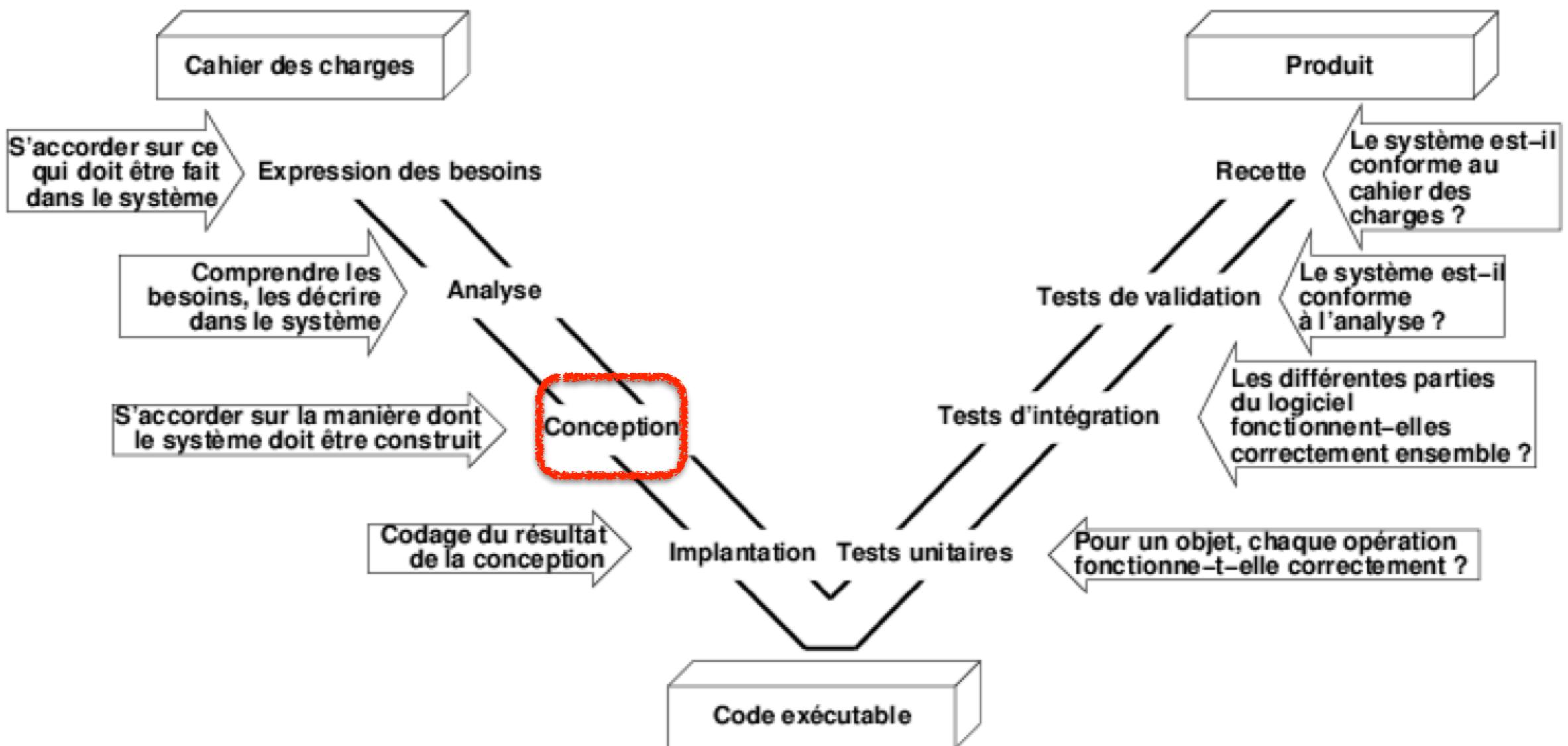
Modèle démographique

Modèle de Systèmes Informatiques

COO: Modélisation



Exemple de cycle de développement



Conception Orientée Objet



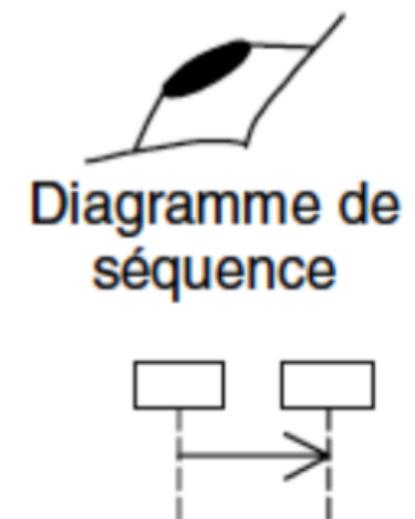
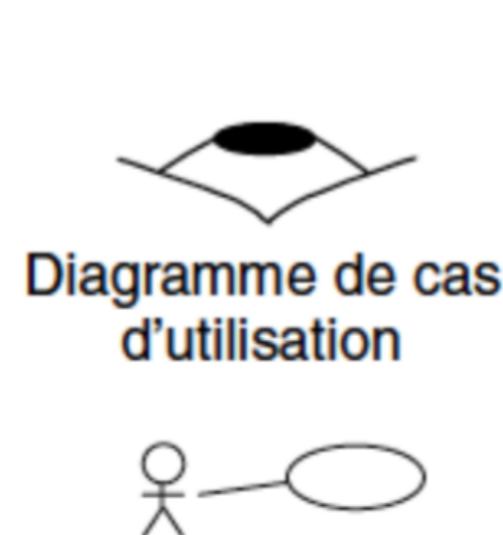
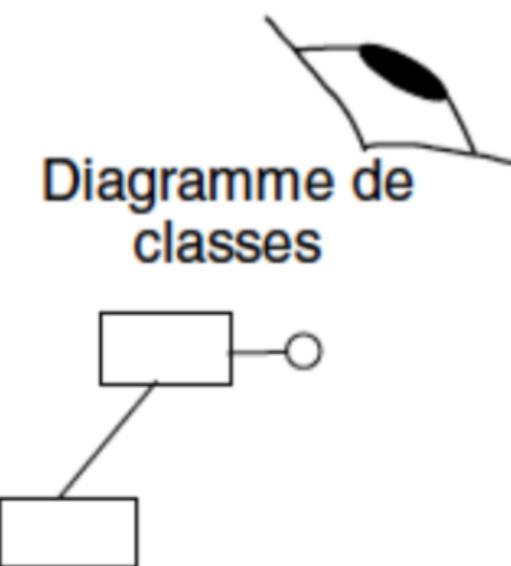
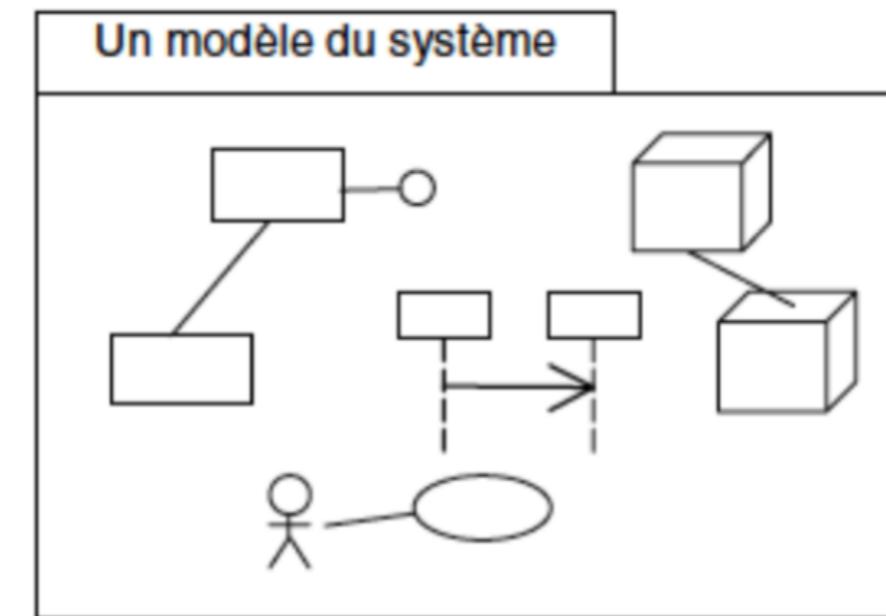
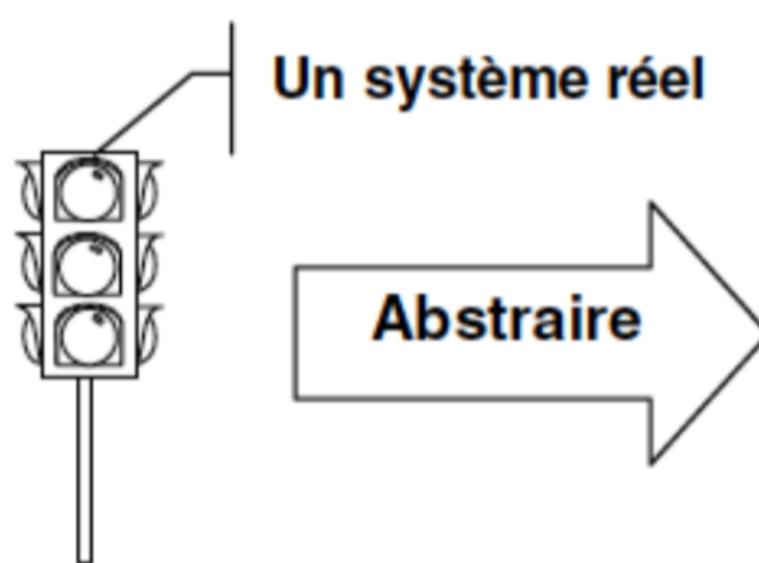
Orienté objet = abstraire et décomposer le système informatique en objets.



Approche orientée objet : on identifie les objets manipulés par le système, avec leurs états et leurs comportements.



Conception Orientée Objet



Conception Orientée Objet



Un objet représente une entité du système qui est caractérisée par:

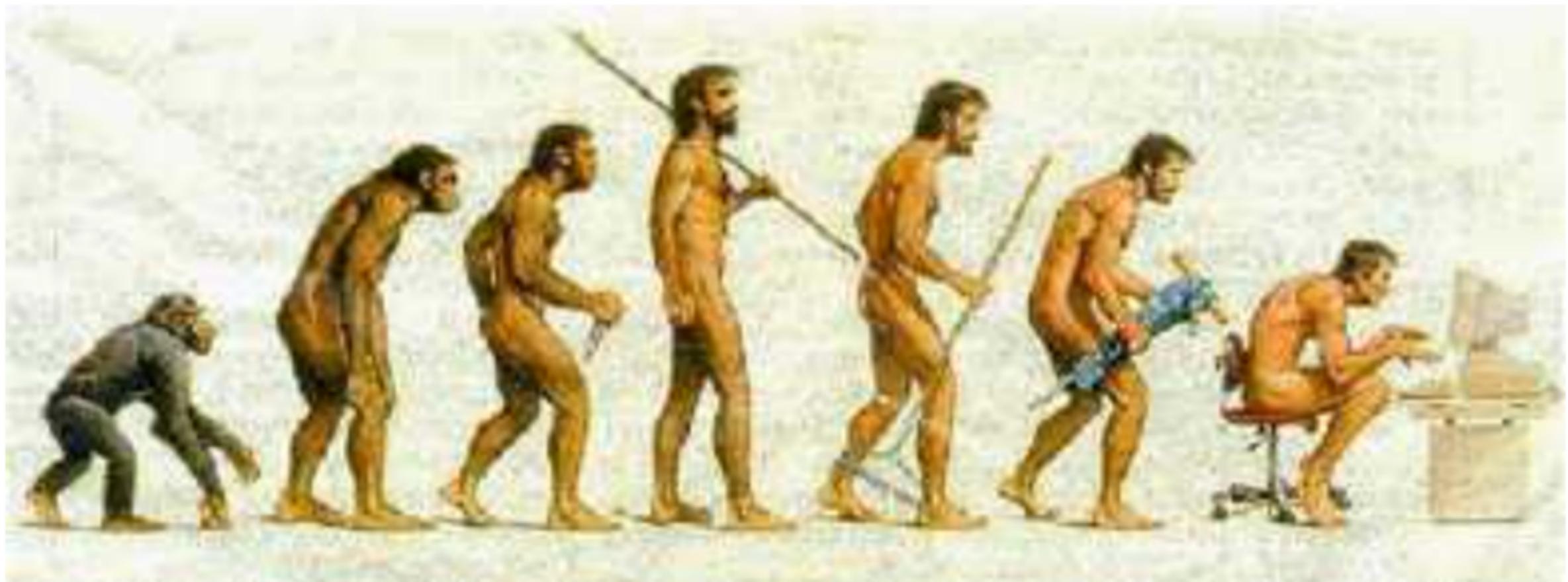
- une identité (ou référence);
- un ensemble d'attributs (propriétés) décrivant son état;
- un ensemble de méthodes (opérations) définissant son comportement.

Conception orientée Objet =>

Construire un modèle basé sur les objets du système.

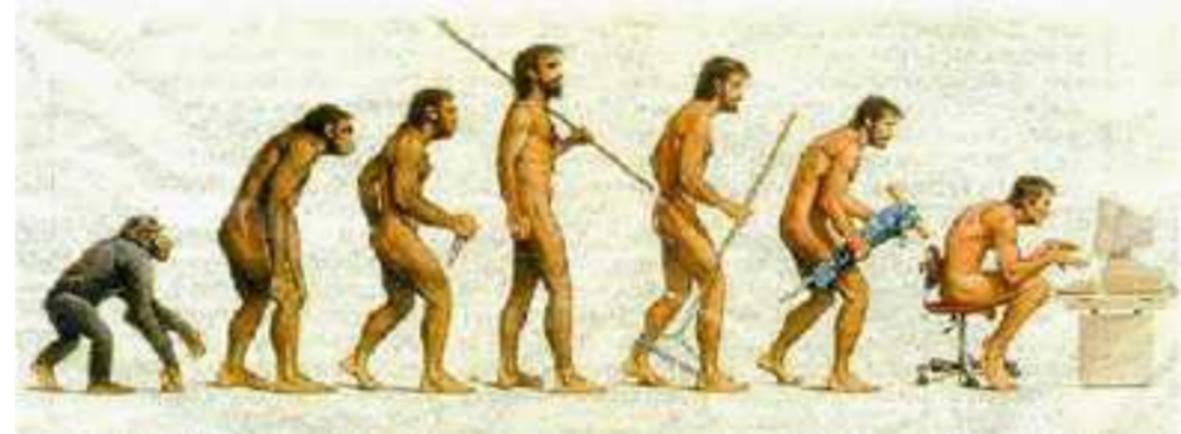


COO: Un peu d'histoire





COO: Un peu d'histoire



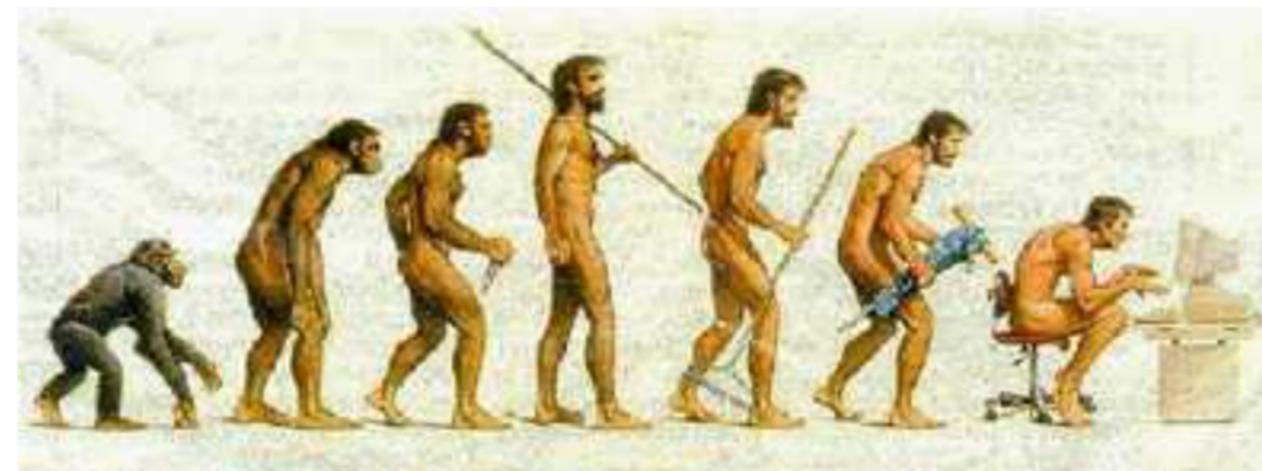
L'idée est connue depuis 1976

Programmation orientée objet : 1980, version industrielle de SmallTalk

Langages de programmation : Simula (1967), Smalltalk (1970), C++ (1985), Eiffel (1988)
...Java(1995), Python, Scala, etc.

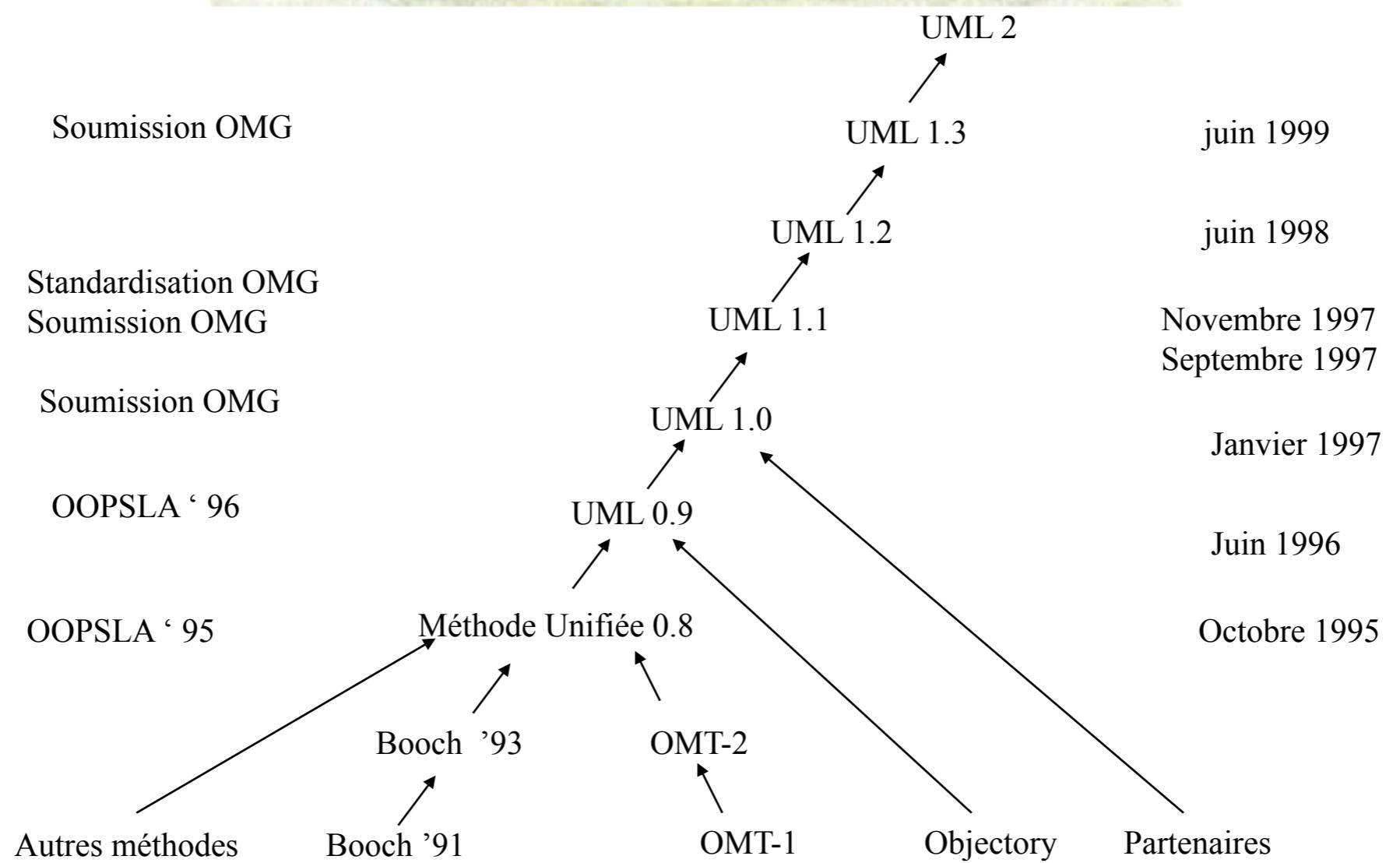
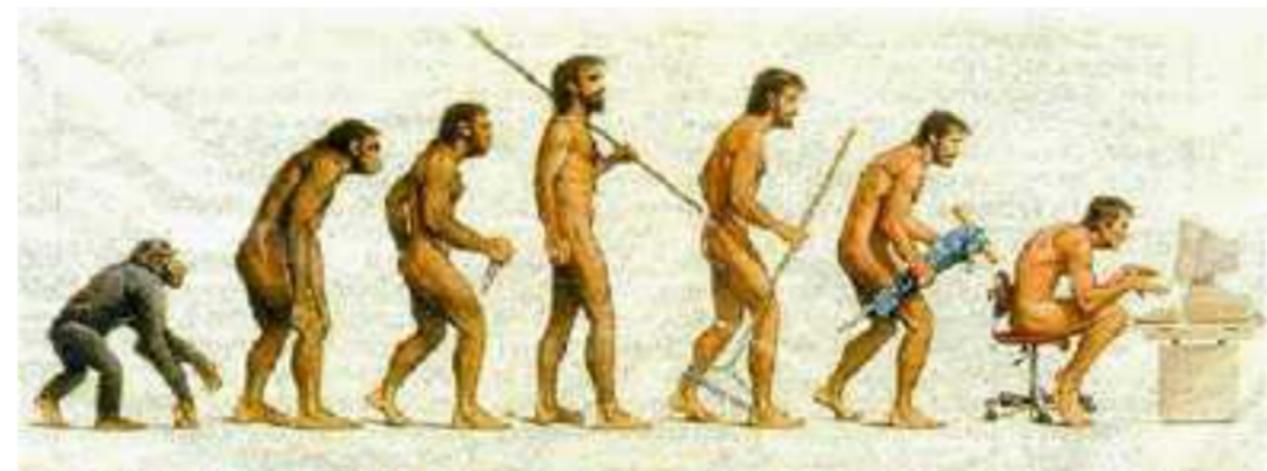
Les outils de POO existent, qu'en est-il de l'analyse et de la conception?

COO: Un peu d'histoire

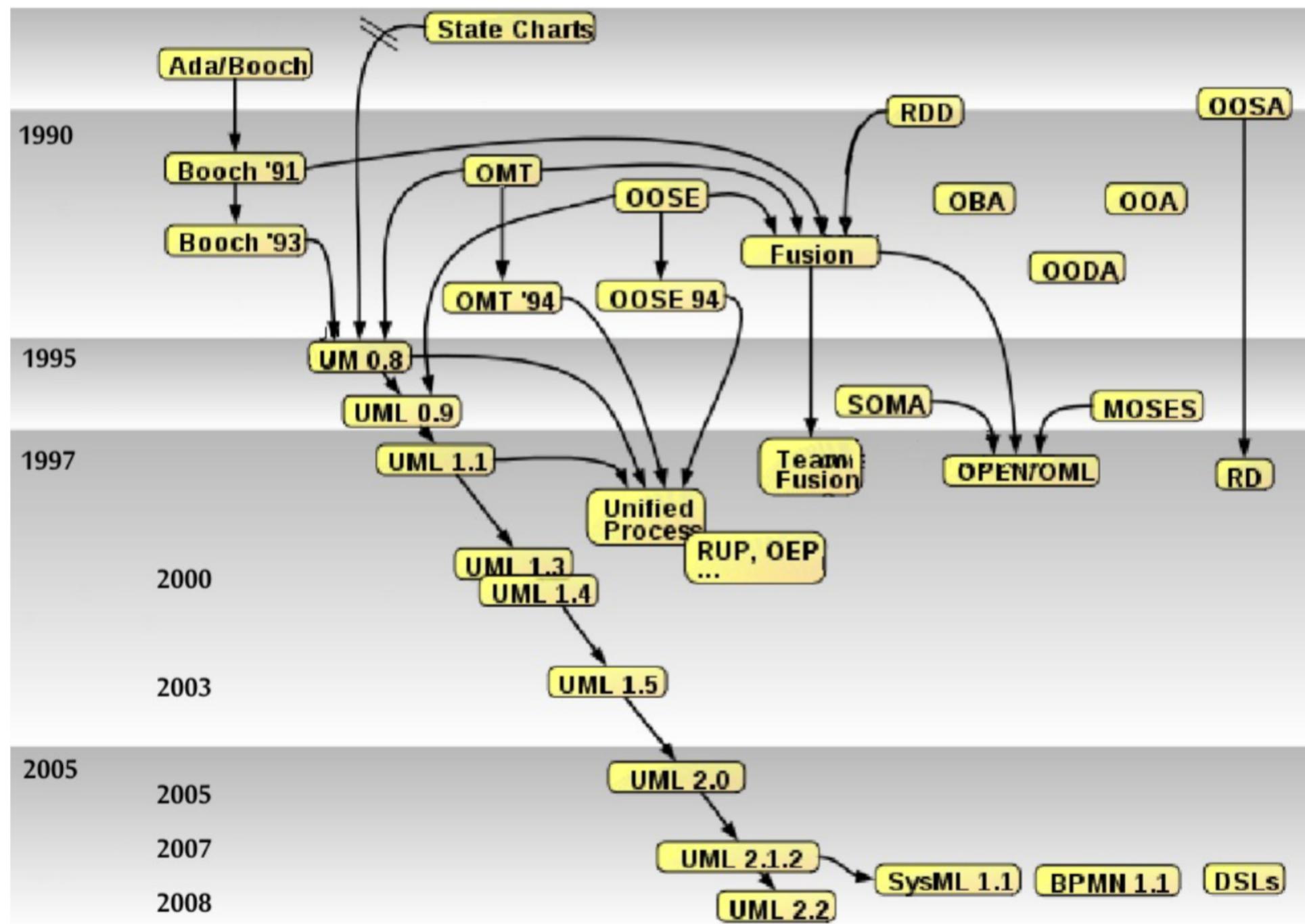
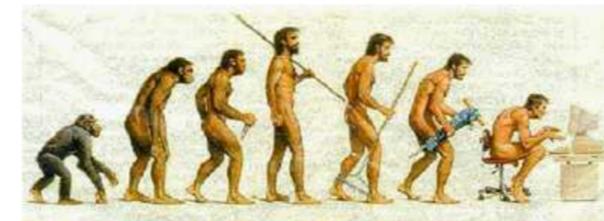


- Méthodes fonctionnelles et Systémiques
 - Années 60 → 90
- Apparitions de méthodes objets : 90
 - Booch, OMT (Rumbaugh), Objectory (Jacobson)
- Insuffisances
 - Méthodes partielles, cantonnées à des domaines distincts
 - « Abondance de biens nuit »

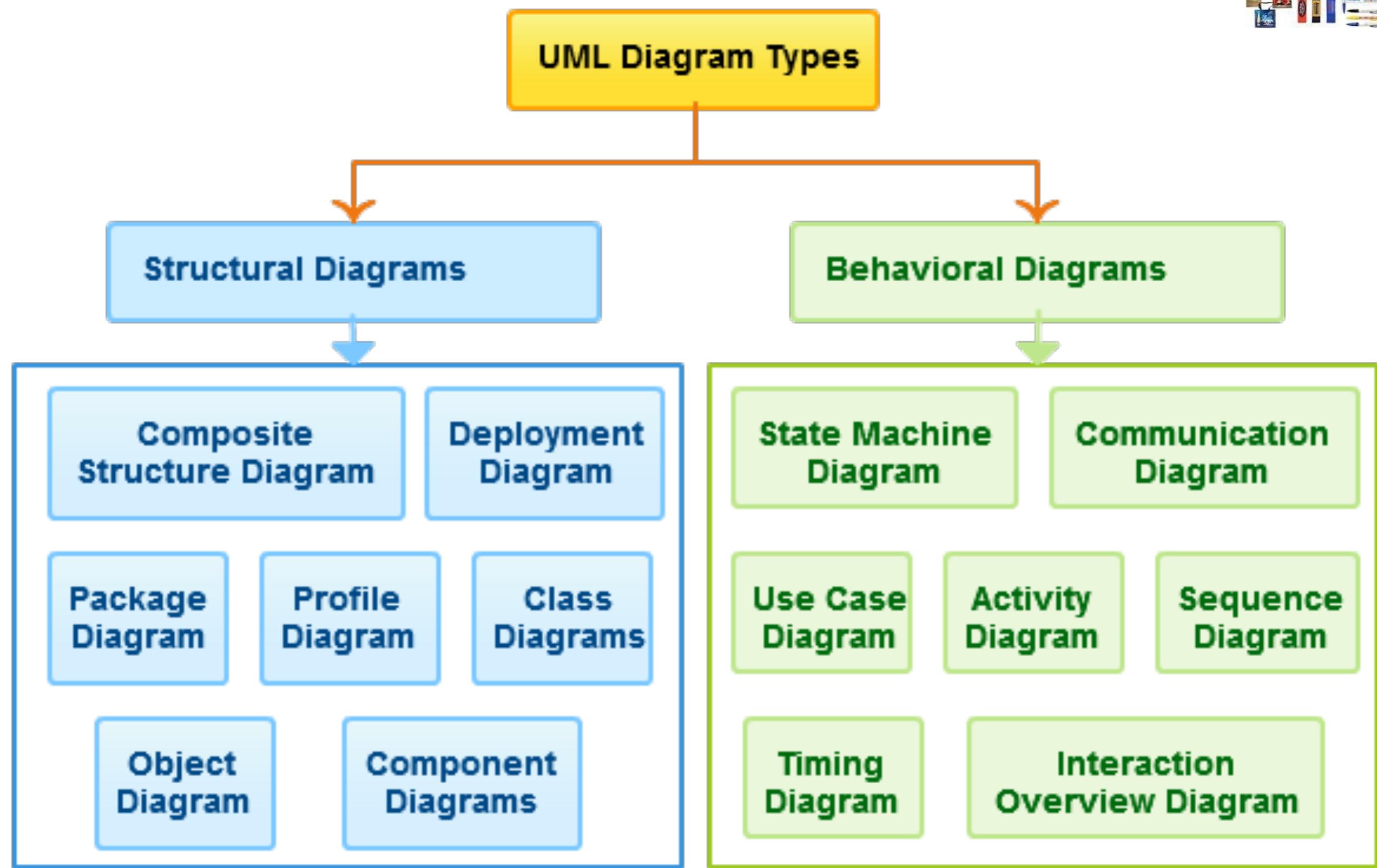
COO: Un peu d'histoire



UML



UML



Take away message



En résumé, l'approche objet c'est :

- Un ensemble de concepts stables, éprouvés et normalisés.
- Une solution destinée à faciliter l'évolution d'applications complexes.
- Une panoplie d'outils et de langages performants pour le développement.

Quelques liens utiles



- <https://www.omg.org/>