

DIGINAMIC, le talent à l'ère digitale

Formation aux métiers du management et du web



« Dynamisez votre carrière digitale ! »

Modélisation des besoins fonctionnels avec UML

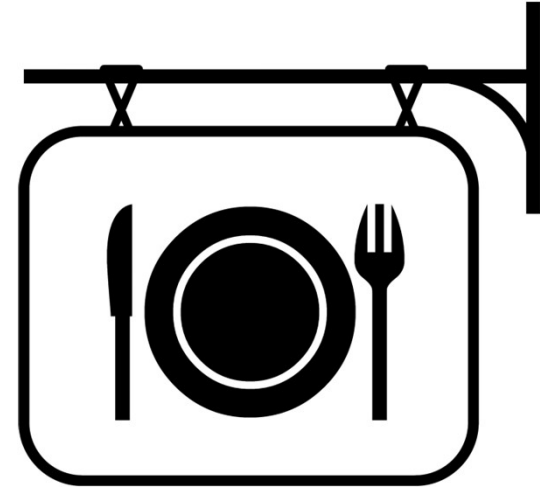
Objectifs pédagogiques du cours

À l'issue de cette formation, vous serez en mesure de :

- Traduire un cahier de charges en concepts UML
- Utiliser les **bases** des diagrammes UML suivants
 - Les diagrammes de cas d'utilisation
 - Les diagrammes de séquence
 - Les diagrammes de classe
 - Les diagrammes d'activités (si le temps nous le permet)

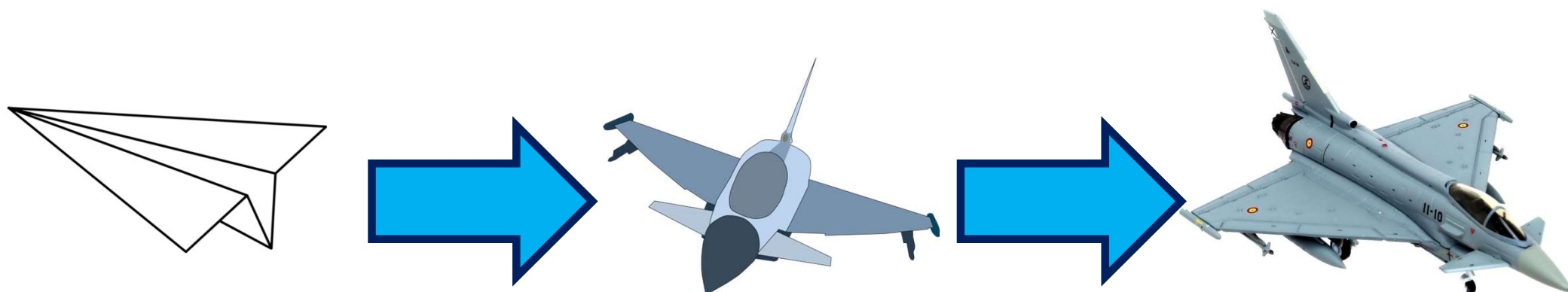


- Au menu du jour :
 - Réflexion sur la modélisation : utilité, outils, méthodes
 - UML... kézako ?
 - Premiers diagrammes :
 - Fonctionnel : le diagramme de cas d'utilisation (*use case*)
 - Statique : le diagramme de classe
 - Dynamique : le diagramme de séquence, le diagramme d'activités
 - Travaux pratiques :
 - La station d'essence « TOTALITY »
 - Un réseau de bibliothèques de communauté de communes
 - Questions et réponses



Réflexion sur la modélisation

- Qu'est ce qu'un modèle?
- Modéliser... pourquoi ?
- Modéliser... avec quel outil?
- Modéliser... avec quelle méthode ?



Qu'est ce qu'un modèle?

Un **modèle** est une représentation abstraite et simplifiée (exclut certains détails) d'une entité (système, phénomène, processus, etc...) du monde réel en vue de le décrire, de l'expliquer ou de le prévoir.

Exemple de modèles:

- Modèle météorologique => prévoir les conditions climatiques pour les jours à venir.
- Modèle économique => évolution du chômage, taux de croissance...
- Plans (de bâtiment, d'écoulement d'eaux...)

Pourquoi modéliser ?

Modéliser un système avant sa réalisation permet de:

- Mieux comprendre son fonctionnement
- Maîtriser sa complexité et assurer sa cohérence
- Communiquer plus efficacement avec les acteurs du projet y compris les développeurs
- Documenter les exigences de manière à faciliter la traçabilité aux artefacts d'architecture et d'implémentation
- Faire des simulations avant la réalisation (MDA...)
- Se distinguer de ses pairs 😊

Modéliser avec quel outil ?

L'idéal est de pouvoir **utiliser un même langage pour décrire les organisations** et ce qui s'y passe, afin de faciliter le travail et la compréhension entre les BA et le reste du monde :

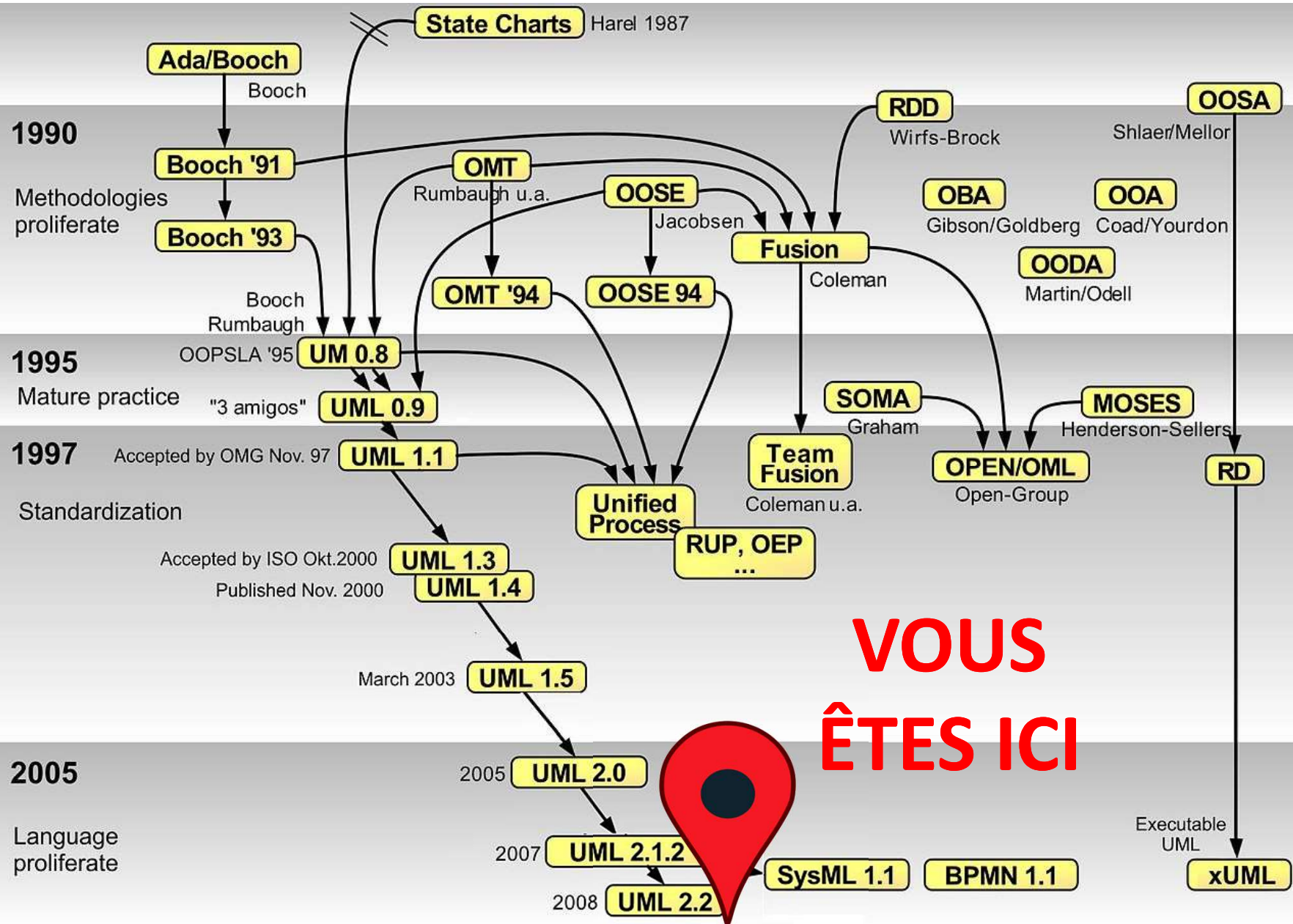
- Qui fait quoi ?
- Dans quel ordre s'effectuent les actions ?
- De quoi dépend chaque action ?
- Que consomme-t-on ? Que produit-on ?

➔ « Un petit dessin vaut mieux qu'un long discours »... et on s'épargne des problèmes de traduction !

Modéliser... avec quel outil ?

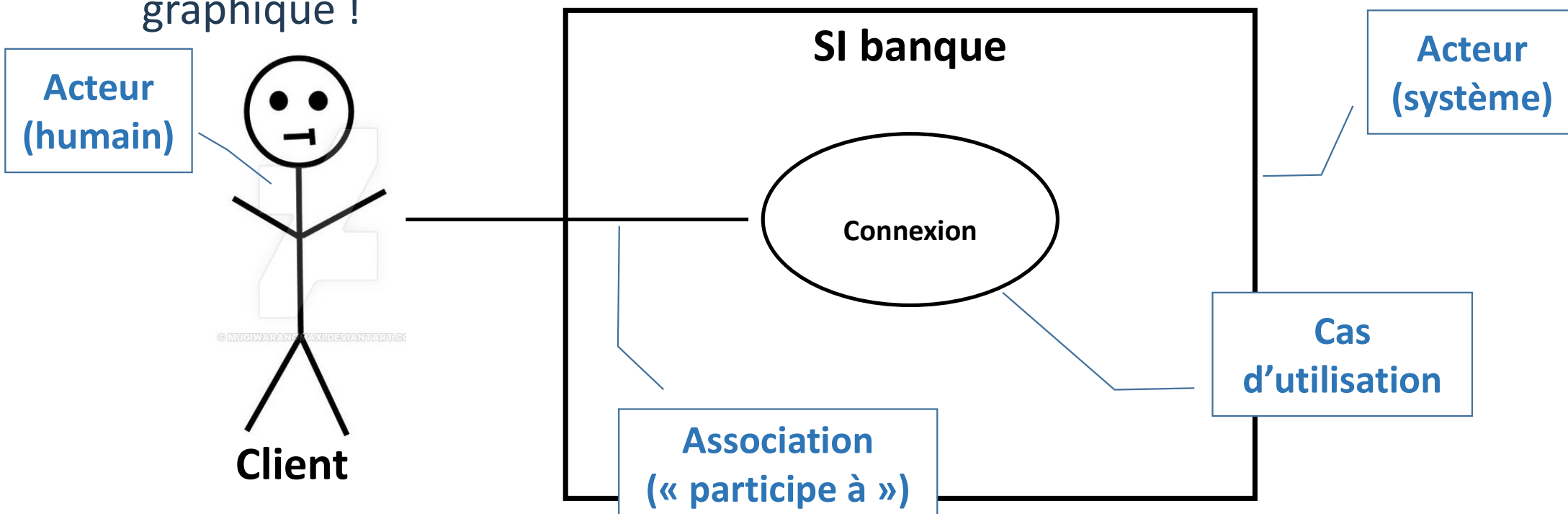
- UML 2 est la version la plus aboutie des travaux visant à produire un langage universel de modélisation des systèmes informatiques.
- UML est l'héritier de nombreux efforts précédents pour modéliser les processus d'entreprise :





Modéliser... avec quel outil ?

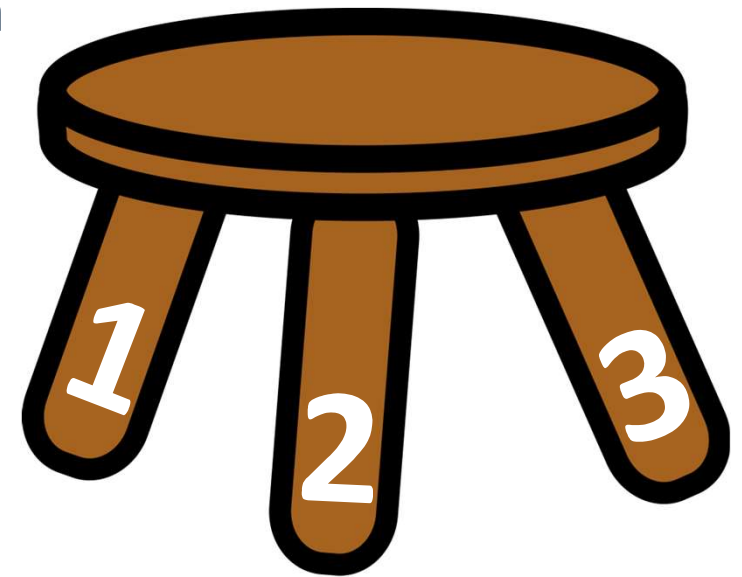
- Le langage UML a sa grammaire, son vocabulaire... essentiellement graphique !



Modéliser... avec quel outil ?

En UML, le minimum à faire pour une modélisation exploitable est de :

1. Réaliser un diagramme de cas d'utilisation
2. Réaliser un diagramme de classes
3. Réaliser un diagramme de séquences



Modéliser... avec quel outil ?

S'il manque un de ces diagrammes...



Modéliser... avec quelle méthode ?

La méthode à employer est fonction des demandes du projet auquel vous collaborez et de l'organisation à laquelle vous participez.

C'est à vous de choisir celle qui s'adaptera aux besoins :
Rational Unified Process, XP, Scrum...

Toutes ont leurs atouts et leurs faiblesses, familiarisez-vous avec et intégrez-les dans votre boîte à outil. Elles ont beaucoup de points communs !

Modéliser... avec quelle méthode ?

En toutes situations, posez-vous les questions suivantes :

- Le coût du travail de modélisation est-il justifié ?



Modéliser... avec quelle méthode ?

En toutes situations, posez-vous les questions suivantes :

- Suis-je bien en train de décrire une solution fonctionnelle ?



Modéliser... avec quelle méthode ?

Quelques conseils de base :

- Allez de la vision la plus large à la vision la plus fine
- Acceptez dès le départ que votre modèle ne sera jamais parfait (et qu'un petit malin y trouvera des défauts 😊)
- Produire un modèle n'est pas la finalité de votre travail : **le modèle est un produit transitoire que l'évolution de l'organisation rend périmé tôt ou tard. Ne vous y accrochez pas !**

Modéliser... avec quelle méthode ?

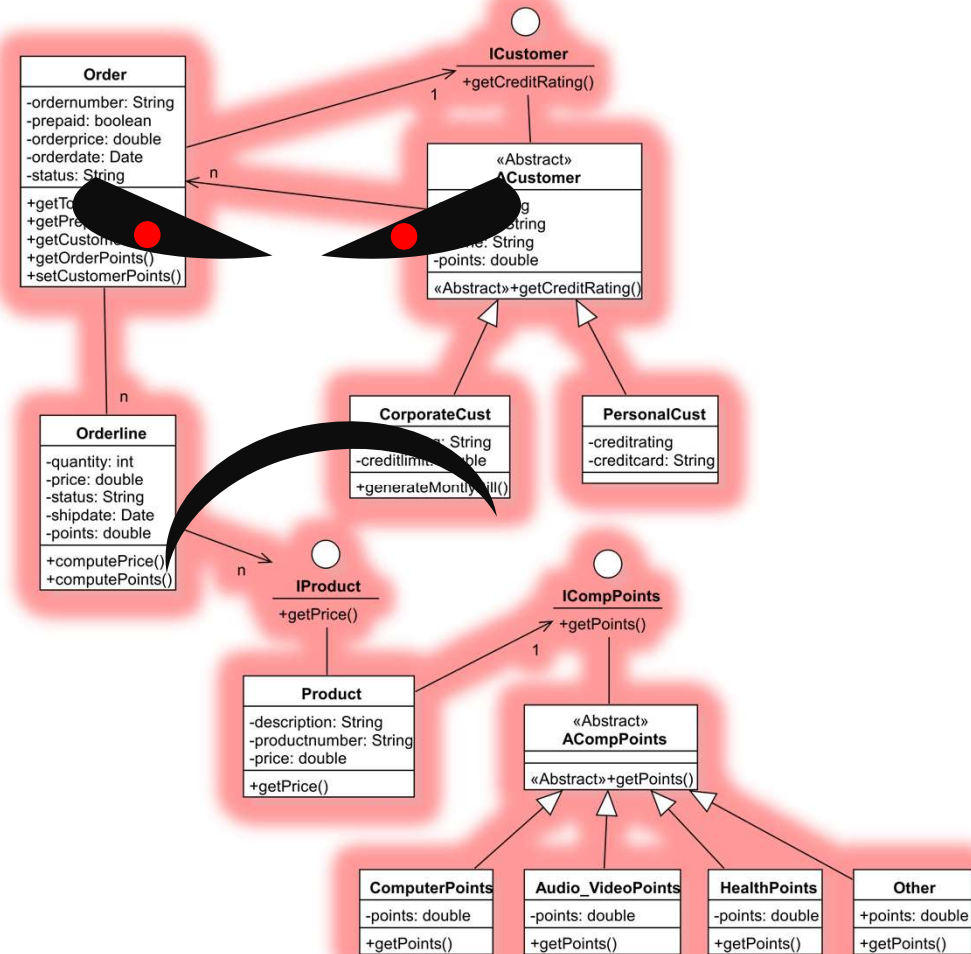
Avoir en mémoire les 3 aspects suivants de la modélisation

- Audience ciblée
 - Détermine le niveau de détails
- But
 - En général 3 buts principaux (Martin Fowler):
 - Croquis, Ebauche (Informel, Incomplet, beaucoup utilisé en agile)
 - **Schéma d'analyse & conception** (Plus complet, plus formel créé avec un outil de modélisation et utilisé comme livrables)
 - Langage de programmation (MDA, Génération de code automatique)
- Périmètre

- Ce qu'est un diagramme
- Ce que n'est pas un diagramme
- Pourquoi il y en a autant dans UML ?



Diagramme... kézako ?



Ce qu'est un diagramme

- Un diagramme UML représente de façon exhaustive une ou plusieurs exigences fonctionnelles d'un système, perçue(s) selon un point de vue déterminé :
 - Point de vue **fonctionnel** : exemple, un cas d'utilisation (ou *use case*) mettant en évidence les acteurs et les différents scénarii du cas d'utilisation
 - Point de vue **statique** : exemple, ensemble des relations d'agrégation et de composition des classes intervenant dans un cas d'utilisation
 - Point de vue **dynamique** : exemple, séquence des actions composant un cas d'utilisation



Ce qu'un diagramme n'est pas

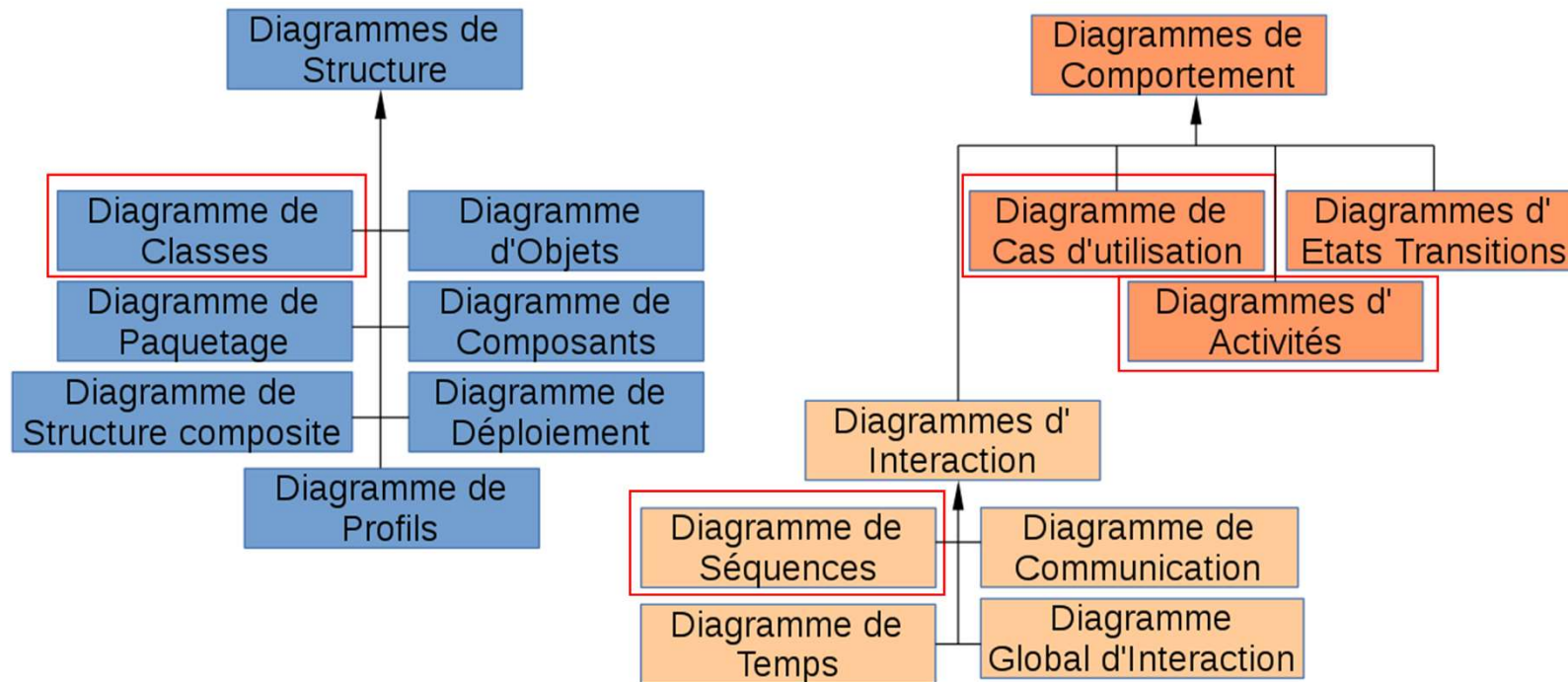
- Un fourre-tout abscons que ni vous ni personne n'est capable de comprendre
- Un objet décoratif à afficher derrière son bureau ou inclure dans un rapport pour faire sérieux
- Un objet sacré que personne d'autre que vous n'a le droit de regarder (et encore moins de critiquer)



Mais pourquoi tant de diagrammes ?

- **Chaque type de diagramme a une raison d'être** : s'ils sont correctement réalisés et saisis dans un outil adéquat, ils permettent d'être (plus ou moins facilement) automatiquement transformés en objets programmables, ce qui raccourcit les temps de développement du système informatique que vous modélisez
- UML v1.3 prévoyait 25 diagrammes différents, **UML v2.5 (2017) n'en comprend plus que 14...** ça progresse !
- Les diagrammes sont dépendants hiérarchiquement et se complètent, **ils constituent un enchaînement logique** et cohérent : ce n'est pas qu'un simple tas de schémas ! (cf illustration ci-après)

Liste des 14 diagrammes UML



Quand les utiliser?

Point de vue/ Diag	Spéc. Fonct	Structure	Spéc. Logique	Matériel	Logiciel	Exploitation	Déploiement	Conf. Logicielle
Classes	●	●	●		●			
Packages		●			●			
Objets		●			●			
Structure Composite		●			●			
Cas d'utilisation	●		●					
Séquence	●		●		●			
Collaboration	●		●		●			
Etats		●			●			
De temps					●			
Activité	●		●		●			
Global d'interactions			●		●			
Composants						●	●	●
Déploiement				●			●	

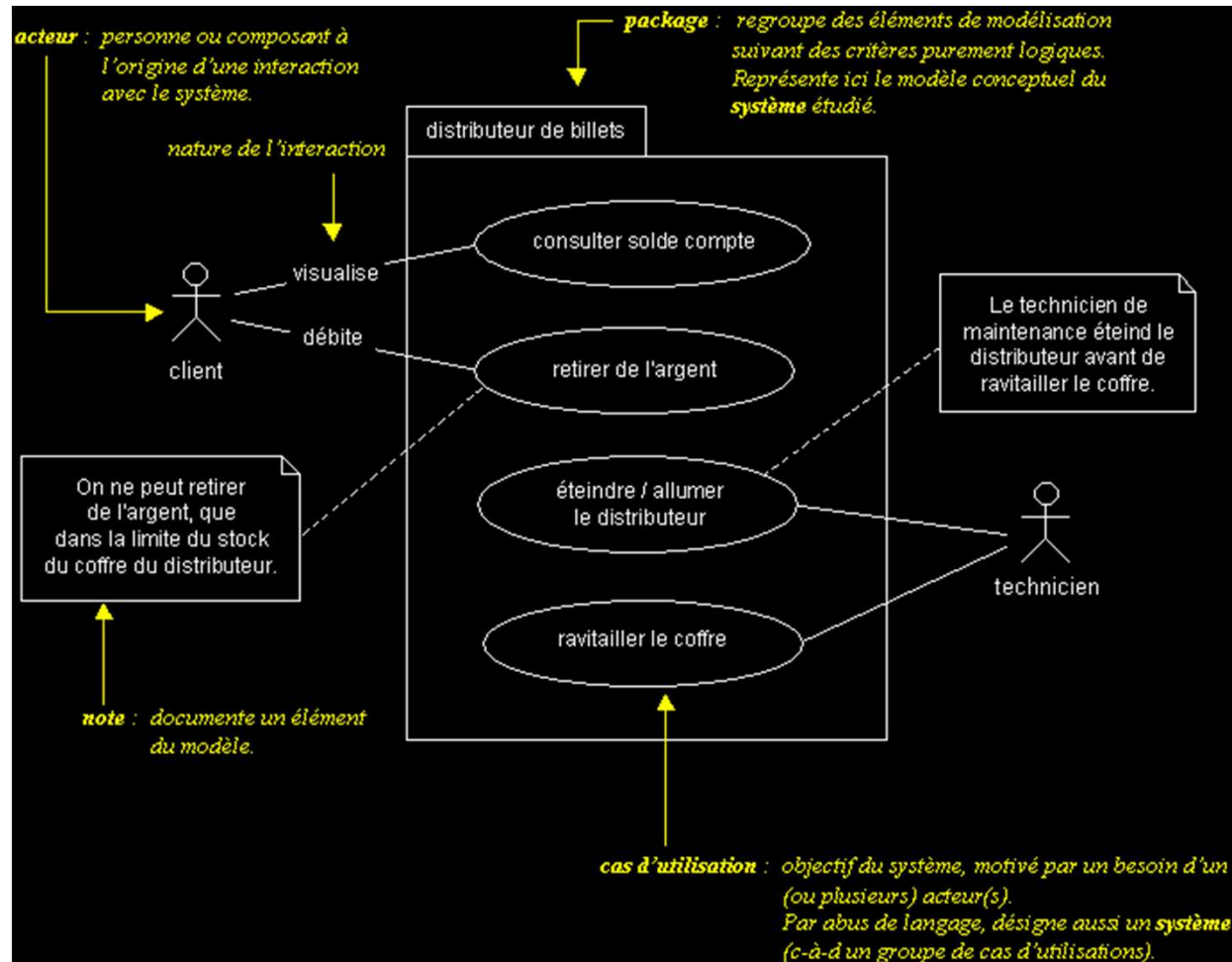
Diag. de cas d'utilisation: Utilité

- Le but principal est de documenter les exigences fonctionnelles d'un système
- Il illustre les limites du métier ou du système ainsi que ses responsabilités
- La description d'un cas d'utilisation décrit les séquences d'étapes entre le système et les différents acteurs (humains et systèmes)
- Illustre les limites du système

Diag. de cas d'utilisation: Concepts

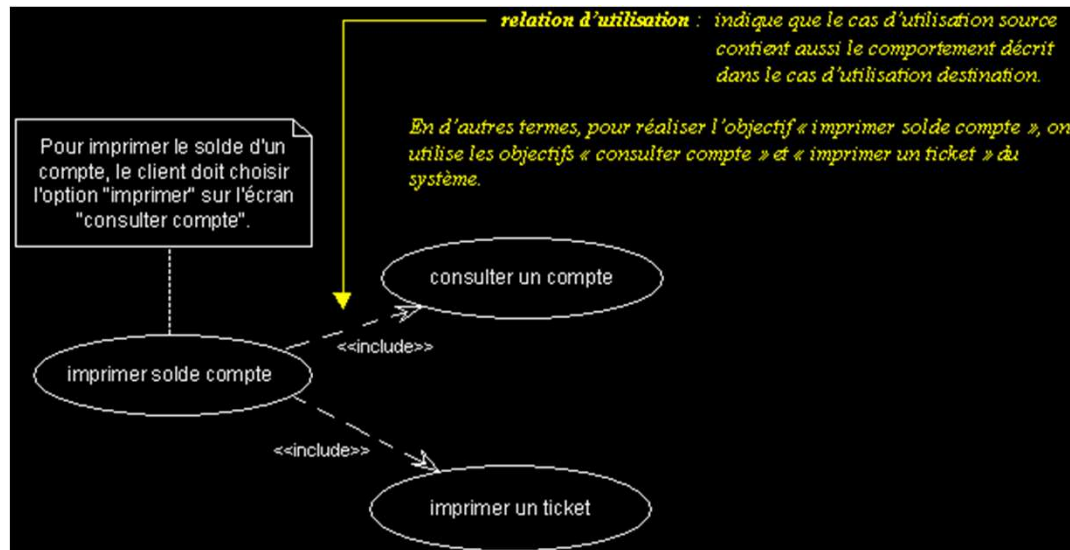
- Acteur
- Cas d'utilisation
- Les relations entre acteurs et cas d'utilisation
- Les relations entre acteurs
- Les relations entre cas d'utilisation
- Les paquetages
- Périmètre du système

Diag. de cas d'utilisation: Exemple

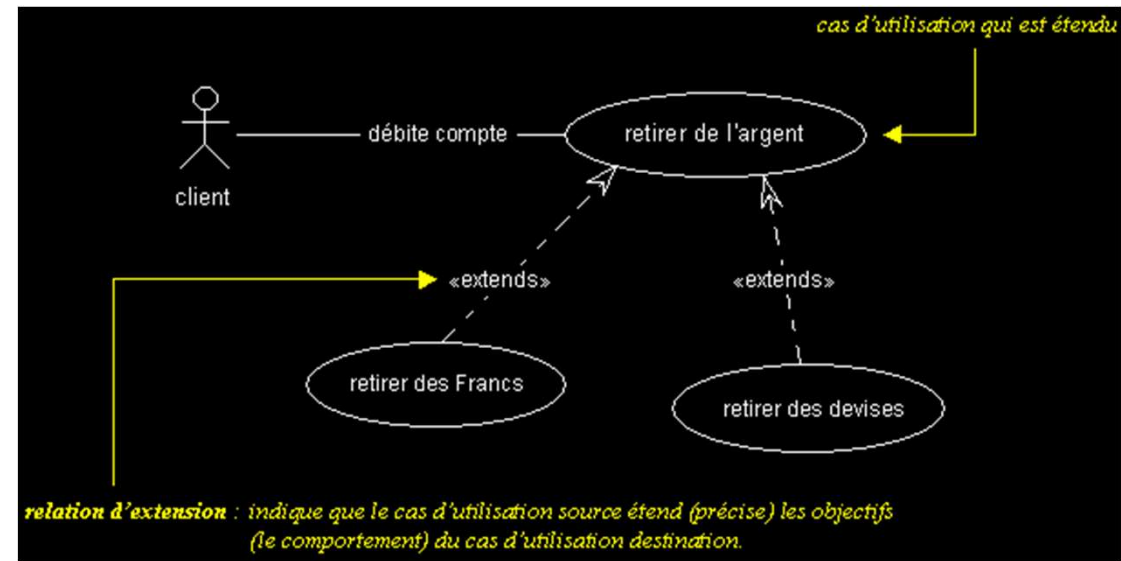


Diag. de cas d'utilisation: Relations

Relations d'inclusion

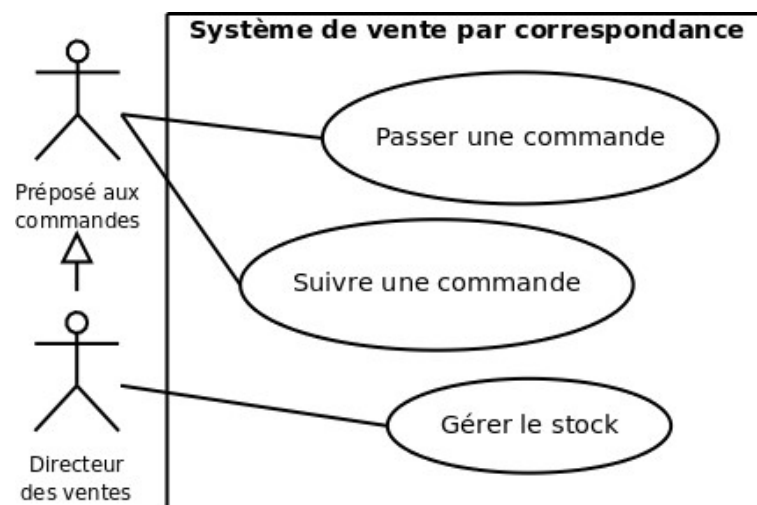


Relations d'extension



Diag. de cas d'utilisation: Relations

Généralisation



Cahier des charges

STATION SERVICE TOTALITY

- Rédiger le diagramme de cas d'utilisation du système « TOTALITY »