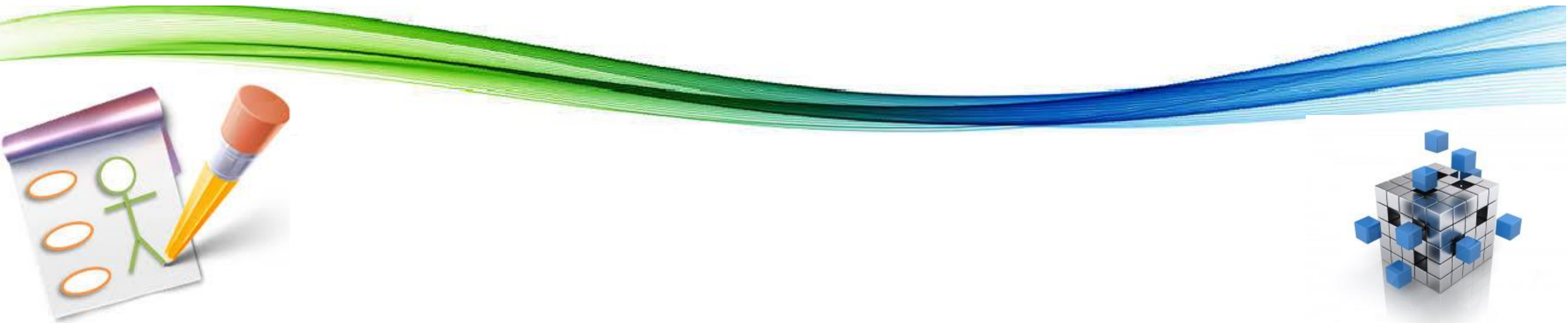


Introduction à UML 2

Zineb BOUGROUN

Bougroun.zineb@gmail.com



Plan

- Cycle de vie
- Modéliser
- UML historique
- Différence entre UML 1 et UML 2



Software life cycle

- Le *cycle de vie d'un logiciel* désigne toutes les étapes du développement d'un logiciel, de sa conception à sa disparition.
- L'origine de ce découpage provient du constat que les erreurs ont un coût d'autant plus élevé qu'elles sont détectées tardivement dans le processus de réalisation.
- Le cycle de vie du logiciel comprend généralement les étapes suivantes :
 - Analyse des besoins et faisabilité
 - Conception générale
 - Conception détaillée
 - Codage (Implémentation ou programmation)
 - Tests unitaires
 - Intégration
 - Qualification (ou recette)
 - Documentation
 - Mise en production
 - Maintenance

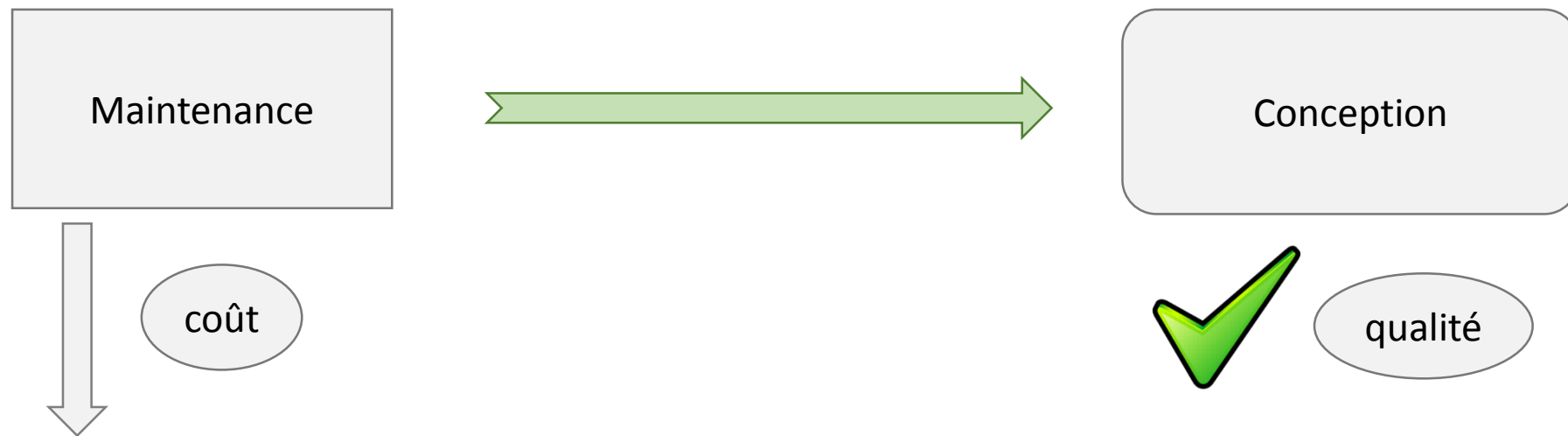


The maintenance

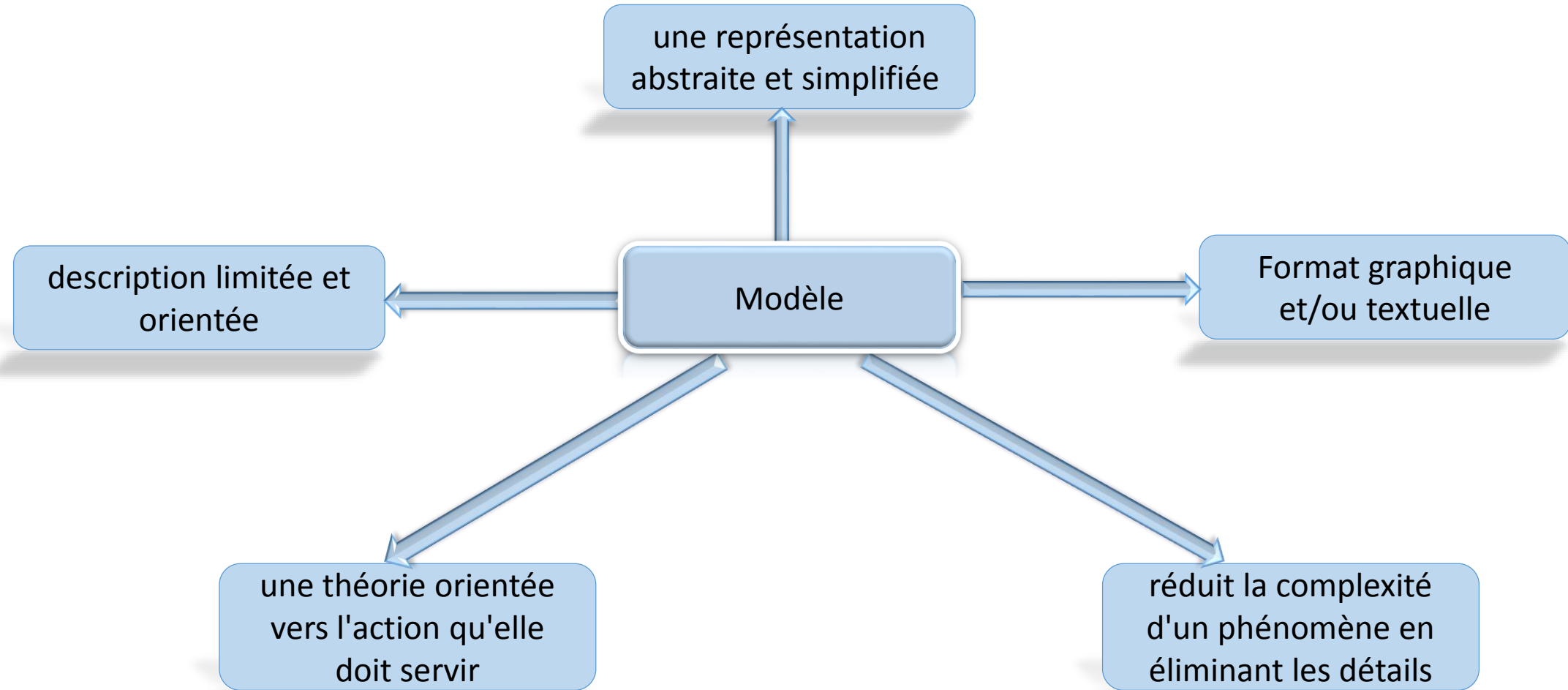
- La maintenance est devenue une facette très importante du cycle de vie d'un logiciel.
- une enquête effectuée aux É.U. en 1986 auprès de 55 entreprises révèle que 53% du budget total d'un logiciel est affecté à la maintenance.
- Ce coût est réparti comme suit :
 - 34% maintenance évolutive (modification des spécifications initiales) ;
 - 10% maintenance adaptative (nouvel environnement, nouveaux utilisateurs) ;
 - 17% maintenance corrective (correction des bogues) ;
 - 16% maintenance perfective (améliorer les performances sans changer les spécifications) ;
 - 6% assistance aux utilisateurs ;
 - 6% contrôle qualité ;
 - 7% organisation/suivi ;
 - 4% divers.



To Save the Budget?



Un modèle, c'est quoi?



Why model?

- Organiser ses idées, les documenter, puis organiser la réalisation en définissant les modules et étapes de la réalisation.
- Cette démarche antérieure à l'écriture est la modélisation ; son produit est un modèle.
- Décrit sous différents angles en utilisant des modèles différents
- On modélise pour avoir une meilleure vue du système qu'on souhaite développer.

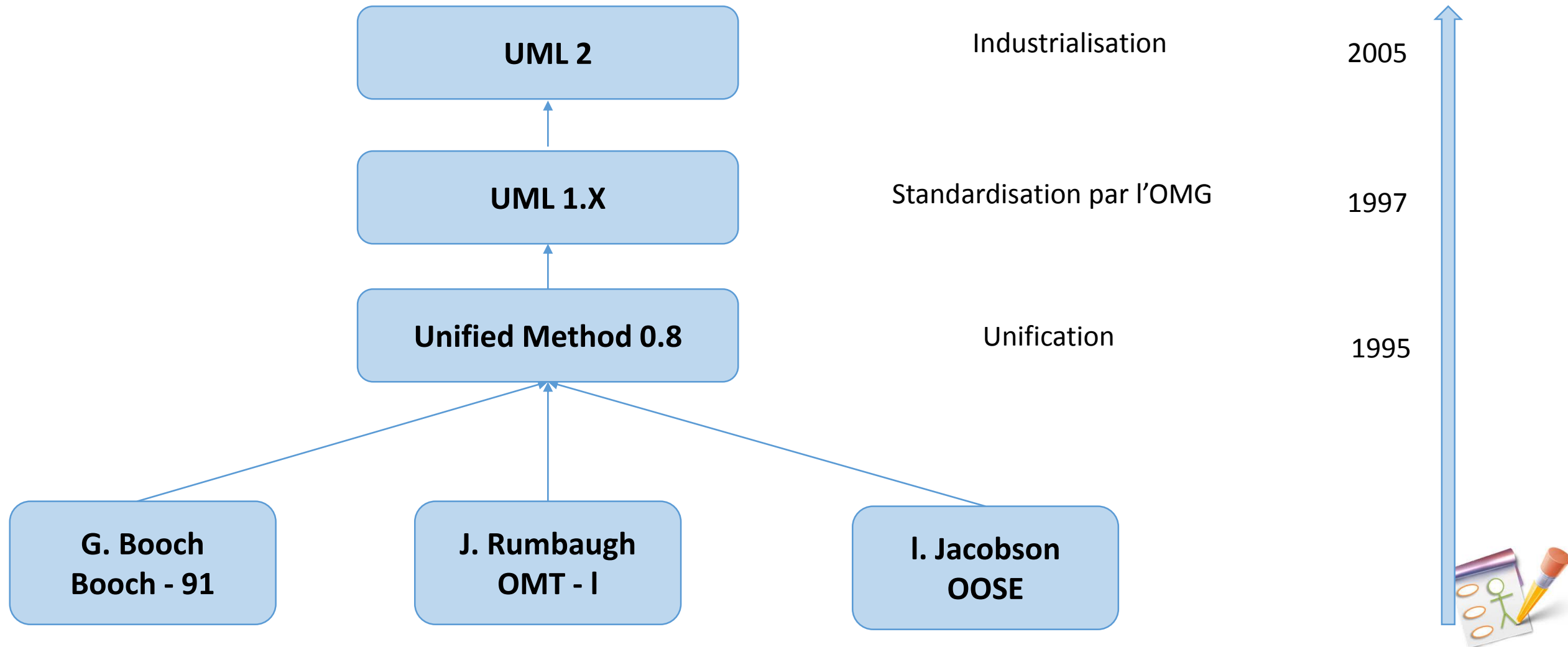


Why model?

- Un modèle décrit toujours une réalité observable, mesurable avec un certain filtre; utilisant des concepts pour classifier cette réalité.
- La réalité observable peut être décrite en répondant aux questions :
 - Quoi ? (Objets)
 - Qui ? (Tiers)
 - Quand ? (Éléments temporels)
 - Où ? (Endroits)
 - Comment ? (Actions)
 - Contraintes des limites du modèle



UML history



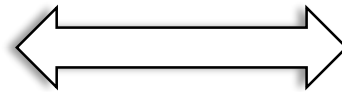
UML Unified Modeling Language

- UML n'est pas une méthode dans la mesure où elle ne présente aucune démarche. A ce titre UML est un formalisme de modélisation objet
- UML n'est donc pas une méthode ou un processus; en raison de la diversité des cas particuliers
- UML propose un ensemble de notations pour que chacun ait à sa disposition les éléments nécessaires à la conception d'une application
- UML est donc un métalangage : il fournit les éléments permettant de construire le modèle;
 - les éléments de modélisation (les concepts manipulés par le langage),
 - la sémantique de ces éléments (leur définition et le sens de leur utilisation).



UML : Diagrams

- UML 1
- structural modeling
 - class diagrams
 - object diagrams
 - component diagrams
 - deployment diagrams
- behavioral modeling
 - use case diagrams
 - collaboration diagrams
 - sequence diagrams
 - State machine diagrams
 - activity diagrams



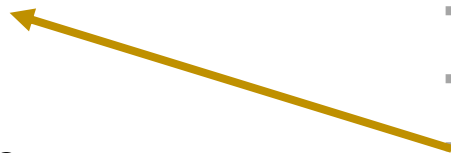
UML 2

structural modeling

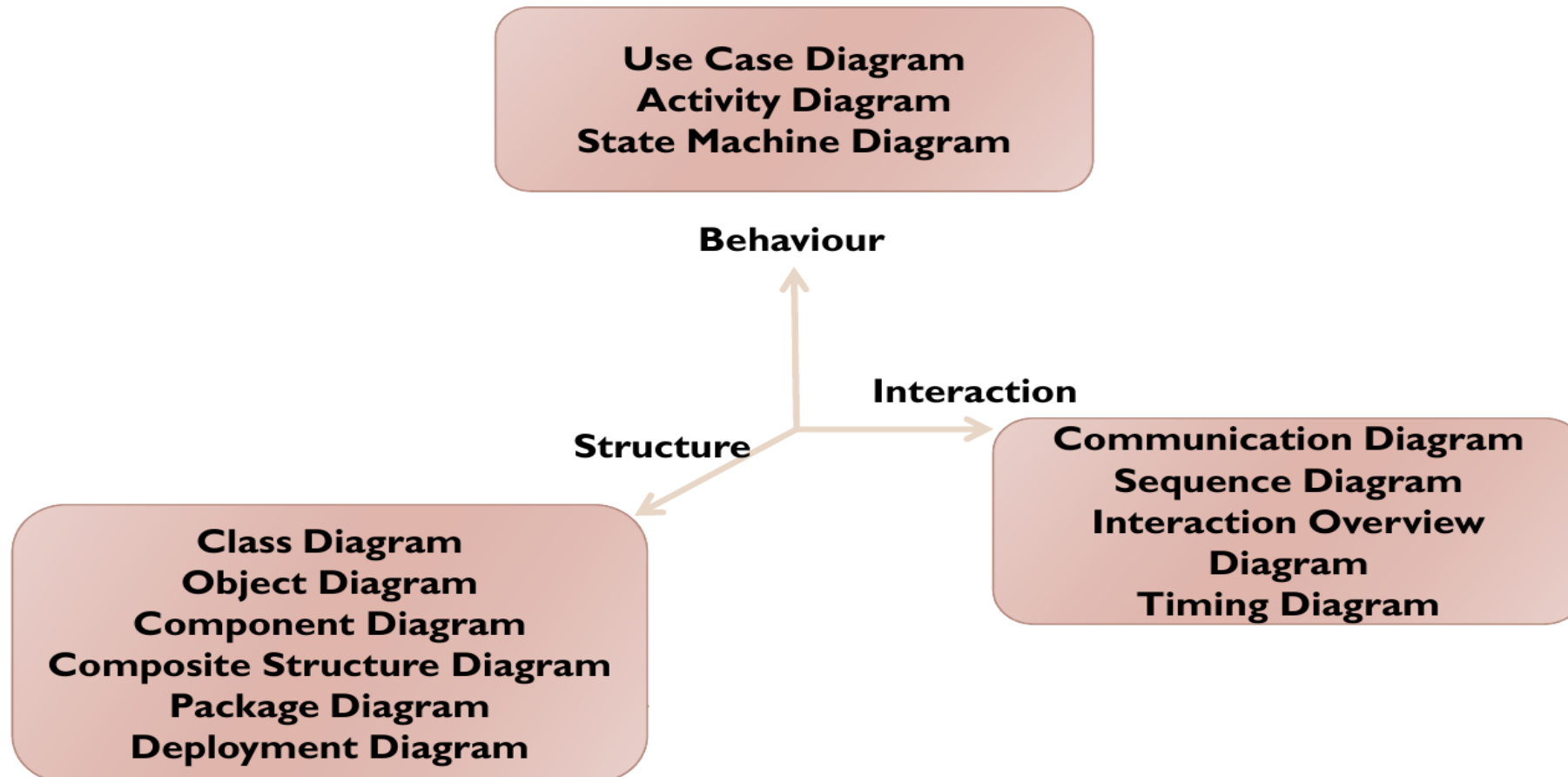
- class diagrams
- object diagrams
- component diagrams
- deployment diagrams
- composite structure diagrams
- package diagrams

behavioral modeling

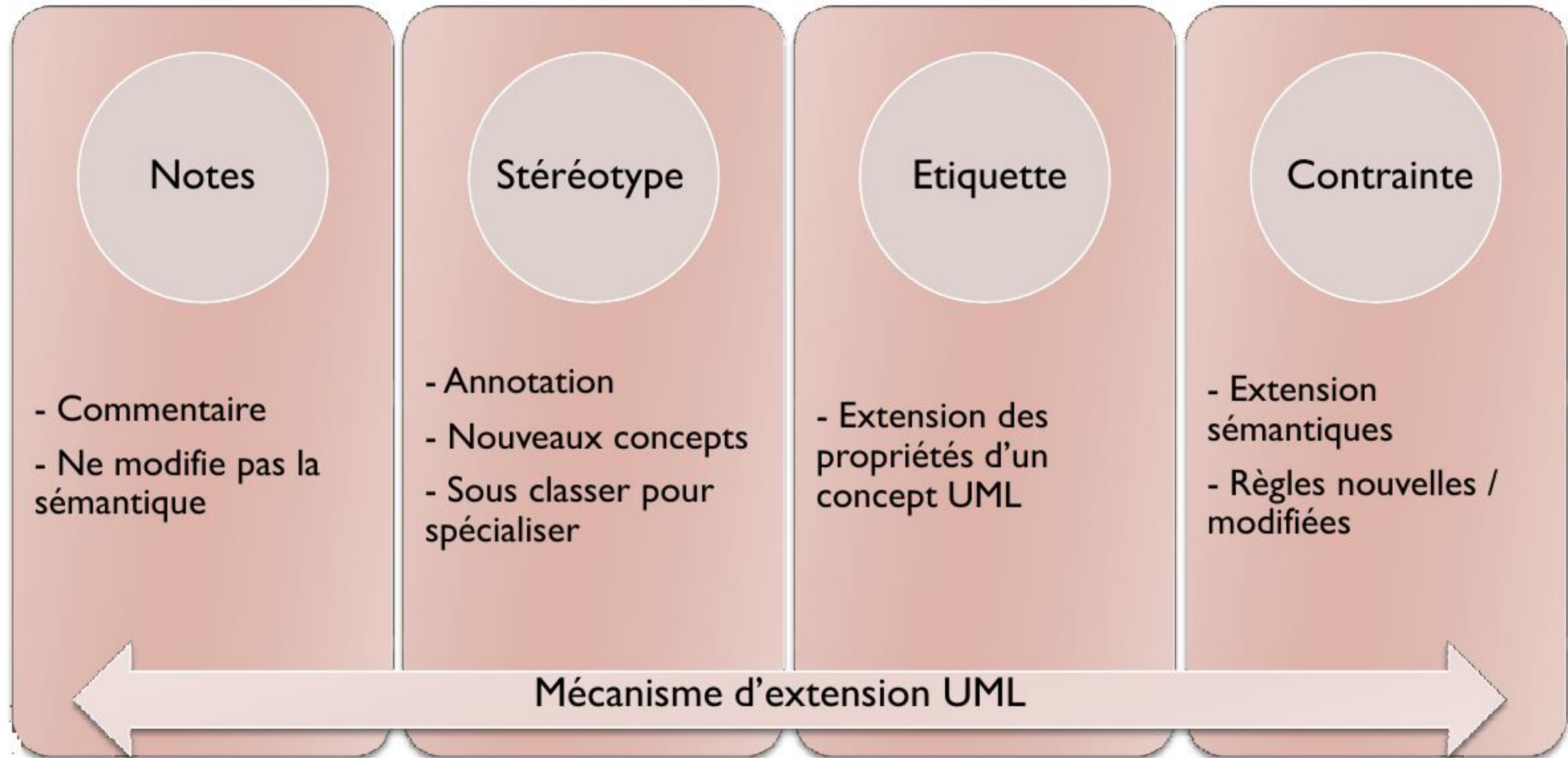
- use case diagrams
- sequence diagrams
- state machine diagrams
- activity diagrams
- communication diagrams
- interaction overview diagrams
- timing diagrams



UML : Diagrams

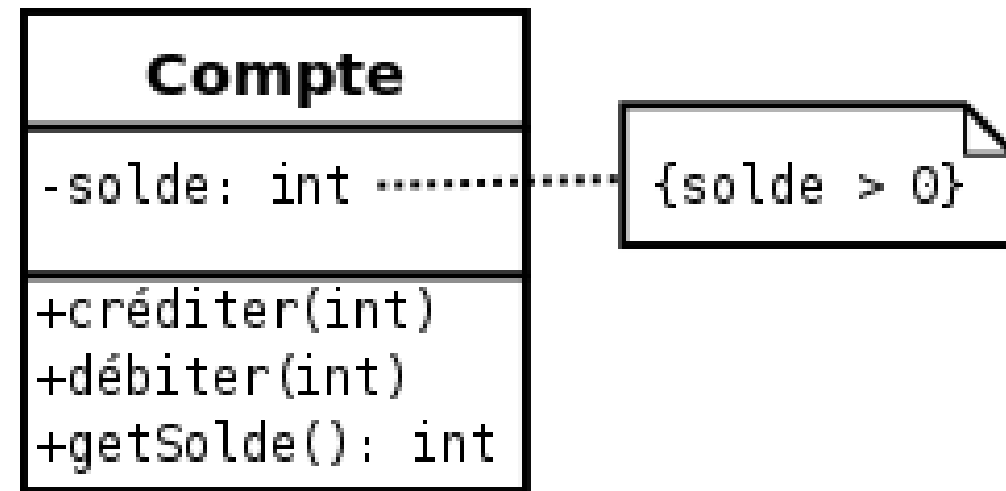


Mécanisme d'extension



Mécanisme d'extension : Notes

- Une note contient une information textuelle comme un commentaire, un corps de méthode ou une contrainte.
- Graphiquement, elle est représentée par un rectangle dont l'angle supérieur droit est plié.
- On peut relier une note à l'élément qu'elle décrit grâce à une ligne en pointillés. Si elle décrit plusieurs éléments, on dessine une ligne vers chacun d'entre eux.



Mécanisme d'extension : Stéréotype

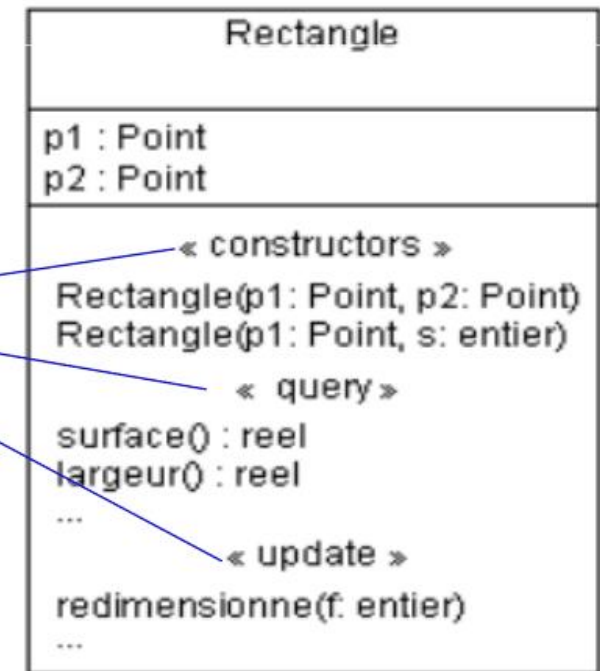
- Définition
- un **stéréotype** est rendu comme un nom encadré par des guillemets et placé au-dessus du nom d'un autre élément. Par exemple, dans un diagramme de classe les **stéréotypes** peuvent être utilisés pour classer les méthodes par comportement telles que « constructor » et « getter ».
- Autrement dit:
- Annotation s'appliquant sur un élément de modèle.
- Mieux caractériser des variétés d'un même concept.
- Sous classer un élément pour le spécialiser en plus fin.
- Ne change pas les caractéristiques fondamentales des éléments auxquels ils sont appliqués.
- Représenté par une chaîne de caractères entre :
 - <<guillemets >>



Mécanisme d'extension : Stéréotype

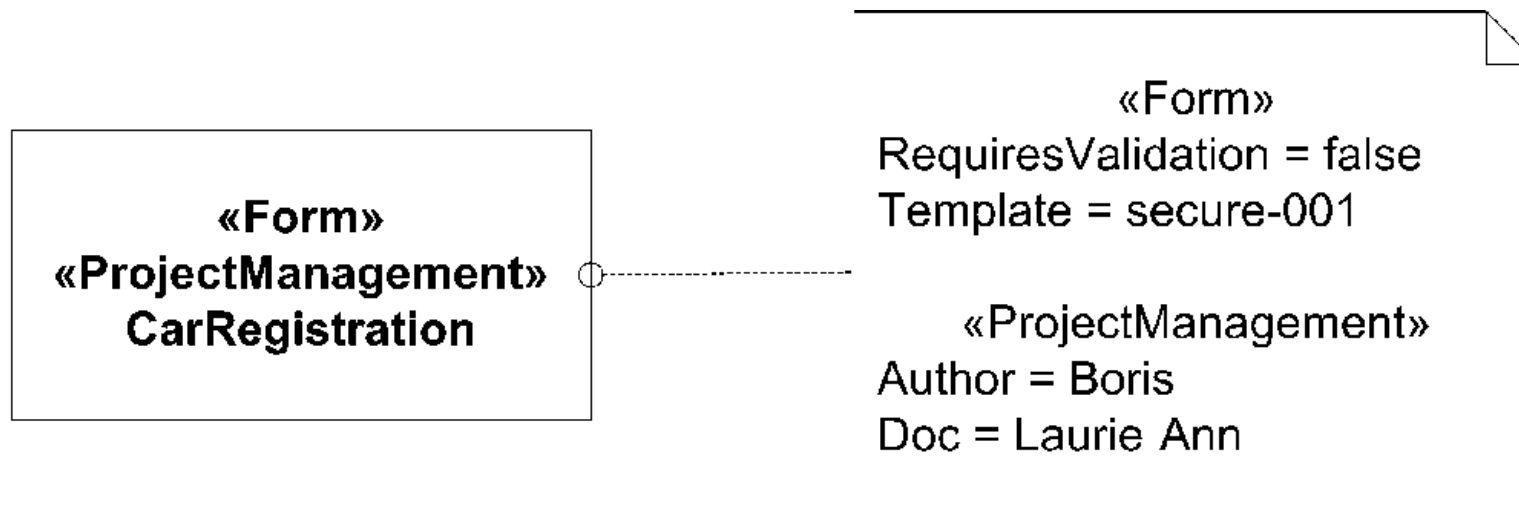
- UML propose des stéréotype prédéfinis :
 - «interface» : interface
 - «énumération» : instances appartiennent à un ensemble fini de littéraux
 - «type primitif» : instances sont des valeurs d'un type primitif
 - «utilitaire» : variables et procédures globales

Stéréotype



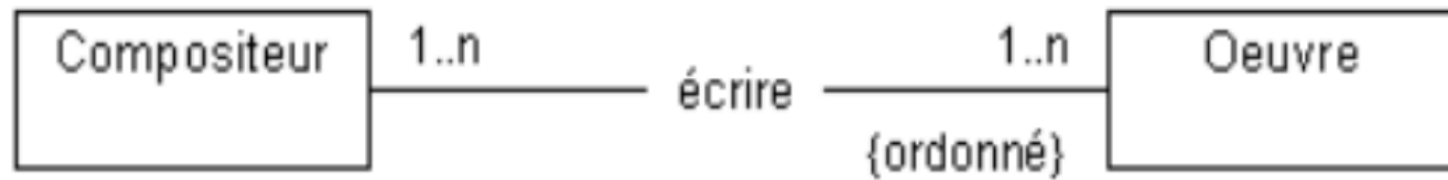
Mécanisme d'extension : Etiquettes

- Les 'tagged values' permettent la création de nouvelles informations dans la spécification d'un élément.
- Représentation : paire {nom, valeur}.
- Elles sont souvent utilisés dans les AGL pour indiquer des propriétés relatives à la génération de code.



Mécanisme d'extension : Contraintes

- Chaîne de caractères associée à un élément UML permet d'ajouter de nouvelles règles ou de modifier des règles existantes
- Elles peuvent être exprimées en texte libre ou en utilisant Object Constraint Language (OCL) d'UML.



Mettre l'accent sur l'importance de l'ordre de création des œuvres d'un compositeur



Mécanisme d'extension

