



UML pour la modélisation de systèmes d'information

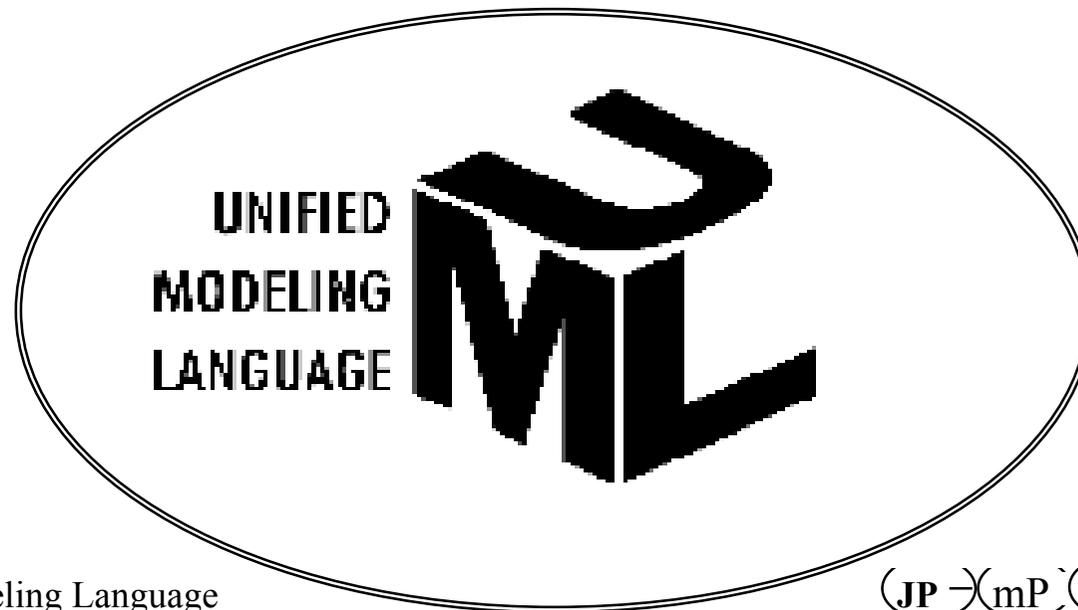
Bilan

Architecture

Processus

Vue globale des éléments

Conclusion



Ar.1. Quelle architecture pour structurer les objets du modèle?

2

- Une architecture pour
 - Le système d'information : diagramme fonctionnel et organisationnel en *sous-systèmes*
 - Le logiciel du système d'information : diagramme de *classes et packages*

mais aussi pour

- La modélisation elle-même : packages de modélisation
- Découpage d'un sous-systèmes en
- Package des use-cases
 - Package des classes et objets

À tous les niveaux : description des utilisations et visibilité entre packages.

Ar.2. Une architecture de modélisation

- Package racine
 - Diagramme des flux entre grandes fonctions
 - Sous-systèmes fonctionnels
- Sous-systèmes fonctionnels
 - Packages des use-cases
 - Diagramme de use-cases
 - Use cases
 - » 1 Collaboration pour 1 scénario
 - diagramme de séquence acteur/système
 - diagramme de collaboration interne
 - diagramme de séquence interne
 - » Éventuellement diagramme d'activité
 - Packages des classe
 - Diagramme de classes
 - Classes
 - Éventuellement Objets

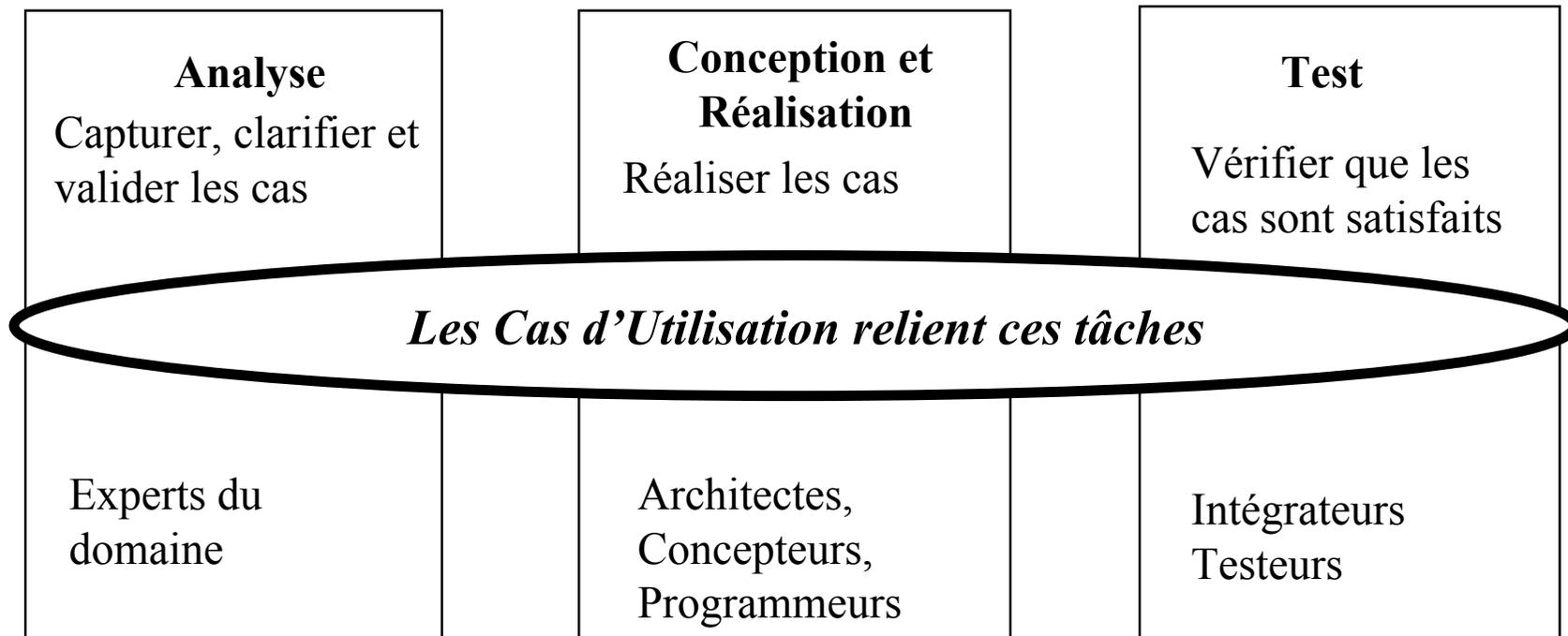


pr. Démarche de modélisation : un processus

pr.2. Un processus dirigé par les cas d'utilisation

5

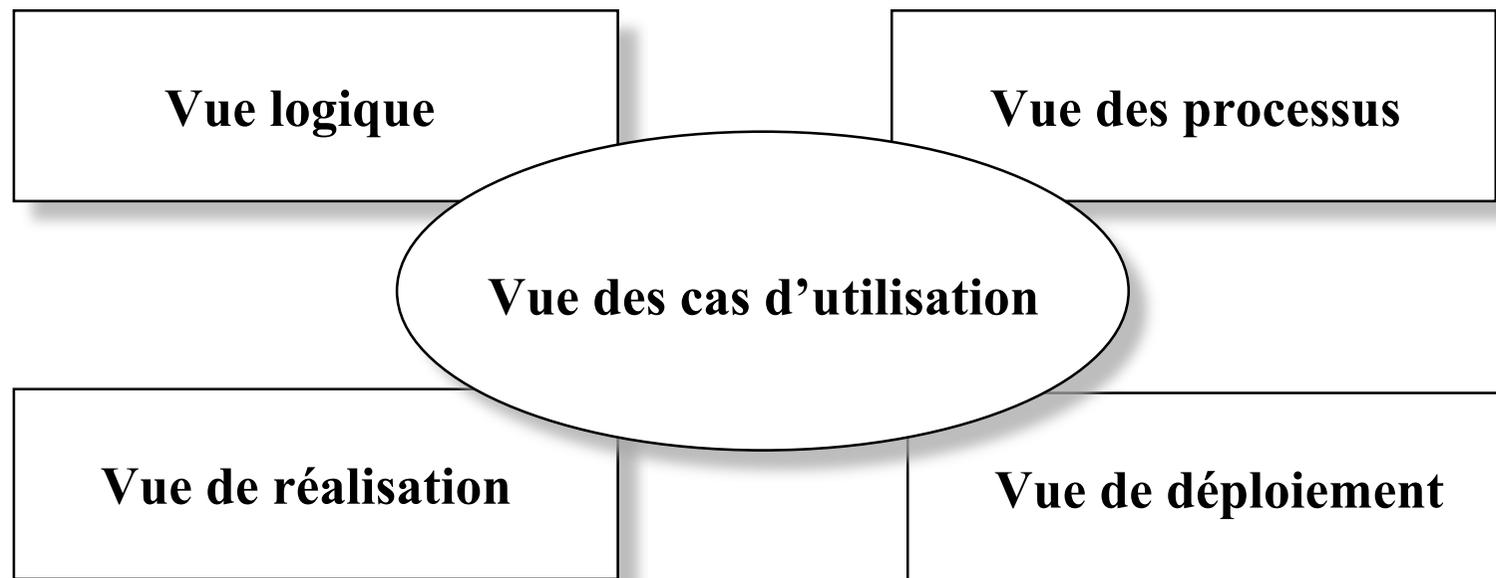
fil conducteur de toutes les activités, le découpage par cas d'utilisation



pr.3. Vision de l'architecte

il n'existe pas une seule manière de regarder un système

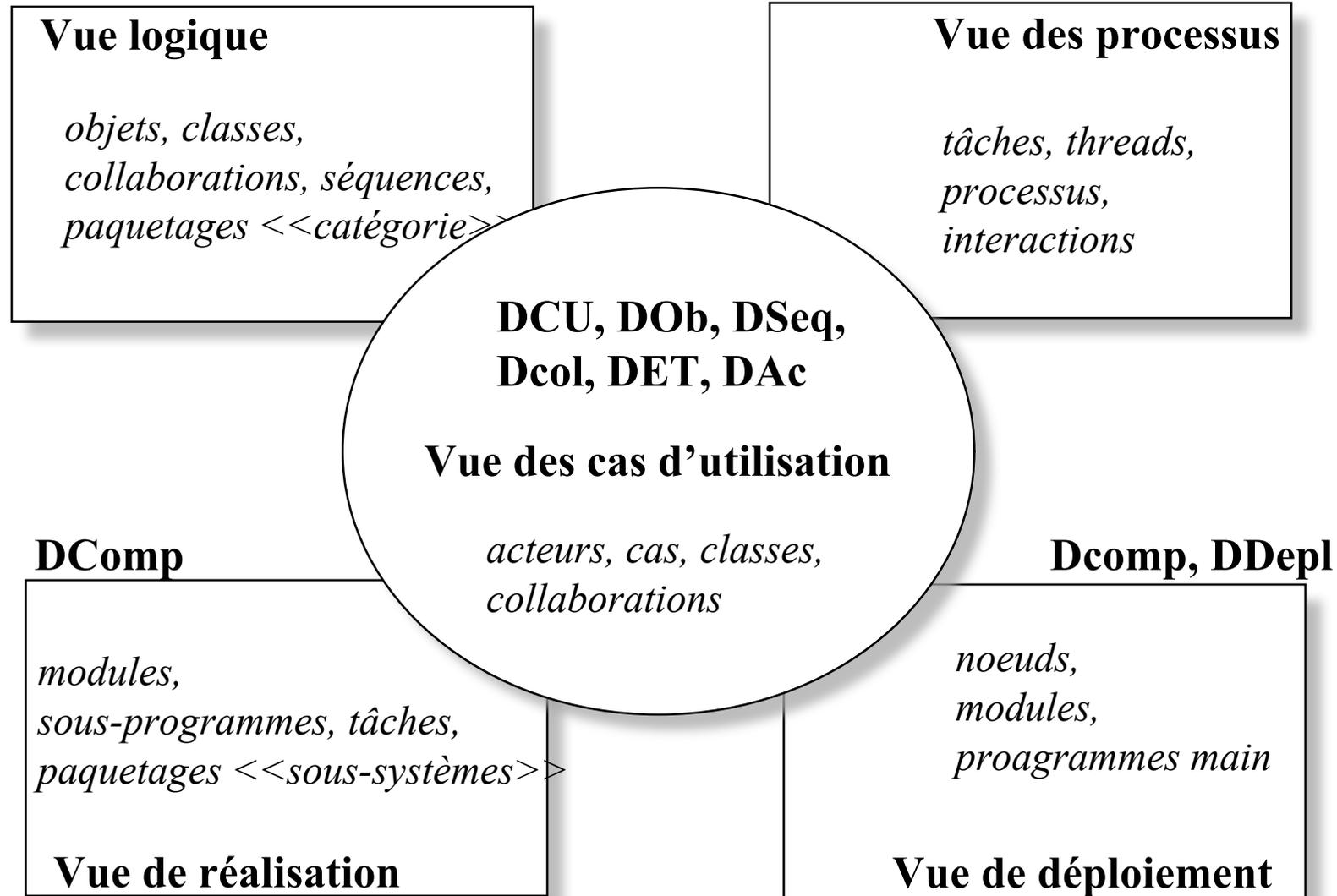
modèle 4 + 1 vues, Ph. Kruchten, IEEE Software nov. 95



pr.4 Diagrammes & processus

DCI, DOb, DSeq, Dcol, DET, DAc

DSeq, Dcol, DET, Dac, DComp



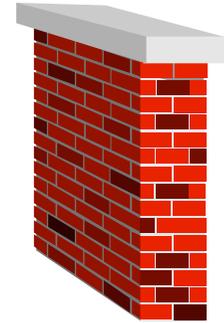
ax : Éléments du langage

- **structurants**
- **comportementaux**
- **agrégatifs**
- **annotations**
- **relations**

ax. Éléments de base

**des Choses, “things”
des Relations,
qui composent les graphes des Diagrammes.**

*“Les choses sont des composants fondamentaux d’un modèle;
les relations les lient ensemble
et les diagrammes regroupent des collections intéressantes de choses” (Booch 98)*



ax.1 Éléments de base : Les Choses (données, concepts : Things)

Ce sont les éléments constitutifs (building blocks)

orientés objet de base d'UML,
nécessaires à l'écriture de modèles corrects.

4 types de bases de Choses (concepts) :

- des concepts organiques
- des concepts fonctionnels
- des concepts agrégatifs
- des annotations, notes et commentaires

A chaque concept est associée une notation graphique qui est un nœud possible d'un ou plusieurs diagrammes du langage... ou un arc

ax.2 Eléments de base : les concepts organiques (au nombre de 7) notation

Classe : Ensemble d'objets

mêmes attributs, mêmes relations, mêmes opérations et même sémantique.

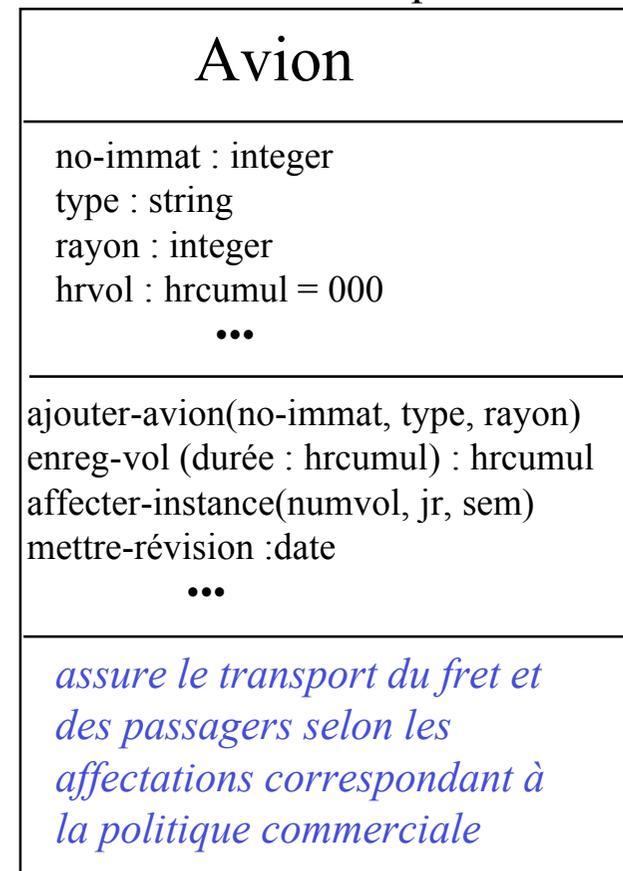
Une classe implémente

une ou plusieurs interfaces.

attributs

opérations

responsabilités



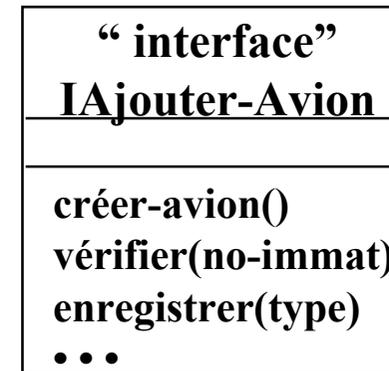
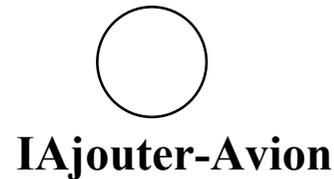
ax.2.1 Éléments de base : les concepts organiques ¹²

Interface : Collection d'opérations
spécifie le service d'un composant ou d'une classe,
càd un comportement externe visible.

Spécification d'opération

=

Signature pas Implémentation



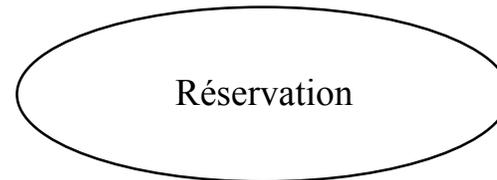
Collaboration : interaction entre objets collaborants ,
rôle social ou de coopération d'éléments

Une classe participe généralement à plusieurs collaborations.
Exprime ce qui intervient dans la réalisation d'une opération.



ax.2.2. Eléments de base : les concepts organiques

Cas d'utilisation, Use Case : Séquence d'*actions* du système, avec un résultat observable, significatif pour **un** *acteur* particulier structurant les comportements et réalisée par les *collaborations*.



en liaison avec les cas d'utilisation, on rencontre des acteurs, des diagrammes de cas d'utilisation, des scénarios: diagrammes de séquence & d'interaction qui peuvent être considérés comme des instances de USE CASE

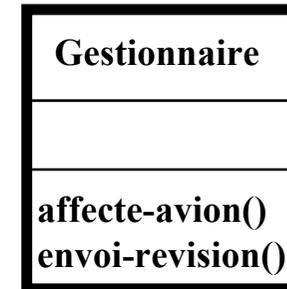
pour décrire les besoins du système

ax.2.3. Eléments de base : les concepts organiques

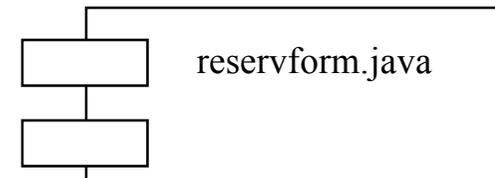
Les trois “concepts organiques” restants sont de “type-classe”:

“ensemble d’objets avec mêmes attributs, opérations, relations & sémantique”

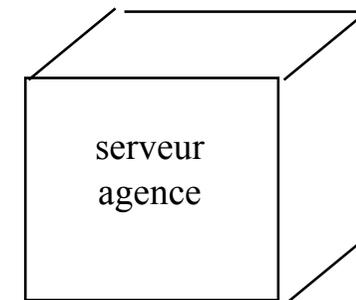
Classe active : initiatrice d’un contrôle d’activité
(et concurrence)



Composant : élément physique (*implémentation*)
remplaçable du système conforme
à un ensemble d’interfaces qu’il réalise.



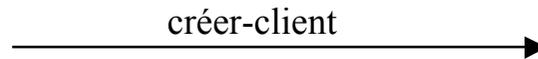
Nœud : ressource physique capable d’exécuter
(*processeur & mémoire*). lieu de résidence
et de migration d’un ensemble de composants.



ax.3. Eléments de base : les concepts fonctionnels

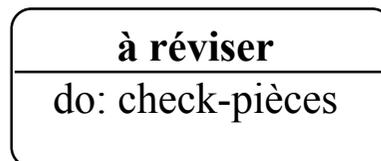
éléments de la dynamique des modèles

Interaction : comportement exprimant un ensemble de messages échangé entre un ensemble d'objets, dans un contexte, pour accomplir un but.



Interviennent dans les diagrammes d'interaction (collaboration, séquences)

Machine d'Etat, state machine : comportement spécifiant les séquences d'états par lesquels passe un objet ou une interaction durant son existence en réponse à des événements (*avec aussi des états, transitions, activités*).



aX.4 Éléments de base : les concepts agrégatifs

16

(grouping things)

Packages : mécanisme de regroupement organisationnel

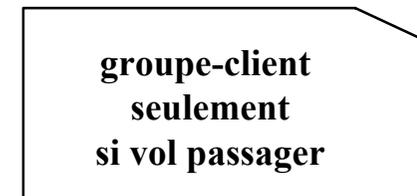
d 'éléments de natures diverses, exprime des “boîtes dans lesquelles un modèle est décrit”.
et aussi des “frameworks, modèles et sous-systèmes”.

*Un élément fondamental de lisibilité
des modèles*



6.5 Éléments de base : les annotations

Notes : commentaires explicatifs additionnels

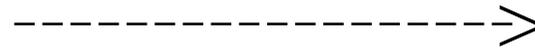


ax.6. Eléments de base : Les Relations

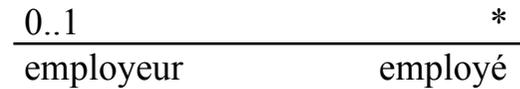
Ce sont des éléments relationnels constitutifs d'UML, nécessaires à l'écriture de modèles corrects.

4 types de bases de Relations :

la dépendance



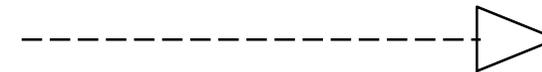
l'association



la généralisation



la réalisation



et aussi l'agrégation

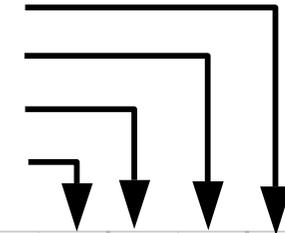


Conclusion : UML

- Un langage de modélisation (ML)
- Unifiant
 - des langages de modélisation de même type
Entité-association, OMT, Booch, etc.
 - des approches de modélisation
 - Fonctionnelle
Cas d'utilisation, Collaboration, Activité
 - Organique
Paquetage, Classe, Objets
 - Génétique
États
- Seulement un langage : abandon de l'idée d'un processus unifié (1996)

Retours OMG sur l'utilisation de UML

- 4.1.5. What parts of the language should be removed?
- 4.1.4. What parts of the language need clarification?
- 4.1.3. What constructs in the language are least used?
- 4.1.2. What constructs in the language are most used?



Class diagrams	●			
Sequence diagrams	●			
Use cases	●			
State charts	○		●	
Stereotypes and tagged values	●			
Package	●			
Implementation diagrams	●	●		
Collaboration diagrams	●	●	●	
Activity diagrams		●	●	
Model management (except package)		●	●	
Component		●	●	
Pattern		●		
Class, type, interface, classifier role, component		●	●	
Whole-part relationships			●	
UML Standard Profiles				●
Notes in semantics section				●
Standard elements without abstract semantics				●



Legend (responses)

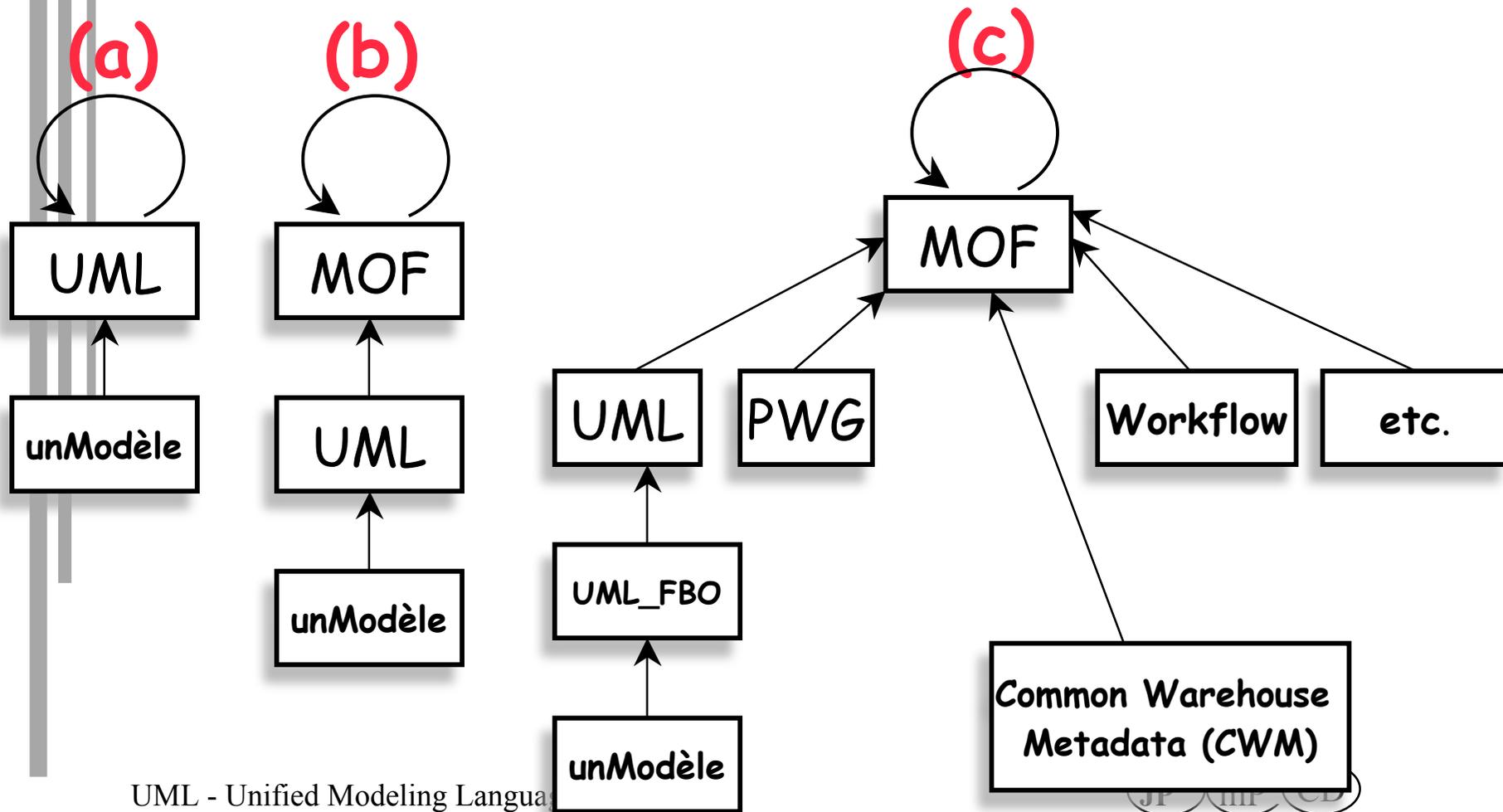
Utilisation de UML dans l'industrie

Modèle	Niveau d'utilisation
Classe et package	5 Le plus utilisé, le plus essentiel
Use case	3 Dépend des cultures. Il y a des domaines où cela a peu d'intérêt
Séquence	3 Assez employé, bien compris
Collaboration	2 Les diagrammes d'objet sont utilisés, mais la modélisation des collaboration et des rôles est encore marginale
Etat/transition	2 Très utilisé dans le temps réel, peu utilisé dans le tertiaire.
Activité	1 Arrivée tardive dans les ateliers, habitude culturelle locale à certaines parties du tertiaire
Composant/Déploiement	1 Peu de personnes l'utilisent, sur peu de parties de leurs applications

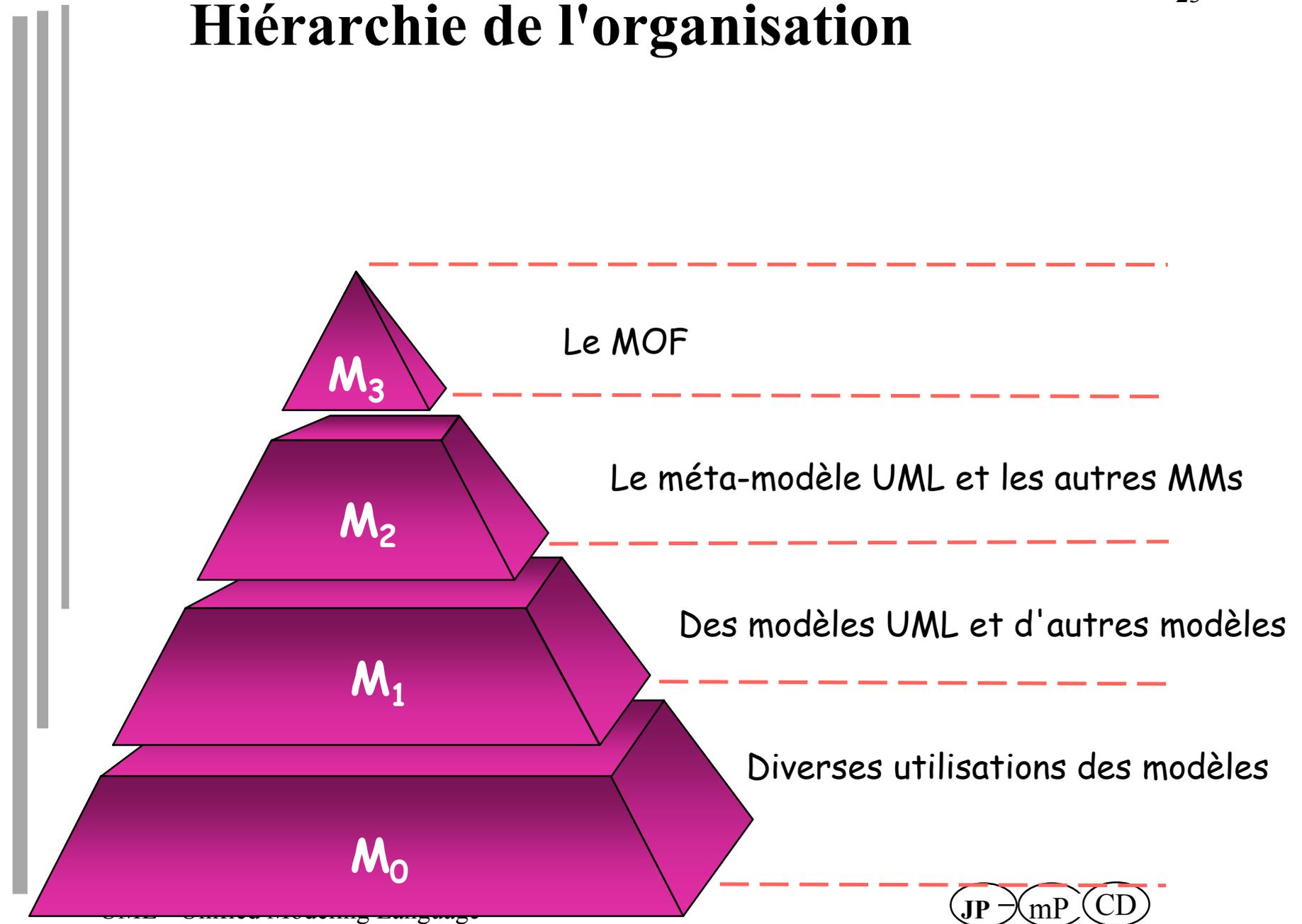
Conclusion : le futur

- De UML 1.0 à UML 2.0
 - Contraintes OCL
 - D'une simplicité efficace à une complexité expressive?
 - Le compromis efficace (?) : spécialisation à des contextes
 - Forte utilisation des stéréotypes
 - » Stéréotypes généraux
 - » Stéréotypes locaux
 - Les profils
 - » Éléments de méthode (processus) pour des contextes spécifiques
 - » Des profils standard, des profils à adapter (méta-modèle MOF)
- Nouveautés 2.0
 - Classes : ports, signal, pièce (« part ») + connecteurs → diagrammes d'architectures (vision organisationnelle)
 - Séquences : alternatives-exceptions, appel de scénarios, etc.
Vers les diagrammes d'activité ...!

Évolution vers le MOF



Hiérarchie de l'organisation



Getting serious with UML

24

**"In short: the time for experimentation
is past;
the time for stability and use
is now."**

*Grady Booch
Chief Scientist
Rational Software Corporation*