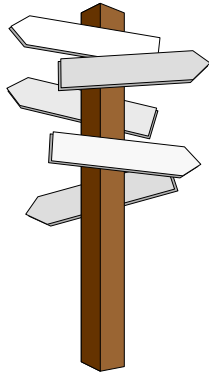


Panorama des différents types de méthodes

Chantal Reynaud

Université Paris X – Nanterre - UFR SEGMI

Maîtrise MIAGE



Plan

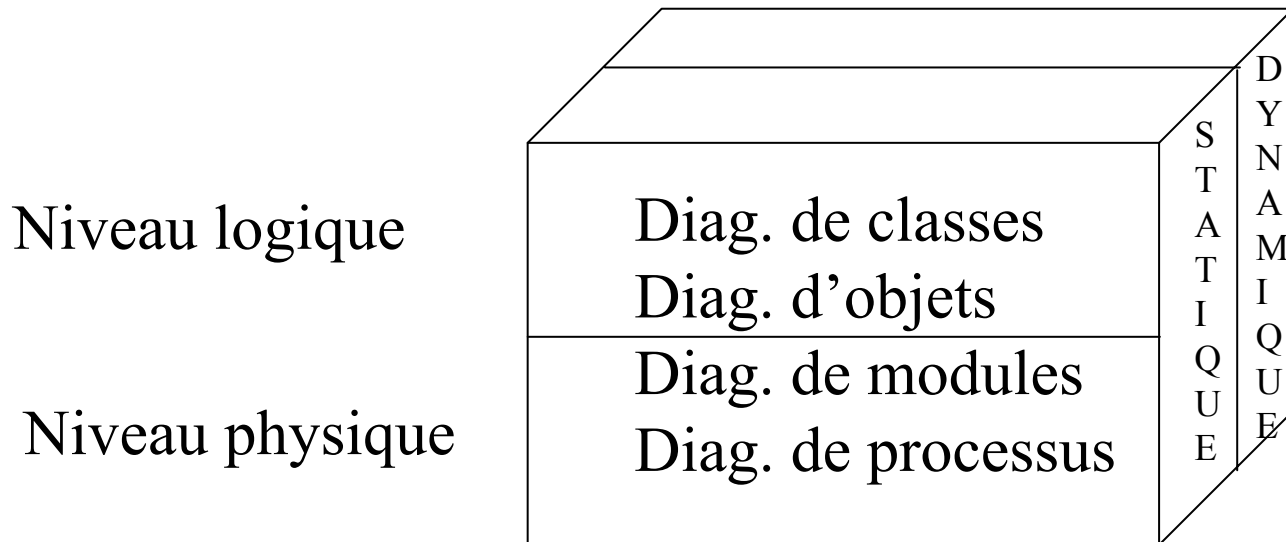
- I. Panorama de quelques méthodes objet
- II. Comparaison approches cartésiennes /
systémiques / objets

Partie I. Panorama de quelques méthodes objets

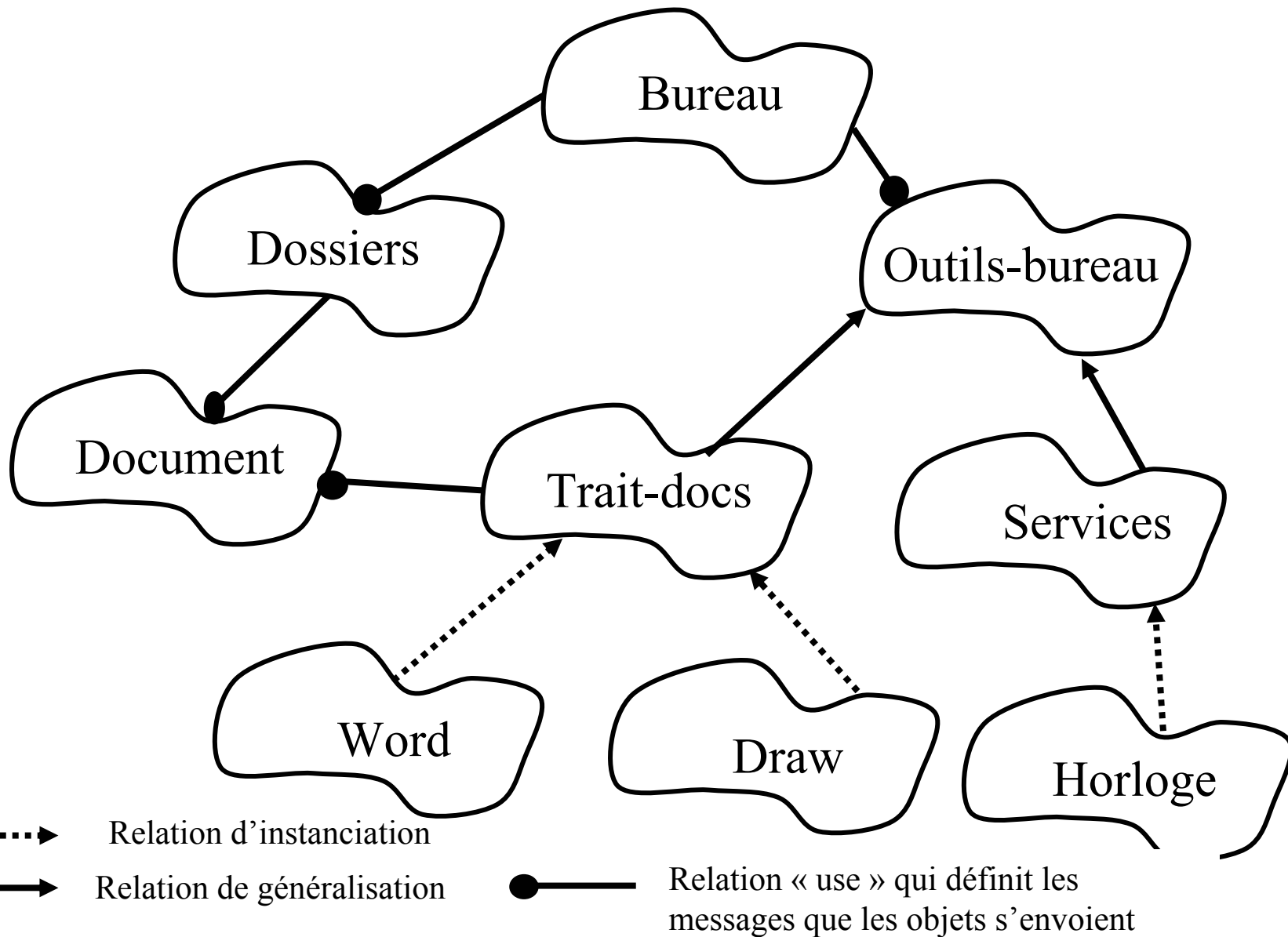
- I. OOD
- II. HOOD
- III. OOA
- IV. OMT

I. Object Oriented Design (OOD - G. Booch)

- Une des méthodes précurseurs de l'approche objet, définie pour le DOD, afin de rationaliser le développement d'applications ADA, puis C++.
- Une méthode de développement, non de conception.
- Niveaux et modèles de OOD :



I. OOD (G. Booch)

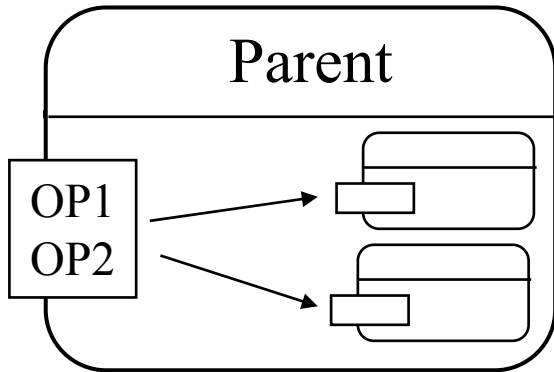


I. OOD (G. Booch)

- Se focalise sur l'aspect statique
- Propose de représenter l'aspect dynamique à l'aide de :
 - Diagrammes Etats - Transitions
 - Diagrammes de Temps
- Un outil de représentation d'objets complexes MAIS :
 - Ne fournit aucune information sur la manière de composer les objets
 - Représentation graphique peu conviviale
 - Passage logique → physique pas clairement formalisé

II. Hierarchical Object Oriented Design (HOOD)

- Une extension de OOD, mise au point dans le cadre du projet ESA, et mise en œuvre au début du projet Hermès.
- Ne couvre que le niveau *logique* et la dimension *statique* des applications.
- Une méthode capable de représenter des objets complexes à l'aide de hiérarchies de *composition* et d'*inclusion*.



Structure d'un objet composite

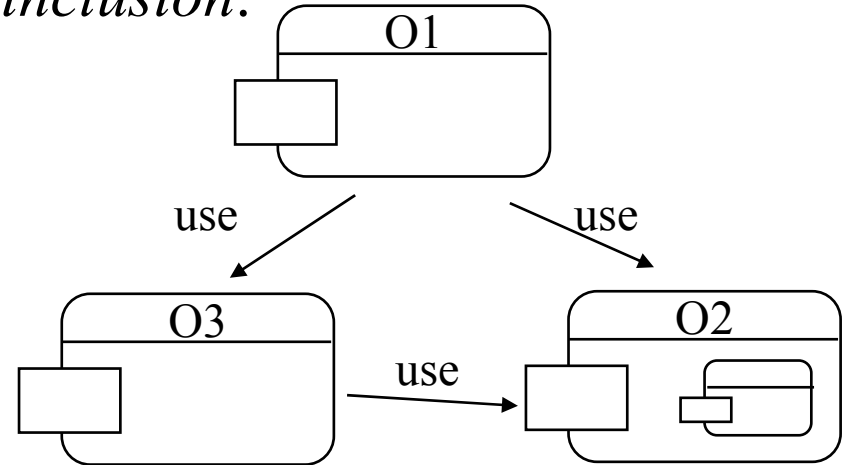


Schéma HOOD

II. Hierarchical Object Oriented Design (HOOD)

- L'objet est une unité de modularité, il encapsule données + opérations, aucun, un ou plusieurs flots de contrôle et fournit une interface d'utilisation à ses utilisateurs.

Mais :

- Absence de relations de généralisation
- N'exploite pas la réutilisation
- Les aspects dynamiques et fonctionnels ne sont pas couverts.

III. Object Oriented Analysis (OOA - Shlaer – Mellors)

- Couvre les 3 aspects : statique, dynamique et fonctionnel
- OOA semble plus proche des méthodes classiques :
 - ✓ sur le plan de la structuration des objets : pas d'avancée significative par rapport au modèle relationnel
 - ✓ la notion d'objet complexe n'existe pas
 - ✓ les objets n'ont que des attributs statiques, pas d'opérations.
(l'accès aux objets ne respecte pas l'encapsulation)

III. Object Oriented Analysis (Shlaer – Mellors)

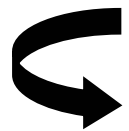
- Toutefois :

On peut assimiler la liste des actions attachées à un diagramme d'états d'un objet comme une liste de méthodes réalisant l'encapsulation de cet objet.

- Atout principal :

Une étude détaillée et exhaustive :

- du cycle de vie des objets
- des spécifications des actions attachées aux états des objets.

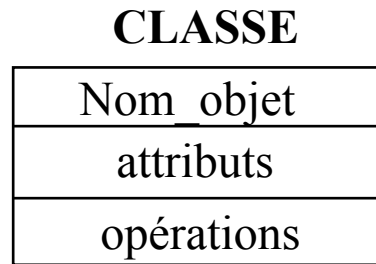


Bien pour décrire un système temps réel ou interactif
Peu adaptée aux applications batch

IV. Object Modeling Technique (OMT – Rumbaugh 1991)

- OMT respecte les 3 dimensions : statique, dynamique et fonctionnelle

↓
D.F.D.



↓
D.E.T.
Une extension du schéma E/A vers les concepts de l'objet

- L'une des méthodes les plus *complètes* et les plus *documentées*.
- Son cycle de développement couvre les étapes de *spécification formelles jusqu'à l'implémentation*.
- Elle *allie* judicieusement l'approche *objet* avec l'approche *fonctionnelle*.

Conclusion

- OOD – HOOD : plus des méthodes de programmation.
- OOA – OMT : même philosophie – S'appuient sur 3 niveaux.
- Les méthodes à objets correspondent mieux aux préoccupations des développeurs actuels :
 - rentabilité du développement,
 - réutilisation ou adaptation des développements,
 - formalismes graphiques,
 - un concept unificateur : l'objet,
 - possibilité de développer incrémentalement des applications,
 - prototypage rapide des applications pour la validation.
- L'approche objet devient pleinement profitable avec l'existence de bibliothèque de composants.

II. Comparaison des approches cartésiennes / fonctionnelles / objet

Approches cartésiennes

Approches objets

Années 70 - SADT, SA, SA/RT

Années 90 - OOD, HOOD, OOA,
OOA/OOD, OMT, ...

Descendante

DEMARCHE

Ascendante

Au plus haut niveau, le système modélisé ne doit remplir qu'une grande fonction. Chaque fonction est décomposée de façon hiérarchique en sous-fonctions

→ *Pas pour des systèmes complexes*

Partage de sous-fonctions à prévoir à l'avance

→ *Limite la réutilisation*

Identification des principaux objets et fabrication d'objets complexes par composition

Des objets autonomes

→ *Modulaire, adapté à la réutilisation*

Dirigée par les fonctions

Cohérence des données négligée.
Besoins d'un utilisateur rapidement et facilement capturés
Fonctions volatiles → perpétuelle reconception.

Dirigée par les données

Intégration facilitée des besoins de plusieurs utilisateurs en faisant partager les objets de base et en garantissant leur évolution.

Approches systémiques

Approches objets

Années 80 - Merise, Axial,...

Années 90 - OOD, HOOD, OOA,
OOA/OOD, OMT, ...

ASPECTS MODELISES

Données + traitements

Aucun concept commun

Des modèles séparés

Aspects statiques, dynamiques et fonctionnels

Expression intégrée des différents aspects → meilleure compréhension de la sémantique des objets

PRISE EN COMPTE DE DIFFERENTS NIVEAUX D'ABSTRACTION

Niveau conceptuel

Niveau logique

Niveau physique

Des méthodes basées sur la transformation de modèles

Une spécification formelle des objets séparée de l'implémentation mais IDENTITE des objets → une implémentation directe des structures de données conceptuelles est possible.

Une transformation entre les différents niveaux simplifiée