



# La genèse d'UML

Pierre-Alain Muller

ENSISA

[pa.muller@uha.fr](mailto:pa.muller@uha.fr)

03.89.33.69.65

# Sommaire

- Les méthodes d'analyse et de conception
  - A quoi sert une méthode
  - Des méthodes fonctionnelles aux méthodes objet
  - La prolifération des méthodes objet
  - Rapprochement de Booch et OMT
- L'unification des méthodes
  - Vers un langage unifié pour la modélisation
  - Modèle et métamodèle

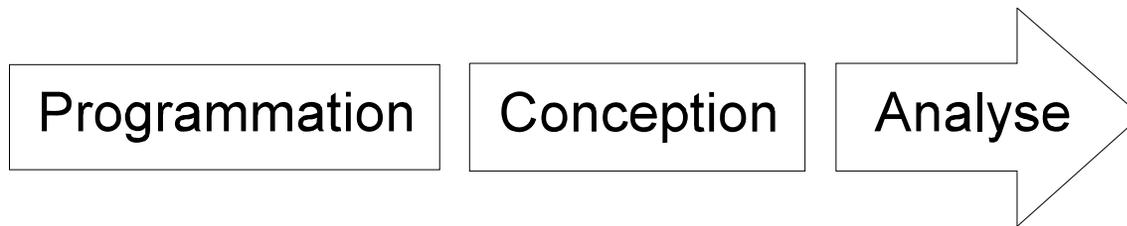


# A quoi sert une méthode

- Modéliser et construire des systèmes logiciels de manière fiable et reproductible
- Une méthode définit
  - Des éléments de modélisation
  - Une représentation graphique
  - Du savoir-faire et des règles

# Des méthodes fonctionnelles aux méthodes objet

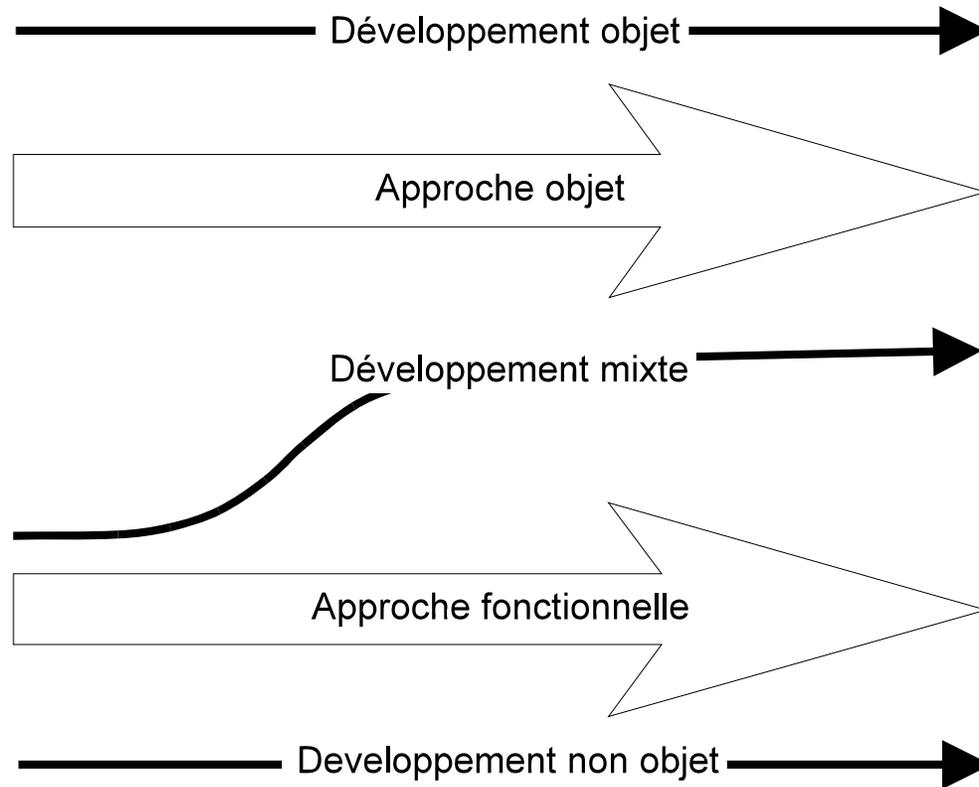
- Méthodes structurées et fonctionnelles
- Méthodes objet



Evolution des méthodes

# Approche mixte

- Logique mais anachronique



# Approche tout-objet

- Couverture de toutes les phases du cycle de vie du logiciel
- Pas de changement de mode de pensée
- Navigation facilitée entre les phases
- Bon support pour les démarches itératives et incrémentales

# La prolifération des méthodes objet

- Une cinquantaine de méthodes objet dans les cinq dernières années
  - Confusion, attentisme
- Consensus autour d'idées communes
  - Objets, classes, associations, sous-systèmes, cas d'utilisation

# Rapprochement de Booch et OMT

- Booch'93 et OMT-2 sont plus ressemblantes que différentes
  - Booch'93 adopte les associations, les diagrammes d'Harel, les traces d'événements
  - OMT-2 introduit les flots de messages et retire les diagrammes de flot de données
- Booch-93 construction
- OMT-2 analyse et abstraction

# Origine des concepts objet

- Souvent une histoire imbriquée

<b>Booch</b>	Catégories et sous-systèmes
<b>Embley</b>	Classes singletons et objets composites
<b>Fusion</b>	Description des opérations, numérotation des messages
<b>Gamma, et al.</b>	<i>Frameworks, patterns</i> , et notes
<b>Harel</b>	Automates ( <i>Statecharts</i> )
<b>Jacobson</b>	Cas d'utilisation ( <i>use cases</i> )
<b>Meyer</b>	Pré- et post-conditions
<b>Odell</b>	Classification dynamique, éclairage sur les événements
<b>OMT</b>	Associations
<b>Shlaer-Mellor</b>	Cycle de vie des objets
<b>Wirfs-Brock</b>	Responsabilités (CRC)

# L'unification des méthodes

- La pratique des méthodes a permis de faire le tri entre les différents concepts
- Jim Rumbaugh, Grady Booch et plus tard Ivar Jacobson décident d'unifier leurs travaux
  - OMT(*Object Modeling Technique*), Booch et OOSE (*Object Oriented Software Engineering*)

# Les objectifs

- Représenter des systèmes entiers
- Etablir un couplage explicite entre les concepts et les artefacts exécutable
- Prendre en compte les facteurs d'échelle
- Créer un langage de modélisation utilisable à la fois par les humains et les machines

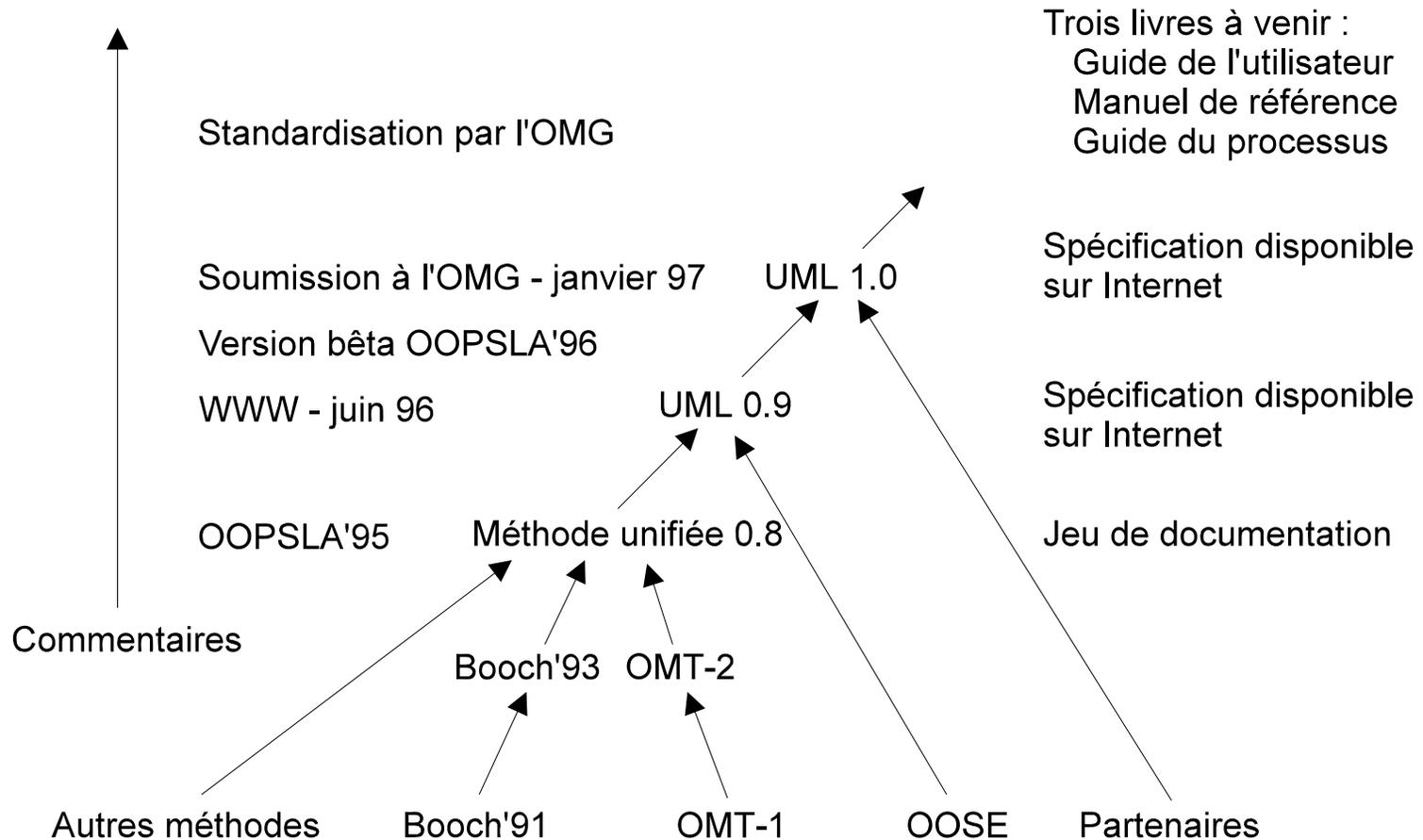
# Les étapes

- Octobre 95
  - *Unified Method V0.8*
- Octobre 96
  - UML V0.91 (*The Unified Modeling Language for Object-Oriented Development*)
- Janvier 97
  - UML 1.0 est soumise à l'OMG

# Les partenaires

- Courant 96 UML devient un enjeu stratégique
- Consortium de partenaires
  - DEC, HP, i-Logix, Intellicorp, IBM, ICON Computing, MCI Systemhouse, Microsoft, Oracle, Rational Software, TI et Unisys

# La définition d'UML



# Vers un langage unifié

- UML est une notation, pas une méthode
- UML est un langage de modélisation objet
- UML convient pour toutes les méthodes objet
- UML est dans le domaine public



# Métamodèle

- Identification des concepts fondamentaux
  - Définition de la sémantique de ces concepts
  - Choix d'une représentation graphique
- Métamodélisation d'UML avec UML
  - Description formelle des éléments de modélisation
- Austère, pas pédagogique
  - Méthodologistes
  - Constructeurs d'outils

# Modèle

- L'unité de base du développement
- La forme du modèle dépend du métamodèle
- Le contenu du modèle dépend du problème
- Relié à une phase précise du développement
- Construit à partir d'éléments de modélisation avec leurs différentes vues associées

# Les modèles d'UML

- Le modèle des classes
- Le modèle des états
- Le modèle des cas d'utilisation
- Le modèle d'interaction
- Le modèle de réalisation
- Le modèle de déploiement



# Les diagrammes d'UML

- Les diagrammes de classes
- Les diagrammes d'interactions
- Les diagrammes de séquences
- Les diagrammes de collaborations
- Les diagrammes d'états-transitions



# Diagrammes (suite)

- Les diagrammes d'activités
- Les diagrammes de cas d'utilisation
- Les diagrammes de composants
- Les diagrammes de déploiement

# Conclusion

- UML est la notation du génie-logiciel objet
- UML est soutenue par les principaux acteurs du monde de l'informatique
- UML est le successeur naturel des notations des méthodes de Booch, OMT et OOSE
- UML est le langage de la transition vers l'objet