La Conception Orientée Objet

- l'AOO Analyse Orientée Objet
- La COO La Conception Orientée Objet
- La POO La Programmation Orientée Objet

Analyse OO (P Coad et E Yourdon)

- L'identification des objets
- l'identification des structures
- la définition des sujets
- la définition des attributs
- la définition des services

La COO

- " *La conception par objets est la méthode qui conduit à des architectures logicielles fondées sur les objets que tout système ou sous-système manipule (plutôt que sur la fonction qu'il est censé réaliser)...Pour beaucoup de programmeurs, ce changement de point de vue est un choc comparable à l'idée de la terre tournant autour du soleil il y a 400 ans."*
  - Bertrand Meyer
- "*baby duck syndrom*"

Objectif numéro 1 : réutiliser ce que l'on produit

- "*Pourquoi le logiciel n'est-il pas comme le matériel ? Pourquoi tout nouveau développement doit-il repartir de zéro ? Il devrait y avoir des catalogues de modules logiciels, comme il y a des catalogues de puces de circuits intégrés : pour construire un nouveau système, on devrait commander des composants dans les catalogues et les combiner, plutôt que de réinventer la roue à chaque fois....."*
  - Extrait de "Conception et programmation par objets". Bertrand Meyer.

Méthode d'analyse : partir du réel

- " *Plutôt que de bâtir un système formé de modules associés à des opérations , on structure le système autour des objets existants dans le modèle du monde réel "*
  - J. L Théron
- *" Perhaps the greatest strenght of an object-oriented approach to development is that it offers a mechanism that captures a model of the real world "*
  - Grady Booch, member IEEE

Reflexion

- En analyse structurée il s'agit de constituer des actions élaborées ou procédure à partir d'actions primitives alors qu'ici il s'agit de constituer des objets complexes à partir d'objets primitifs.

L'approche objet

- Il s'avère que l'orienté objet offre une manière claire de concevoir une architecture faite de modules autonomes, facilitant l'implémentation multi-plateforme. Elle permet une flexibilité technique accrue et une meilleure ouverture aux nouvelles technologies (telles que le multimédia)

Etapes classiques identifiées

- ANALYSE
  - consiste à étudier les attentes des utilisateurs, afin d'identifier les objets de gestion ou "objets Métier" qu'ils invoquent dans leur activité
- DESIGN (Conception)
  - consiste à concevoir l'organisation des modules (le classes) du logiciel et du stockage des données persistantes
- IMPLEMENTATION (Réalisation)

GRANULARITE

Essentiel

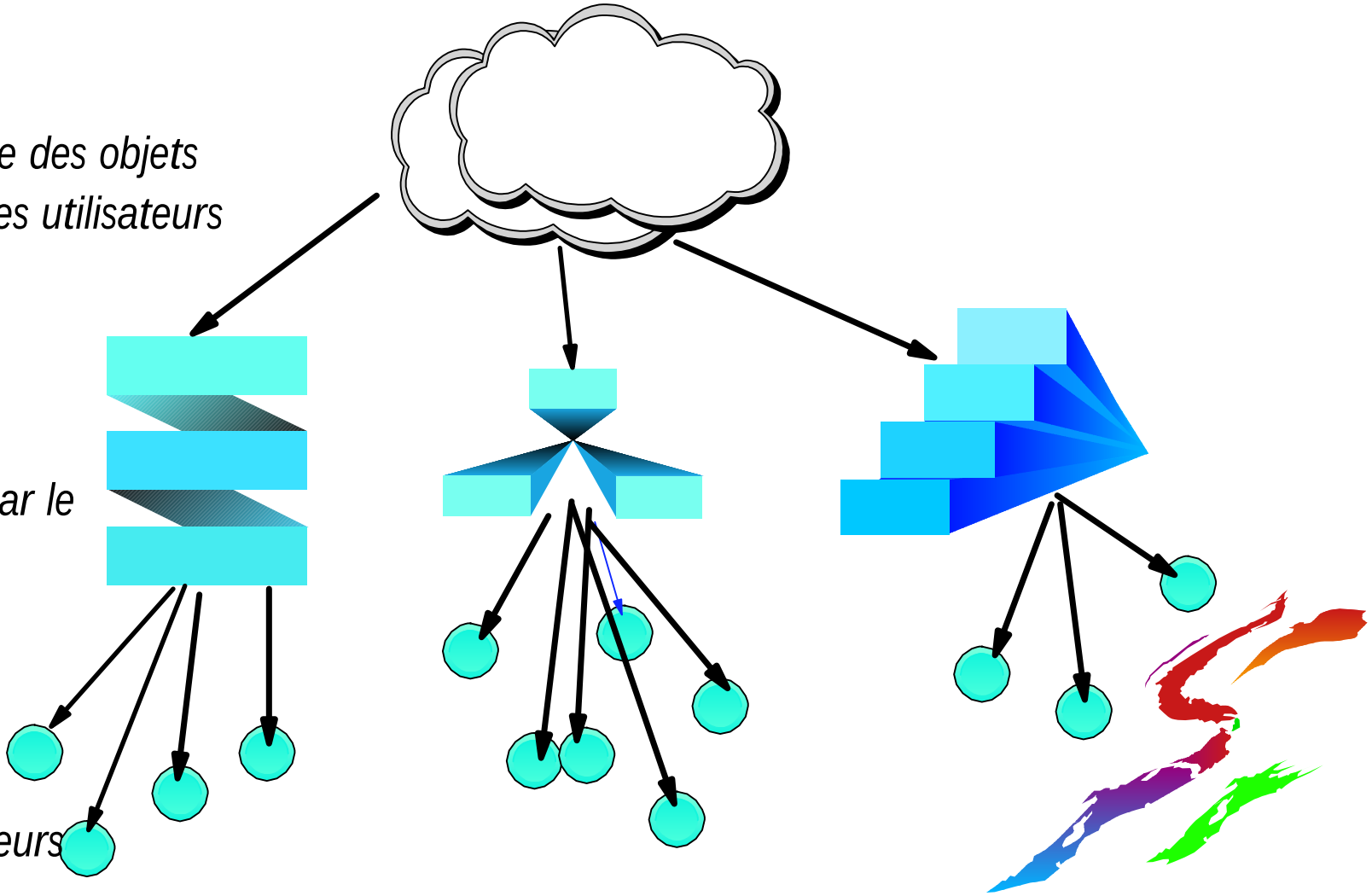
Vision globale des objets de gestion des utilisateurs

Conceptuel

Construction par le concepteur

Opérationnel

Implémentation par les développeurs



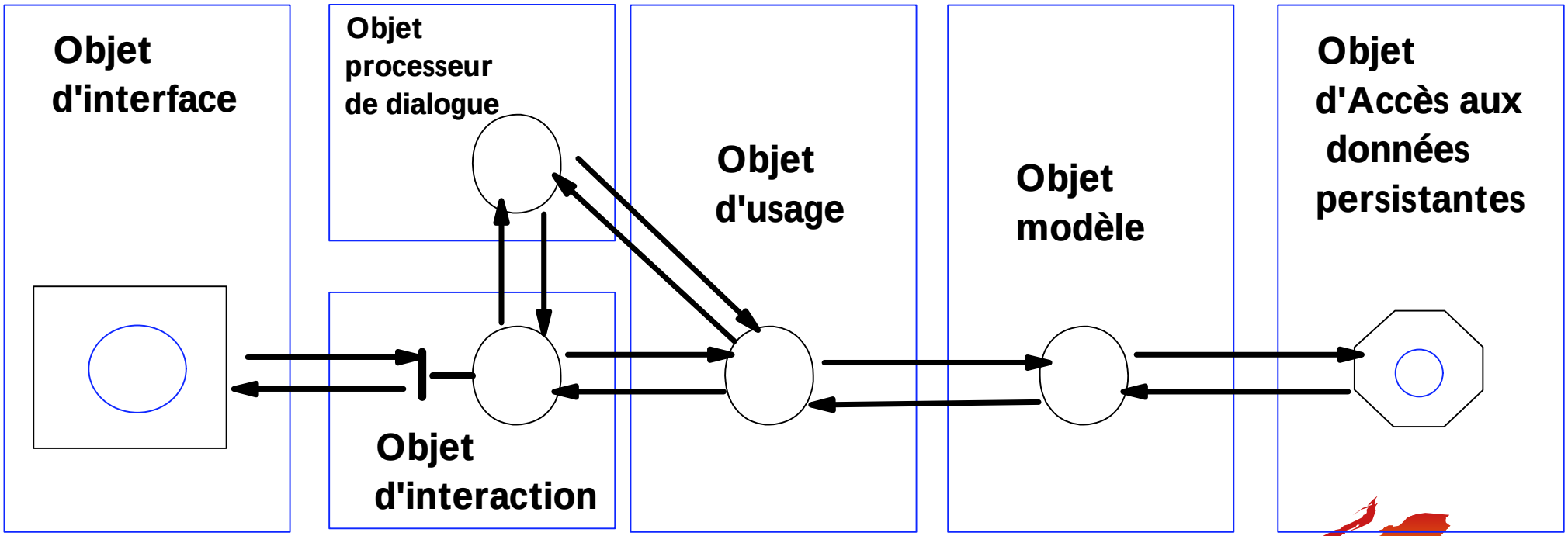
Les objets que perçoit l'utilisateur

- Actes de gestion
  - portion de l'activité de l'utilisateur, bornée par une sollicitation "externe" (client demande une réservation sur un vol pour NYK) et par la réponse à celle-ci (billet)
- objets essentiels
  - ceux que l'utilisateur voit, consulte, crée, etc (le client, le billet, le vol,...)objets essentiels
- Ces objets essentiels encapsulent des services
  - le client doit pouvoir donner son nom
  - Le vol doit pouvoir founir une place

Les objets que le concepteur définit

- le concepteur doit construire une architecture de composants logiciel qui permette :
  - au système de présenter des informations sur une interface utilisateur
  - d'interpréter les actions demandées par les utilisateurs sur cette interface
  - de mettre en oeuvre les règles de gestion attendues
  - de consulter ou mettre à jour une ou plusieurs bases de données.

Exemple d'objets de l'architecture



Les objets que manipule le développeur

- Le développeur fabrique et manipule des composants logiciels, des modules, des classes, langage ou CASE
- Chacun de ces modules implémente un ou plusieurs type d'objets Conceptuels
- Les classes génériques sont implémentées une fois pour toute dans l'environnement de développement
- Pour chaque application, les classes applicatives implémentent le fonctionnel et héritent de ces classes génériques

La Difficulté

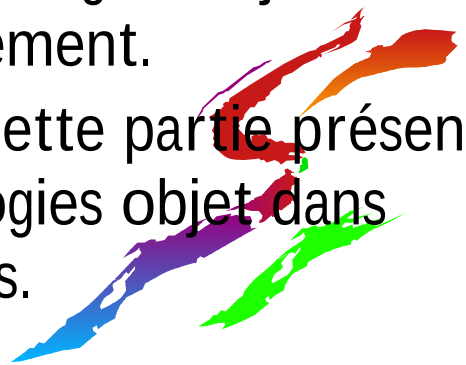
- En approche objet :
  - De quoi parle-t-on ?
- En approche structurée :
  - Que veut-on faire ?

Synthèse

- Approche structurée
  - Top down
- Approche objet
  - itération
- Modélisation objet :
  - structuration : mécanisme d'héritage et d'instantation
  - Lisibilité, cohérence

Le passage du structuré vers l'objet

- La comparaison des concepts. Cette partie met en exergue les principales caractéristiques qui différencient le monde structuré du monde objet.
- L'approche méthodologique. Après avoir comparé quelques méthodes, cette partie différencie les principes de base de la méthode Merise avec ceux d'UML (décrit dans la précédente partie de ce document).
- La mise en œuvre organisationnelle de la transition. Cette partie décrit différents scénarios d'intégration des technologies objet dans l'entreprise afin d'améliorer la conduite du changement.
- La transition dans les métiers de l'informatique. Cette partie présente l'évolution des métiers avec l'arrivée des technologies objet dans l'entreprise en réalisant un zoom sur trois métiers.



Le Chef de Projet

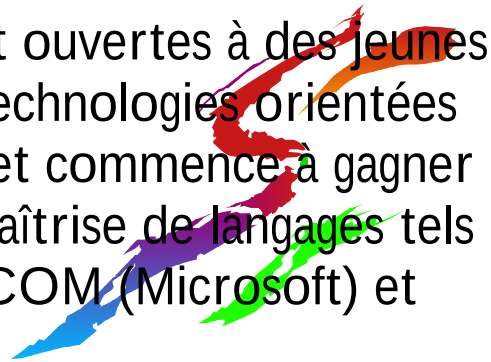
- Le chef de projet objet assure le pilotage des projets. Il doit être expérimenté dans le métier car il y a beaucoup de différences entre la démarche structurée et la démarche objet. Par exemple, la phase de conception fonctionnelle et technique est plus longue. Tout travail de planification doit en tenir compte.
- Le chef de projet est souvent amené à arbitrer entre les demandes des différents sous-projets, en fonction des objectifs assignés au logiciel. Il doit toujours se recentrer sur ces objectifs ; aussi, il ne peut pas être un « distributeur de programmes » à développer. Il est décideur, animateur et coordinateur sans remplacer l'architecte. Il doit savoir déléguer une partie des tâches de contrôle à l'architecte. D'autre part, il est amené à travailler plus souvent avec les utilisateurs ce qui nécessite une composante relationnelle importante et plus d'implication dans son activité.

L'Architecte

- Il intervient dans les premières phases du projet pour y concevoir l'architecture. C'est un rôle primordial dans le processus de mise en œuvre du projet. Il doit posséder une grande expérience technique dans les nouveaux domaines informatiques : objet et client/serveur. C'est un point clé pour la réussite du projet dont il s'occupe.
- Il doit construire une architecture vivante et un système d'information performant. Il doit permettre de développer une communication rapide et performante au sein du projet dans lequel il travaille, une réactivité maximale des équipes qui nécessite un gros effort de coordination.
- Il travaille de façon transversale. Il n'est pas nécessairement un bibliothécaire centralisant les objets outils ou métiers.

Le Spécialiste du développement par composants

- Le spécialiste du développement par composant est chargé selon le cas de fabriquer ou d'assembler des composants techniques et/ou métiers au sein d'une équipe de développement. Il administre les composants et gère leur cycle de vie (évolution, maintenance) dans l'entreprise.
- Se présentant comme une alternative au développement spécifique et à l'approche tout logiciel, le développement à base de composants réutilisables apporte aux entreprises la possibilité de développer plus rapidement et à moindre coût. C'est une approche assez jeune des entreprises qui y voient un moyen de s'affranchir des solutions toutes faites qui empêchent la diversification et aussi de la mainmise des éditeurs sur leur système d'information. Elle génère déjà de nouvelles méthodes de travail au sein des équipes de développement.
- Les fonctions de développement à base de composants sont ouvertes à des jeunes diplômés disposant d'une solide formation de base sur les technologies orientées objet. Le métier peut s'exercer en SSII ou chez un éditeur, et commence à gagner les services informatiques des entreprises utilisatrices. La maîtrise de langages tels que C++ et de l'environnement Java ou des Middleware DCOM (Microsoft) et CORBA (OMG) sont des atouts essentiels.



Les méthodes

- OMT de RUMBAUGH, M Blaha, W Premerlani, F Eddy **1991**
- OOSE de IVAR JACOBSON, M Christerson, P Jonson, G Overgaard **1992**
- OOD de GRADY BOOCH
- CRCde REBECCA WIRFS-BROCK **1991**
- OOM A ROCHFELD, M Bouzeghoud **1993**
- OAD S SCHAER , S MELLOR **1988, 1992**
- OOA/OOD E YOURDON et T COAD **1991**

Approche fondée

- Objets, événements, opérations (méthode Remora, Colette Roland)
- Information, état, processus (Schaer et Mellor

Un peu d'histoire et quelques méthodes

- Bailin
- Booch
- Coad et Yourdon
- Colbert
- Edwards
- Food
- Gibson
- Jacobson
- Kurtz, Woodfield et Embley
- Odell
- Pages-Jones et Weiss
- Rumbaugh
- Shlaert et Mellor
- Wirfs-brock, Wilkerson et Wiener

FERBER

- Identifier les entités du domaine **(statique)**
  - à partir des classes évidentes
  - à partir d'une liste exhaustive
  - regroupement de propriétés
- Structurer le domaine en analysant les propriétés de ces entités et leurs relations **(statique)**
- Identifier les opérations que savent effectuer ces entités **(dynamique)**
- Décrire précisément ces opérations en les reliant à des messages

Introduction

- **BOOCH**
  - Analyse et Design
  - Relation d'héritage
  - Diagramme d'objets, de classes, de modules de processus de transition
  - outil Rose
  - interface avec C++ ADA Smalltalk, Eiffel, ...
- **COAD et YOUDON**
  - héritage
  - Envoi de messages
  - temps réel, tables état évènements
  - diagramme de transition
- **JACOBSON**
- **RUMBAUGH**
  - Analyse, Design
  - diagramme flux de données
  - Modélisation fonctionelle et dynamique
  - diagramme d'états et de transition
  - agrégat d'objet

Comparatif des méthodes

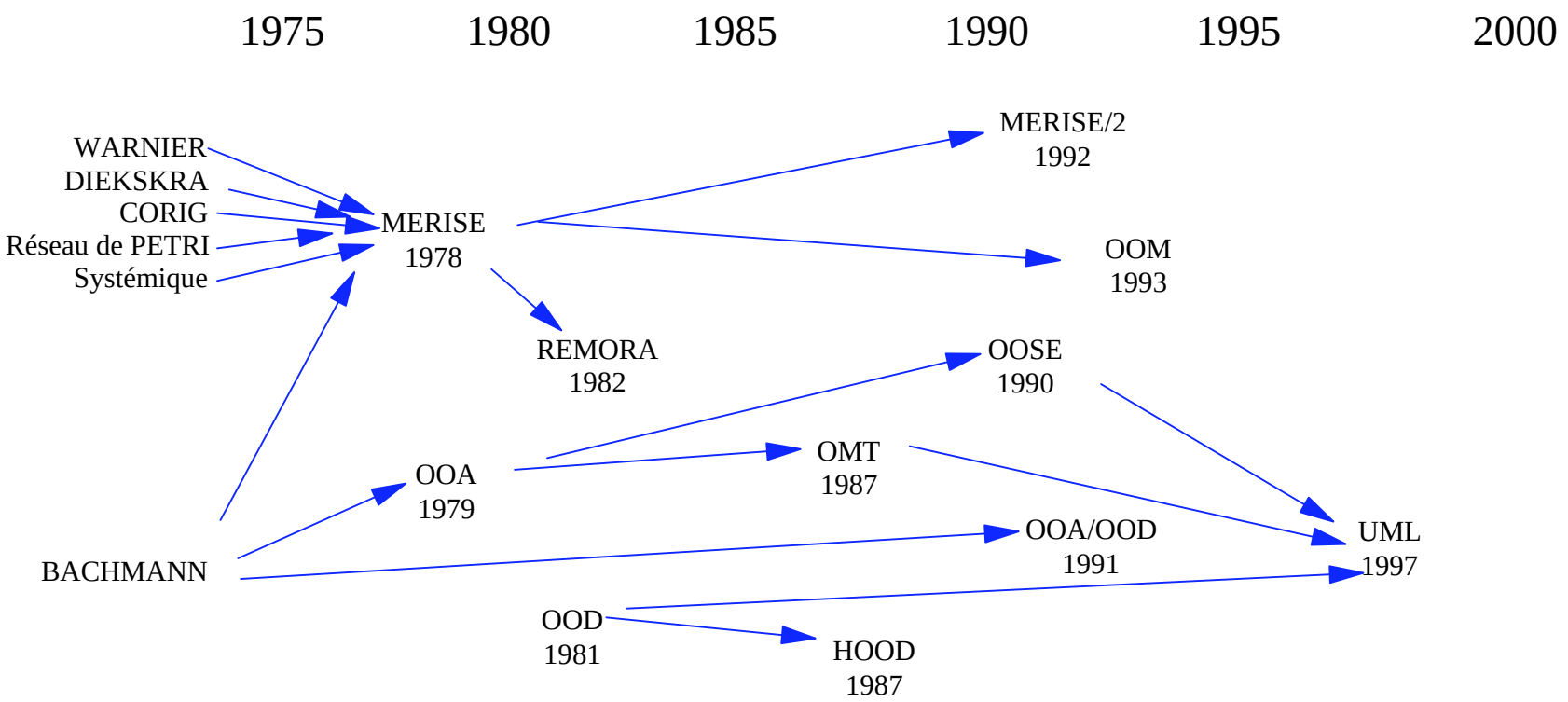
- **OOD : Object Oriented Design de Booch (Diffusion : 1985)**. Elle propose deux étapes de formalisation : la vue logique et la vue physique. Cette méthode suit un cycle en cascade sans intégrer l'analyse
- **OMT : Object Modeling Technique de Rumbaugh (Diffusion : 1991).** Elle couvre l'analyse, la conception et l'implémentation en quatre phases : l'analyse, la conception du système, la conception des objets et l'implémentation. La démarche suit un cycle en cascade en décomposant la notion de conception système et conception logiciel.. La méthode utilise les modèles objet, dynamique et fonctionnel.
- **OOSE : Object Oriented Software Engineering (appelée aussi Objectory) de Ivar Jacobson (Diffusion : 1992)**. La méthode suit d'une part une approche systémique (décomposition du système d'information en sous-systèmes pour aboutir au niveau de l'objet) et d'autre part, une approche structurée : analyse, construction et test. La phase de test suit un cycle ascendant du cycle en V.
- **OOM : Orienté Objet dans Merise de Rochfeld & Bouzeghoub (Diffusion :1993)**. C'est une méthode basée sur les trois cycles de vie de Merise et une adaptation à l'objet.

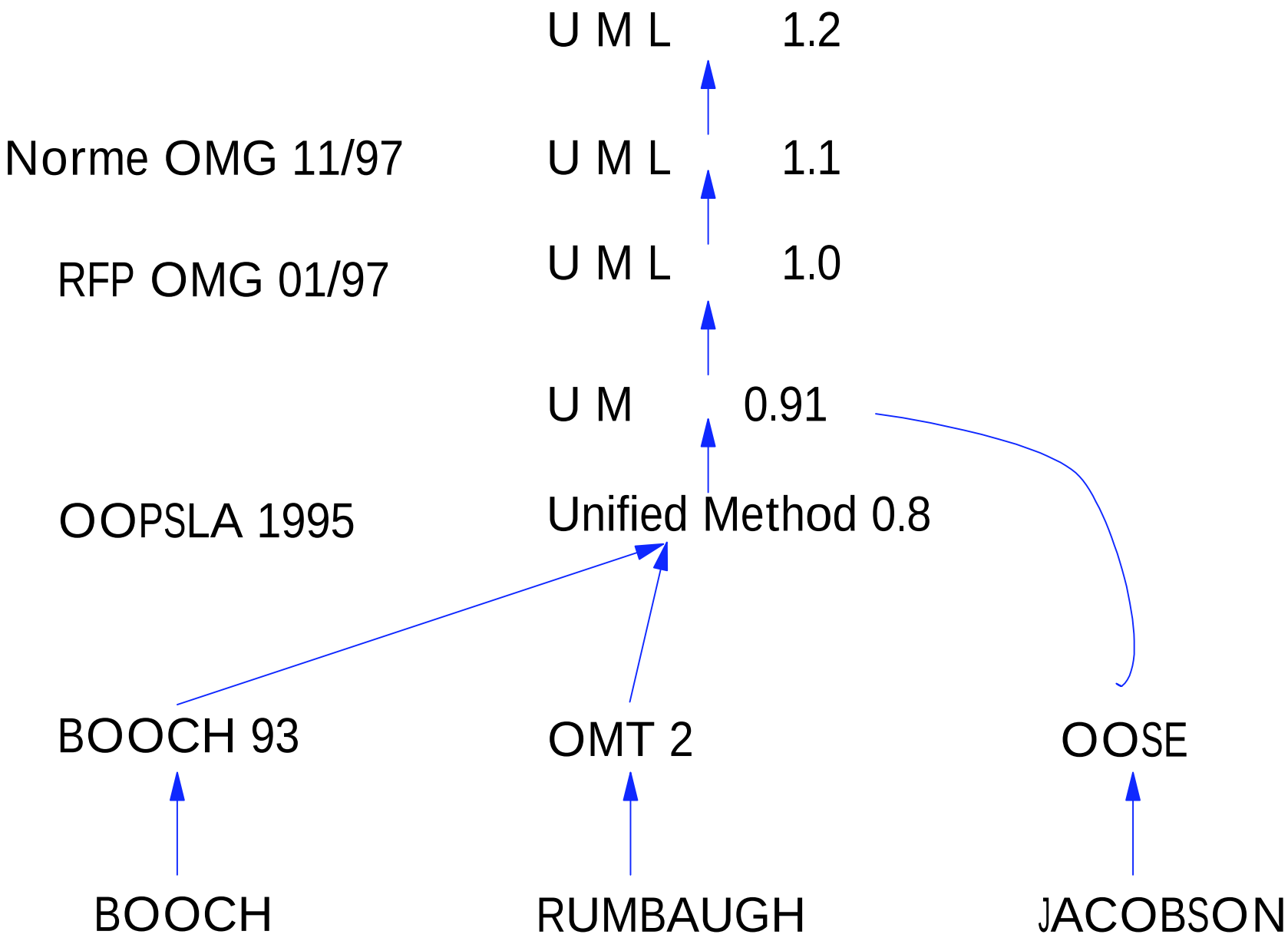
Comparatif des méthodes

Etapes du cycle traditionnel	OOD	OMT	OOSE	UML	OOM
Planification					X
Analyse du besoin		X	X	X	X
Conception système		X	X	X	X
Conception logiciel		X	X	X	X
Spécification logiciel	X	X	X	X	X
Spécification technique	X	X	X	X	X
Implémentation	X	X	X	X	X
Tests et intégration	X		X	X	
Maintenance	X		X	X	

ATTENTION UML : Unified Modeling Language (Diffusion : 1997). UML n'est pas à priori une démarche, ni une méthode mais bien un langage de modélisation.

Historique des méthodes




U M L 1.2
Norme OMG 11/97
U M L 1.1
RFP OMG 01/97
U M L 1.0
U M 0.91
OOPSLA 1995
Unified Method 0.8
BOOCH 93
OMT 2
OOSE
BOOCH
RUMBAUGH
JACOBSON