

UML : un peu d'histoire

A thick, horizontal yellow brushstroke underline that spans the width of the slide, positioned directly beneath the title.

UML



- ⌘ UML : Unified Modeling Language. Langage de modélisation unifié (sous entendu : orienté objet)
- ⌘ Formalisme graphique de modélisation basé sur les concepts orientés objet.
- ⌘ UML est issu des formalismes utilisés dans certaines méthodes nées dans les années 90. Parmi ces méthodes, on peut citer : OMT, OOSE, Booch,

UML



- ⌘ UML est un langage universel basé sur une notation graphique largement admise par la communauté OO. Il est indépendant par rapport aux langages de programmation, aux domaines d'application et aux processus de développement. Proposition d'un processus unifié.
- ⌘ Il améliore et facilite :
 - ☑ la communication
 - ☑ la représentation
 - ☑ la compréhension de solutions objet

UML



⌘ UML est devenu un standard incontournable car il répond à un besoin fondamental : bien que les concepts orientés objet (OO) soient les mêmes, il existait un grand nombre de formalismes graphiques (et textuels) différents pour les représenter.

➔ **besoin d'un langage universel**

⌘ Depuis novembre 1997, UML, version 1.1, est une norme de l'OMG (Object Management Group, www.omg.org).

⌘ Actuellement, ils en sont à la version 2.0 (avec XMI qui spécifie une structure pour l'échange de modèles utilisant XML, etc.)

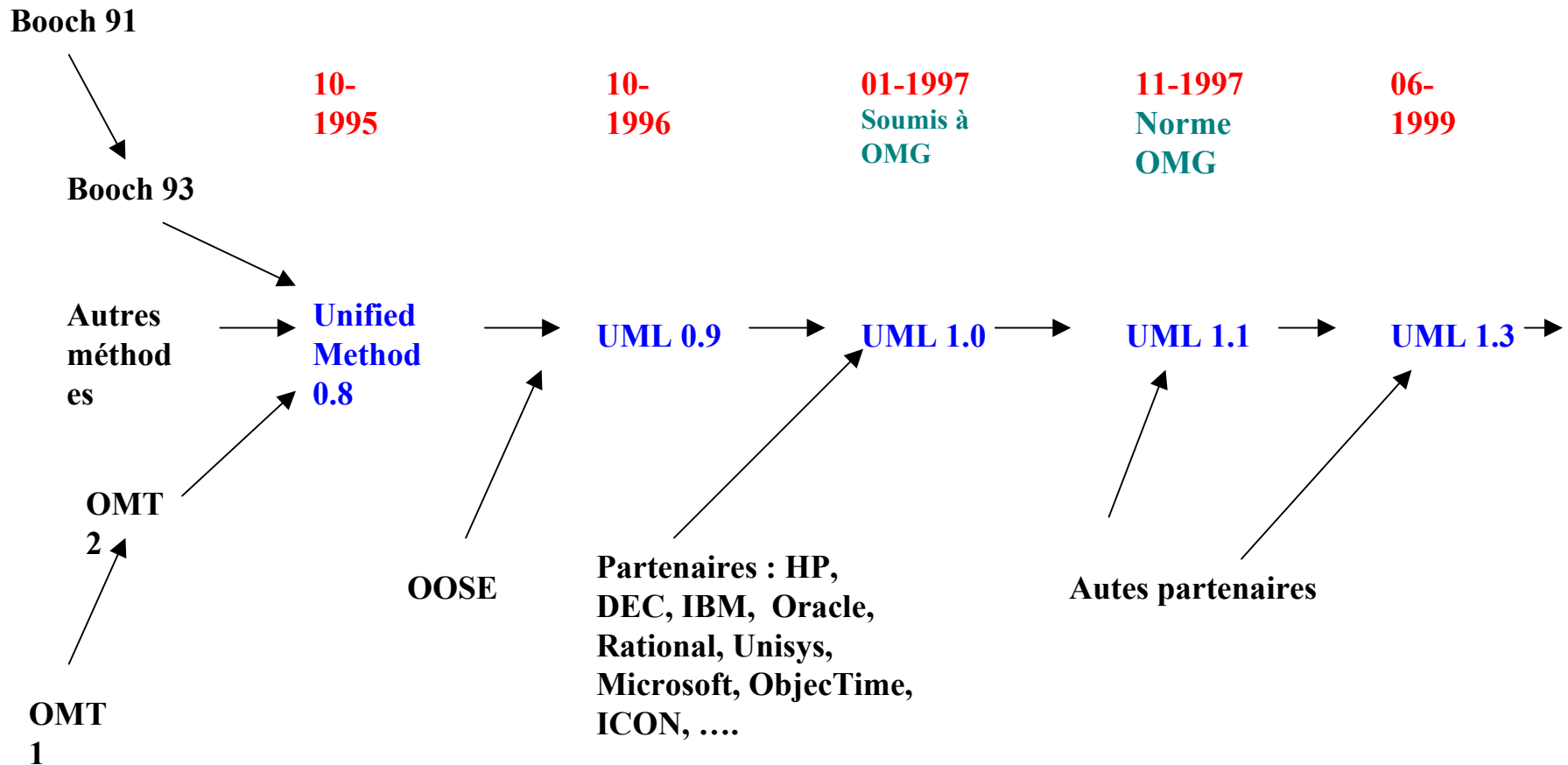
OMG



- ⌘ **OMG** : un organisme international à but non lucratif, créé en 1989 à l'initiative de grandes compagnies informatiques et autres (HP, Sun, Unisys, American Airlines, Philips...). Aujourd'hui, l'OMG fédère plus de 860 acteurs du monde informatique.
- ⌘ Il œuvre essentiellement à promouvoir des normes qui permettent **l'interopérabilité** entre applications OO communicant sur des réseaux hétérogènes.
- ⌘ L'OMG propose notamment l'architecture CORBA (Common Object Request Broker Architecture), pour la construction d'applications OO réparties dans le cadre de l'OMA (Object Management Architecture).
- ⌘ UML est intégré à l'OMA.

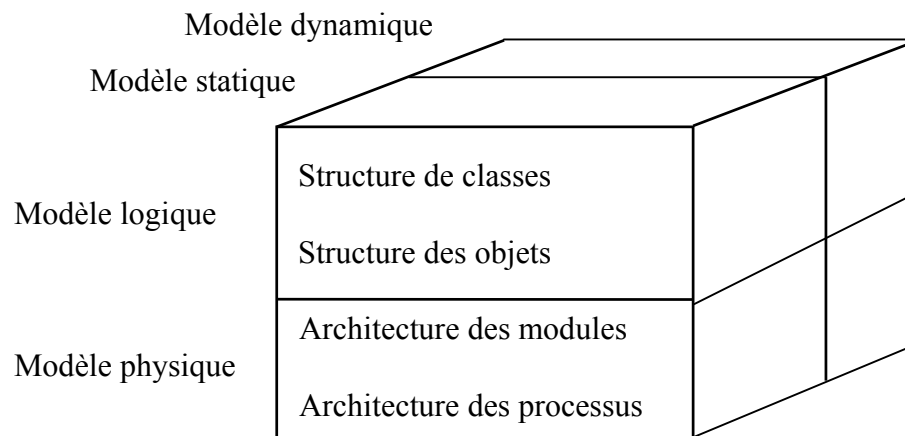
Genèse

← Discussions, propositions d'amélioration →



OOD [Booch 94]

- ⌘ Le but de la méthode est de supporter toutes les étapes du développement du logiciel jusqu'à la conception détaillée et même jusqu'à la programmation ;
- ⌘ La modélisation du système à développer prend en compte différents aspects



OOD [Booch 94]



- ⌘ Le modèle de classes représente le point de vue logique/statique ;
- ⌘ Le point de vue logique/dynamique est représenté par les trois types de diagrammes : (1) diagramme des objets, (2) diagramme de transition d'états et (3) le diagramme d'interaction ;
- ⌘ La méthode de Booch est essentiellement pragmatique. Elle est une collection de techniques ;

OMT [Rumbaugh 91, 96]



- ⌘ La principale caractéristique d'OMT est que l'emphasis est mis sur l'identification et l'organisation des concepts de l'application et du domaine plutôt que sur l'implantation de ces concepts ;
- ⌘ La méthode couvrent aussi bien les étapes d'analyses que celle de la conception et de l'implantation. Cependant cette dernière étape n'est couverte que partiellement ;
- ⌘ Les notations proposées par OMT permettent de décrire un système avec les trois points du vue: statique, dynamique et fonctionnel.

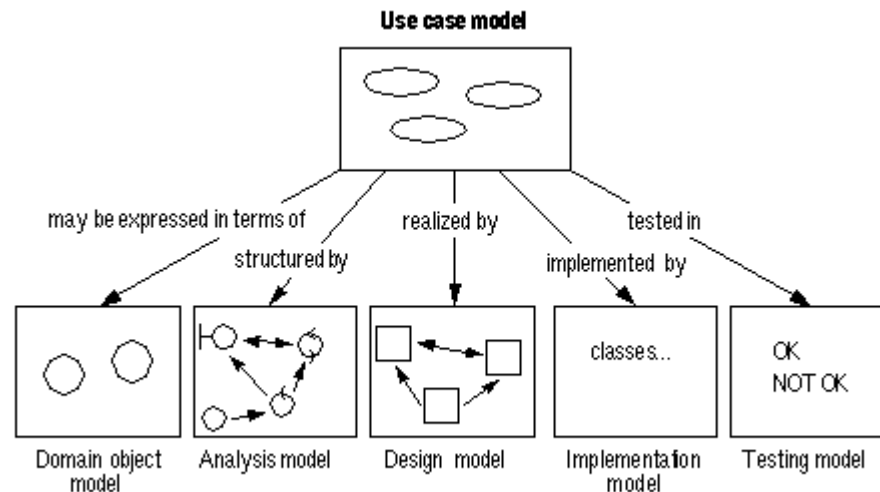
OMT [Rumbaugh 91, 96]



- ⌘ Le point de vue statique est représenté par le modèle objet. Le but de ce modèle est de décrire la structure statique des classes et leurs relations ;
- ⌘ Le point de vue dynamique est représenté par les diagrammes d'états des classes ;
- ⌘ Le point de vue fonctionnel est décrit par un ou plusieurs diagrammes de flux de données (DFD) ;
- ⌘ Le processus mis en œuvre dans la méthode OMT est une succession de trois étapes : l'analyse, la conception système et la conception objet (ou détaillée).

OOSE [Jacobson 92, 94]

- ⌘ OOSE est une approche guidée par les cas d'utilisation. Tous les modèles construits, sont en fait dérivés du modèle des cas d'utilisation. Il est la base des phases d'analyse, de conception et de test ;



Les diagrammes d'UML



- ⌘ **Cas d'utilisation**
- ⌘ **Classes**
- ⌘ **États-Transitions**
- ⌘ **Séquence**
- ⌘ **Collaboration**
- ⌘ **Objets**
- ⌘ **Activité**
- ⌘ **Composants**
- ⌘ **Déploiement**

Quelques questions sur UML



- ⌘ UML est-il un langage OO ?
- ⌘ Quelles différences existe-t-il entre une méthode et un formalisme ?
- ⌘ UML vs. OMT ?
- ⌘ Projets de gestion ou techniques ?
- ⌘ Qu'apportent les outils de modélisation à UML ?