



Modélisation UML

Michel MALDONADO
INGESUP

OBJECTIFS

- **PRINCIPES UML**
- **VUE GENERALE**
Diagrammes
- **PRISE EN MAIN**
Cas d'utilisation
Diagramme de classes
Diagramme d'objets
Diagramme de séquence
Diagramme d'états-transition

PRE REQUIS

- **CONCEPTS OBJETS**
classe, encapsulation, association, héritage



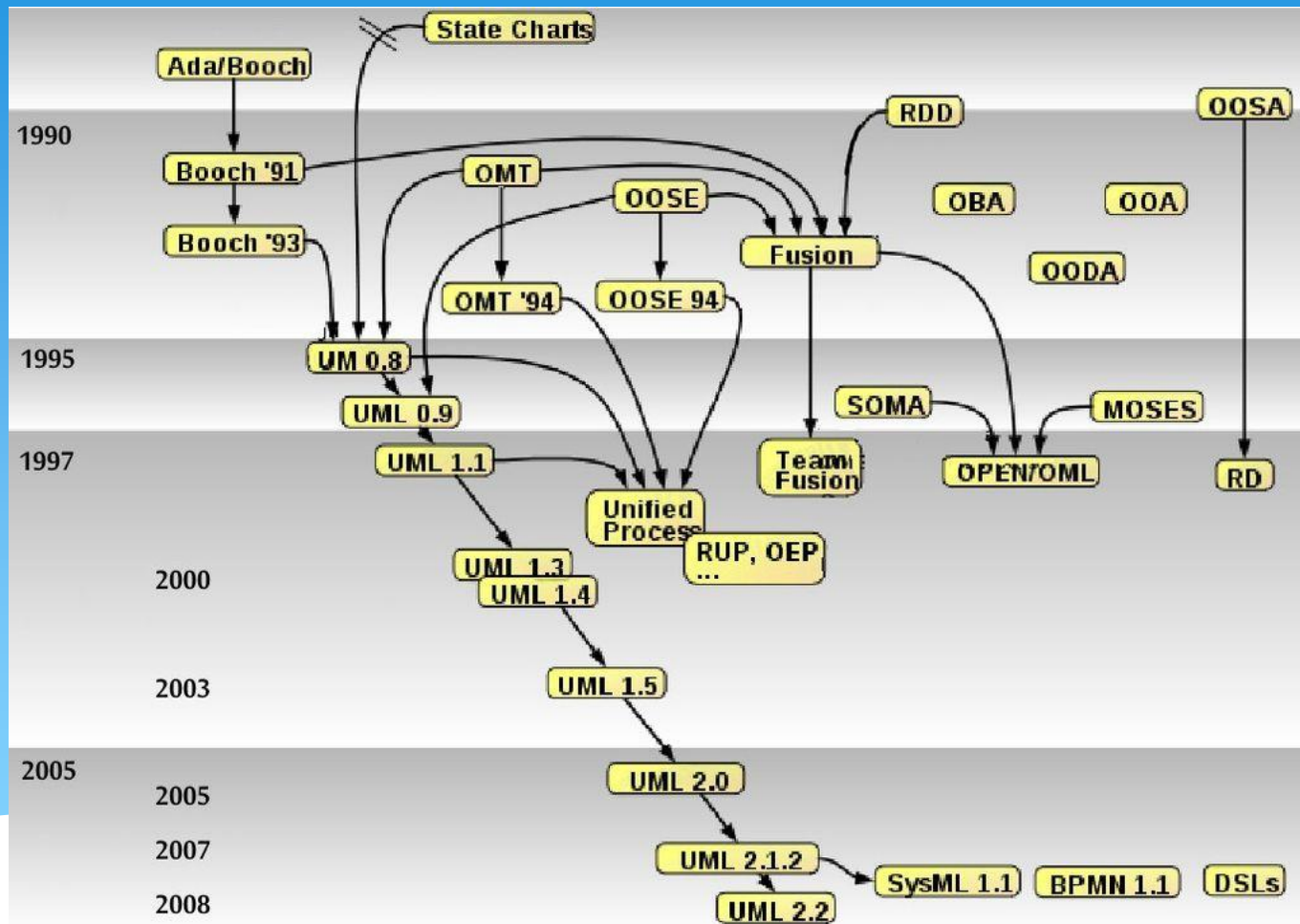
Modélisation UML

1- INTRODUCTION

HISTORIQUE

- **70's : approche fonctionnelle (top-down)**
 - **80's : approche systémique (Merise)**
 - **Early 90's : émergences des méthodes objets**
 - *approche spécifique objet (données+traitement)*
 - *Booch, Classe-Relation, Fusion, HOOD, OMT, OOA, OOD, OOM, OOSE...*
 - *aucune*
 - **Late 90's : convergence**
 - *OMT (James Rumbaugh)*
 - *OOD (Grady Booch)*
 - *OOSE (Ivar Jacobson)*
- ➔ **UML (Unified Modeling Language)**

UML Tree



source : wikipedia

UML

- **Unified Modeling Language**

Langage de modélisation graphique à base de pictogrammes. Il est utilisé en développement logiciel, et en conception orientée objet.

http://fr.wikipedia.org/wiki/UML_%28informatique%29

- **Standard ISO**

UML 2.4.1(2012)

UML 2.5 (bêta 2 - déc 2013)

- **Object Management Group (OMG)**

Corba

www.omg.org / www.uml.org

AVANTAGES

- **Langage formel et normalisé**
gage de cohérence, précision, stabilité
standard de l'industrie
- **Éléments de conception accessibles au client**
vulgarisation
support de communication
- **Centré sur les besoins du client**
application en adéquation avec les besoins (cas d'utilisation)
- **Auto-descriptif (Meta modèle UML)**
décrit le modèle UML (diagramme) par des diagramme UML

INCONVENIENTS

- **Période d'apprentissage**
formalisme
diagrammes
outils
- **Cycle de développement ralenti**
dépassement du délai de livraison
- **UML est un langage, pas une méthode**
« Le tournevis permet de monter l'armoire, mais ne dira pas comment »
nécessité d'un processus de développement
(RUP ou Rational Unified Process)

Vues UML



Vue des cas d'utilisation

- **QUI ? QUOI ?**
- **Besoins des clients du système, objectifs du système**
- **Acteurs**
utilisateurs du système
- **Scénario, cas d'utilisations**

Vue logique

- **COMMENT ?**
Définition du système de l'intérieur
- **Eléments du domaine**
*abstraction et encapsulation
liés aux métiers de l'entreprise*

Vue des composants

encore Vue de réalisation, vue d'implémentation

- **Modules physiques**
fichiers sources, bibliothèques dynamiques, bases de données, exécutables,...
- **Dépendances entre modules**
- **Organisation en sous-systèmes**
interfaces
dépendances

Vue des processus

- **COMMENT ?**
vue temporelle et technique
- **Importance dans environnements multi-tâches**
- **Processus ou tâches concurrentes**
décomposition
interaction
synchronisation

Vue de déploiement

- **OU ?**
- **Importance dans le cas d'environnements distribués**
- **Ressources matérielles**
 - nature*
 - implantation géographique*
 - distribution du logiciel sur ces ressources*
- **Performances**
 - temps de réponse, tolérance aux pannes*

Outils logiciels

- **Rational Rose** (<http://www.rational.com>)
Outil de plus important du marché
Racheté par IBM
- **Together** (<http://www.togethersoft.com>)
Outil fortement couplé avec Java
Racheté par Borland
- **ArgoUML** (<http://argouml.tigris.org>)
Outil Open Source
- **Visio** (<http://www.microsoft.com/office/visio>)
Outil non complet de Microsoft
- **SAP Power AMC** (<http://www.sybase.fr/>)
monde MERISE
- **Objecteering** (<http://www.objecteering.fr/>)
évolution : Modelio

