

Petite histoire d'UML

Pierre-Alain Muller

ESSAIM

pa.muller@uha.fr

03.89.59.69.65

Prolifération des méthodes objet

- Une cinquantaine de méthodes objet au début des années 90
 - Confusion, attentisme
- Consensus autour d'idées communes
 - Objets, classes, associations, sous-systèmes, cas d'utilisation

Unification des méthodes

- « *La guerre des méthodes ne fait plus avancer la technologie des objets* »
 - Ne permet pas de dégager un marché suffisant
- Recherche d'un langage commun unique
 - Utilisable par toutes les méthodes
 - Adapté à toutes les phases du développement
 - Compatible avec toutes les techniques de réalisation

De quoi a-t-on besoin ?

- Un langage de modélisation
 - Notation claire
 - Sémantique précise
- Une démarche de génie logiciel

Méthode = Langage + Démarche

Langage de modélisation

- Générique
- Expressif
- Flexible (configurable, extensible)
- Syntaxe et sémantique
- Unification par convergence aujourd'hui

Démarche

- Générique
- Impossible à standardiser
 - Personnes, applications, cultures...
- Cadre configurable
- Unification par convergence dans le futur

Différentes sortes de systèmes

- Logiciels
 - Ingénierie des logiciels
- Logiciels et matériels
 - Ingénierie des systèmes
- Organisations
 - Ingénierie des organisations

Unification sur plusieurs domaines d'applications

Rapprochement Booch / 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

Notation unifiée UML

- Basée sur les méthodes de BOOCH, OMT et OOSE



Grady Booch James Rumbaugh Ivar Jacobson

- Influencée par les bonnes idées des autres méthodes
- Mûrie par le travail en commun

Principales influences

■ Souvent une histoire imbriquée

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

Objectifs

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

Approche retenue

- Identifier la sémantique des concepts de base
- Classer les concepts
- Construire un métamodèle
- Choisir une notation graphique
- Regrouper par niveau d'abstraction, complexité et domaine

Métamodèle

- Identification des concepts de base
 - 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 et constructeurs d'outils

Modèles et les vues

- Modèle = unité de développement
 - Cohérence interne forte
 - Couplage faible avec les autres modèles
 - Relié à une phase de développement
- Vue = projection (modèle)
 - Graphique
 - Peut englober plusieurs modèles

Unification des méthodes objet

- Appel aux propositions de l'OMG
- Démarche d'unification
- UML (*Unified Modeling Language*)

- OMG (Object Management Group)
 - Consortium ouvert, sans but lucratif
 - Edition de spécifications techniques
 - >800 membres

Les grandes 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 grandes étapes

- Décembre 97
 - UML 1.1 est normalisée par l'OMG
 - RTF (Création Revision Task Force)
- Juin 1999
 - UML 1.3 publié par UML RTF
 - 458 problèmes / UML 1.1
- Septembre 2001
 - UML 1.4

Actuellement UML 1.4

- Computer Associates International Inc.
- Electronic Data Systems Corporation
- Hewlett-Packard Company
- IBM Corporation
- I-Logix
- IntelliCorp
- Microsoft Corporation
- Object Management Group
- Oracle Corporation
- Ptech Inc.
- Rational Software Corporation
- Reich Technologies
- Softeam
- Taskon A/S
- Unisys Corporation

Les grandes étapes

- Septembre 2001
 - RFP UML 2.0 divisé en 4 documents
 - Infrastructure
 - Superstructure
 - Object Constraint Language
 - Diagram Interchange
- En janvier 2003
 - U2 Partners , 2U, UML4MDA
 - Tiraillement entre vite/mieux

Portée d'UML

- Standardiser les artefacts du développement
 - Modèles, notation et diagrammes
- Ne pas standardiser le processus
 - Dirigé par les cas d'utilisation
 - Centré sur l'architecture
 - Itératif et incrémental

Acceptation d'UML

- UML est dans le domaine public
- Successeur naturel des méthodes de Booch, OMT et OOSE
- UML est le fruit de l'expérience et des besoins de la communauté des utilisateurs

UML est **la notation** pour documenter les modèles objets

Introduction au métamodèle

- UML a été d'abord défini avec UML
 - Dessins + textes
 - Méta-modèle informel
- Promotion du noyau d'UML
 - Alignement avec le MOF
 - Un langage pour exprimer des métamodèles

Ingénierie des modèles

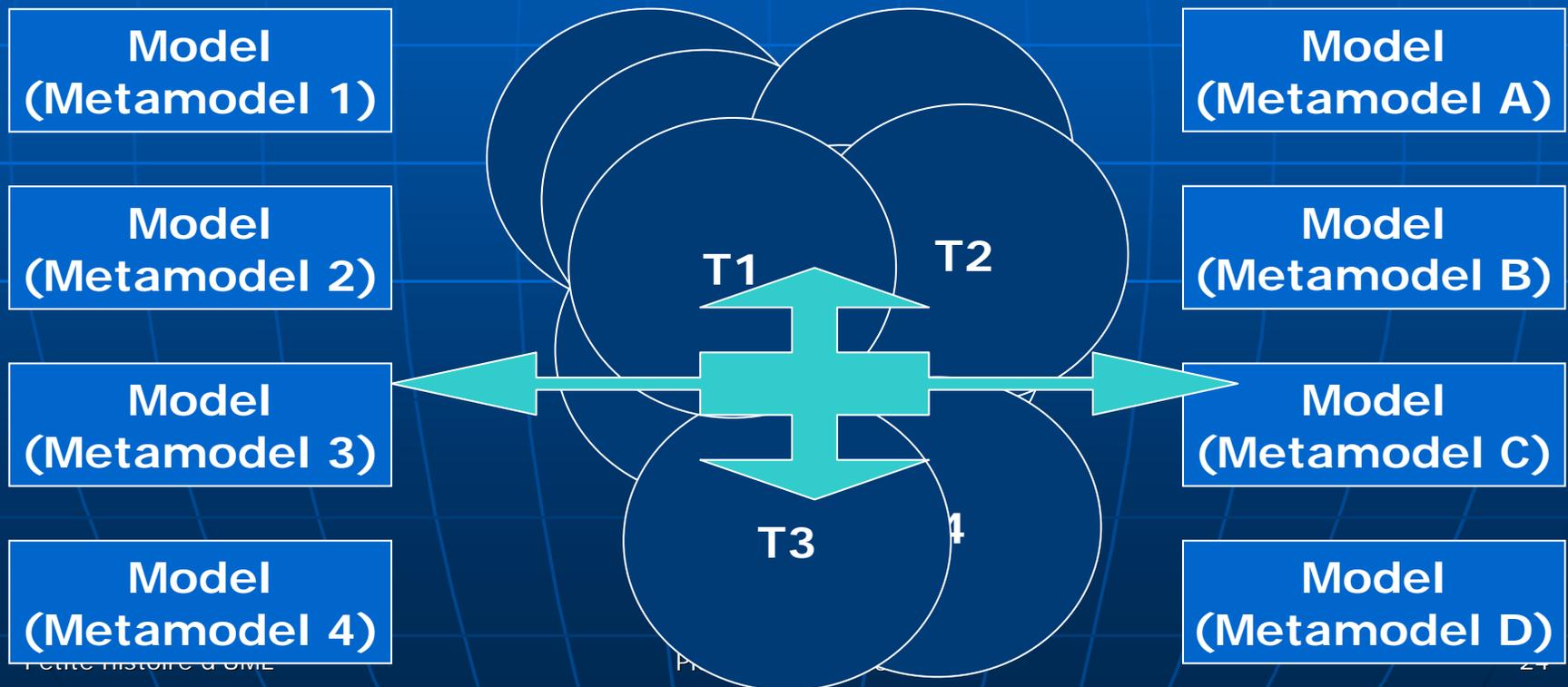
- Architecture de métamodélisation de l'OMG

M3	Méta-Métamodèle	MOF, XMI, OCL, QVT...
M2	Métamodèle	UML, CWM, SPEM, ..., Java, ...
M1	Modèle	
M0	Domaine	



Modèles et transformations

Aspects Transformations Aspects



Le minimum

- Entités (classes) + types de base
- Relations
- Paquetages

En résumé

- UML est une norme de langage de modélisation objet
- UML n'est pas une méthode
- UML convient pour toutes les méthodes objet
- UML est en évolution continue