

L
A
N
G
A
G
E

U
M
L

ENSAF
2^{ème} année Filière : Génie Informatique
Année Universitaire 2016/2017

Langage de Modélisation UML

Mohammed Berrada
mohammed.berrada@gmail.com

ENSAF1M.BERRADA

L
A
N
G
A
G
E

U
M
L

Sommaire

- Introduction au génie logiciel
- Présentation d'UML
- Diagramme de cas d'utilisation
- Diagramme de classes
- Diagramme d'objets
- Diagramme d'états-transitions
- Diagramme d'activité
- Diagrammes d'interaction (séquence et collaboration)
- Diagrammes de composants et de déploiement

ENSAF2M.BERRADA

LANGAGE
UML

ENSAF

Langage de Modélisation UML (Partie 1)

Mohammed Berrada
mohammed.berrada@gmail.com

ENSAF3M.BERRADA

LANGAGE
UML

Introduction au Génie Logiciel

ENSAF4M.BERRADA

LANGAGE UML

Plan 1

- Introduction
- Définitions
- Le Génie Logiciel : Genèse et Objectifs
- Les Cycles de vie de développement industriel de Logiciels
- Les facteurs de qualité logiciel
- Des méthodes fonctionnelles aux méthodes “Objet”

ENSAF 5 M.BERRADA

LANGAGE UML

Introduction

- Programmer n'est pas Concevoir un système informatique
- La technique ? nécessaire, mais pas si importante que ça !
- Le VRAI problème difficile : l'organisation, la gestion
 - difficulté de formalisation
 - multitude de paramètres, facteurs
 - gestions des ressources (financiers, humaines, matériel, temps)

ENSAF 6 M.BERRADA

L
A
N
G
A
G
E

U
M
L

Introduction (2)

- Comment acquérir/développer un système **sur mesure** ?
 - Que le logiciel soit
 - développé en interne
 - acheté, sous-traité
- Comment avoir/donner **confiance**
 - respect des coûts, du calendrier
 - respect des besoins fonctionnels

ENSAF 7 M.BERRADA

L
A
N
G
A
G
E

U
M
L

Définitions

- Le génie logiciel (software engineering) est une science de [génie industriel](#) qui étudie les méthodes de travail et les bonnes pratiques des ingénieurs qui [développent des logiciels](#).
- Le génie logiciel s'intéresse en particulier aux procédures systématiques qui permettent d'obtenir des logiciels de grande taille, qui correspondent aux attentes du client, sont fiables, ont un coût d'entretien réduit et de bonnes performances tout en respectant les délais et les coûts de construction.[Wikipédia]
- Ensemble de moyens (techniques, méthodes) mis en œuvre pour la construction
 - de systèmes informatiques
 - de logiciels

ENSAF 8 M.BERRADA

LANGAGE UML

Genèse et Objectifs (1)

- Difficulté de maîtrise des coûts
- Difficulté de réalisation de plannings
- Difficulté de maîtrise des délais de réalisation
- Difficulté d'amélioration de la productivité et de la qualité des logiciels

ENSAF 9 M.BERRADA

LANGAGE UML

Genèse et Objectifs (2)

- Difficulté de gestion de projets logiciels de grande ampleur (Programming in the Large)
- Nombreux échecs : résultats fournis par les logiciels insatisfaisants pour les clients finaux.

Tout ceci dans un contexte de compétition internationale sévère

ENSAF 10 M.BERRADA

LANGAGE UM L

Difficultés liées à la nature du logiciel

- un logiciel ne s'use pas, sa fiabilité ne dépend que de sa conception
- mais, pour rester utilisé un logiciel doit évoluer
- pas de direction clairement exprimée,
- changements fréquents,
- Contradictions des besoins

ENSAF 11 M.BERRADA

LANGAGE UM L

Difficultés liées aux personnes

- ne savent pas toujours ce qu'elles veulent, ou ne savent pas bien l'exprimer
- communication difficile entre personnes de métiers différents (jargons)
- l'informaticien est souvent perçu comme introverti, peu solidaire du groupe (...ça change...)
- beaucoup d'autodidactes qui croient savoir...

ENSAF 12 M.BERRADA

L
A
N
G
A
G
E

U
M
L

Difficultés technologiques

- courte durée de vie du matériel,
- beaucoup de méthodes, langages, ...
- Évolution des outils de développement
 - adaptation
 - formation
 - Investissement lourds

ENSAF
13
M.BERRADA

L
A
N
G
A
G
E

U
M
L

Le coût de développement...

- Répartition : (Ref. Boehm)
 - Analyse/Conception
 - 33-34 % : Système d'exploitation, Aérospatiale
 - 44-46 % : Contrôle et Régul. indus., Calcul scientifique, Gestion
 - Codage
 - 17-20 % : Système d'expl., Contrôle et Régul. indus., Aérospatiale
 - 26-28 % : Calcul scientifique, Gestion
 - Test/Intégration
 - 28-34 % : Contrôle et Régul. indus., Calcul scientifique, Gestion
 - 46-50 % : Système d'exploitation, Aérospatiale
 - Maintenance
 - coûts très importants...
- Peu de capitaux d'investissement nécessaires
- Frais de personnel élevés

ENSAF
14
M.BERRADA

LANGAGE UML

Quelques thèmes tirés par le Génie Logiciel

- Qualification du personnel par la formation
- Procédures de gestion de la qualité logiciel
- Outils dédiés au GL
- Langages et environnements de programmation
- Prototypage

*Il n'y a pas de remède miracle,
mais quelques voies à creuser...*

- Méthodes formelles et semi-formelles
- Réutilisabilité
- L'approche Objet

ENSAF 15 M.BERRADA

LANGAGE UML

Les cycles de vie

- **qu'est-ce qu'un cycle de vie logiciel?**
 - Enchaînement des activités de développement logiciel
 - Définition des Pré et Post conditions pour chaque phase
 - Procédures de gestion et d'encadrement
 - Procédures de mesures
 - Cycle de vie logiciel : synonyme de méthodologie logiciel

ENSAF 16 M.BERRADA

Les cycles de vie (2)

Etapes d'un cycle de vie

- Analyse : opportunité fonctionnelle et faisabilité technique
- Conception : choix tactiques de réalisation et d'architecture
- Codage : réalisation informatique du détail des opérations
- Test : tests unitaires et d'intégration
- Exploitation / Maintenance

Les deux grandes catégories de cycles de vie :

- Les cycles linéaires : succession d'étapes ordonnées
- Les cycles itératifs : réalisation incrémentale par évolutions

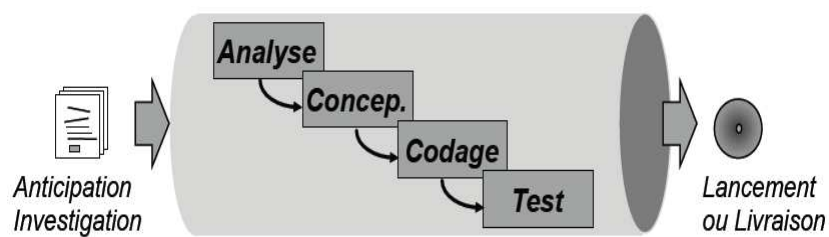
ENSAF

17

M.BERRADA

Les cycles de vie linéaires

- Cycle en cascade : ouvre des points de visibilité



ENSAF

18

M.BERRADA

L
A
N
G
A
G
E

U
M
L

Limites du modèle linéaire

- Les projets présentent bien souvent une part d'inconnu et donc de risques.
 - Méconnaissance des besoins par le client
 - Incompréhension des besoins par le fournisseur
 - Instabilité des besoins
 - Choix technologiques
 - Mouvements de personnels

=> Le processus de développement d'un logiciel n'est pas naturellement linéaire...

ENSAF
19
M.BERRADA

L
A
N
G
A
G
E

U
M
L

Les cycles de vie itératifs

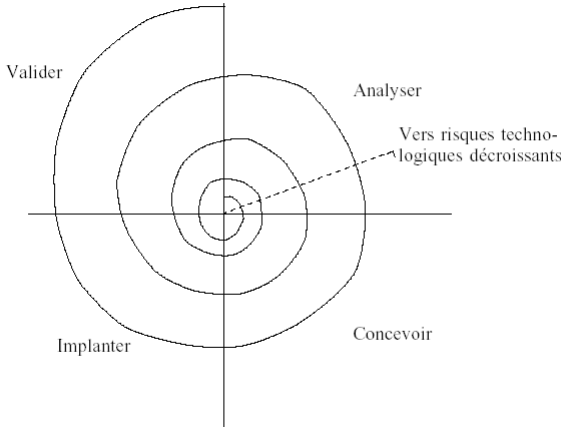
- Evaluation d'éléments concrets au cours du développement : élimination de l'effet tunnel
 - basé sur l'évolution de prototypes exécutables, mesurables
 - diminution de l'importance des documents de spéc. Détaillée
 - livraisons intermédiaires => résultats concrets réguliers de l'équipe de développement
 - meilleurs anticipation et prise en compte des problèmes
 - meilleurs gestion de la prise en compte de modifications de spécification qui peuvent être intégrées dans une itération future
 - intégration progressive de composants
 - ...

ENSAF
20
M.BERRADA

LANGAGEL

Le cycle en spirale (Boehm)

- A chaque spire, il y a itération complète sur les phases :
 - Analyse
 - Conception
 - Codage
 - Test



ENSAF 21 M.BERRADA

LANGAGEL

Le cycle en spirale (2)

- A chaque itération, le logiciel doit être dans un état quasi commercialisable
- Grand intérêt en prototypage incrémental
- Très utilisé sur les projets reposant sur l'objet.
- La première spire doit comprendre les éléments les plus abstraits et Le coeur fonctionnel minimum du système

«Design a little, code a little»

ENSAF 22 M.BERRADA

LANGAGE UML

Facteurs de qualité logiciel (1)

Facteurs externes (visibles par le client)

- **Validité** : aptitude d'un produit logiciel à remplir exactement ses fonctions, définies par le cahier des charges et les spécifications.
- **Exactitude** : le logiciel fournit les bons résultats
- **Robustesse** : le logiciel réagit correctement à des données fausses
- **Fiabilité** : exactitude + robustesse
- **Stabilité** : possibilité d'intégrer des modif. de spécification légères

ENSAF 23 M.BERRADA

LANGAGE UML

Facteurs de qualité logiciel (2)

Facteurs externes (visibles par le client)

- **Efficacité** : Utilisation optimales des ressources matérielles (performances d'exécution, encombrement mémoire,...)
- **Facilité d'emploi** : facilité d'apprentissage, d'utilisation, de préparation des données, d'interprétation des erreurs et de rattrapage en cas d'erreur d'utilisation.
- **Compatibilité** : facilité avec laquelle un logiciel peut être combiné avec d'autres logiciels.

ENSAF 24 M.BERRADA

LANGAGE UML

Facteurs de qualité logiciel (3)

Facteurs internes

- **Vérifiabilité** : facilité de préparation des procédures de test.
- **Maintenabilité** (support du temps..., testabilité, traçabilité)
- **Portabilité** : facilité avec laquelle un logiciel peut être transféré sous différents environnements matériels et logiciels.
- **Intégrité** : aptitude d'un logiciel à protéger son code et ses données contre des accès non autorisés.
- **Cohésion** : forte cohésion dans les modules
- **Faible couplage** entre les modules

ENSAF 25 M.BERRADA

LANGAGE UML

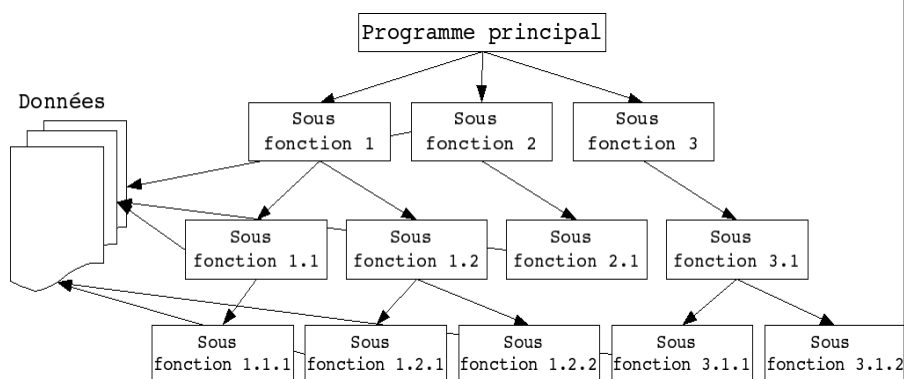
Des méthodes fonctionnelles aux méth. objet

- **L'approche fonctionnelle**
 - Raisonnement en terme de fonctions du système
 - l'accent est mis sur les fonctions et non sur les données
 - Séparation des données et du code de traitement
 - transposition dans les méthodes des contraintes du matériel
 - Diffusion des responsabilités
 - intégrité des données non garanties
 - ajout possible de nouvelles opérations à tout moment
 - Décomposition fonctionnelle descendante

ENSAF 26 M.BERRADA

L'approche fonctionnelle

- Représentation graphique d'une approche fonctionnelle

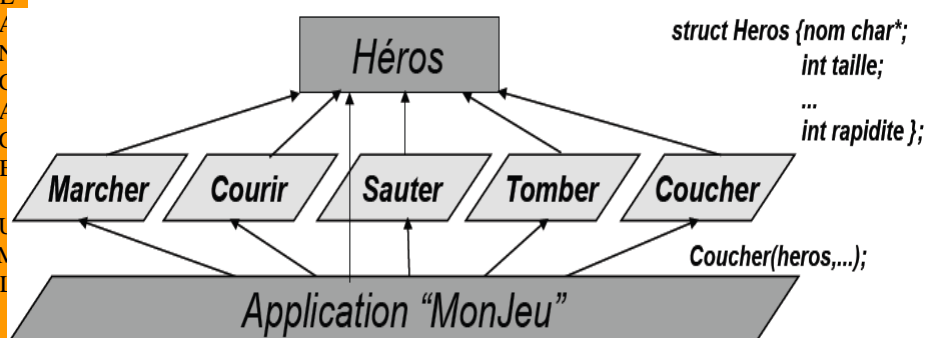


ENSAF

27

M.BERRADA

L'approche fonctionnelle (Exemple)



ENSAF

28

M.BERRADA

L
A
N
G
A
G
E

U
M
L

L'approche Objet

- Regroupement données-traitements
- Diminution de l'écart entre le monde réel et sa représentation informatique (approche naturelle)
 - Les informaticiens sont pervers : le monde est avant tout objet
- Localisation des responsabilités : encapsulation
- Décomposition par identification des relations entre objets :
 - association, composition , généralisation/spécialisation

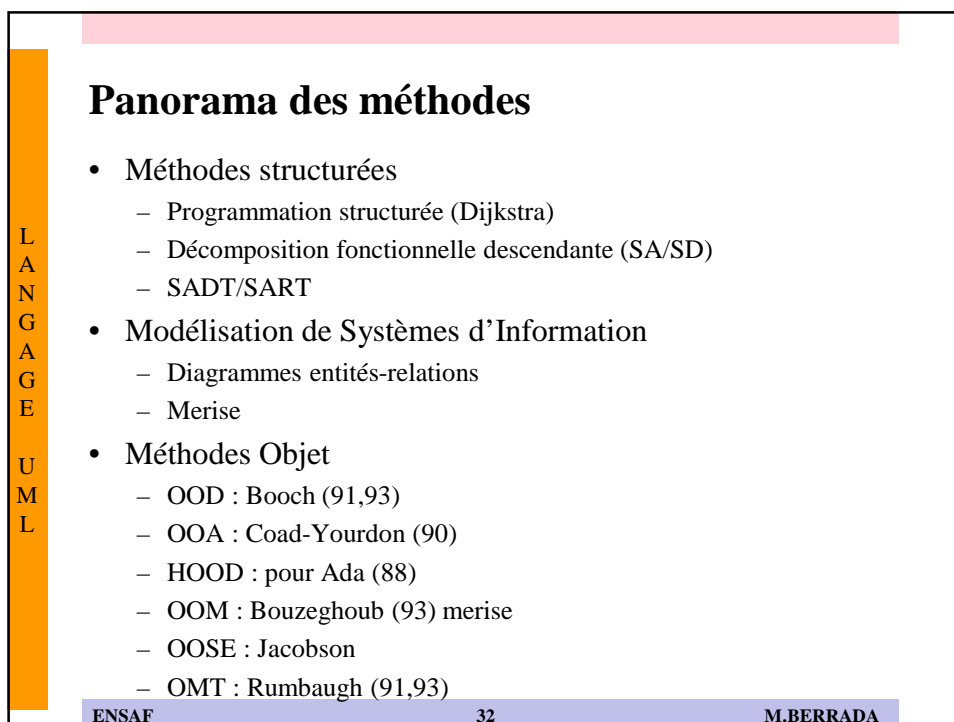
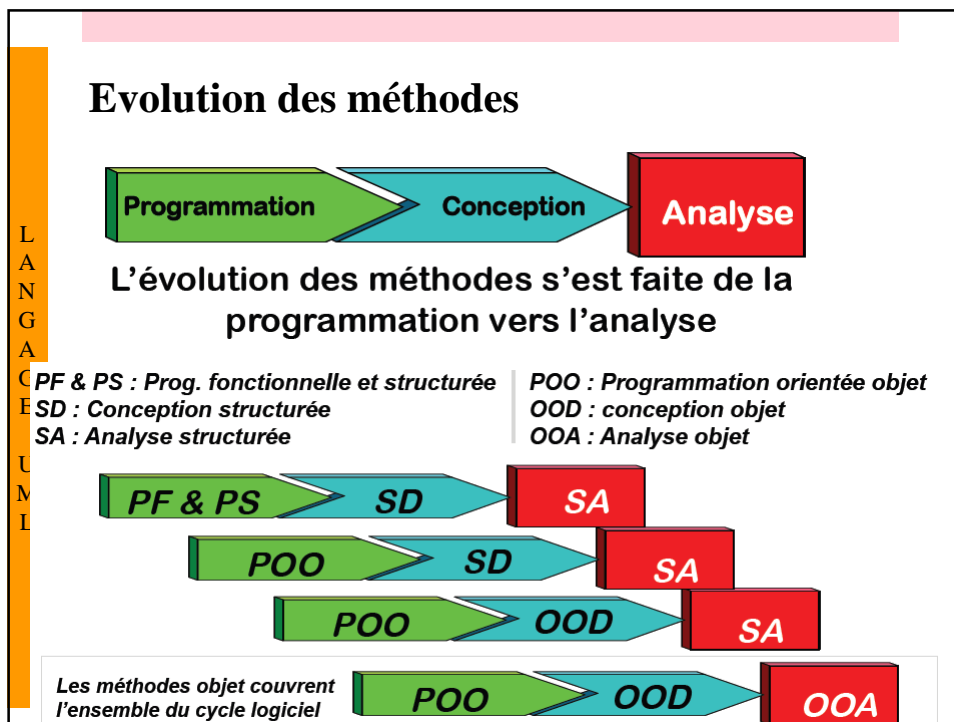
ENSAF
29
M.BERRADA

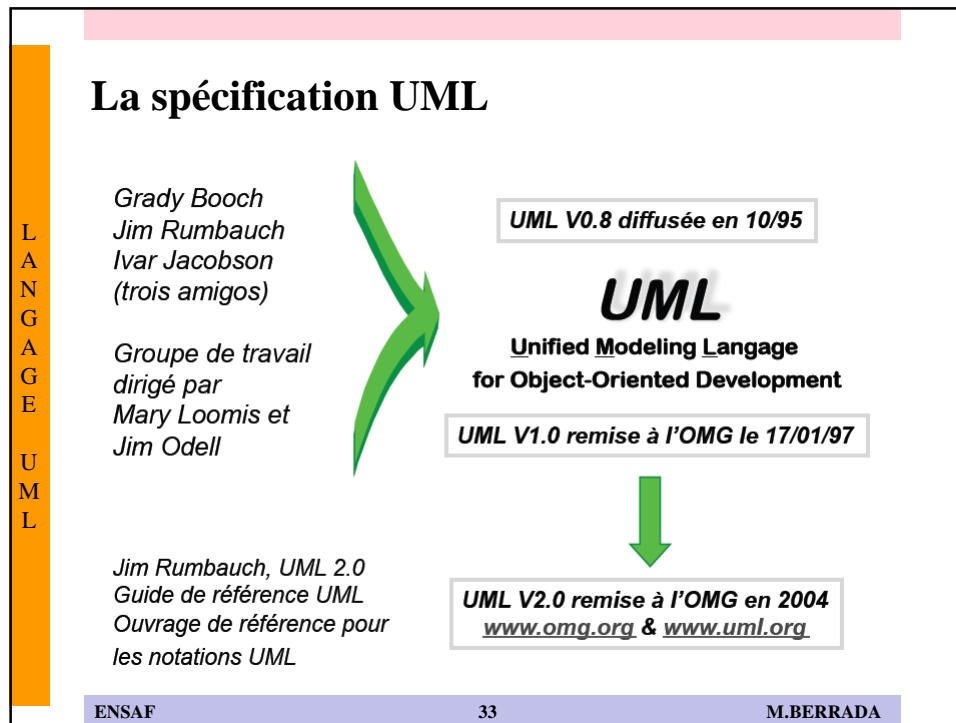
L
A
N
G
A
G
E

U
M
L

L'approche Objet (Exemple)

ENSAF
30
M.BERRADA





L
A
N
G
A
G
E

U
M
L

Plan 2

- Introduction
- La modélisation
- Concepts de l'approche Objet
- Historique d'UML
- Diagrammes d'UML
- Classification des digrammes

ENSAF
35
M.BERRADA

L
A
N
G
A
G
E

U
M
L

La modélisation

- Elle est essentielle pour :
 - Comprendre le fonctionnement d'un système
 - Maîtriser la complexité
 - Faciliter la communication au sein de l'équipe
- Et particulièrement en génie logiciel :
 - Être un facteur de réduction des **coûts** et des **délais**
 - Être un facteur d'accroissement de la **qualité** du produit

{

Validité, fiabilité, extensibilité,
réutilisabilité, compatibilité, intégrité,
portabilité, vérifiabilité, efficacité...
 - Permettre d'assurer une maintenance facile et efficace
 - Permettre de contrôler l'avancement d'un projet

ENSAF
36
M.BERRADA

LANGAGE UML

Concepts de l'approche Objet

- **Classe** : type de donnée abstrait
 - Caractéristiques : attribues et méthodes
- **Encapsulation** : masquer les détails d'implémentation d'un l'objet
 - Facilite l'évolution d'une application
 - Garantit l'intégrité des données
- **Héritage** : transmission les caractéristiques d'une classe vers une/plusieurs sous classe(s)
 - Spécialisation et généralisation
- **Polymorphisme**: faculté d'appliquer une méthode à des objets de classes différentes
- **Agrégation** : composition des objets, plus complexes, d'une classe à partir des objets d'autres classes de base

ENSAF
37
M.BERRADA

LANGAGE UML

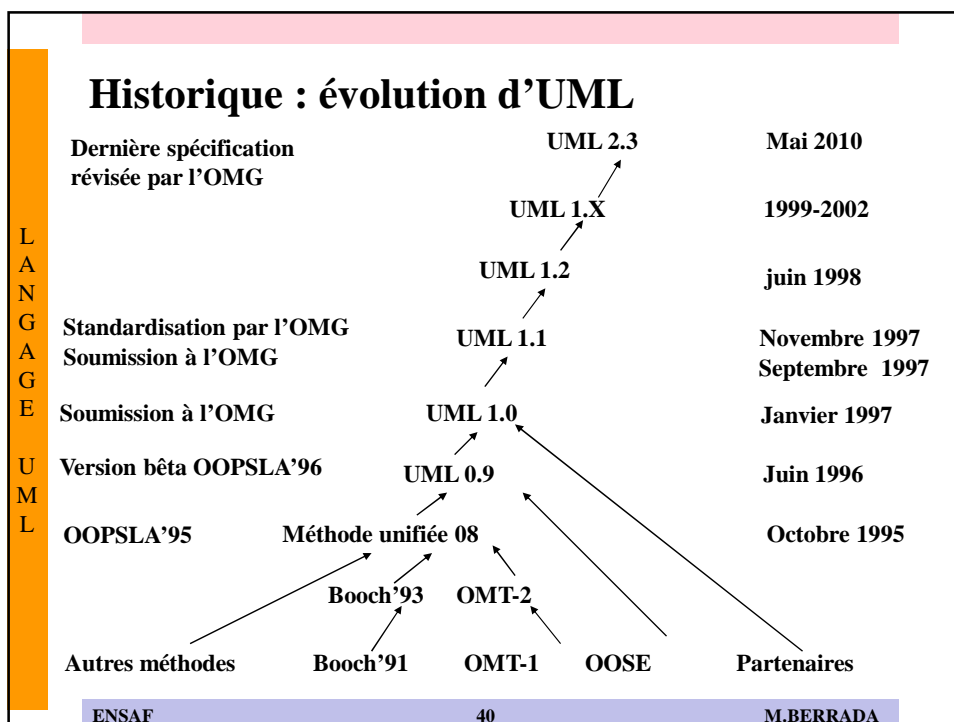
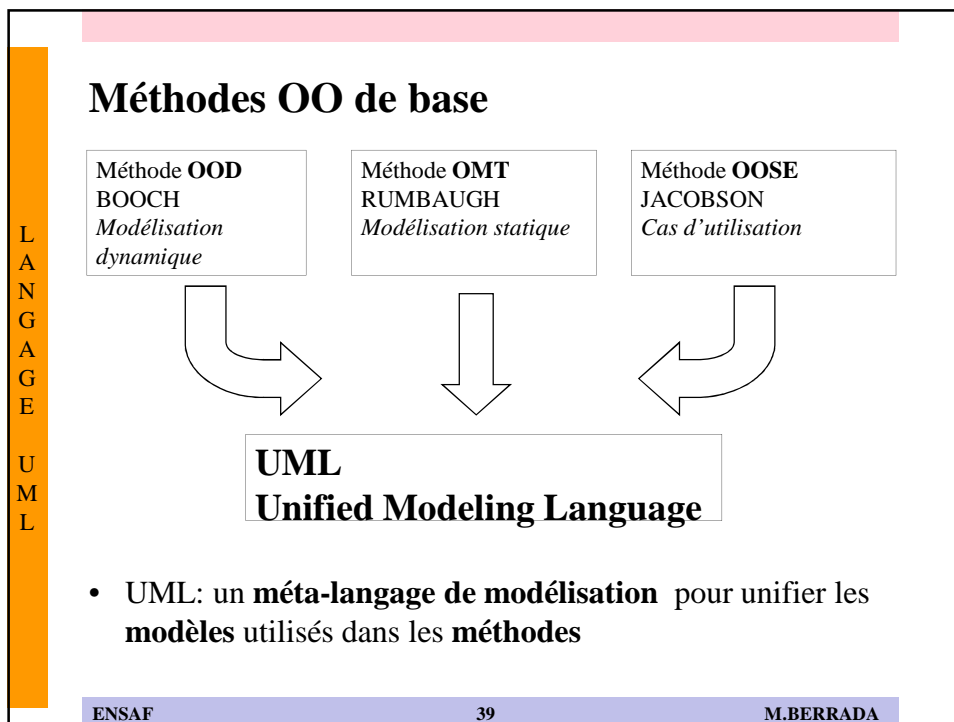
Méthodologie OO

- Les méthodes de modélisation 'classiques' sont basées sur :
 - Un modèle de données et un modèle des traitements séparés
 - Une modélisation de flots de données

Insatisfaisantes pour modéliser des systèmes objet
- L'émergence des méthodologies objets (c.1990)
 - Plus de 50 méthodes objet sont apparues
 - Exemples : Booch, Classe-Relation, Fusion, HOOD, OMT, OOA, OOD, OOM, OOSE
 - Toutes les méthodes avaient pourtant d'énormes points communs (objets, méthode, paramètres,...)

Aucune ne s'est réellement imposée

ENSAF
38
M.BERRADA



L
A
N
G
A
G
E

U
M
L

Présentation

- UML est un langage graphique qui permet de modéliser tous les types de systèmes d'informatiques.
 - Comprendre et de décrire les besoins
 - Concevoir et construire des solutions
 - Documenter un système tout au long du cycle de développement
 - Communiquer entre les membres de l'équipe de projet
- C'est une notation qui laisse la liberté de conception

ENSAF
41
M.BERRADA

L
A
N
G
A
G
E

U
M
L

Les diagrammes d 'UML

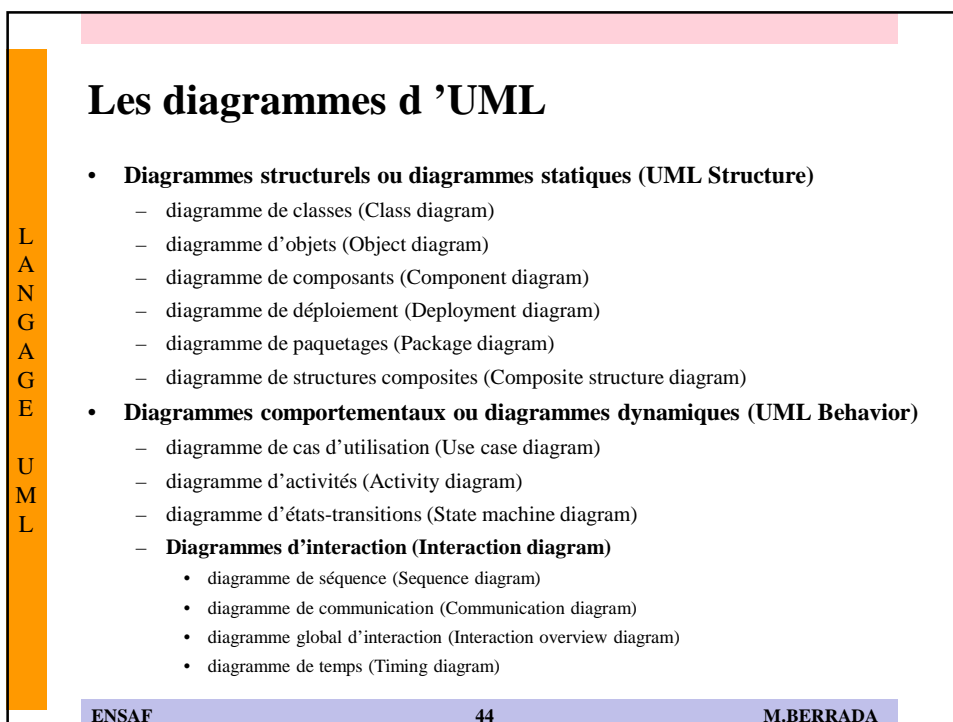
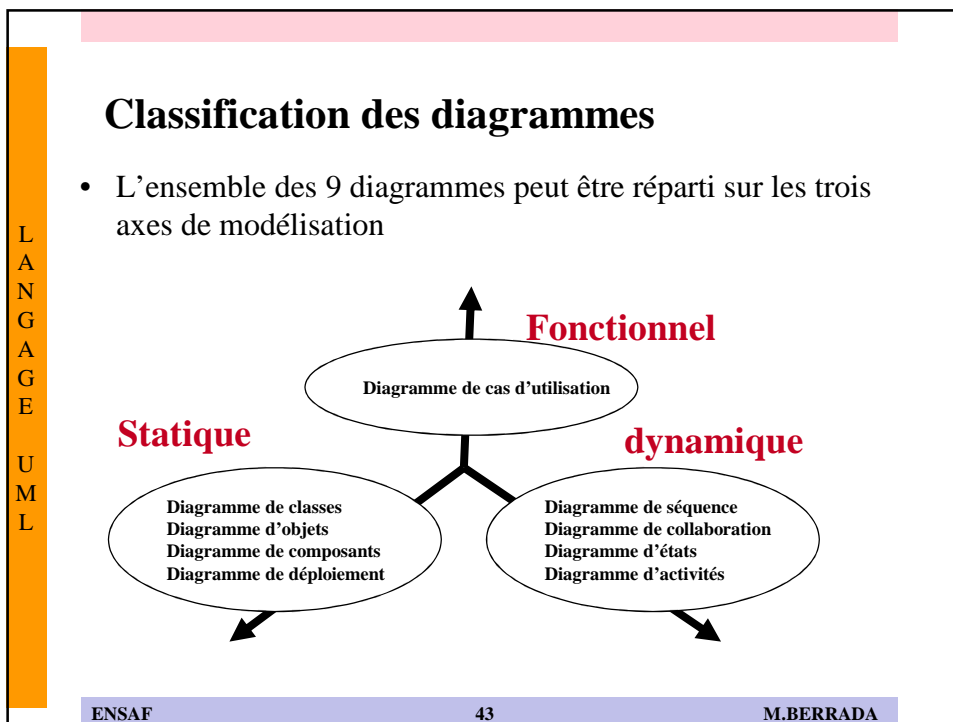
9 types de diagrammes

<ul style="list-style-type: none"> – Classes : les classes et les relations statiques – Objets : les objets et les liens – Cas d'utilisation : les acteurs et l'utilisation du système – Séquence : vision temporelle des interactions – Collaboration : vision spatiale des interactions – Etats-Transitions : le comportement des objets – Activité : le flot de contrôle interne aux opérations – Composants : les composants d'implémentation et leurs relations – Déploiement : la structure matérielle et la distribution des objets et des composants 	<div style="font-size: 2em;">}</div> <div style="font-size: 2em;">}</div> <div style="font-size: 2em;">}</div> <div style="font-size: 2em;">}</div>	Structurel Interaction Comportement Implémentation
--	---	---

4 nouveaux diagrammes sont ajoutés depuis la version 2.0

- Diagrammes de paquetages, de structures composites, global de collaboration, et de temps

ENSAF
42
M.BERRADA



LANGAGE UML

Les diagrammes d 'UML

- Diagramme **Use Case** (ou de Cas d'utilisation) : Il répond à la question "qu'est-ce que les utilisateurs attendent ?", c'est-à-dire les relations entre les acteurs et les fonctionnalités du système d'information. Niveau fonctionnel, point de vue externe.
- Diagramme **de Classe** est l'ensemble des éléments qui constituent le monde réel et les relations qui existent entre ces éléments.
- Diagramme **d'objets** : Il représente les objets et les liens qui les relient, permet de préciser un aspect particulier du diagramme de classe.
- Diagramme **de séquence** : Il représente les messages échangés entre les objets qui s'enchaînent de façon séquentielle.

ENSAF
45
M.BERRADA

LANGAGE UML

Les diagrammes d 'UML (2)

- Diagramme **State Chart** (ou d'états/transitions) : Il définit les règles d'évolution, soit le cycle de vie des objets d'une classe
- Diagramme **d'activité** : Il décrit les phases d'évolution du système et modélise les actions effectuées sur le système. Niveau abstrait
- Diagramme **de communication** : montre les relations sémantiquement faibles entre les objets
- Diagramme **de composants** : Montre le découpage du systèmes en unités pouvant être distribuées (logiciels).
- Diagramme **de déploiement** : répartition des matériels : machines, systèmes d'exploitation et les liens réseaux entre ces machines

ENSAF
46
M.BERRADA

L
A
N
G
A
G
E

U
M
L

Les diagrammes d 'UML (3)

- UML2 diagramme "interaction overview" ou **synthèse des interactions** : c'est un mélange des diagrammes de séquence et d'activité.
- UML2 diagramme **de timing** : Il décrit les contraintes temporelles sur l'évolution du système.
- UML2 diagramme **des packages** : Il montre les dépendances des éléments au niveau de la compilation.
- UML2 diagramme **des structures composites** : Comme le diagramme des packages, il montre les dépendances des éléments, mais au niveau de l'exécution.

ENSAF47M.BERRADA