

Introduction à UML

Jean-Philippe Domenger
Université Bordeaux I & LaBRI

Objectifs de ce cours

- Présenter la notation UML
 - ▮ Historique
 - ▮ Les motivations
 - ▮ Les différents diagrammes

JP Domenger, G.Eyrolles, F.Pellegrini

2

Historique

La floraison des méthodes objet pénalise la communication et la réutilisation des solutions.

Constat sur l'objet

- L'objet à 30 ans (Simula 1967)
 - ▮ atteint sa maturité
 - ▮ augmentation des logiciels objets
 - ▮ stabilité des concepts
- Concepts complexes
 - ▮ Approche objet plus délicate que l'approche fonctionnelle
 - ▮ confusion entre langage et concepts

JP Domenger, G.Eyrolles, F.Pellegrini

4

Constat sur l'objet

- Prolifération des méthodes objets
 - ▮ une cinquantaine dans les 10 dernières années
 - ▮ Aucune ne s'est imposée
 - ⇒ frein à l'essor des technologies objets
 - ▮ Difficultés d'échanger et de comparer des découpages objets
 - ▮ Apparition de modèles complémentaires

JP Domenger, G.Eyrolles, F.Pellegrini

5

Que faut-il ?

- Un langage
 - ▮ définition non ambiguë des concepts
 - ▮ extensible et générique
 - ▮ Couverture complète des aspects du développement allant de l'analyse jusqu'au déploiement.
 - ▮ Stabilisation par convergence
 - ▮ Indépendant des langages de programmation objet

JP Domenger, G.Eyrolles, F.Pellegrini

6

Que faudrait-il ?

- Un processus: formalisation et attribution des phases du développement
 - | Description des phases
 - | Enchaînement des phases
 - | Attribution des phases
- Difficile à standardiser
 - Freins: Domaine, Entreprise, Personne

JP Domenger, G.Eyrolles, F Pellegrini

7

Convergence des notations

- Des notations ont tout de même émergées
 - 1980 - OMT (Rumbaugh) maturité en 1991
 - | Modèle statique, Modèle dynamique, Modèle fonctionnel
 - 1980 - OOD (Booch) maturité en 1991
 - | Modèle statique, Modèle dynamique
 - 1990 - OOSE (Jacobson) maturité en 1992
 - | Modèle des besoins, Modèle d'analyse, Modèle de conception, Modèle d'implémentation, Modèle de test

JP Domenger, G.Eyrolles, F Pellegrini

8

Convergence des notations

- Des notations ont tout de même émergées
 - 1994- FUSION (Coleman) intégration de certains modèles provenant d'autre notations.
 - | Un premier pas vers UML

JP Domenger, G.Eyrolles, F Pellegrini

9

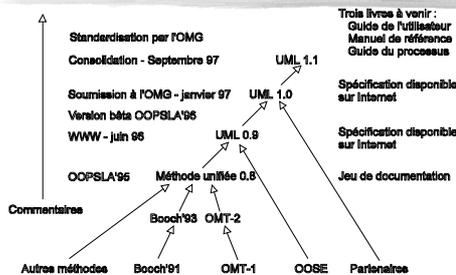
L' Emergence d'UML

- 1995 *Unified Method V0.8*
- 1996 UML V0.91 (*The Unified Modeling Language for Object-Oriented Development*)
- 1997 UML 1.0 est soumise à l'OMG
- 1997 UML 1.1 est acceptée par l'OMG
- 1999 UML 1.3

Pierre-Alain Muller

10

Emergence d'UML



Pierre-Alain Muller

11

UML: une convergence

■ Les contributions

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)

Pierre-Alain Muller

12

Le consortium UML

- Adhésion des plus grands
 - Rational, Microsoft , HP, Oracle, Sterling, MCI Systemhouse, Unisys, ICON, i-Logix, Intellicorp, IBM, ObjectTime, Platinum, Ptech, Taskon, Reich Technologies, Softeam.
- UML: un enjeu stratégique
 - Les trois fondateurs chez Rational

JP Domenger, G.Eyrolles, F Pellegrini

13

Motivation

Une documentation pour représenter l'approche objet.

UML: un langage

- UML: un langage visuel et universel
 - UML est dans le domaine public
 - Notation graphique expressive
 - Basé sur un méta-modèle
 - Formalisation de la sémantique des concepts
 - non ambiguë
 - Développement d'outils basé sur le méta-modèle
 - XMI format d'échange standard de modèle UML.

JP Domenger, G.Eyrolles, F Pellegrini

15

UML: fédère l'approche objet

- UML permet de représenter des modèles, pas de les définir.
 - Modèle: Description abstraite d'un système
 - compréhension du système
 - abstraction du système
 - simulation du système

JP Domenger, G.Eyrolles, F Pellegrini

16

UML:fédère l'approche objet

- UML permet de représenter les vues d'un système.
 - Vue utilisateur:
 - Spécifie les besoins de l'utilisateur vis à vis du système.
 - Vue logique:
 - identification des entités du domaine,
 - abstraction de ces entités,
 - définition des relations et interactions entre entités
 - répartition des entités

JP Domenger, G.Eyrolles, F Pellegrini

17

UML:fédère l'approche objet

- Vue des composants:
 - implémentation de la vue logique
 - contraintes logicielle (langage, bibliothèque)
- Vue des processus:
 - décomposition en processus
 - synchronisation et communications des processus
- Vue de déploiement
 - répartition des processus sur architecture parallèle.

JP Domenger, G.Eyrolles, F Pellegrini

18

UML: fédère l'approche objet

- UML: un outil de documentation de l'approche objet
 - ▮ Représentation des phases de développement
 - ▮ Analyse, Conception, Implémentation
 - ▮ Représentation multi-échelle
 - ▮ Raffinement d'une vue de l'analyse à l'implémentation

JP Domenger, G.Eyrolles, F Pellegrini

19

UML et le processus de développement

- UML est un langage, pas une méthode
 - ▮ pas de spécification du processus de développement
- mais UML favorise un processus
 - ▮ guidé par les besoins des utilisateurs
 - ▮ focalisé sur l'architecture logicielle
 - ▮ itératif et incrémental

JP Domenger, G.Eyrolles, F Pellegrini

20

En résumé

- Langage visuel basé sur un méta-modèle
- Langage utilisable en largeur et en profondeur
- Langage standardisé et consensuel
- Indépendant des langages de programmation
- Utilisable par tous les processus de développement

JP Domenger, G.Eyrolles, F Pellegrini

21

UML

Les différents diagrammes d'UML et leur signification.

Les diagrammes d'UML

- Diagrammes capturant l'aspect statique du système
 - ▮ Diagramme des cas d'utilisation
 - ▮ Diagramme de classe et de paquetage
 - ▮ Diagramme d'objet
 - ▮ Diagramme de composants
 - ▮ Diagramme de déploiement

JP Domenger, G.Eyrolles, F Pellegrini

23

Les diagrammes d'UML

- Diagrammes capturant l'aspect dynamique du système
 - ▮ Diagramme de collaboration
 - ▮ Diagramme de séquence
 - ▮ Diagramme d'activité
 - ▮ Diagramme d'état-transition

JP Domenger, G.Eyrolles, F Pellegrini

24

Les cas d'utilisation (1)

- Les cas d'utilisation permettent de définir un système en fonction des besoins exprimés par les utilisateurs.
 - Recenser et clarifier les besoins des clients
 - ┆ Définition des acteurs (Rôle).
 - ┆ Description de leur actions vis à vis du système
 - ┆ Délimitation de la frontière du système
 - ┆ Définition des actions/réactions du système

JP Domenger, G.Eyrolles, F Pellegrini

25

Les cas d'utilisation (2)

- Un **acteur** est un **rôle** joué par un utilisateur du système.
 - Plusieurs utilisateurs peuvent tenir le même rôle.
 - ┆ Pour une salle des marchés, il y a plusieurs personnes qui jouent le rôle de négociant
 - Un utilisateur peut avoir plusieurs rôles.
 - ┆ Un négociant peut acheter pour son propre compte.
 - Un acteur n'est pas nécessairement une personne physique, il peut être un système externe

JP Domenger, G.Eyrolles, F Pellegrini

26

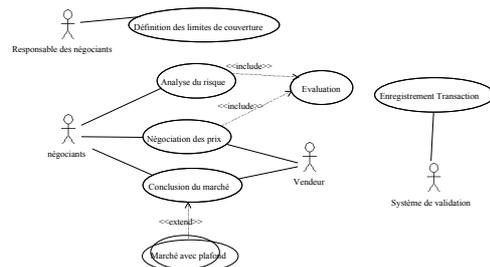
Les cas d'utilisation (3)

- L'acteur et le système:
 - L'acteur consulte ou modifie l'état du système.
 - ┆ Quelles informations sont communiquées au système ?
 - Le système répond à une action de l'acteur.
 - ┆ Quelles sont les actions de l'acteur ?
 - ┆ Quelles informations sont communiquées par le système ?

JP Domenger, G.Eyrolles, F Pellegrini

27

Les cas d'utilisation (4)

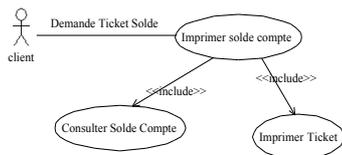


JP Domenger, G.Eyrolles, F Pellegrini

28

Les cas d'utilisation (5)

- Relation d'utilisation: un cas d'utilisation contient le comportement d'un autre

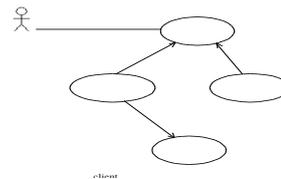


JP Domenger, G.Eyrolles, F Pellegrini

29

Les cas d'utilisation (6)

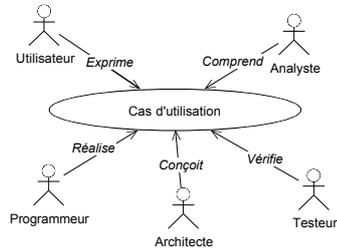
- Relation d'extension: un cas d'utilisation précise un autre cas.



JP Domenger, G.Eyrolles, F Pellegrini

30

Démarche guidée par les besoins



Pierre-Alain Muller

31

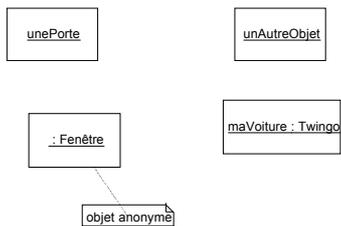
Diagramme d 'objet (1)

- Ce diagramme présente
 - l'identité des objets
 - leur classe (qui peut être indéfinie)
 - leur état (pas nécessairement la valeur de leur variable d'instance)
 - les liens entre les objets
 - une qualification des liens

JP Domenger, G.Eyrolles, F Pellegrini

32

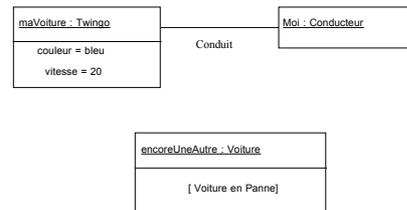
Diagramme d 'objet (2): exemple



JP Domenger, G.Eyrolles, F Pellegrini

33

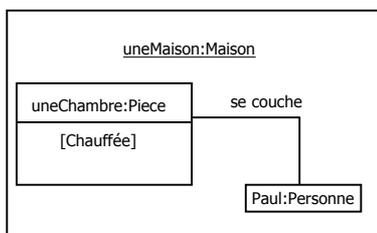
Diagramme d'objet (3): exemple



JP Domenger, G.Eyrolles, F Pellegrini

34

Diagramme d'objet (4)



JP Domenger, G.Eyrolles, F Pellegrini

35

De l'objet à la classe

- Une classe représente un ensemble d'objets.
- Un objet sera obligatoirement attaché à une classe.
- Les objets similaires sont regroupés dans la même classe.

JP Domenger, G.Eyrolles, F Pellegrini

36

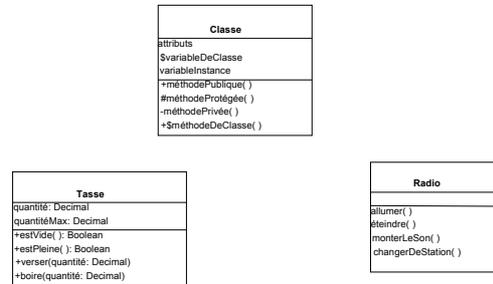
Diagramme de classe (1)

- Une classe est la déclaration statique
 - Attributs:
 - | Définition des variables d'instances
 - | Définition des variables de classes
 - Comportement:
 - | Définition des méthodes des objets
 - | Définition des méthodes de classes
 - Visibilité (public, private, protected)

JP Domenger, G.Eyrolles, F Pellegrini

37

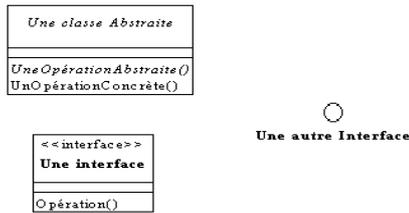
Diagramme de classe (3)



JP Domenger, G.Eyrolles, F Pellegrini

38

Diagramme de classe (4)



JP Domenger, G.Eyrolles, F Pellegrini

39

Diagramme de classe (5)

- Relation entre classes
 - Associations sont déclinées en
 - | Association simple
 - | Agrégation
 - | Composition
 - Généralisation/Spécialisation
 - Réalisation

JP Domenger, G.Eyrolles, F Pellegrini

40

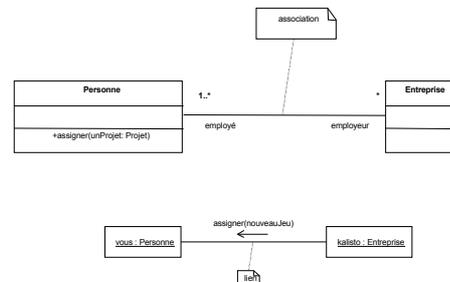
Diagramme de classe (6)

- Une association représente les liens qui existent entre les instances des classes associées. Plusieurs liens entre objet peuvent se projeter sur une même association
- L'association exprime une connexion sémantique bidirectionnelle entre classe

JP Domenger, G.Eyrolles, F Pellegrini

41

Diagramme de classe (7)

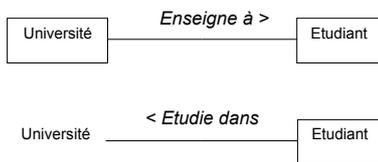


JP Domenger, G.Eyrolles, F Pellegrini

42

Diagramme de classe (7)

- Le sens de lecture d'une association peut être précisé



JP Domenger, G.Eyrolles, F Pellegrini

43

Diagramme de classe (8)

- Le rôle d'une classe pour une association peut être défini.

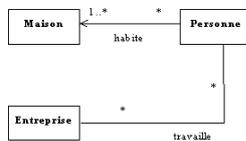


JP Domenger, G.Eyrolles, F Pellegrini

44

Diagramme de classe (9)

- Une association peut être précisée par les cardinalités



JP Domenger, G.Eyrolles, F Pellegrini

45

Diagramme de classe (10)

- La représentation des cardinalités

1	Un et un seul
0..1	Zéro ou un
M .. N	De M à N (entiers naturels)
*	Plusieurs
0 .. *	De zéro à plusieurs
1 .. *	D'un à plusieurs

JP Domenger, G.Eyrolles, F Pellegrini

46

Diagramme de classe (11) Dépendance

- On peut raffiner une association en lui donnant une direction. On exprime alors une relation de dépendance.
- Le mobile ne connaît pas le labyrinthe



JP Domenger, G.Eyrolles, F Pellegrini

47

Diagramme de classe (12) Agrégation

- Raffinement de l'association
- Correspond à une relation : *partie-de* ou *composant-composé*
 - La classe A fait partie de la classe B
 - L'état d'une instance de la classe B contient temporairement une instance de la classe A
 - une voiture a 4 roues
- La relation est orientée (navigation)

JP Domenger, G.Eyrolles, F Pellegrini

48

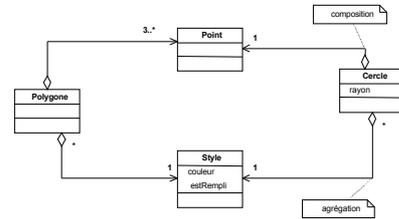
Diagramme de classe (13) Composition

- Relation partie-de plus stricte : pas de partage
- Le composé encapsule ses composants
- Durée de vie du composant est égale à celle du composé.

JP Domenger, G.Eyrolles, F Pellegrini

49

Diagramme de classe (14)

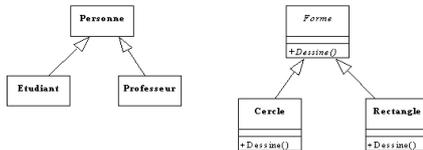


JP Domenger, G.Eyrolles, F Pellegrini

50

Diagramme de classe (15) Généralisation/Spécialisation

- La relation Généralisation/spécialisation se traduit par « est une sorte de ». Elle implique le partage de code et le sous-typage



JP Domenger, G.Eyrolles, F Pellegrini

51

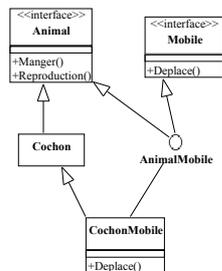
Diagramme de classe (15) Interface

- L'interface est la définition d'un rôle (type abstrait). Elle matérialise une partie du comportement d'un objet. Elle définit un ensemble de signature.

JP Domenger, G.Eyrolles, F Pellegrini

52

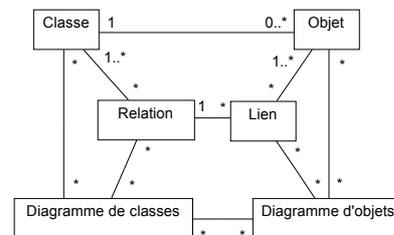
Diagramme de classe (15) Interface



JP Domenger, G.Eyrolles, F Pellegrini

53

Correspondances entre diagrammes



Pierre-Alain Muller

54

Comportement

- Le comportement dépend de l'état de l'objet
- Les objets interagissent entre eux par envois de messages
 - Un objet réalise une opération lorsqu'il reçoit un message d'un autre objet (client)
 - L'envoi de message : un concept très général pouvant être mis en oeuvre de différentes manières (par exemple par des évènements).

JP Domenger, G.Eyrolles, F Pellegrini

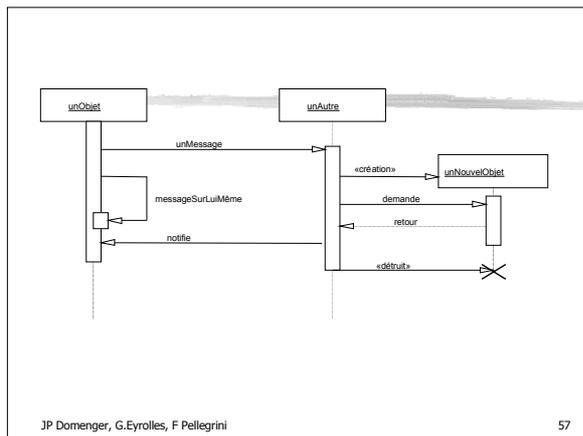
55

Diagramme de séquence

- Les objets sont représentés par une droite verticale en pointillé (axe du temps)
- Le temps s'écoule de haut en bas
- Un rectangle indique quand un objet est actif

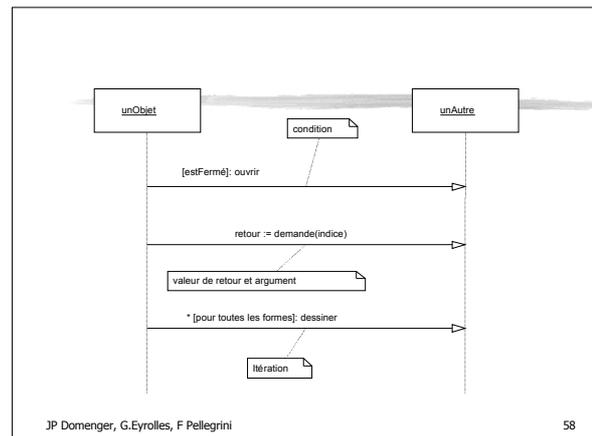
JP Domenger, G.Eyrolles, F Pellegrini

56



JP Domenger, G.Eyrolles, F Pellegrini

57



JP Domenger, G.Eyrolles, F Pellegrini

58

Démarche UML

- Définir le fonctionnel du programme, recherche du fonctionnel de haut niveau
 - Charger, Sauvegarder, Lancer/Jouer, Démos
- Trier le fonctionnel en fonction de l'apport à l'architecture.

JP Domenger, G.Eyrolles, F Pellegrini

59