



UML 2.0

Prof :Y.ABOUQORA

Introduction

UML(Unified Modeling Language):

- Langage de modélisation d'un système orientée objet ;
- Représentation graphique établie par des diagrammes ;
- UML n'est pas une méthode, est une norme qui s'appuie sur des notations et des règles syntaxiques spécifiées par l'OMG (Object Management Group) ;

Pourquoi modéliser

- Pour visualiser le système comme il est ou comme il devrait l'être.
- Pour valider le modèle vis à vis des clients
- Pour spécifier les structures de données et le comportement du système.
- Pour fournir un guide pour la construction du système.
- Pour documenter le système et les décisions prises.

Qu'est-ce qu'un modèle ?

- C'est une représentation abstraite et simplifiée de la réalité qui permet de mieux comprendre le système à développer ;
- Représenté par des schémas graphiques (diagrammes) et des textes ;
- Formalisé par un méta-modèle qui définit ses éléments et ses règles d'utilisation ;

Histoire de la naissance d'UML

- Apparition de Génie Logiciel à la fin des années 60 pour répondre à la "la crise du logiciel" ;
- Symptôme de cette crise :
 - Le logiciel livré ne satisfait pas les besoins de l'utilisateur,
 - il consomme plus de ressources que prévu et il est à l'origine de pannes.
 - Temps de réponse trop lents.
 - Non respect des délais et des coûts
 - Maintenance difficile, coûteuse et souvent à l'origine de nouvelles erreurs.

Histoire de la naissance d'UML

- Décomposition fonctionnelle du système (années 70) ;
- Diagrammes entité-relation pour la modélisation des données(années 80) ;
- L'émergence des méthodes objet (1990-1995) , plus de 50 méthodes objet sont apparues;

Histoire de la naissance d'UML

- UML est la fusion de 3 langages de modélisation objet des années 90 :
 - OMT (James Rumbaugh) : vues statiques dynamiques et fonctionnelles d'un système,
 - Booch (Grady Booch) : utilisation de 7 diagrammes pour représenter un système orienté objets,
 - OOSE (Ivar Jacobson) : analyse fondée sur la description des besoins des utilisateurs (use case),
 - 1995 : fusion OMT, Booch (Unified method 0.8),
 - 1996 : OOSE est inclus (UML 0.9), nouveaux acteurs : DEC, HP, Microsoft, Oracle, etc.
 - 1997 : l'OMG accepte UML, proposé par Rational, comme standard de modélisation objet.
 - 2007 : La version UML 2.0 constitue une évolution majeure où le langage a mûri

UML 2.0

Pourquoi UML 2.0 :

- Permettre la représentation de tous les aspects d'un problème avec un ensemble de concepts ;
- spécifier tous les éléments de la solution à mettre en place ;
- Définir, analyser et construire des diagrammes pour la modélisation des aspects du système ;

Diagrammes d'UML 2

- UML 2.0 comporte treize types de diagrammes se répartissent en deux grands groupes :
 - Diagrammes statiques(structurels)
 - Diagrammes dynamiques(comportementaux)

Diagrammes d'UML 2

- **Diagrammes statiques(structurels) :**
 - Diagramme de classe : définit les blocs de bases d'un modèle (classe, interface, types,...) ;
 - Diagramme d'objet : montre comment les instances sont reliés et utilisées pendant l'exécution ;
 - Diagramme de composant : décrit les élément physique et leurs relations dans l'environnement de réalisation, montre les choix de réalisation ;

Diagrammes d'UML 2

- **Diagrammes dynamiques(comportementaux)**
 - Diagramme de cas d'utilisation : décrit les besoins de l'utilisateur ;
 - Diagramme d'activités : définit les flux de bases, points de décision et d'action d'un processus généralisé ;
 - Diagramme d'état-transitions : exprime le comportement dynamique des objets(cycle de vie d'une classe) ;

Diagrammes d'UML 2

- **Diagrammes dynamiques(comportementaux)**
 - Diagrammes d'interactions :
 - Diagramme de séquence : montre la séquence des messages échangés entre des objets utilisant une ligne de temps verticale ;
 - Diagramme de communication : montre séquence de communication entre objets pendant l'exécution durant une instance de collaboration ;
 - Diagramme d'interaction vue globale : fusionne diagramme d'activité et de séquence pour combiner les points de décisions et des flux ;
 - Diagramme de temps : fusionne les diagrammes de séquences et d'états pour fournir une vue de l'état d'un objet dans le temps et les messages modifiant son état ;
- **Diagrammes les plus utilisés dans MOA sont diagrammes de cas d'utilisation, classes, séquence, d'activités et d'état-transition.**

Langage des contraintes d'objet (OCL)

- OCL permet de :
- Noter une condition ou une restriction sémantique exprimée grâce à un formalisme textuel simple ou composé ;
- Désigner une restriction qui doit être appliqué par une implémentation correcte ;

Conclusion

- L'apport de langage modélisation objet pour la modélisation informatique ;
- Evolution d'UML dans les années 90 ;
- Nécessité de suivre un ensemble des étapes pour faciliter la modélisation ;
- D'où le besoin de définition d'un cycle de vie d'un logiciel et des méthodes de modélisation d'UML .