

## Cours de Génie Logiciel

Sciences-U  
P. Parrend  
2005-2006

UML, Design Pattern et MDA pour la gestion de projet informatique

Plan :

### I. Introduction au Génie Logiciel

#### A. Motivation et Contenu du Cours

- i. Gestion de Projet Informatique
  - a) Quant une application est mal conçue
  - b) Outils conceptuels
  - c) Principes de conception
  - d) Gérer le cycle de vie du logiciel
- ii. Modélisation UML
  - a) Principe
  - b) Diagrammes de Cas d'utilisation
  - c) Diagrammes de Classes
  - d) Diagrammes de Séquences
- iii. Les Design Patterns
  - a) Principe
  - b) Exemples
- iv. MDA : Model Driven Architecture
  - a) Principe
  - b) Les langages de MDA

#### B. La gestion de Projet Informatique

- i. Cycle de vie du logiciel
  - a) Objectif
  - b) Etapes de création
- ii. Modèles de développement
  - a) En cascade
  - b) Itératif
  - c) En V
  - d) En spirale
  - e) Développement Agile
  - f) RAD
  - g) ISO 12207
- iii. Méthode de Conception

## II. Introduction à UML

- i. Ce qu'est UML
  - a) Cycle de vie de la conception d'une application
  - b) UML comme langage
  - c) Les diagrammes
  - d) Méthodes d'usage d'UML en entreprise
  
- ii. L'Approche Objet (1)
  - a) Avant : l'approche fonctionnelle
  - b) Raisons d'être de l'approche Objet
  - c) Quelques notions
  
- iii. Les Notations UML
  - a) vue statique
  - b) vue dynamique
  - c) Cas d'utilisation
  - d) Classes
  - e) Déploiement
  - f) Séquences
  - g) Collaboration
  - h) Etats-Transition
  - i) Activité

### B. Les diagrammes UML (1)

- i. Les Niveaux de Modèles
  
- ii. Eléments Communs aux Diagrammes
  
- iii. Diagrammes de Cas d'Utilisation
  - a) Objectifs
  - b) Intervenants du projet
  - c) Acteurs du diagramme
  - d) Cas d'utilisation
  
- iv. Diagrammes de Classes
  - a) Présentation
  - b) Classes
    - Attributs et opérations
    - Classes paramétrables
    - Stéréotypes
    - Mots clés
  - c) Associations
    - Principe
    - Représentation
    - Nommage
    - Rôles
    - Associations multiples
    - types d'associations : Généralisation

- types d'associations : agrégation
- types d'associations : composition
- Classe d'association
- Arité
- Multiplicité
- Contraintes
- Navigabilité
- Dépendances
- d) Autres types de classes
  - Classes actives
- e) Exemple

#### v. Diagrammes d'Objets

- a) Principe
- b) Représentation

#### vi. Diagrammes de Séquences

- a) Principe
- b) Elements des diagrammes de séquences : instances
- c) Elements des diagrammes de séquences : messages
- d) Types de messages

### C. Les diagrammes UML (2)

#### i. Diagrammes de Collaboration

- a) Principe
- b) Exemple
- c) Définition
- d) Représentation
- e) Niveaux de granularité
- f) Rôles
  - Niveau spécification
  - Niveau Instance
- g) Envoi de messages
- h) Interactions
- i) Place de l'utilisateur

#### ii. Diagrammes d'Etats-Transitions

- a) Généralités
  - Exemple
  - Principe
  - Représentation
- b) Etats
  - Initial et final
  - Composites
  - Synchronisation
  - Historiques
- c) Transitions
  - Principe

- Transitions composites
- d) Evènements
  - Exemple
  - Caractéristiques
  - Types d'évènements
  - Spécification
  - Evènements différés
- e) Gardes
- f) Réalisation d'automates
  - Actions
  - Points de réalisation des actions

### iii. Diagrammes d'Activités

- a) Exemple
- b) Principe
- c) Etats
- d) Transitions
- e) Flots entre actions et objets
- f) Partition de diagrammes

### iv. Diagrammes de Composants

- a) Exemple
- b) Principe
- c) Ce qu'est un composant
- d) Dépendances
- e) Processus et tâches

### v. Diagrammes de Déploiement

- a) Exemple : disposition physique du matériel
- b) Principes
- c) Les noeuds
- d) Les Supports de Communication

## III. Les Design Pattern

### A. Introduction aux Design Patterns

#### i. Un Design Pattern

- a) Problème
- b) Solution
- c) Code
- d) Exemple
- e) Analyse de l'exemple

#### ii. Motivations

- a) Ingénierie Logicielle
- b) Réutilisation de l'existant : principe
- c) Réutilisation du code
- d) Réutilisation des modèles
- e) Réutilisation d'applications

- iii. Les Design Pattern
  - a) Définition
  - b) Synonymes
  - c) Intérêt
  - d) Familles de Design Pattern
  - e) Description d'un Design Pattern

- iv. Les Framework
  - a) Définition
  - b) Caractéristiques
  - c) Types de Frameworks
  - d) Avantages
  - e) Inconvénients
  - f) Design Pattern et Frameworks

## B. Utilisation des Design Patterns

- i. Introduction
  - a) La références : Les DP du 'Gang of Four'
  - b) Définition
  - c) Types principaux
  - d) Patterns de création
  - e) Pattern de structure
  - f) Pattern de comportement
  - g) Un autre exemple
  - h) Autres familles de Patterns
- ii. Concepts Objets (2)
  - a) Le paradigme Objet
  - b) Granularité des objets
  - c) Spécification de l'interface des objets
  - d) Implémentation d'objets
  - e) Rôle des interfaces
  - f) Difficultés de la programmation Objet
  - g) Héritage et composition
  - h) Agrégation et association
- iii. Réutilisation
  - a) Conception pour la réutilisation
  - b) Causes de difficultés d'évolution
  - c) Environnement de réutilisation
- iv. Utilisation des Design Pattern
  - a) Sélection d'un Design Pattern
  - b) Usage d'un Design Pattern
  - c) A ne pas faire

## C. Trois Exemples

- i. Création
  - a) Les Design Pattern de Création
    - Liste

- Principes
- b) Le constructeur (Builder)
  - Objectif
  - Motivations
  - Applicabilité
  - Structure
  - Participants
  - Collaboration
  - Conséquences
  - Code
  - Utilisations connues
  - Patterns liés

## ii. Structure

- a) Les Design Pattern de Structure
  - Liste
  - Principes
- b) Le Composite
  - Objectif
  - Motivations
  - Applicabilité
  - Structure
  - Participants
  - Collaboration
  - Conséquences
  - Code
  - Utilisations connues
  - Patterns liés

## iii. Comportement

- a) Les Design Pattern de Comportement
  - Liste
  - Principes
- b) L'itérateur
  - Objectif
  - Motivations
  - Applicabilité
  - Structure
  - Participants
  - Collaboration
  - Conséquences
  - Code
  - Utilisations connues
  - Patterns liés

## iv. Bilan

- a) Encapsulation des variations
- b) Objets comme arguments
- c) Encapsulation ou distribution/communication
- d) Découplage émission/réception
- e) Pattern complémentaires entre autres
- f) Qu'attendre des Design Pattern ?

## D. Catalogue de Design Pattern

- i. Création
- ii. Structure
- iii. Comportement

## IV.MDE (Model Driven Engineering)

### A. Introduction à la Conception Orientée Modèle

- i. MDE : Principe
  - a) Spécifications
  - b) Principe
- ii. MDE et le Génie Logiciel
  - a) Contrôle de la conception
  - b) Evolution
  - c) Cycle de vie
  - d) Modélisation
  - e) Mort des logiciels
  - f) Approches du génie logiciel
- iii. MDE et UML
  - a) Classes et modèles
  - b) Ce qu'est UML
  - c) Ce qu'est MDE
  - d) Le niveau Méta
  - e) La méta-programmation
- iv. MDE et les Design Pattern
  - a) Intégration des Design Pattern dans MDE
- v. Le Processus MDE
  - a) Principe
  - b) Modification des modèles
  - c) Langages de modèles
  - d) Chaîne de transformation
  - e) Outils de support
  - f) Processus Complet