

Table des matières

Avant-propos	11
1 Introduction à la modélisation objet	17
1.1 Génie logiciel et modélisation	19
1.1.1 Problématique du génie logiciel	19
1.1.2 Pourquoi et comment modéliser ?	20
1.2 Cycle de vie d'un logiciel	23
1.2.1 Étapes du cycle de vie d'un logiciel	23
1.2.2 Modèles de cycles de vie linéaires	24
1.2.3 Modèles de cycles de vie itératifs	26
1.3 De la programmation structurée à l'approche orientée objet	29
1.3.1 Approche fonctionnelle ou structurée	29
1.3.2 L'approche orientée objet	30
1.3.3 Approche fonctionnelle <i>vs</i> approche objet	31
1.3.4 Concepts importants de l'approche objet	32
1.3.5 Historique de la programmation orientée objet	33
1.4 UML (<i>Unified Modeling Language</i>)	34
1.4.1 Historique de la modélisation orientée objet	34
1.4.2 Diagrammes UML	35
1.5 Travaux Dirigés – Introduction à la modélisation objet	37
2 Diagramme de cas d'utilisation & Notions transversales d'UML	41
2.1 Notions transversales du langage UML	43
2.1.1 Classeur	43
2.1.2 Paquetage	43
2.1.3 Espace de noms	44
2.1.4 Stéréotype	45
2.1.5 Note	45
2.2 Éléments des diagrammes de cas d'utilisation	46
2.2.1 Qu'est-ce qu'un diagramme de cas d'utilisation ?	46
2.2.2 Acteur	46

2.2.3	Cas d'utilisation	47
2.2.4	Représentation d'un diagramme de cas d'utilisation	47
2.3	Relations dans les diagrammes de cas d'utilisation	48
2.3.1	Relations entre acteurs et cas d'utilisation	48
2.3.2	Relations entre cas d'utilisation	49
2.3.3	Relations entre acteurs	52
2.3.4	Exemple complet de diagramme de cas d'utilisation	52
2.4	Modélisation des besoins avec les diagrammes de cas d'utilisation	53
2.4.1	Comment identifier les acteurs ?	53
2.4.2	Comment recenser les cas d'utilisation ?	54
2.4.3	Description textuelle des cas d'utilisation	55
2.4.4	Remarques concernant les diagrammes de cas d'utilisation	57
2.5	Travaux Dirigés – Diagramme de cas d'utilisation	57
3	Diagramme de classes	63
3.1	Principes fondamentaux des diagrammes de classes	65
3.1.1	Notions de classe, d'objet et d'instance	65
3.1.2	Notions d'association	65
3.1.3	Caractéristiques et responsabilités d'une classe	66
3.1.4	Notions de diagramme de classes	67
3.1.5	Notions de diagramme d'objets	68
3.2	Classe	68
3.2.1	Représentation graphique	68
3.2.2	Nom d'une classe	69
3.2.3	Encapsulation, visibilité, interface	70
3.2.4	Attributs	71
3.2.5	Opérations	72
3.2.6	Classe active	74
3.2.7	Énumération	74
3.3	Relations entre classes	75
3.3.1	Généralisation et Héritage	75
3.3.2	Association binaire et n-aire	76
3.3.3	Terminaison d'association : possession, multiplicité et navigabilité	77
3.3.4	Agrégation et composition	81
3.3.5	Classe-association	81
3.3.6	Qualification	82
3.3.7	Considérations avancées sur les associations	84
3.4	Relations de dépendance, de réalisation et notions d'interfaces	88

3.4.1	Relation de dépendance	88
3.4.2	Relation de réalisation et notions d'interfaces	88
3.5	Diagramme d'objets	90
3.6	Élaboration d'un diagramme de classes	93
3.7	Implémentation d'un diagramme de classes	94
3.7.1	Implémentation en Java	94
3.7.2	Implémentation en SQL	98
3.8	Travaux Dirigés – Diagramme de classes	102
4	<i>Object constraint language (OCL)</i>	107
4.1	Introduction aux contraintes en UML	109
4.1.1	Expression et représentation graphique des contraintes	109
4.1.2	Exemples de contraintes prédéfinies	109
4.1.3	Intérêt d'un langage de contraintes comme OCL	111
4.1.4	OCL – illustration par l'exemple	112
4.1.5	Diagramme support des exemples illustratifs	113
4.2	Typologie des contraintes OCL	114
4.2.1	Contexte (context)	114
4.2.2	Invariants (inv)	115
4.2.3	Préconditions et postconditions (pre , post)	115
4.2.4	Résultat d'une opération (body)	116
4.2.5	Définition d'attributs et d'opérations (def et let ... in)	117
4.2.6	Initialisation (init) et évolution des attributs (derive)	118
4.3	Types et opérateurs utilisables dans les expressions OCL	119
4.3.1	Types et opérateurs prédéfinis	119
4.3.2	Types du modèle UML	120
4.3.3	Hierarchie des types	120
4.3.4	Collections	120
4.4	Accès aux objets et à leurs caractéristiques	121
4.4.1	Accès aux attributs et aux opérations (self)	121
4.4.2	Navigation via une association	121
4.4.3	Navigation via une association qualifiée	122
4.4.4	Navigation vers une classe association	123
4.4.5	Navigation depuis une classe association	123
4.4.6	Accéder à une caractéristique redéfinie (oclAsType())	124
4.4.7	Opérations prédéfinies sur tous les objets	124
4.4.8	Opérations sur les classes	125
4.5	Opérations sur les collections	125

4.5.1	Introduction : «.», «->», «::» et self	125
4.5.2	Opérations de base sur les collections	126
4.5.3	Opérations sur les éléments d'une collection	127
4.5.4	Règles de priorité des opérateurs	130
4.6	Exemples de contraintes	130
4.7	Travaux Dirigés – OCL	132
5	Diagramme d'états-transitions	135
5.1	Introduction au formalisme	137
5.1.1	Notion d'automate à états finis	137
5.1.2	Diagrammes d'états-transitions	137
5.2	États	138
5.2.1	État dans un diagramme d'états-transitions	138
5.2.2	État global d'un objet, ou du diagramme d'états-transitions	138
5.2.3	État initial et final	139
5.3	Événements	139
5.3.1	Notions d'événement	139
5.3.2	Événement de type signal (signal)	139
5.3.3	Événement d'appel (call)	140
5.3.4	Événement de changement (change)	140
5.3.5	Événement temporel (<i>after</i> ou <i>when</i>)	141
5.4	Transitions	141
5.4.1	Définition et syntaxe	141
5.4.2	Transition externe	142
5.4.3	Transition d'achèvement	142
5.4.4	Transition interne	142
5.5	Points de choix	143
5.5.1	Point de jonction	143
5.5.2	Point de décision	145
5.6	États composites	145
5.6.1	Présentation	145
5.6.2	Transition	146
5.6.3	État historique	147
5.6.4	Interface : les points de connexion	148
5.6.5	Concurrence	149
5.7	Travaux Dirigés – Diagramme d'états-transitions	150
6	Diagramme d'activités	153

6.1	Action, activité, nœud et transition	155
6.1.1	Action	155
6.1.2	Activité	156
6.1.3	Groupe d'activités	156
6.1.4	Nœud d'activité	156
6.1.5	Nœud exécutable et nœud d'action	156
6.1.6	Transition	158
6.2	Nœuds de contrôle	159
6.2.1	Nœud initial	160
6.2.2	Nœud final	160
6.2.3	Nœuds de décision et de fusion	160
6.2.4	Nœuds de bifurcation et d'union	161
6.3	Nœuds d'objet	162
6.3.1	Introduction	162
6.3.2	Pins d'entrée/sortie	162
6.3.3	Pin de valeur	162
6.3.4	Flot d'objet	163
6.3.5	Nœud tampon central	163
6.3.6	Nœud de stockage des données	164
6.4	Nœud d'activité structurée	164
6.4.1	Nœud d'activité structurée	164
6.4.2	Structures de contrôle	165
6.4.3	Zone d'expansion	166
6.5	Partitions, exceptions et interruptions	167
6.5.1	Partitions	167
6.5.2	Gestion des exceptions	168
6.5.3	Région interruptible	170
6.6	Travaux Dirigés – Diagramme d'activités	171
7	Diagrammes d'interaction & diagramme de structures composites	175
7.1	Diagramme de structures composites	177
7.1.1	Notions de diagramme de structures composites	177
7.1.2	Éléments constitutifs d'un diagramme de structures composites	178
7.1.3	Classeur structuré	180
7.1.4	Collaboration	180
7.2	Diagrammes d'interaction : présentation d'ensemble	181
7.2.1	Interactions et lignes de vie	181
7.2.2	Représentation graphique	182

7.3	Diagramme de communication	183
7.3.1	Représentation des lignes de vie	183
7.3.2	Représentation des connecteurs	183
7.3.3	Représentation des messages	184
7.4	Diagramme de séquence	184
7.4.1	Représentation des lignes de vie	184
7.4.2	Représentation des messages	185
7.4.3	Fragments d'interaction combinés	189
7.4.4	Utilisation d'interaction	192
7.5	Travaux Dirigés – Diagramme d'interaction	192
8	Diagrammes de composants & de déploiement	197
8.1	Diagrammes de composants	198
8.1.1	Pourquoi des composants?	198
8.1.2	Notions de composant	198
8.1.3	Diagramme de composants	200
8.2	Diagramme de déploiement	200
8.2.1	Objectifs du diagramme de déploiement	200
8.2.2	Représentation des nœuds	201
8.2.3	Notions d'artefact (<i>artifact</i>)	201
8.2.4	Diagramme de déploiement	203
9	Patrons de conception	205
9.1	Introduction aux patrons de conception	207
9.1.1	Qu'est-ce qu'un patron de conception	207
9.1.2	Intérêt des patrons de conception	207
9.1.3	Patrons de conception fondamentaux	208
9.2	Principaux patrons de conception GRASP	209
9.2.1	Patron <i>Créateur</i>	209
9.2.2	Patron <i>Expert en information</i> ou <i>Expert</i>	210
9.2.3	Patron <i>Faible Couplage</i>	211
9.2.4	Patron <i>Forte Cohésion</i>	212
9.2.5	Patron <i>Contrôleur</i>	213
9.3	Principaux patrons de conception du GoF	213
9.3.1	Patron <i>Façade</i>	213
9.3.2	Patron <i>Singleton</i>	214
9.3.3	Patron <i>Patron de Méthode</i>	215
9.3.4	Patron <i>Stratégie</i>	216

9.3.5	Patron <i>Composite</i>	218
9.3.6	Patron <i>Observateur</i>	219
9.4	Autres patrons	221
9.4.1	Patron <i>Énumération</i>	221
9.4.2	Patron <i>Métaclasse</i> ou <i>Type-Exemplaires</i>	222
9.4.3	Patron <i>Couche</i>	224
9.4.4	Patron <i>Modèle-Vue-Contrôleur</i>	226
10	Mise en œuvre d'UML	229
10.1	Méthodes de développement orienté objet	230
10.1.1	Fondements du Processus Unifié	230
10.1.2	Disciplines et phases du Processus Unifié	231
10.1.3	Méthodes Agiles	233
10.1.4	Scrum	235
10.1.5	<i>eXtreme Programming</i> (XP)	236
10.1.6	Choix d'une méthode simple et générique	238
10.2	Identification des besoins et spécification des fonctionnalités	240
10.2.1	Identification et représentation des besoins	240
10.2.2	Spécification détaillée des besoins	241
10.2.3	Maquette de l'IHM	241
10.3	Phase d'analyse	242
10.3.1	Modèle du domaine	242
10.3.2	Diagramme de classes participantes	244
10.3.3	Diagrammes d'activités de navigation	246
10.4	Phase de conception	247
10.4.1	Diagrammes d'interaction	247
10.4.2	Diagramme de classes de conception	250
	Annexes	253
A	Corrections des exercices	253
A.1	Correction du TD 1.5 – Introduction à la modélisation objet	254
A.2	Correction du TD 2.5 – Diagramme de cas d'utilisation	258
A.3	Correction du TD 3.8 – Diagramme de classes	264
A.4	Correction du TD 4.7 – OCL	275
A.5	Correction du TD 5.7 – Diagramme d'états-transitions	277
A.6	Correction du TD 6.6 – Diagramme d'activités	282
A.7	Correction du TD 7.5 – Diagramme d'interactions	284

B Bibliographie	289
Index	293