

# Table des matières

<b>Introduction</b>	<b>xi</b>
<b>I Les concepts</b>	<b>1</b>
<b>1 Les modèles dans le développement logiciel</b>	<b>3</b>
1.1 Principes généraux . . . . .	3
1.1.1 Modèle . . . . .	4
1.1.2 Modèles, aspects et lignes de produits . . . . .	6
1.1.3 Métamodèle . . . . .	10
1.2 L’approche MDA . . . . .	11
1.3 Langages dédiés de modélisation . . . . .	13
<b>2 La métamodélisation</b>	<b>17</b>
2.1 Qu’est-ce qu’un langage ? . . . . .	17
2.2 Syntaxe abstraite . . . . .	20
2.2.1 Le langage de métamodélisation MOF de l’OMG . . . . .	20
2.2.2 Le langage de contrainte OCL de l’OMG . . . . .	21
2.3 Syntaxe concrète . . . . .	23
2.4 Sémantique . . . . .	24
2.4.1 Rappels . . . . .	24
2.4.2 Sémantique axiomatique . . . . .	25
2.4.3 Sémantique opérationnelle . . . . .	26
2.4.4 Sémantique dénotationnelle (ou <i>par traduction</i> ) . . . . .	28
2.4.5 Bilan . . . . .	29
<b>3 Vers des modèles productifs</b>	<b>33</b>
3.1 La transformation de modèle . . . . .	33

3.1.1	Historique . . . . .	35
3.1.2	Standards et langages pour la transformation de modèle . . . . .	37
3.2	La composition de modèle . . . . .	38
3.2.1	Chaînes de compilation . . . . .	39
3.2.2	Co-évolution ou synchronisation de modèles . . . . .	39
3.2.3	Interopérabilité de systèmes . . . . .	39
3.2.4	Évolution de système . . . . .	40
3.2.5	Bilan . . . . .	40
<b>II</b>	<b>La pratique</b>	<b>43</b>
<b>4</b>	<b>Processus de métamodélisation</b>	<b>45</b>
4.1	Différents profils d'utilisateurs . . . . .	45
4.2	Éléments d'un environnement de modélisation dédié . . . . .	47
4.3	Étapes d'un processus de métamodélisation . . . . .	47
4.4	Quelques outils supports à l'IDM . . . . .	51
4.4.1	Eclipse Modeling Project . . . . .	51
4.4.2	Kermeta . . . . .	52
4.5	Le format XMI pour la sérialisation des modèles . . . . .	53
<b>5</b>	<b>Définition d'un domaine métier</b>	<b>55</b>
5.1	Langages de métamodélisation . . . . .	55
5.1.1	Le langage de métamodélisation Ecore . . . . .	55
5.1.2	Le langage de métamodélisation exécutable Kermeta . . . . .	57
5.2	Exemple de domaine métier : Logo . . . . .	59
5.3	Metamodélisation de Logo . . . . .	60
5.4	Bonnes pratiques . . . . .	62
5.4.1	Arbre de contenance . . . . .	62
5.4.2	Composition/extension de domaine métier . . . . .	63
5.4.3	Autres bonnes pratiques issues de la modélisation objet . . . . .	64
<b>6</b>	<b>Vérification statique de modèle</b>	<b>65</b>
6.1	Utilisation d'OCL dans la métamodélisation . . . . .	66
6.2	Expression de la sémantique statique de Logo . . . . .	67
6.3	Définition de la sémantique statique de Logo . . . . .	68
6.4	Intégration du vérificateur statique de modèle dans un AGL . . . . .	70

<b>7</b>	<b>Édition de modèle</b>	<b>71</b>
7.1	Éditeur arborescent . . . . .	73
7.2	Éditeur textuel . . . . .	74
7.3	Éditeur graphique . . . . .	76
7.4	Bonnes pratiques . . . . .	79
<b>8</b>	<b>Simulation de modèle</b>	<b>81</b>
8.1	Kermeta pour décrire le comportement des modèles . . . . .	82
8.2	Définition du domaine sémantique de Logo . . . . .	84
8.3	Définition de la sémantique dynamique de Logo . . . . .	86
8.4	Définition d'un interpréteur de programme Logo . . . . .	88
8.5	Définition d'un simulateur de programme Logo . . . . .	89
8.6	Autres techniques de simulation . . . . .	91
<b>9</b>	<b>Compilation, <i>refactoring</i>, etc.</b>	<b>93</b>
9.1	Transformations Modèle-vers-Texte et Modèle-vers-Modèle . . .	93
9.2	Compilateur de modèle . . . . .	95
9.2.1	Choix d'une plateforme d'exécution cible . . . . .	95
9.2.2	Définition d'un compilateur par l'utilisation d'un visiteur	96
9.2.3	Définition d'un compilateur par l'utilisation de templates	98
9.3	Outils de refactoring . . . . .	101
	<b>Conclusion</b>	<b>103</b>
	<b>Livres de références</b>	<b>105</b>
	<b>Bibliographie</b>	<b>113</b>
	<b>Table des figures</b>	<b>113</b>
	<b>Listings</b>	<b>117</b>
	<b>Index</b>	<b>123</b>