

Table des Matières

Table des Figures	7
Introduction Générale	9
1. Outils et plate-formes de construction d'application	9
2. Intégration de paradigmes de conception dans le cycle de vie	10
2.1. Equilibrage de charge et équilibrage d'application	11
2.2. Applications parallèles et applications réparties	12
2.3. Unification des deux problématiques	13
2.4. Synthèse	14
3. Positionnement et démarche méthodologique	14
3.1. Intégration de la notion d'architecture	15
3.2. Approche méthodologique	15
4. Organisation du mémoire	16
5. Références bibliographiques	20
Chapitre 1 - Langages de description d'architectures matérielles hybrides	23
1. Classifications des architectures matérielles multi-processeurs	25
1.1. La classification de Flynn	25
1.2. Vers des classifications alternatives	26
1.3. Synthèse : définition d'une architecture hybride	27
2. Langage de description d'architectures matérielles	29
2.1. La description de l'architecture matérielle dans les langages de programmation parallèle	30
2.2. La description de l'architecture matérielle dans les langages de configuration	32
2.3. La description de l'architecture matérielle dans les logiciels de simulation	34
2.4. La description de l'architecture matérielle dans les logiciels de répartition de charge	40
2.5. La description de l'architecture matérielle dans les logiciels d'administration de systèmes et de réseaux	41
2.6. Conclusion	43
3. Hadel : un langage de description d'architectures matérielles Hybrides	44
3.1. Description du langage HADEL	45
3.2. Conclusion	50
3.3. Perspectives	51
4. Références bibliographiques	52
Chapitre 2 - Architecture logicielle et construction d'applications client-serveur	55
1. Définition et description d'architectures logicielles	58
1.1. Séparation des préoccupations et vues sur l'architecture	59
1.2. Style architectural	60
1.3. Gabarit de conception	63

1.4.	Squelette d'implémentation	66
1.5.	Langages de description d'architectures logicielles	67
1.6.	Synthèse	71
1.7.	Notre vision d'une architecture logicielle dans MEDEVER	73
2.	Les architectures logicielles de type client-serveur	74
2.1.	Classification des architectures client-serveur.....	74
2.2.	Les environnements de développement client-serveur	77
2.3.	Le médiateur : colle des architectures logicielles client-serveur	80
2.4.	Conclusion	86
3.	Les architectures client-serveur selon un axe langage	87
3.1.	Rappels sur le modèle objet.....	87
3.2.	Classification des langages objets.....	89
3.3.	Révision du modèle objet en vue de l'intégration de la concurrence	90
3.4.	Mise en oeuvre de la concurrence	93
3.5.	Conclusion	95
4.	Les architectures client-serveur selon un axe système	96
4.1.	Les architectures de systèmes ouverts.....	97
4.2.	Les plate-formes d'intégration.....	99
4.3.	Les architectures logicielles à base de composants logiciels	100
4.4.	Les architectures logicielles basées sur des ORB.....	104
4.5.	Les Architectures d'applications client-serveur s'appuyant sur des Systèmes Répartis	109
4.6.	Synthèse	112
5.	Conclusion	113
6.	Références bibliographiques	114

**Chapitre 3 - VODEL, un langage de description d'architectures logicielles
statiques et dynamiques** **121**

1.	Le langage VODEL	122
1.1.	Style architectural définit dans VODEL	124
1.2.	Description des entités logicielles (ou DVSC)	126
1.3.	Les entités de communication dans VODEL : DVSL et DVGL	128
1.4.	Synthèse	134
2.	Intégration de la dynamique dans VODEL : VODEL-D	134
2.1.	Les entités logicielles de définition : étendre les DVSC	135
2.2.	Les entités logicielles d'exécution	136
2.3.	Modélisation des contraintes d'évolution dynamique d'un canal	136
2.4.	Les plans d'exécution	137
2.5.	Intégration d'éléments de description de HADEL dans VODEL-D	139
2.6.	Synthèse	141
3.	Un exemple de description d'une application répartie	142
3.1.	Description des entités passives physiques et logiques.....	143
3.2.	Description des entités actives.....	144

3.3.	Description des entités logicielles de définition.....	145
3.4.	Descriptions des liens de communication entre entités logicielles de définition.....	146
3.5.	Description des entités logicielles d'exécution.....	148
3.6.	Description des plans d'exécution.....	149
3.7.	Analyse près exécution.....	149
3.8.	Discussion.....	150
4.	Conclusion	151
5.	Références bibliographiques	152
Chapitre 4 - Placement de composants logiciels : Application à l'équilibrage d'application		153
1.	Les méthodes de prototypage	155
1.1.	Première classification des techniques de prototypage.....	155
1.2.	Autre classification des techniques de prototypage.....	156
1.3.	Synthèse : le prototypage par raffinements.....	156
2.	Contexte de l'étude : la méthode MARS	157
2.1.	Les trois formalismes de MARS.....	158
2.2.	Génération d'applications et MEDEVER.....	159
3.	Génération d'applications à partir de réseaux de Petri	160
3.1.	Etat de l'art.....	160
3.2.	Notre approche de la génération d'applications.....	162
3.3.	Conclusion.....	169
4.	Génération d'applications à partir de H-COSTAM	170
4.1.	Présentation de H-COSTAM.....	171
4.2.	Placement des objets H-COSTAM.....	174
4.3.	Conclusion.....	180
5.	Application à un exemple de chaîne d'assemblage automobile	180
5.1.	Le modèle initial.....	180
5.2.	Prototypage à partir du réseau de Petri.....	181
5.3.	Modélisation avec H-COSTAM.....	184
5.4.	Comparaison des deux approches.....	187
6.	Conclusion	188
6.1.	Génération d'applications à partir d'un modèle formel.....	188
6.2.	Génération d'applications à partir d'un modèle opérationnel.....	189
6.3.	Mise en oeuvre et perspectives.....	189
7.	Références bibliographiques	190
Chapitre 5 - Placement de composants logiciels : Application à l'équilibrage de charge		195
1.	Placement sur des systèmes parallèles et répartis	196
1.1.	Représentation des programmes.....	197
1.2.	Les algorithmes de placement.....	198
1.3.	Du placement dynamique vers le placement hybride.....	207

1.4. Synthèse	210
2. Equilibrage de charge et placement hybride dans IDEFIX	212
2.1. Gestion de l'équilibrage de charge	213
2.2. Intégration de l'environnement dynamique.....	214
2.3. Gestion des applications dynamiques : VODEL-D.....	216
3. Conclusion	216
4. Références bibliographiques	217
Chapitre 6 - Mise en oeuvre	221
1. Description et gestion de l'architecture matérielle	222
1.1. La description à l'aide de Macao	223
1.2. Gestion de bibliothèques d'architectures matérielles	224
1.3. Gestion des requêtes.....	225
1.4. Conclusion	226
2. Présentation de Gatostar	226
2.1. Environnement système de Gatostar	226
2.2. Le modèle d'application et le langage GEL	227
2.3. Placement dynamique multi-critères	228
2.4. Synthèse	228
3. IDEFIX : intégration de Gatostar, HADEL et VODEL-D	228
3.1. Placement Statique : de VODEL-D vers GEL	229
3.2. Intégration de la gestion des applications dynamiques : VODEL-D	230
3.3. Intégration de VODEL-D et de Gatostar	231
3.4. Synthèse	232
4. Mise en oeuvre d'IDEFIX	232
4.1. Le segment de mémoire partagée.....	233
4.2. Le gestionnaire de cohérence	235
4.3. Sessions coopératives	239
4.4. Les compilateurs de VODEL-D	240
4.5. Deux utilitaires développés dans IDEFIX	241
4.6. Performances	242
5. Perspectives	242
6. Références bibliographiques	243
Conclusion Générale	245
1. MEDEVER et sa mise en oeuvre	245
1.1. Description des architectures	246
1.2. Recherche de la granularité des composants et de leur placement.....	246
1.3. Exécution répartie.....	247
1.4. Analyse de l'exécution et optimisation	247
1.5. Conclusion	248
2. Perspectives	250

2.1. Equilibrage d'application et génération de code	250
2.2. Equilibrage de charge et gestion de la mobilité	251
3. Références bibliographiques	254
Annexe A : Bibliographie commentée	259
Langages de programmation objets concurrents	260
Langages de description d'architectures logicielles	270
Normes et spécifications	281
Protocoles d'interactions	292
Environnements	299
Systèmes répartis et systèmes agents	317
Annexe B : BNF du langage HADEL	327
Annexe C : BNF du langage VODEL-D	333
Annexe D : BNF du formalisme H-COSTAM	341
Annexe E : Comparaison des bibliothèques de communication hétérogènes	351
Annexe F : Les jeux d'essai (benchmark)	357
Index Général	361

