

Table des Figures

Figure 1 : Le cycle de vie d'une application constituée de composants logiciels	11
Figure 2 : La méthode MEDEVER	16
Figure 3 : Notre vision de l'évolution des architectures parallèles	28
Figure 4 : Le modèle de placement à deux niveaux de HPF	32
Figure 5 : Description Macro d'un réseau hybride avec HADEL	46
Figure 6 : Propriétés au niveau Macro d'une H-Machine	46
Figure 7 : Propriétés au niveau Macro d'un H-Lien	48
Figure 8 : Description Micro de la machine multi-processeur pc_parallele avec HADEL	49
Figure 9 : Propriétés au niveau Micro d'une H-Machine	49
Figure 10 : Des gabarits de conception aux langages de gabarits.	65
Figure 11 : Sept niveaux de description d'une architecture logicielle	72
Figure 12 : Taxinomie des architectures client-serveur	75
Figure 13 : Architecture client-serveur pour l'accès aux informations sur le WWW	79
Figure 14 : Les trois composants de base du médiateur	82
Figure 15 : Accès aux systèmes de gestion de bases de données par médiateur	82
Figure 16 : Les catégories de langages décrivant des objets	89
Figure 17 : Architecture d'intégration de type grille pain	100
Figure 18 : Groupes d'entités actives fortement et faiblement couplées	130
Figure 19 : Deux DVSC reliés par un DVSL	130
Figure 20 : Communication sur groupe de canaux et commutation	131
Figure 21 : Communication d'une application data-parallèle au moyen du renommage	132
Figure 22 : Communication entre les instance d'une même classe	132
Figure 23 : Communications dans un groupe de processus	133
Figure 24 : Synchronisation par rendez-vous	133
Figure 25 : Synchronisation avec un moniteur	133
Figure 26 : Les principales entités définies dans VODEL	134
Figure 27 : Exemple d'un réseau local comprenant trois machines et un disque partagé	140
Figure 28 : Représentation d'un réseau via des entités passives issues de la classe machine	140
Figure 29 : Architecture logicielle composée de ressources passives	141
Figure 30 : Les principales entités définies dans VODEL-D	142
Figure 31 : Un exemple de programme réparti	142
Figure 32 : Description des entités passives et physiques de l'application AR	144
Figure 33 : DVSL et DVGL reliant C, D, D1 et D2	148
Figure 34 : Le prototypage par raffinements	157
Figure 35 : La méthode Mars	158
Figure 36 : La méthode MEDEVER adaptée à l'équilibrage d'application	159
Figure 37 : Description des trois étapes de la génération de code	163
Figure 38 : Gestion de compte utilisateur dans une banque	165
Figure 39 : Décomposition en objets de génération du modèle de la Figure 38	165
Figure 40 : Architecture d'un prototype généré par CPN/Tagada	167
Figure 41 : Description des phases du calcul du placement lors de la génération d'une application	169
Figure 42 : Description des étapes de notre méthode d'équilibrage d'application	171
Figure 43 : Les composants élémentaires de H-COSTAM au niveau Macro	173
Figure 44 : Les composants élémentaires de H-COSTAM au niveau Micro	173
Figure 45 : Un exemple de création dynamique de processus en H-COSTAM	174
Figure 46 : Les différentes phases de placement	174
Figure 47 : Architecture d'un prototype généré par H-TAGADA	177
Figure 48 : Architecture répartie du prototype généré par H-TAGADA	179
Figure 49 : Etapes de construction d'une carrosserie automobile	180
Figure 50 : Modélisation de l'atelier de montage avec des réseaux de Petri	181
Figure 51 : Transformation des objets de génération en objets VODEL-D	182
Figure 52 : Regroupement des DVSC avant placement.	183
Figure 53 : Modèle d'assemblage d'une voiture en H-COSTAM	185
Figure 54 : Modèle déduit automatiquement de H-COSTAM	185
Figure 55 : Placement de composants logiciels issus d'un modèle H-COSTAM	186

Figure 56 : Sémantique de communication entre processus H-COSTAM	188
Figure 57 : La méthode MEDEVER adaptée à l'équilibrage d'application	189
Figure 58 : Classification générale des algorithmes d'allocation	198
Figure 59 : Algorithmes d'allocation statique sur une architecture statique optimale	200
Figure 60 : Algorithmes d'allocation statique sur une architecture statique non optimale	202
Figure 61 : Régulation dynamique de la charge	203
Figure 62 : Classification des algorithmes de placement dynamique	205
Figure 63 : Tentative de représentation de la problématique globale du placement dynamique.	207
Figure 64 : L'équilibrage de charge et MEDEVER	211
Figure 65 : IDEFIX = HADEL, VODEL-D et une plate-forme d'équilibrage de charge	212
Figure 66 : Architecture de la gestion système du placement dynamique	213
Figure 67 : Description d'une architecture matérielle avec HADEL	215
Figure 68 : La méthode MEDEVER appliquée à l'équilibrage de charge	217
Figure 69 : La méthode MEDEVER	222
Figure 70 : Architecture de description persistante d'architectures matérielles avec HADEL	223
Figure 71 : L'environnement de gestion d'architectures matérielles basé sur le langage HADEL	223
Figure 72 : Importation et exportation de description d'architectures matérielles avec HADEL	224
Figure 73 : Description d'une application simple avec GEL	227
Figure 74 : Mise en oeuvre de la méthode MEDEVER dans IDEFIX	229
Figure 75 : De VODEL-D à GEL	229
Figure 76 : Plans d'exécution du programme défini à la Figure 75	230
Figure 77 : Trois modes d'intégration de VODEL-D dans Gatostar	231
Figure 78 : Architecture d'IDEFIX	233
Figure 79 : Exemple de trois sessions coopératives	239
Figure 80 : Temps d'exécution du ping-pong en mode système et en mode utilisateur	242
Figure 81 : La méthode MEDEVER et les langages supports de la méthode	245
Figure 82 : De l'architecture logicielle virtuelle vers l'architecture logicielle réelle	248
Figure 83 : Un environnement de prototypage idéal	249
Figure 84 : Une nouvelle vision de l'équilibrage d'application et de la génération de code	250
Figure 85 : Architecture d'IDEFIX	252
Figure 86 : Les différents éléments du modèle des Territoires et leurs relations	253
Figure 87 : Intégration de l'environnement et du placement dynamique dans IDEFIX	254
Figure 88 : L'environnement d'exécution OLAN	275
Figure 89 : Multiplexage VCC/VPC/Liaison	282
Figure 90 : Commutation dans un réseau ATM	282
Figure 91 : Les composants de l'architecture fonctionnelle CORBA	285
Figure 92 : L'architecture de référence de l'OMA	290
Figure 93 : Génération des programmes à partir des interfaces	291
Figure 94 : Technologies de base d'ActiveX	292
Figure 95 : ActiveX : l'architecture OLE de Microsoft adaptée à Internet	292
Figure 96 : L'architecture de la mise en oeuvre des références Java dans Windows	293
Figure 97 : Les trois types d'accès possibles à un serveur DDE	293
Figure 98 : L'architecture fonctionnelle OLE/COM	295
Figure 99 : L'architecture fonctionnelle mise en oeuvre dans X-Open-CAE	301
Figure 100 : L'architecture fonctionnelle mise en oeuvre dans CORDS.	303
Figure 101 : L'architecture fonctionnelle mise en oeuvre dans OSF-DCE	305
Figure 102 : Architecture fonctionnelle de Guide-2	307
Figure 103 : L'architecture fonctionnelle de Esprit-ISA	308
Figure 104 : Description de la plate-forme Java	309
Figure 105 : Architecture Logicielle d'exécution d'un programme Java sur un navigateur	310
Figure 106 : Les squelettes d'implantations de RSA	313
Figure 107 : L'architecture fonctionnelle de UI-ATLAS	319
Figure 108 : Architecture fonctionnelle de Spring	322
Figure 109 : L'architecture fonctionnelle du système Taligent	324
Figure 110 : Visibility rules of declarative items in H-COSTAM.	343