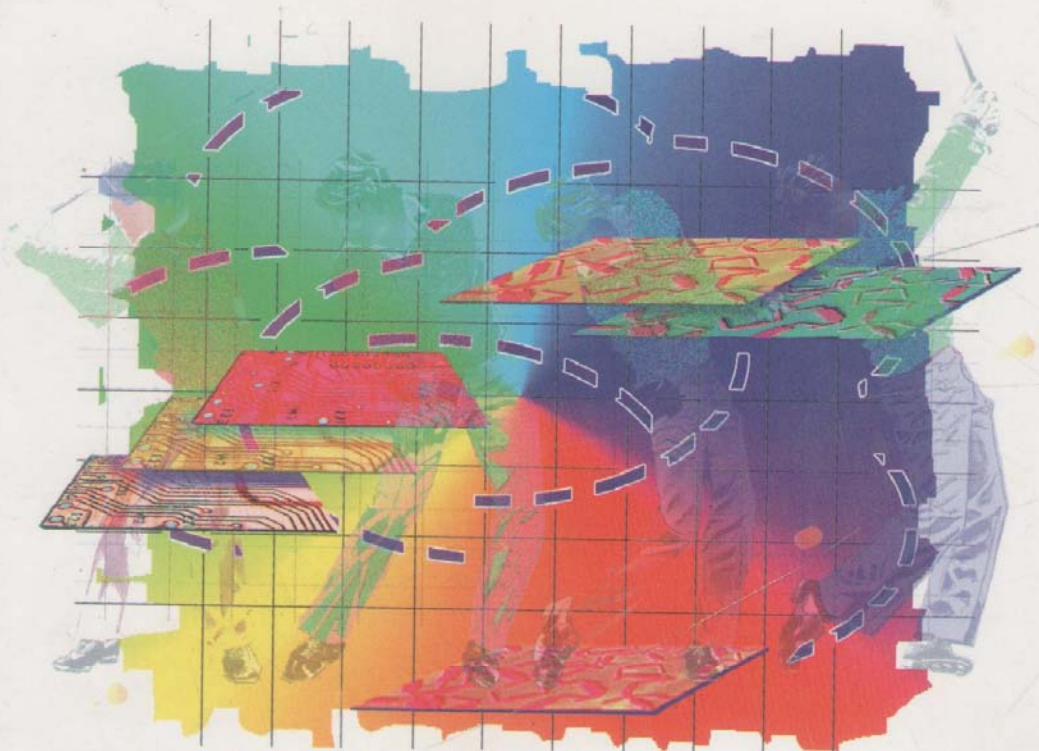


Ammar ATTOUI

Préface de Jean-Louis IMBERT

Les **S**YSTÈMES
MULTI-**A**GENTS
et le **T**EMPS **R**ÉEL



 Eyrolles

SOMMAIRE

Préface	13
Introduction	15
CHAPITRE 1 : PROBLÉMATIQUE DES SYSTÈMES RÉPARTIS	21
1. Définition d'une application distribuée multi-agents	22
2. Évolution des systèmes téléinformatiques	25
2.1. Connexion des terminaux à distance	26
2.2. Utilisation des concentrateurs	27
2.3. Utilisation des frontaux	28
2.4. Les réseaux généraux d'ordinateurs	29
3. Architecture des systèmes téléinformatiques	31
4. Les applications réparties	35
4.1. Les systèmes bancaires	36
4.2. Les systèmes bureautiques	37
4.3. Systèmes informatiques d'une entreprise de production	38
4.5. Conception des systèmes d'information répartis	39
5. Propriétés des systèmes répartis	42
5.1. Les avantages	43
5.2. Les inconvénients	43
5.3. Absence de mémoire commune	44
5.4. Les délais de transmission	46
6. Gestion des activités dans un contexte réparti	47
6.1. Les structures de contrôle de l'exécution répartie	48
6.2. Lancement d'une activité à distance	52
6.3. Gestion du dialogue entre les entités coopérantes	55
6.4. Terminaison et validation globale des applications réparties	57
7. Synchronisation	58
7.1. Mise en œuvre d'un ordre total	59
7.2. Allocation de ressources dans un contexte réparti	62
8. Protocoles de validation des applications globales	71
9. Tolérances aux pannes	77

CHAPITRE 2 : MODÈLES D'INTERACTION	83
1. Architecture distribuée	84
2. La programmation concurrente et distribuée orientée événements	85
3. Les systèmes massivement parallèles et la programmation distribuée	86
3.1. La programmation distribuée et les réseaux hétérogènes d'ordinateurs	88
3.2. Les réseaux à haut débit et la programmation distribuée	89
4. Principes de base de la programmation distribuée	91
4.1. Les processus filtres	91
4.2. Les processus clients	92
4.3. Les processus acteurs	92
5. Méthodologie d'analyse d'une application multi-agents	94
5.1. Décomposition orientée données d'une application distribuée	94
5.2. Décomposition fonctionnelle des applications distribuées	95
6. Les canaux de communication	96
7. Éléments d'un langage de programmation distribuée de haut niveau	97
7.1. Les canaux asynchrones	98
7.2. Primitives de manipulation de canaux	100
8. Les communications anonymes	104
8.1. Éléments de mise en œuvre du modèle de communications anonymes	105
9. Les moniteurs de Hoare	107
9.1. Réalisation des moniteurs avec les sémaphores	109
9.2. Solution du problème des Lecteurs-Rédacteurs avec les sémaphores	110
9.3. Solution du problème des Lecteurs-Rédacteurs avec les moniteurs	111
9.4. Exemple de réalisation d'un canal de communication synchrone	113
9.5. Les processus serveurs et les moniteurs	113
Exercice	118
CHAPITRE 3 : PVM, OUTILS DE PROGRAMMATION PARALLÈLE	121
1. Présentation de PVM	121
2. Installation de PVM	124
2.1. Comment obtenir PVM	125
2.2. Création d'une machine virtuelle	125
3. L'interface utilisateur ou les primitives PVM	127
3.1. Contrôle de l'exécution répartie sous PVM	129
3.2. Configuration dynamique des machines virtuelles	131
3.3. Mise en œuvre des interactions entre les tâches	131
3.4. Le regroupement dynamique de tâches	138
CHAPITRE 4 : LA PROGRAMMATION DISTRIBUÉE SOUS UNIX	145
1. Unix et le modèle client-serveur	145
2. Les outils Unix pour l'informatique répartie	147

2.1. Le protocole TCP/IP	147
2.2. Désignation et adressage des machines sur Internet.....	148
3. Les interfaces IP	150
3.1. Le routage des datagrammes.....	152
4. Les protocoles de transport de données TCP et UDP	153
4.1. Le protocole UDP (User Datagramme Protocol).....	154
4.2. Le protocole TCP (Transport Control Protocol).....	154
5. Les outils Unix pour la programmation distribuée	155
5.1. Création et manipulation de processus à partir d'un programme	156
5.2. Les opérations d'entrée/sortie asynchrones	161
5.3. Les interruptions logicielles	162
5.4. Les tubes, un moyen de communication intramachine	165
6. Les sockets	172
6.1. Les applications client-serveur en mode « connecté domaine AF_UNIX »	174
6.2. Les applications client-serveur en mode « connecté domaine AF_INET »	182
6.3. Échange de messages structurés.....	196
CHAPITRE 5 : PRINCIPES DE L'INFORMATIQUE TEMPS RÉEL	205
1. Structure et fonctionnement d'un système automatisé	206
2. Les systèmes automatisés combinatoires	211
3. Les systèmes automatisés séquentiels.....	212
4. Le Grafcet : outil de synthèse des automatismes industriels.....	212
5. Choix technologiques pour la mise en œuvre de la partie commande.....	214
6. Les systèmes numériques de commande de procédés	216
7. Nécessité d'une méthodologie de spécification bien adaptée	216
7.1. Introduction à la méthode SA/RT	217
7.2. Exemple de spécification d'un système avec SA/RT	219
8. Dynamique des procédés.....	221
8.1. Les procédés à dynamique lente	222
8.2. Les procédés à dynamique rapide	224
9. Les tâches périodiques et les tâches apériodiques	226
9.1. Les tâches périodiques	226
9.2. Les tâches apériodiques	228
10. Généralités sur les noyaux temps-réel.....	230
10.1. Fonctionnalités d'un noyau.....	231
10.2. Structure interne d'un noyau temps réel.....	233
10.3. L'ordonnancement des tâches	233
10.4. Les moyens de communication	235
10.5. La réentrée dans les exécutifs temps réel	235
10.6. Les exceptions, les interruptions et les timers	237
11. Les réseaux locaux industriels.....	240

11.1. Le réseau de terrain FIP	243
11.2. Les principaux services de la couche application du modèle FIP	245
12. L'environnement de développement d'applications temps réel VxWorks	248
12.1. Gestion des tâches	250
12.2. Exemple de création de tâches	251
12.3. Les communications	252
12.4. Gestion des interruptions	253
12.5. Gestion des timers	253
12.6. Exemple d'exclusion mutuelle et de synchronisation entre tâches	254
13. L'environnement d'une applications temps réel multi-agents	257
13.1. Pilote de la carte d'E/S analogique / numérique	257
13.2. Pilote en mode direct	259
13.3. Pilote en mode asynchrone	262
14. Outils de développement et de covérification des systèmes embarqués	265
CHAPITRE 6 : PROLOG III ET L'INGÉNIÉRIE DES SYSTÈMES TEMPS RÉEL	269
1. Impact des spécifications formelles	271
1.1. Généralités sur les méthodes formelles	272
1.2. Les techniques de spécification formelle	273
1.3. Vérification, implémentation et réutilisation	279
2. Rappel sur les systèmes à événements discrets	282
2.1. Les systèmes discrets	282
2.2. Les systèmes à événements discrets	283
2.3. Les automates Max-Plus	284
3. Prolog III et l'étude des systèmes temps réel simples	287
3.1. Modélisation et simulation du comportement des systèmes combinatoires	289
3.2. Modélisation et simulation du comportement des systèmes discrets	295
4. Méthodologie de spécification des systèmes complexes MSS	300
4.1. Analyse et spécification des systèmes complexes avec (MSS)	301
4.2. Exemple d'analyse d'un système de production	303
4.3. L'environnement support de MSS	307
4.4. Simulation du comportement des systèmes complexes	320
5. Conclusion	331
6. Projet 1 : Modélisation des systèmes répartis	332
7. Bibliographie	341
CHAPITRE 7 : TRAITEMENTS PARALLÈLES D'IMAGES	345
1. Présentation	345
1.1. Fonctions de l'agent maître	345
1.2. Fonctions des agents esclaves	346
2. Implémentation	348
2.1. Le code de l'agent maître	348
2.2. Code des agents « newton.fjd »	354

2.3. Modules de manipulation de complexes	356
2.4. Mise en œuvre de la fonction de calcul de la couleur d'un pixel	357

CHAPITRE 8 : MISE EN ŒUVRE D'UNE APPLICATION DISTRIBUÉE SELON LE MODÈLE CLIENT-SERVEUR

1. Présentation	359
1.1. Architecture fonctionnelle du système à mettre en œuvre	361
1.2. Protocole de coopération client-serveur	362
2. Implémentation	362
2.1. Ordonnancement des contraintes temporelles	362
2.2. Structure de données utilisée	363
2.3. Un scénario d'échanges entre un client et le serveur	364
2.4. Primitives implémentées	365
2.5. Mise en œuvre de l'interaction clients-serveur	366
2.6. Le fonctionnement du client	366
2.7. Le fonctionnement du serveur	367
3. Mini projet 1 : « Un système de gestion d'événements »	368
4. Mini-projet 2 : « Une version Vx-Works du gestionnaire de contraintes temporelles »	370
5. Les sources du système de gestion de contraintes temporelles	370
5.1. Le fichier include	370
5.2. Fichier des fonctions internes de service	372
5.3. Code du serveur	377
5.4. Jeux d'essai	383

CHAPITRE 9 : PILOTAGE D'UNE CHAÎNE DE CONDITIONNEMENT DE PRODUITS DANS UN ATELIER

1. Fonctionnement	385
1.1. Descriptif de la partie opérative	386
1.2. Cahier des charges	386
2. Analyse et spécification avec la méthode SA/RT(SA/SD)	387
2.1. La démarche d'analyse	388
2.2. Étape « construction d'un DFD initial »	392
2.3. Étape « élimination des agents superflus et regroupement des agents coopérants »	394
2.4. Étape de mise à jour du dictionnaire	395
2.5. Étape de spécification du comportement des agents élémentaires	396
2.6. Étape de décomposition hiérarchique des agents du DFD0	399
3. Conception et mise en œuvre de la partie commande	407
3.1. Choix d'une architecture de la partie commande	407
3.2. Mise en œuvre des agents	414

CHAPITRE 10 : UN SYSTÈME DE GESTION D'OBJETS PERSISTANTS

1. Présentation	423
------------------------------	------------

2. Le gestionnaire d'objets GO	424
2.1. Une application multi-agents avec des objets persistants GO	425
2.2. Architecture client-serveur	432
2.3. Déroulement d'une session d'utilisation de GO	434
2.4. Les outils MO2	437

ANNEXE 1 : PROBLÈMES DE COUPLAGE COUCHE MATÉRIELLE - COUCHE LOGICIELLE DANS UN SYSTÈME INFORMATIQUE

455

1. La machine cible	455
1.1. Organisation interne de la famille X86	457
1.2. Appel de procédure et changement de contexte	461
1.3. Coordination entre le processeur et les contrôleurs de périphériques	462
1.4. Les interruptions	466

ANNEXE 2 : PRINCIPES DE BASE DES PILOTES DE PÉRIPHÉRIQUES

471

1. Présentation	471
2. Remarques générales de programmation	474
3. Communications entre le noyau et le pilote	475
3. Procédure d'installation des pilotes sous Linux	477

ANNEXE 3 : GESTION DES PROCESSUS

481

1. Introduction	481
2. Principe de mise en œuvre de l'ordonnanceur	482
3. L'ordonnanceur et la synchronisation interne des pilotes	484

ANNEXE 4 : GESTION DE LA MÉMOIRE CENTRALE

487

1. La partie supportée par le matériel	487
2. Mécanisme de pagination	488
3. Mécanisme de segmentation	489
4. Caractéristiques des segments systèmes	490
5. Les selecteurs dans le 386	491
6. Les descripteurs de segment	492
7. La partie supportée par le noyau	493

ANNEXE 5 : INTRODUCTION À PROLOG III

495

1. Fonctionnement d'un programme logique	496
2. Prolog III	498
3. Les expressions acceptées	500
4. Conclusion	502
5. Références	502

Index	503
--------------------	------------