

THESE

présentée par

Rodrigo SCIOVILLE GARCIA

pour obtenir le titre de **DOCTEUR**

de L'UNIVERSITE JOSEPH FOURIER - GRENOBLE 1

(arrêté ministériel du 5 juillet 1984)

Spécialité : **INFORMATIQUE**

**GESTION DES INFORMATIONS PERSISTANTES DANS
UN SYSTEME REPARTI A OBJETS**

Thèse soutenue le 23 juin 1989

Composition du Jury :

Président : M. Jacques MOSSIERE

Rapporteurs : M. Michel ADIBA
M. Marc SHAPIRO

Examineurs : M. Roland BALTER
M. Sacha KRAKOWIAK

Thèse préparée au sein du Laboratoire de Génie Informatique
à l'Université Joseph Fourier - Grenoble 1

Je tiens à remercier :

Monsieur Jacques Mossière, Directeur de l'Ecole Nationale Supérieure d'Informatique et Mathématiques Appliquées de Grenoble, de m'avoir fait l'honneur de présider le jury de cette thèse.

Monsieur Michel Adiba, Professeur à l'Université Joseph Fourier, pour les conseils qu'il m'a donnés lors de la rédaction de ce document et pour avoir accepté de participer au jury.

Monsieur Marc Shapiro, Directeur de Recherche à l'Institut National de Recherche en Informatique et Automatique, d'avoir bien voulu rapporter ce travail et de faire partie du jury.

Monsieur Roland Balter, Directeur du Centre de Recherche Bull à Grenoble, pour l'intérêt qu'il a porté à la réalisation de ce travail et pour sa participation au jury. Je tiens à lui témoigner toute ma reconnaissance pour le soutien permanent qu'il m'a manifesté.

Monsieur Sacha Krakowiak, Professeur à l'Université Joseph Fourier, pour m'avoir accepté dans son équipe et constamment soutenu dans mes recherches.

Monsieur Gérard Vandôme, Ingénieur au Centre de Recherche Bull, pour les conseils qu'il m'a prodigués lors de la réalisation de ce travail. Sans les discussions que nous avons eues, cette thèse n'aurait sans doute jamais vu le jour.

Messieurs André Freyssinet et François Exertier, Thésards au Centre de Recherche Bull, pour l'aide inappréciable qu'ils m'ont apportée durant l'élaboration de cette thèse.

Monsieur Philippe Chol pour la patience de relire et de corriger la dernière version de ce document.

Mes collègues des projets Guide et Comandos, du Centre de Recherche Bull et du Laboratoire de Génie Informatique, pour leur aide et leur sympathie. Je voudrais remercier particulièrement Messieurs Christian Lenne et Jacques Bernadat du Centre de Recherche Bull et Monsieur Bernard Cassagne du Laboratoire de Génie Informatique.

Le Centre Régional des Oeuvres Universitaires de Grenoble, l'Ambassade de France en Colombie et le Departamento de Sistemas y Computación de la Universidad de los Andes de Bogotá, qui ont rendu possible ma venue et mon séjour en France.

Je tiens à exprimer toute ma reconnaissance et ma gratitude à :

Monsieur Eric Paire, Ingénieur au Centre de Recherche Bull, non seulement pour son support technique mais surtout pour la compagnie et le soutien constants qu'il m'a donnés lorsque nous partagions le même bureau. Sa présence m'a permis de supporter mieux les longues soirées d'hiver consacrées à la rédaction de cette thèse.

Madame Marie Meysembourg-Männlein, Thésard au Laboratoire de Génie Informatique, pour avoir lu et commenté tous mes écrits, pour les agréables discussions que nous avons eues, et surtout pour ses encouragements, ses conseils et sa grande sympathie.

Mes copains Latinoaméricains et Européens, pour la patience, la solidarité et l'amitié qu'ils m'ont toujours témoignées.

Résumé

Ce travail aborde quelques-uns des problèmes de la gestion d'objets dans un système informatique réparti.

Guide est un système d'exploitation expérimental réparti sur un réseau local. Les informations qu'il manipule sont structurées selon un modèle à objets. Le concepteur d'applications voit le noyau de ce système comme une machine virtuelle multi-sites et multi-processeurs dans laquelle le parallélisme est apparent et la distribution est cachée. Cette machine virtuelle réalise la gestion d'objets.

Ce travail décrit l'un des composants de la machine virtuelle : la mémoire d'objets. Il présente les choix de conception, sa réalisation et une analyse de l'expérience acquise. La mémoire d'objets est persistante. En effet, les objets ne sont pas détruits tant qu'ils restent liés à un objet particulier, persistant par définition. La mémoire d'objets est répartie et comporte deux niveaux : la Mémoire Virtuelle d'Objets, qui constitue le support d'exécution des objets, et la Mémoire Permanente d'Objets, qui se charge de leur conservation.

Certaines fonctions de gestion d'objets n'ont pas été intégrées dans la machine virtuelle mais elles sont mises en œuvre comme des applications. On appelle ces applications les services du système. La conception et la réalisation des services de désignation symbolique et de gestion de versions sont décrites dans cette thèse.

Table des Matières

Introduction	1
1. Description du cadre de travail	1
Persistence 'classique'	1
Les systèmes à objets	2
Programmation persistante	4
Le modèle de la mémoire persistante	5
La répartition	8
2. Le système Guide	9
3. Le travail réalisé	10
4. Plan de la thèse	13
Première partie : Des systèmes de gestion de fichiers aux systèmes de gestion d'objets	15
Avant-propos	17
1. Gestion classique de la persistance	21
1.1 Systèmes de gestion de fichiers	21
1.1.1 NFS	21
1.1.1.1 L'espace de désignation des fichiers	22
1.1.1.2 Le client	24
1.1.1.3 Le serveur	24
1.1.1.4 Mise en oeuvre	25
1.1.2 Le système Andrew	25

1.1.2.1	L'espace de désignation de fichiers	26
1.1.2.2	La gestion des fichiers partagés	26
1.1.2.3	L'accès aux fichiers	27
1.1.2.4	Mise en oeuvre	28
1.2	Oracle : Un système de gestion de bases de données	29
1.2.1	Organisations physique et logique des données	29
1.2.2	Contrôle de concurrence	30
1.2.3	La vue de l'utilisateur	30
1.3	Etude comparative et conclusions	31
2.	Tendances actuelles	35
2.1	Le serveur d'objets ObServer	35
2.1.1	Mise en oeuvre de la mémoire du serveur	36
2.1.2	Partage des objets	36
2.1.2.1	Contrôle de concurrence	36
2.1.2.2	Transactions	37
2.1.2.3	L'accès aux objets	38
2.1.3	Les clients	40
2.1.4	Mise en oeuvre	40
2.2	Mémoires persistantes	40
2.2.1	Une mémoire persistante pour Smalltalk-80	41
2.2.1.1	LOOM : Une réalisation centralisée de la mémoire d'objets	41
Le ramassage de miettes	42	

2.2.1.2	La réalisation répartie de Decouchant	44
2.2.1.2.1	Structure d'un gestionnaire local d'objets	44
2.2.1.2.2	Les deux niveaux de mémoire	45
2.2.1.2.3	La mise en oeuvre de la répartition	46
2.2.1.2.4	La récupération de l'espace des miettes	46
2.2.1.3	Mise en oeuvre de la mémoire persistante de Smalltalk- 80	48
2.2.2	La mémoire persistante de Thatte	48
2.2.2.1	Le support à la mémoire persistante	49
2.2.2.1.1	Résistance aux pannes	49
2.2.2.1.2	Récupération d'espace	51
2.2.2.2	La gestion d'objets	51
2.2.2.3	Les applications de la mémoire persistante	52
2.3	Etude comparative et conclusions	53
Deuxième partie : La gestion d'objets persistants dans le système Guide		57
3.	Introduction au système Guide	59
3.1	Le modèle de données	60
3.1.1	Types et classes	60
3.1.2	Persistance des objets	64
3.1.3	Désignation des objets	64
3.1.4	Composition d'objets	65
3.2	Le modèle d'exécution	66

3.2.1	Aspects dynamiques du modèle	68
3.2.2	Vue des objets selon le modèle d'exécution	69
3.3	Le modèle de la mémoire	69
3.4	La machine virtuelle de Guide	70
3.4.1	La machine à objets	72
3.4.2	Le gestionnaire des domaines et des activités	72
3.5	Les services du système Guide	73
4.	Le modèle de la mémoire	77
4.1	Description de la mémoire d'objets	77
4.1.1	Description des objets en MVO	78
4.1.1.1	Images d'objet	79
4.1.1.2	Création et destruction d'images	80
4.1.2	Description des objets en MPO	81
4.1.2.1	Systèmes d'objets	82
4.1.2.2	Objets stockés	83
4.1.2.3	Création d'objets stockés	84
4.1.3	Cycle de vie des objets	84
4.1.4	Hierarchie de la mémoire d'objets	90
4.2	Désignation d'objets	93
	Références-système	94
	Références d'exécution	95
4.3	Localisation d'objets	96
	Le support de la localisation d'objets	96

4.3.1	Localisation d'images	97
4.3.2	Localisation d'objets stockés	98
4.4	Récupération d'espace	99
4.4.1	Ramasse-miettes sur l'espace global d'objets	99
4.4.2	Ramasse-miettes en MPO seulement	100
4.5	Duplication d'objets	101
5.	Désignation symbolique et gestion de versions	103
5.1	Le service de désignation symbolique	103
5.1.1	L'espace de noms	104
5.1.2	Les objets répertoires	106
5.1.3	La désignation symbolique	108
5.2	Le service de versions	110
5.2.1	Création de versions	111
5.2.2	Désignation de versions	112
5.2.3	Le gestionnaire de versions	116
5.3	Caractéristiques communes aux services	118
6.	Réalisation du modèle de la mémoire	121
6.1	Description d'un site Guide	121
6.1.1	Le système support	123
6.1.2	Support à la réalisation d'images	124
6.1.2.1	Version mono-segment de la gestion de la mémoire partagée	125
6.1.2.2	Version multi-segments de la gestion de la mémoire partagée	126

6.1.3	Support à la réalisation d'objets stockés	127
6.1.3.1	Gestion du cache	127
6.1.3.2	Gestion des blocs d'un système d'objets	128
6.2	Principes de la réalisation répartie du modèle d'exécution	129
	Répartition des domaines et des activités	129
	Réalisation des processeurs virtuels de la machine à objets	130
	Domaines locaux	131
	Le descriptif d'un domaine	132
	Le démon Guide	132
	Exécution d'une méthode d'objet par une activité	133
6.3	Réalisation du modèle de la mémoire	134
6.3.1	Principes de la réalisation de la gestion d'images	135
6.3.2	Principes de la réalisation de la gestion d'objets stockés	136
6.3.3	Verrouillage, déverrouillage et validation	139
	Verrouillage et déverrouillage d'un objet	139
	Validation d'une image d'objet	140
	Les transitions d'état pour un objet	140
6.3.4	Gestion d'images relative à un site	141
6.3.4.1	Description de la TMVO	143
6.3.4.2	Création, modification et destruction d'images	143
6.3.4.3	Primitives	144
6.3.5	Gestion d'objets stockés relative à un site	145

6.3.5.1	Descripteur en MPO d'un objet	147
6.3.5.2	Création et destruction d'objets stockés	147
6.3.5.3	Primitives	148
6.3.6	Réalisation des interfaces de la machine à objets	149
	L'interface que la MPO offre à la MVO	149
	L'interface de la machine à objets	151
6.3.7	Les processus qui exécutent la gestion de la mémoire d'objets	151
	Le démon	152
	Les activités	153
	Les processus de sauvegarde et serveurs	153
6.4	L'interface de la machine virtuelle de Guide	154
	La primitive guideCall	154
	La primitive guideCreate	155
7.	Réalisations des services	157
7.1	Description d'un objet répertoire	157
7.2	Description d'un gestionnaire de versions	159
Conclusions	163
1.	Les acquis du travail	163
	Problèmes de la réalisation actuelle	165
	Fonctions non encore mises en oeuvre	166
2.	Perspectives	167
	Hétérogénéité	168

Images de taille variable	168
Aspects 'bases de données' dans Guide	168
Utilisation de la MPO par un système différent de Guide	170
Références	173
Annexes	183
Annexe A : Algorithme de verrouillage et validation	185
Annexe B : Réalisation du service de désignation symbolique	187
Annexe C : Réalisation du service de gestion de versions des objets	191