

---

# Projet RODIN

## Systèmes de Bases de Données

---

**Localisation :** *Rocquencourt*

**Mots-clés :** base de données, gestion de transaction, modèle de base de données, base de données actives, optimisation, gestion d'objets, distribution, hétérogénéité, parallélisme.

### 1 Composition de l'équipe

**Responsable du projet**

Patrick Valduriez, directeur de recherche Inria

**Responsable permanent**

Eric Simon, directeur de recherche Inria

**Secrétaire**

Elisabeth Baqué, assistant-ingénieur

**Personnel INRIA**

Marcin Skubiszewski, chargé de recherche

**Ingénieurs experts**

Claude Darrieumerlou, jusqu'au 31 mai  
Anthony Tomasic

**Conseillers scientifiques**

Mokrane Bouzeghoub, professeur, université de Versailles  
Georges Gardarin, professeur, université de Versailles  
Mikal Ziane, maître de conférences, université Paris 5

**Professeurs invités**

Michael Franklin, université du Maryland, USA, 1 mois  
Louïqa Raschid, université du Maryland, USA, 1 mois

### **Chercheurs doctorants**

Luc Bouganim, boursier Inria, université de Versailles  
Eric Dujardin, boursier Inria, université Paris 6, jusqu'au 31 mars  
Daniela Florescu, boursière Inria, université de Versailles  
Jean-Robert Gruser, boursier MENESR, université de Versailles  
Olga Kapitskaia, boursière Inria, université Paris 6  
François Llibat, boursier MENESR, université de Versailles  
Maja Matulovic, boursière MENESR, université de Versailles  
Hubert Naacke, boursier MENESR, université de Versailles  
Esther Pacitti, boursière du Brésil  
Dimitri Tombroff, boursier Inria, université Paris 6, jusqu'au 30 juin  
Florian Xhumari, boursier Inria, université Paris 6

### **Collaborateurs extérieurs**

Françoise Fabret, chargée de travaux, CNAM

### **Stagiaires**

Mokrane Amzal, DEA, université Paris 6  
Seweryn Golinski, Ecole Polytechnique Franco-Polonaise de Poznan  
Marie-José Montrieux, DESS, université Paris 11  
Didier Postel, DESS, université Paris 11  
Robert Walkowski, Ecole Polytechnique Franco-Polonaise de Poznan

## **2 Présentation du projet**

L'objectif général du projet Rodin est de concevoir et d'expérimenter des techniques (nouvelles) afin d'améliorer les fonctionnalités et les performances de la gestion de bases de données. Ces techniques prennent la forme de langages ou d'algorithmes qui peuvent être implémentés dans des outils ou des composants de bases de données.

Aujourd'hui, le projet a accumulé un savoir-faire technologique suffisant pour pouvoir expérimenter avec des applications pilotes. Cette activité nous permet de mieux comprendre les besoins d'applications complexes et d'identifier les problèmes de recherche prioritaires qui nous concernent.

En 1996, un effort important a porté sur l'étude des problèmes de bases de données qui se posent dans deux grands domaines d'application: le commerce électronique (en particulier, la bourse d'affaires électronique) et les systèmes d'information pour l'environnement. Ce travail a permis de recentrer le projet autour de trois actions de recherche :

- Aide à la conception d'applications de base de données, dans lesquelles les contrôles d'intégrité et les règles de gestion ont été intégrés.
- Découverte et recherche d'informations pertinentes dans des sources de données hétérogènes et autonomes dans un réseau global.
- Gestion d'objets persistants.

Deux autres actions de recherche se sont terminées naturellement, avec les fins de thèses.

- Extension de la flexibilité et de l'expressivité des modèles de données objet.
- Optimisation de requêtes.

La valorisation industrielle est importante pour le projet Rodin. D'abord, elle permet de confronter des solutions développées à l'Inria dans un contexte d'utilisation réel. D'autre part, la collaboration avec des partenaires industriels ou des grands utilisateurs est à la base de la définition de nouveaux problèmes de recherche pour l'Inria. Nos partenaires industriels principaux sont Bull et O2 Technology, avec qui nous menons des opérations de transfert de technologie.

Notre activité est complémentaire de celle du projet Verso avec qui nous entretenons une collaboration étroite, notamment sur les problèmes d'accès à des bases de données hétérogènes. Du point de vue systèmes, nous coopérons aussi avec le projet Sor.

En 1996, les faits suivants ont marqué l'activité du projet Rodin : la démonstration dans des conférences internationales du prototype d'optimiseur Flora, la démonstration du prototype Acto (bases de données actives), le développement de la version 1 du prototype Disco (outil de recherche d'informations dans des sources de données hétérogènes), le transfert de technologie dans le produit O2, et enfin les premiers succès du GIE Dyade (Bull-Inria) dirigé par P. Valduriez.

## 3 Actions de Recherche

### 3.1 Extension des modèles de données objet

Cette action avait pour objectif d'étendre l'expressivité et la flexibilité des modèles de données supportés par les SGBD existants [233]. Cette action s'est terminée cette année en se concentrant sur le support efficace des multi-méthodes, c'est à dire des méthodes dont la sélection du code associé à une invocation s'effectue en fonction du type de tous les arguments de l'invocation.

#### 3.1.1 Sélection efficace de multi-méthodes

*Participants* : Eric Dujardin, Eric Simon

Un des problèmes majeurs qui freine l'introduction des multi-méthodes dans les SGBD à objets est le manque d'algorithmes efficaces pour effectuer la sélection du code d'une méthode en fonction du type de tous les arguments de l'invocation. L'approche avec des tables de sélection, utilisée par exemple en C++ pour la sélection de mono-méthodes, est la seule mise en œuvre efficace qui garantisse un coût en temps constant (c.à.d, qui n'utilise pas de mécanisme de cache). Dans le cas des multi-méthodes, les tables de sélection occupent une place mémoire trop importante pour être utilisables. Nous avons proposé et mis en œuvre un algorithme de compression des tables de sélection très efficace [257] : son coût est au pire de l'ordre de  $k^2 \times K \times n$ , où  $K$  est le nombre de types dans le schéma,  $n$  est le nombre d'arguments et  $k$  est le nombre maximum de super-types directs qu'un type a dans la hiérarchie de types. En général,  $K$  est de l'ordre de plusieurs centaines alors que  $k$  est de l'ordre de 2 ou 3. Les mesures effectuées sur une bibliothèque existante comportant plusieurs milliers de méthodes, dont environ 10% de multi-méthodes, ont révélé que le taux de compression moyen est d'environ 500 pour les multi-méthodes à 2 arguments, de 200 000 pour les multi-méthodes à 3 arguments et la taille moyenne d'une table de sélection est d'environ 2 Koctets [257]. Enfin, nous avons montré que l'utilisation d'arbres de sélection, au lieu de tables, permettaient d'améliorer le taux de compression lorsque les méthodes avaient plus de trois arguments [261].

#### 3.1.2 Désambiguïsation explicite des multi-méthodes

*Participants* : Eric Dujardin

L'héritage multiple et la sélection multiple représentent deux sources d'ambiguïté dans les langages et modèles à objets. Les ambiguïtés peuvent être résolues automatiquement par le système, par exemple en utilisant un ordre total entre les supertypes et un ordre des arguments dans les multi-méthodes. Ces désambiguïsations implicites ont l'inconvénient d'être difficiles à comprendre par le programmeur et

de masquer des erreurs potentielles de conception. A l'inverse, la résolution d'ambiguïtés peut être laissée à la charge du programmeur, par exemple en ajoutant de nouvelles méthodes pour les invocations ambiguës. Il est cependant difficile d'isoler les invocations ambiguës lorsqu'on utilise des multi-méthodes. Nous avons montré qu'il existe toujours un ensemble unique minimal de redéfinitions de multi-méthodes permettant de désambiguïser explicitement un schéma composé de multi-méthodes. Nous avons proposé et mis en œuvre [244] un algorithme pour calculer cet ensemble minimal ainsi que des explications associées : l'ensemble des méthodes causant l'ambiguïté est associé à chaque méthode à redéfinir. Pendant la construction de la table de sélection des multi-méthodes, le coût ajouté par l'algorithme de désambiguïsation est de l'ordre de  $M' \times (M + M')^2$ ,  $M$  étant le nombre de multi-méthodes définies par l'utilisateur et  $M'$  le nombre de multi-méthodes nouvelles.

## 3.2 Optimisation de requêtes

Cette action sur l'optimisation de requêtes objet et l'optimisation de requêtes pour exécution parallèle s'est terminée cette année, avec la soutenance de trois thèses et la fin du projet Esprit IDEA.

### 3.2.1 Optimisation de requêtes objet

*Participants* : Daniela Florescu, Georges Gardarin, Jean-Robert Gruser, Hubert Naacke, Patrick Valduriez, Mikal Ziane

Le langage Flora [235] offre un modèle à objets général et un langage algébrique qui sert de support pour l'optimisation de requêtes objet. Nous avons conçu l'optimiseur Flora de manière modulaire [230] afin d'avoir un environnement très flexible pour l'étude de l'optimisation de requêtes objet. L'espace de recherche est défini par des règles d'équivalence qui permettent d'exprimer uniformément les transformations logiques et physiques des requêtes.

Nous avons développé un modèle de coût générique pour les SGBDO [231] qui prend en compte les méthodes définies par l'utilisateur. Il permet de donner le coût d'exécution des expressions de chemins en tenant compte du regroupement des objets sur disque et de la taille mémoire disponible [252]. Nous avons validé ce modèle avec le SGBD O2 et le banc d'essai OO7. Nous avons aussi adapté ce modèle de coût aux systèmes distribués et parallèles.

Nous avons étendu le prototype d'optimiseur Flora pour supporter le langage OQL, incorporer le nouveau modèle de coût, et améliorer l'interface graphique. L'optimiseur est réalisé avec l'outil Ilog View pour l'interface graphique et le système O2 pour l'exécution des programmes Flora ou OQL optimisés. Ce prototype a été démontré à la revue du projet IDEA en juin et a fait l'objet de démonstrations dans des conférences internationales.

### 3.2.2 Optimisation pour exécution parallèle

*Participants* : Luc Bouganim, Daniela Florescu, Patrick Valduriez

L'exécution de requêtes peut bénéficier naturellement du parallélisme afin de réduire les temps de réponse [237]. En 1996, nous avons étendu les concepts utilisés dans DBS3, le SGBD parallèle que nous avons réalisé avec Bull sur KSR1, afin de prendre en compte une architecture hiérarchique, composée d'un ensemble de machines multiprocesseurs à mémoire partagée, reliés par un réseau rapide. Dans ce contexte, la répartition de charge doit être effectuée à deux niveaux : localement sur chaque noeud à mémoire partagée, puis globalement entre les noeuds. Nous avons proposé, implémenté et évalué un modèle d'exécution réalisant cette répartition de charge à deux niveaux, et cherchant à maximiser le niveau local (moins coûteux) afin de réduire le besoin de redistribution globale de la charge. Les résultats de nos expériences sur un prototype de ce modèle, avec notre KSR1 (72 processeurs) ont démontré d'excellentes performances, y compris avec des données très biaisées [248], [249].

Nous avons aussi reconsidéré l'exécution de requêtes complexes sur des architectures à mémoire non uniforme et proposé un modèle d'exécution adapté à ce genre d'architecture [259], [229].

### 3.3 Conception d'applications

Cette action a pour objectif de développer des techniques nouvelles d'aide à la conception et à la génération d'applications efficaces de bases de données dans lesquelles les contrôles d'intégrité et les règles de gestion ont été intégrés.

#### 3.3.1 Mise en œuvre des règles de gestion

*Participants* : Mokrane Bouzeghoub, Françoise Fabret, François Llibat, Maja Matulovic, Eric Simon

La mise en œuvre des règles de gestion (“business rules” en anglais) pose des problèmes de conception et de performances [238]. On peut les implanter de façon ad-hoc et efficace à l'intérieur des programmes d'application (par exemple, en utilisant des procédures stockées). Mais cette solution est peu fiable et la maintenance de l'application devient difficile. L'autre possibilité est d'implanter les règles sous forme de déclencheurs (“triggers” en anglais) et d'assertions gérés par un SGBD *actif*.

Un déclencheur possède trois composantes qui correspondent à : la détection d'un événement  $E$  au cours de l'exécution d'une application, l'évaluation d'une condition  $C$  sur la base de données et l'exécution d'un programme de bases de données  $A$ . Intuitivement, un déclencheur a l'interprétation suivante : si l'événement  $E$  survient et la condition  $C$  est satisfaite, alors le programme  $A$  peut être exécuté. En fait, l'exécution des déclencheurs obéit à un modèle d'exécution qui précise les modes d'invocation, d'exécution et de synchronisation des composantes  $E, C, A$ . Or, chaque SGBD actif met en œuvre un modèle d'exécution spécifique différent. Ceci pose plusieurs problèmes : (1) les méthodes d'optimisation des déclencheurs, qui dépendent du modèle d'exécution, sont spécifiques à un SGBD, (2) la spécification des règles de gestion d'une application est difficile si le modèle d'exécution du SGBD est inadapté aux besoins de l'application, et (3) les spécifications de déclencheurs ne sont pas portables d'un SGBD actif à l'autre.

Pour répondre à ces problèmes, nous avons développé un modèle d'exécution paramétré qui capture les possibilités d'exécution de la plupart des SGBD actifs existants. Ce modèle est mis en œuvre dans le système ACTO avec un double but [253]. Premièrement, il s'agit de générer un programme d'application, appelé moteur d'exécution, à partir de la spécification des composantes  $E, C, A$  des déclencheurs et de la sémantique de chaque déclencheur, ou module de déclencheurs, définie à l'aide du modèle paramétré. Ce moteur prend en charge la gestion du contexte d'exécution des déclencheurs (par ex., la gestion de l'historique des événements) et contrôle l'exécution des déclencheurs. Le moteur peut s'exécuter sur tout SGBD capable d'assurer la détection d'événements simples, c.à.d, tout SGBD ayant des capacités actives minimales. Deuxièmement, il s'agit de développer des techniques génériques d'analyse du comportement des déclencheurs dans le modèle d'exécution paramétré. Ces techniques sont utilisables soit pour l'optimisation soit pour l'aide à la conception des déclencheurs [262].

#### 3.3.2 Optimisation des Applications Actives

*Participants* : Françoise Fabret, François Llibat, Eric Simon, Dimitri Tombroff

Pour répondre au problème de performances des applications de bases de données actives, nous avons développé deux techniques complémentaires d'optimisation. La première permet une vérification efficace des contraintes d'intégrité (règles de gestion particulières) en fin de transaction. L'exécution de ces contraintes réduit le débit transactionnel pour deux raisons : (i) les verrous en lecture pris par les contraintes obligent des transactions écrivains à attendre (et inversement), et (ii) les verrous obtenus par une transaction ne sont libérés qu'après l'exécution des contraintes, ce qui crée des attentes avec des transactions concurrentes. Nous proposons de réduire considérablement cette perte de débit en utilisant un protocole spécial de contrôle des accès concurrents basé sur des versions [264], [232]. Ce protocole permet aux contraintes d'effectuer leurs lectures sans poser de verrous et donc sans bloquer d'autres transactions. La correction de ce protocole a été prouvée et son implantation est achevée. Des mesures effectuées dans des configurations très diverses montrent une amélioration du débit transactionnel significative par rapport à l'utilisation du protocole de verrouillage strict à deux phases. Une

excellente propriété du protocole est que sa réalisation ne nécessite que de très légères modifications des algorithmes de contrôle d'accès concurrents multi-versions déjà réalisés par de nombreux SGBD.

La seconde technique permet d'optimiser l'exécution répétée de déclencheurs en évitant de générer des calculs redondants dans les composantes condition et action des déclencheurs [263]. Notre technique a été présentée dans le cadre de déclencheurs se conformant au modèle d'exécution de la proposition de standard SQL3. Nous avons d'abord défini un modèle permettant de décrire des programmes, des déclencheurs et leurs interactions. Des algorithmes ont ensuite été proposés afin d'extraire des expressions de requêtes invariantes dans les composantes condition et action des déclencheurs en fonction du résultat de l'analyse de leurs contextes d'exécution. Des heuristiques simples s'appuyant sur des modèles de coût existants sont utilisées afin d'isoler les invariants les plus rentables à anté-mémoriser. Enfin, nous avons développé une méthode permettant de générer différentes versions d'un même déclencheur en fonction de son contexte d'exécution et des invariants mémorisés, puis de gérer ces versions au moment de l'exécution d'une application. L'ensemble de cette technique est en cours de prototypage au dessus du SGBD Oracle.

### 3.3.3 Contraintes d'intégrité réparties

*Participants* : Esther Pacitti, Eric Simon

Un problème fréquent des applications de bases de données réparties est de vérifier efficacement une contrainte d'intégrité portant sur des données appartenant à différentes bases (ou sites). Chaque fois qu'une transaction locale modifie les données d'un site, on doit accéder aux données des autres sites et vérifier la contrainte. Une solution simple consiste à traiter la transaction locale comme une seule transaction distribuée qui effectue la mise à jour et accède aux données des autres sites en suivant le protocole de validation en deux étapes (2PC). Mais cette solution est chère à mettre en œuvre, car il faut fiabiliser les serveurs de données et les connections entre sites, et peu performante dès que le nombre de sites augmente.

Nous nous intéressons à une autre solution qui consiste à utiliser des copies des données des autres sites, souvent appelées répliquas, que l'on stocke localement. Cette dernière solution, économique et performante, a l'inconvénient de n'être pas sûre, c.à.d. que des incohérences peuvent être introduites dans les bases de données. Nous supposons qu'une seule copie (dite primaire) des données peut être mise à jour et nous étudions les performances de plusieurs protocoles de gestion des répliquas en faisant varier la façon dont les mises à jour des copies primaires sont répercutées vers les autres copies (dites secondaires) et la façon dont les copies secondaires sont rafraichies. Un modèle de simulation est en cours de développement au dessus du SGBD Oracle afin de comparer les performances de ces protocoles dans des configurations variées de mise à jour des sites sources et de contraintes d'intégrité. Les mesures prennent en compte le niveau de qualité des copies, c.à.d. l'écart existant entre les copies primaires et secondaires et la possibilité d'introduction d'incohérences irréparables entre copies.

Une variante intéressante du problème ci-dessus est la conception d'un entrepôt de données ("data warehouse", en anglais). La démarche consiste à copier des données provenant de plusieurs bases de données, appelées sites sources, puis à les recomposer dans une base de données particulière sur le site de l'entrepôt. Cet entrepôt doit servir à répondre à des requêtes décisionnelles tout en garantissant des contraintes de qualité de l'entrepôt, c.à.d. que les données de l'entrepôt diffèrent des données initiales dans des limites spécifiées à l'avance. Les résultats de notre travail de simulation sur la vérification de contraintes d'intégrité réparties seront appliqués aux entrepôts de données dans le cadre du projet LTR DWQ.

## 3.4 Recherche d'informations dans des sources de données hétérogènes : Disco

Le nombre, la variété et la taille des sources de données accessibles dans un réseau global comme Internet sont en augmentation constante. Cette expansion rapide introduit deux problèmes majeurs. D'abord, il est difficile de savoir quelles sont les données pertinentes et leur localisation. Ensuite, même si les sources de données pertinentes sont identifiées, il reste difficile d'accéder et d'intégrer rapidement des

données hétérogènes sur un réseau global à partir de critères de recherche complexes. Ces problèmes sont particulièrement sensibles dans le domaine des systèmes d'information pour l'environnement [241]. Pour aborder ces problèmes, nous avons lancé en 1995 le projet Disco (Distributed Information Search COmponent) dont l'objectif est de construire un système de recherche d'informations dans des sources de données hétérogènes.

Cette action s'est concentrée sur plusieurs aspects complémentaires : (i) définition de l'architecture de Disco, (ii) reformulation de requêtes hétérogènes, et (iii) exécution de requêtes réparties.

### 3.4.1 Définition de l'architecture de Disco

*Participants* : Olga Kapitskaia, Hubert Naacke, Anthony Tomasic, Patrick Valduriez

Pour uniformiser la représentation des données, Disco s'appuie sur le modèle standard de l'ODMG en étendant le langage ODL avec la définition de vues globales [256]. L'architecture de Disco repose sur trois types de composants : les *médiateurs* qui encapsulent la représentation des sources de données, les *adaptateurs* qui convertissent les requêtes sur les sources de données locales, et les *catalogues* qui répertorient tous les composants du système réparti et leur localisation. Un médiateur est un serveur pour les applications ou d'autres médiateurs.

Pour faciliter l'ajout de sources de données dans un médiateur, Disco modélise les connexions aux sources de données par des objets et supporte les transformations de types (du niveau local au niveau global). Les médiateurs gèrent aussi des répertoires de méta-données et des index qui sont utilisés pour optimiser l'accès aux informations. Un médiateur fournit donc des fonctions bases de données pour gérer l'accès concurrent aux méta-données et intégrer les résultats des requêtes. Le traitement des requêtes repose sur des techniques de reformulation et d'exécution répartie, et prend en compte l'indisponibilité des sources de données pendant l'exécution.

Disco est en cours de conception. Un premier prototype de médiateur a été construit en Java. Il supporte la définition de schéma global, de schéma local, et de vue et peut décomposer les requêtes depuis le niveau global vers le niveau local. Une interface HTML devra permettre l'accès sur le Web. La technologie des adaptateurs est développée par Bull dans le contexte du GIE Dyade et doit être intégrée avec un médiateur pour une application de bourse d'affaires électronique.

### 3.4.2 Reformulation de requêtes hétérogènes

*Participants* : Daniela Florescu, Patrick Valduriez

Avec un système comme Disco, l'utilisateur a une vision commune des données hétérogènes grâce au modèle ODMG étendu et peut les interroger de manière transparente via un médiateur avec le langage OQL. Les sources de données sont décrites au niveau Disco comme des vues par des *règles d'équivalence*.

Dans ce contexte, le traitement des requêtes hétérogènes, exprimées en OQL, procède en deux étapes essentielles : (i) reformulation de la requête en des requêtes équivalentes portant sur les interfaces des sources de données locales, et (ii) décomposition des requêtes reformulées en des requêtes locales pouvant être traitées par les traducteurs et une requête de composition du résultat.

L'étape de reformulation est compliquée par la diversité des connaissances sémantiques qui décrivent les données au niveau Disco et par la complexité du langage OQL. A partir du module de transformation de requêtes de l'optimiseur Flora, nous avons développé un algorithme de reformulation [230, 236] qui exploite ces connaissances exprimées sous forme de règles de réécriture avec une technique de pattern-matching. L'algorithme est suffisamment général pour pouvoir réutiliser les résultats de requêtes pré-calculées.

Nous avons proposé [251] une solution pour exploiter les vues matérialisées, définies par des expressions OQL, et éviter le calcul systématique de certaines expressions d'une requête donnée. Nous donnons un algorithme polynomial qui établit les conditions suffisantes pour décider de l'utilisation d'une vue matérialisée, et proposer une réécriture non triviale de la requête.

### 3.4.3 Exécution de requêtes réparties

*Participants* : Anthony Tomasic

L'exécution de requêtes réparties sur un réseau global comme Internet pose un problème nouveau : le temps de réponse de la requête peut varier grandement selon la charge du réseau et la disponibilité des sources de données accédées. Pour résoudre ce problème, nous avons proposé un algorithme de modification dynamique de plan d'exécution, ou *query scrambling* [245]. Cet algorithme modifie les plans d'exécution pendant l'exécution de la requête pour compenser les retards imprévus lors de l'accès aux données. Les modifications de plan concernent l'ordonnancement des opérations et l'introduction de nouveaux opérateurs. Les résultats de simulation ont montré que l'algorithme cache effectivement bien les retards de réception des premiers éléments de réponse de la requête.

La technique de *query scrambling* est étendue dans [258] pour prendre en compte l'arrivée des données en provenance des sources à un taux imprévu et irrégulier. L'extension principale consiste à matérialiser les données lorsqu'elles arrivent trop vite.

## 3.5 Gestion d'objets persistants

*Participants* : Seweryn Goliński, Marcin Skubiszewski

Cette action a pour objectif la conception de mécanismes et d'outils système performants qui seront à la base des futurs SGBDO. Cette année, notre effort a porté sur le ramasse-miettes et la génération efficace de *makefiles* pour SGBDO.

### 3.5.1 Ramasse-miettes concurrent

Nous avons poursuivi notre travail sur un ramasse-miettes (GC) qui soit à la fois performant et *concurrent*. Concurrent signifie que le GC peut s'exécuter en parallèle avec les autres activités du système, sans être gêné par elles, et sans les gêner.

Notre ramasse-miettes fonctionne selon le principe bien connu de marquage-balayage, mais utilise une technique nouvelle pour résoudre le problème le plus délicat, à savoir pour donner au GC une vision cohérente du système malgré son caractère concurrent. Ce mécanisme s'appelle *coupes GC-cohérentes* ; il ressemble aux *coupes causales*, bien connues dans le monde des systèmes répartis. Ce mécanisme consiste à mémoriser l'ancien état des pages dont la modification risque d'introduire des incohérences. Le GC utilise alors l'état mémorisé à la place de l'état courant.

Les coupes GC-cohérentes présentent deux avantages [265]. D'abord, contrairement à toutes les autres techniques, elles n'influencent pas le fonctionnement du système (et, notamment, n'impliquent aucun surcoût) pendant que le GC ne s'exécute pas. Cette qualité est plus importante que l'efficacité du GC car dans les applications réelles, le GC ne s'exécute que pendant une faible fraction du temps total de fonctionnement d'un système. Ensuite, elles sont très efficaces (peu d'entrées-sorties) lorsque le système utilise un *journal des images après*, ce qui est le cas de plusieurs SGBDO, comme par exemple de O<sub>2</sub>.

Cette année a été consacrée à une mise en œuvre industrielle du ramasse-miettes [255].

### 3.5.2 Compilation de *makefiles*

*Participants* : Marcin Skubiszewski, Robert Walkowski

En 1996, nous avons commencé à nous intéresser aux problèmes pratiques de génie logiciel, qui sont importants pour les développeurs de SGBD, qui sont des logiciels de grande taille. Notamment, nous avons démarré un projet visant à améliorer la vitesse d'exécution du programme *make*. A cette fin, nous avons introduit la notion de *compilation de makefiles*, qui consiste à transformer automatiquement

des makefiles compliqués en d'autres, sémantiquement équivalents mais plus simples et plus rapides à exécuter.

Notre intérêt pour la compilation de makefiles vient de la constatation que dans de grands projets logiciels, l'exécution de la commande *make* prend un temps considérable, et que ce temps est principalement dû à la complexité des makefiles utilisés. Par exemple, le système O<sub>2</sub> que nous connaissons bien comporte environ 3000 makefiles, qui sont tous lus lors d'une exécution de *make*. *Make* s'exécute en 15 minutes sur ce projet (il s'agit du temps d'exécution de *make* lui-même, sans compter le temps de compilation). Or, sur un projet d'une complexité équivalente, mais muni d'un makefile spécialement écrit pour être efficace (projet créé exprès pour faire des mesures), *make* s'exécute en 35 secondes. Ceci nous fait penser que la compilation de makefiles peut grandement accélérer l'exécution de *make*. Telle que mise en œuvre en novembre 1996, avec très peu d'optimisations, la compilation de makefiles fait baisser le temps d'exécution de 34 % sur un exemple réaliste que nous avons étudié. Nous espérons atteindre 90 % en 1997.

## 4 Actions industrielles

L'équipe collabore étroitement avec des partenaires industriels pour deux raisons. D'abord, ces collaborations permettent de se concentrer et d'expérimenter sur des applications réelles, généralement stressantes pour les SGBD. Ensuite, elles offrent la possibilité de transférer nos résultats dans des prototypes ou produits plus complets. Nos partenaires industriels essentiels sont Bull et O2 Technology.

### 4.1 GIE Dyade (Bull-Inria)

Le GIE Dyade a été officiellement créé le 1er avril 1996 par Bull et l'Inria pour une durée de cinq ans. Il est présidé par Thierry Breton (Bull) et dirigé par Eric Bantégny (directeur du développement), Claude Boule (directeur pour Bull) et de P. Valdriez (directeur pour l'Inria). Ce GIE s'inscrit naturellement dans le plan stratégique de l'Inria dont la volonté est de d'accroître la valorisation industrielle de ses résultats de recherche. L'objectif est de conjuguer l'excellence scientifique de l'Inria et la capacité industrielle de Bull afin de définir et d'exploiter de nouvelles filières technologiques satisfaisant aux besoins des systèmes d'information et d'intermediation.

Dans ce contexte, la mission du GIE est d'identifier les domaines d'actions à fort potentiel innovant, de démontrer la maturité des technologies sélectionnées par des expérimentations avec des utilisateurs pilotes, et enfin de transférer et de valoriser les résultats dans les unités opérationnelles de Bull ou dans des sociétés filiales ou start-up de Bull ou de l'Inria.

Le programme de travail initial s'articule autour de quatre grands domaines : évolution des infrastructures de communication, systèmes d'information distribués, serveurs d'inforoutes et vérification formelle des systèmes distribués. Le programme de travail n'est pas figé et doit pouvoir évoluer dynamiquement pour répondre aux orientations stratégiques du partenariat. Le GIE compte aujourd'hui une soixantaine de personnes (moitié Bull, moitié Inria). Le premier succès enregistré par le GIE a été le transfert en octobre 1996 de l'action Safetech (technologie de tolérance aux fautes) dans une start-up interne dans l'unité opérationnelle Telco de Bull.

### 4.2 ESPRIT R&D

L'équipe a participé au projet Esprit3 IDEA (Bull, INRIA, ECRC, Politecnico di Milano, TXT, etc.) qui a commencé en juin 1992 pour se terminer en juin 1996. Ce projet avait pour objectif de développer un SGBD offrant des capacités déductives et objets pour un système d'exécution parallèle à mémoire distribuée. Dans ce projet, l'INRIA était responsable de la spécification de l'interface interne du système (Flora), de la conception du compilateur et de la réalisation de l'optimiseur du langage Flora. En 1996, nous avons produit la version finale de l'optimiseur Flora, qui est à la base d'un transfert

de technologie dans le produit Validity développé par Bull. En particulier, le modèle de coût de notre optimiseur a été réutilisé.

### 4.3 O<sub>2</sub> Technology

Nous collaborons avec la société O<sub>2</sub> Technology (Versailles) qui développe et commercialise le SGBD O<sub>2</sub>. Le principal thème de notre collaboration est le développement d'un ramasse-miettes concurrent pour bases de données. La collaboration nous apporte une connaissance des exigences techniques qui existent dans l'industrie et des besoins pratiques des utilisateurs. Elle nous permet de développer le ramasse-miettes directement comme un produit industriel (intégré au produit O<sub>2</sub>). Il s'agit donc d'une collaboration plus directe et plus efficace que l'habituel transfert de technologie vers l'industrie, où un industriel redéveloppe pour ses besoins des techniques mises en œuvre sous la forme d'un prototype de recherche.

Cette année, une première version du ramasse-miettes a été intégrée au SGBD O<sub>2</sub>, et est maintenant commercialisée. Cette version est partiellement concurrente (pendant une partie du ramassage des miettes, le SGBD peut être utilisé par des clients). La version suivante, qui est totalement concurrente, est en cours de développement.

Par ailleurs, nous avons fait des mesures de performances concernant la création de bases de données, puis nous avons modifié le produit O<sub>2</sub> afin d'améliorer ces performances. Nous participons aussi à la conception et à la réalisation d'un logiciel qui permet de mémoriser des objets Java dans un SGBD relationnel ou, au choix, dans le SGBD O<sub>2</sub>.

## 5 Actions nationales et internationales

### 5.1 Systèmes d'information pour l'environnement

Cette action, débutée en 1995 par E. Simon, a poursuivi l'analyse de la problématique de recherche liée aux systèmes d'information pour l'environnement. Elle a bénéficié de la mise à disposition à temps partiel d'Eric Simon au service de la recherche du Ministère de l'environnement depuis le 1er janvier 1996. Plusieurs contacts ont été établis avec des organismes publics comme l'IFREMER, le BRGM, l'IFEN, l'ADEME et l'Office International de l'Eau qui sont à la fois producteurs et utilisateurs de données sur l'environnement. Deux réunions ont été organisées afin de dégager l'expression des besoins en matière de technologies de l'information pour l'environnement et de dresser un bilan des efforts entrepris autour des systèmes d'information pour l'environnement. Ces réunions, préparées conjointement par les ministères de l'Environnement, de la Recherche et de l'Industrie et des représentants français aux programmes européens de recherche, ont rassemblé environ 150 participants à chaque fois. Une problématique commune est ressortie: (1) faciliter la recherche, l'extraction, l'intégration et l'analyse des données sur l'environnement, et (2) organiser un "commerce électronique" de ces données sur des réseaux globaux. Bien que cette problématique se retrouve dans plusieurs autres domaines d'application (e.g., "bibliothèques digitales"), le domaine de l'environnement apporte une spécificité à ces problèmes. Leur analyse nous a permis de définir une première architecture de système d'information réparti à base de médiateurs destinée à faciliter la recherche et l'interprétation de données environnementales [241]. Cette architecture s'appuie sur la technologie développée dans le prototype DISCO. Une proposition de projet européen basée sur ces idées est en fin de montage avec le FORTH en Grèce, l'équipe Air de l'Inria Rocquencourt et plusieurs utilisateurs et producteurs de données environnementales. Une autre proposition de projet, en préparation avec le BRGM et l'association européenne des services géologiques, vise à exploiter la technologie que nous développons sur les médiateurs et les entrepôts de données.

## 5.2 Actions Nationales

L'équipe prend une part importante dans l'animation du GDR Bases de Données, créé en 1993 et dirigé par Nicole Bidoit. G. Gardarin et P. Valduriez sont membres du comité de pilotage. L'équipe participe aux pôles "Gestion de la Dynamique" et "Noyaux Transactionnels" de ce GDR.

Dans le cadre des projets du Ministère de la Recherche lancés en 1996, l'équipe participe au projet Artica sur l'interopérabilité des SGBD objet. Les partenaires de ce projet sont le laboratoire PRISM (Univ. de Versailles), l'université de Montpellier et l'Inria.

## 5.3 Actions Internationales

### 5.3.1 Projet LTR DWQ

L'équipe participe au projet Esprit Long Term Research DWQ (Foundations of Data Warehouse Quality) lancé en octobre 1996 pour une durée de trois ans. Ce projet est coordonné par l'université d'Athènes (Y. Vassiliou) avec pour partenaires les universités de Rome (M. Lenzerini), Aachen (M. Jarke), Saarbruck (W. Nutt) et l'IRST de Trento (E. Franconi). L'objectif général du projet est de développer des techniques et des outils permettant une conception et une mise en œuvre rigoureuse d'entrepôts de données basées sur des facteurs de qualité de données bien définis. Les outils seront prototypés et validés en coopération avec de grands utilisateurs européens. La contribution de l'équipe Rodin est d'apporter des solutions à deux problèmes clés des entrepôts de données: le maintien à jour efficace et fiable des données d'un entrepôt en fonction des changements des sources de données, et la conception du schéma de la base de données de l'entrepôt de façon à optimiser à la fois l'exécution des requêtes décisionnelles et le maintien à jour de l'entrepôt. Ce travail s'appuie sur l'expérience que nous avons déjà acquise sur les bases de données actives avec le prototype Acto et sur la vérification de contraintes d'intégrité réparties.

### 5.3.2 Relations internationales

L'équipe entretient aussi des relations scientifiques avec les meilleurs centres de recherche étrangers en systèmes de bases de données. Citons en particulier :

- l'université de Maryland ; Laurent Amsaleg, ancien membre du projet, Michael Franklin et Louiqa Raschid ont visité le projet pendant le mois de juillet tandis qu'Anthony Tomasic et Eric Simon ont visité l'université en avril et en octobre.
- IBM, Almaden, Californie (Rakesh Agrawal, Michael Carey, Olivier Gruber).
- IBM, Santa Teresa, Californie (Jerry Kiernan).
- AT&T Research, New Jersey (Daniela Florescu, ancien membre du projet).
- Bell Labs, New Jersey (Narain Gehani, Rick Hull) ; Eric Dujardin, ancien membre du projet, a rejoint ce laboratoire cette année.
- HP Labs, Palo Alto, Californie (Umesh Dayal).
- l'université de Stanford (Hector Garcia-Molina, Jeff Ullman, Gio Wiederhold, Jennifer Widom).
- l'université de Madison, Wisconsin (David DeWitt, Miron Livny).
- NYU, New York; Dennis Shasha a visité notre équipe pendant une semaine cet été.
- l'université d'Alberta, Edmonton, Canada (Tamer Özsu).
- l'université de Glasgow (Malcolm Atkinson, Laurent Daynès).

- l'université de Darmstadt (Alex Buchmann).
- l'université NTUA d'Athènes, Grèce (Timos Sellis, Yannis Vassiliou) avec qui nous collaborons dans le projet LTR DWQ.
- l'université d'Aachen, Allemagne (Matthias Jarke) avec qui nous collaborons dans le projet LTR DWQ.
- l'université de Rio de Janeiro (Rozana Lanzelotte) et l'université de Recife (Décio Fonseca), avec qui nous avons un projet de coopération franco-brésilien.
- l'université de Jérusalem (Catriel Beerli, Yehoshua Sagiv, Eliezer Lozinskii); E. Simon était responsable de la partie française du contrat AFIRST entre l'INRIA et l'université de Jérusalem qui s'est terminé le 31 juillet 1996.

## 6 Diffusion des résultats

### 6.1 Actions d'enseignement

P. Valduriez a donné un cours de DEA sur les bases de données réparties à Paris 6. F. Fabret et D. Florescu ont donné des cours sur les concepts et les outils de programmation au CNAM de Versailles en troisième année. D. Florescu a assuré un cours sur les bases de données réparties à l'école franco-polonaise de Poznan. F. Lirbat a assuré des travaux pratiques en CAML et ADA au CNAM de Versailles. Il a donné un cours en bases de données actives à l'ENST Bretagne, et au DEA MISI de l'Université de Versailles. Il a également assuré un cours complet en bases de données (72 heures) à l'ISTM (Institut Supérieur de Technologie et de Management) du pôle Léonard de Vinci. J.-R. Gruser a donné des TD d'algorithmique ADA au CNAM de Versailles. E. Dujardin a donné des TD sur les bases de données objet à l'IUT d'Orsay.

### 6.2 Conférences invitées, tutoriels, cours

P. Valduriez a donné une conférence sur les bases de données parallèles à l'Ecole Polytechnique Fédérale de Lausanne. Il a été conférencier invité dans deux colloques internationaux: *International Conference on Parallel and Distributed Computing Systems (PDCS'96)* à Dijon, et la Conférence Africaine sur la Recherche en Informatique (CARI'96) à Libreville. F. Lirbat a assuré un tutoriel d'une journée à la conférence CARI'96 sur les bases de données actives et les bases de données à objets.

Mokrane Bouzeghoub a visité l'équipe du Prof. Tore Risch de l'Université de Linköping (Suède) et a effectué une présentation des travaux de l'équipe sur les bases de données actives.

E. Simon a participé à deux réunions du groupe de travail technique MITWG du projet ENRM (Environmental and Natural Resource Management) du G7 sur les autoroutes de l'information. Ce groupe a pour tâche de définir l'architecture du système GILS permettant de faciliter l'accès aux données environnementales en utilisant des méta-informations sur les sources de données accessibles. Eric Simon participe également en tant qu'expert aux travaux du groupe de travail ENTAG (ENvironment Telematics Advisory Group) de la DG XIII chargé de définir les orientations du programme Environment Telematics dans l'appel d'offre du V PCRD (1998 - 2002).

### 6.3 Organisation de colloques et de cours

M. Bouzeghoub et G. Gardarin ont été respectivement président et président de l'organisation de la conférence internationale EDBT qui s'est déroulée à Avignon en mars 1996. L'Inria y a tenu un stand où a été démontré notre prototype Flora. P. Valduriez est le président du comité de programme de la conférence internationale ACM SIGMOD en 1997. E. Simon a participé à l'organisation du colloque

des “Entretiens de Ségur” sur les inforoutes pour l’environnement organisé à la demande de C. Lepage, ministre de l’environnement, et avec la participation de F. Fillon, ministre délégué de la Poste et des Télécommunications. Il co-préside le comité de programme de la conférence européenne sur les technologies de l’information pour l’environnement qui est co-organisée par l’INRIA en septembre 1997 à Strasbourg. Mokrane Bouzeghoub a participé à l’organisation du second workshop NLDB qui a eu lieu à Amsterdam et dont il a lancé la première édition en 95 à Versailles. Il est l’éditeur invité d’un numéro spécial de la revue Data and Knowledge Engineering consacré à ce workshop.

L’équipe a participé aux comités de programme des colloques suivants :

- Int. Conf. on Extending Database Technology, Avignon (P. Valduriez),
- Int. Conf. on Very Large Data Bases, Bombay, India (E. Simon),
- Int. Conf. of ACM SIGMOD, Montréal, Canada (E. Simon).
- IEEE Int. Conf. on Metadata, Silver Spring, USA (E. Simon).
- ACM Symp. on Applied Computing, USA (P. Valduriez).
- Int. Workshop on Advanced Transaction Models and Architectures, Goa, Inde (P. Valduriez).
- Int. Conf. of the Austrian Chapter on Parallel Computing, Klagenfurt, Autriche (P. Valduriez).
- Conférence Bases de Données Avancées, Cassis (P. Valduriez).

L’équipe contribue aussi à des comités de lecture et associations:

- Transactions on Database Systems, ACM (P. Valduriez).
- Int. Journal on Intelligent and Cooperative Database Systems, World Scientific (P. Valduriez).
- Int. Journal on Distributed and Parallel Database Systems, Kluwer Academic Publishers (E. Simon, P. Valduriez)
- VLDB Journal (G. Gardarin, P. Valduriez).
- VLDB Endowment (P. Valduriez).
- Journal of Data and Knowledge Engineering, North Holland (G. Gardarin).
- Ingénierie des Systèmes d’Information, Hermes (M. Bouzeghoub, rédacteur en chef ; G. Gardarin, E. Simon, P. Valduriez.)

## 7 Publications

### Livres et monographies

- [226] P. APERS, M. BOUZEGHOUB, G. GARDARIN, *Proceedings de la Conférence EDBT’96, Avignon, France*, Springer Verlag, Mars 1996.
- [227] M. BOUZEGHOUB, G. GARDARIN, P. VALDURIEZ, *Objets : Concepts, Langages, Bases de Données, Architectures*, Eyrolles, Janvier 1997, 3e Edition.
- [228] G. GARDARIN, O. GARDARIN, *Maîtriser le Client-Serveur*, Eyrolles, Mars 1996.

## Thèses

- [229] L. BOUGANIM, *Equilibrage de charge lors de l'exécution de requêtes sur des architectures multiprocesseurs hybrides*, thèse de doctorat, Université de Versailles, Décembre 1996.
- [230] D. FLORESCU, *Espaces de Recherche pour l'Optimisation de Requêtes Objet*, Thèse de doctorat, Université Paris VI, Octobre 1996.
- [231] J. GRUSER, *Modèles de coût pour l'Optimisation de Requêtes Objet*, thèse de doctorat, Université de Paris VI, Décembre 1996.
- [232] D. TOMBROFF, *Techniques pour la conception et l'exécution efficace de transactions*, Thèse de doctorat, Université Paris VI, Juin 1996.

## Articles et chapitres de livre

- [233] E. AMIEL, M.-J. BELLOSTA, E. DUJARDIN, E. SIMON, «Type-Safe Relaxing of Schema Consistency Rules for Flexible Modelling in OODBMS», *VLDB Journal* 5, 2, Avril 1996.
- [234] M.-J. BELLOSTA, G. LECORGNE, B. NICOLAS, P. VALDURIEZ, «Extending an ODBMS to Support ISO Network Management», *Int. Journal of Intelligent Systems for Electrical Engineering and Communication, special issue on Databases and Telecom*, juillet 1996.
- [235] D. FLORESCU, J. GRUSER, H. NAACKE, M. ZIANE, Z. TANG, «Flora - Un optimiseur pour Bases de Données à Objets», *Ingénierie des systèmes d'information*, Janvier 1996.
- [236] D. FLORESCU, L. RASCHID, P. VALDURIEZ, «Query Modification in Multidatabase Systems», *Int. Journal of Intelligent and Cooperative Information Systems, special issue on Formal Methods in Cooperative Information Systems: Heterogeneous Databases* 5, 4, Décembre 1996.
- [237] W. HASAN, D. FLORESCU, P. VALDURIEZ, «Open Issues in Parallel Query Optimization», *SIGMOD Record*, Septembre 1996.
- [238] A. KOTZ-DITTRICH, E. SIMON, «Promises and Realities of Active Database Systems», in: *Active Database Systems*, N. Paton et O. Diaz (réd.), Springer Verlag, 1997.
- [239] T. ÖZSU, P. VALDURIEZ, «Distributed and Parallel Database Systems», *ACM Computing Surveys* 28, 1, Mars 1996.
- [240] A. TOMASIC, L. GRAVANO, C. LUE, P. SCHWARZ, L. HAAS, «Data Structures for Efficient Broker Implementation», *ACM Transactions on Information Systems*, 1997, To Appear.
- [241] A. TOMASIC, E. SIMON, «Improving Access to Environmental Data Using Context Information», *SIGMOD Record, Special Issue on Environmental Information Systems* 26, 1, Janvier 1997.
- [242] P. VALDURIEZ, M. ZAIT, M. ZIANE, *Parallel Database Techniques*, IEEE Press, 1997, ch. Designing an Optimizer for Multi-processor Relational Systems.
- [243] M. ZAÏT, D. FLORESCU, P. VALDURIEZ, «Benchmarking the DBS3 Parallel Query Optimizer», *IEEE Parallel and Distributed Technology* 4, 2, 1996.

## Communications à des congrès, colloques, etc.

- [244] E. AMIEL, E. DUJARDIN, «Supporting Explicit Disambiguation of Multi-Methods», in: *European Conference on Object-Oriented Programming*, Linz, Autriche, Juillet 1996.
- [245] L. AMSALEG, M. FRANKLIN, A. TOMASIC, U. TOLGA, «Scrambling Query Plans to Cope with Unexpected Delays», in: *International Conference on Parallel and Distribution Information Systems (PDIS)*, Miami Beach, Florida, 1996.
- [246] L. BOUGANIM, B.DAGEVILLE, P. VALDURIEZ, «Adaptive Parallel Query Execution in DBS3», in: *5th International Conference on Extending Database Technology*, Avignon, Mars 1996.

- [247] L. BOUGANIM, B. DAGEVILLE, D. FLORESCU, «Skew handling in DBS3», in : *Proc. 3rd Int. Conf. of the Center for Parallel Computation, ACPC'96*, Klagenfurt, Austria, Septembre 1996.
- [248] L. BOUGANIM, D. FLORESCU, P. VALDURIEZ, «Dynamic Load Balancing in Hierarchical Parallel Database Systems», in : *Proc. 22nd Int. Conf. on Very Large Data Bases*, Bombay, India, Septembre 1996.
- [249] L. BOUGANIM, D. FLORESCU, P. VALDURIEZ, «Répartition dynamique de la charge dans un système de bases de données parallèle hiérarchique», in : *12ème journées Bases de Données Avancées*, Cassis, France, Août 1996.
- [250] L. DAYNÈS, M. ATKINSON, P. VALDURIEZ, «Efficient Support for Customizing Concurrency Control in Persistent Java», in : *Int. Workshop on Advanced Transaction Models and Architectures*, Goa, Inde, Septembre 1996.
- [251] D. FLORESCU, L. RASCHID, P. VALDURIEZ, «Answering Queries Using OQL View Expressions», in : *Proc. Int. Workshop on Materialized Views in cooperation with ACM SIGMOD*, Montreal, Canada, Juin 1996.
- [252] G. GARDARIN, J. GRUSER, Z. TANG, «Cost-Based Selection of Path Expression Processing in Object-Oriented Databases», in : *Proc. of 22nd Int'l. Conf. on Very Large Data Bases*, Bombay, India, Septembre 1996.
- [253] M. MATULOVIC, F. FABRET, F. LLIRBAT, E. SIMON, «Architecture d'un Système de Règles à la Sémantique Paramétrable», in : *12ème journées Bases de Données Avancées*, Cassis, France, Août 1996.
- [254] B. SALZBERG, D. TOMBROFF, «DSDT: Durable Script Containing Database Transactions», in : *Int. Conf. on Data Engineering*, New Orleans, Louisiana, Février 1996.
- [255] M. SKUBISZEWSKI, N. PORTEIX, F. XHUMARI, «Développement d'un ramasse-miettes concurrent pour O2», in : *Actes des Journées INFORSID*, Bordeaux, Juin 1996.
- [256] A. TOMASIC, L. RASCHID, P. VALDURIEZ, «Scaling Heterogeneous Database and the Design of DISCO», in : *The IEEE International Conference on Distributed Computing Systems (ICDCS-16)*, Hong Kong, 1996.

## Rapports de recherche et publications internes

- [257] E. AMIEL, E. DUJARDIN, E. SIMON, «Fast algorithms for compressed multi-method dispatch tables generation», *Rapport de Recherche n°2977*, INRIA, Septembre 1996.
- [258] L. AMSALEG, M. FRANKLIN, A. TOMASIC, «Query Scrambling for Bursty Data Arrival», *rapport de recherche*, Octobre 1996.
- [259] L. BOUGANIM, D. FLORESCU, P. VALDURIEZ, «Parallel Query Execution in NUMA Multiprocessors», *rapport de recherche*, Octobre 1996.
- [260] L. BOUGANIM, D. FLORESCU, P. VALDURIEZ, «Performance Experimentation on Parallel Architectures», *Rapport IDEA n°IDEA.DD.24B.004*, 1996.
- [261] E. DUJARDIN, «Efficient Dispatch of Multimethods in Constant Time With Dispatch Trees», *Rapport de Recherche n°2892*, INRIA, Mai 1996.
- [262] F. FABRET, F. LLIRBAT, E. SIMON, «Analyzing Repetitive Evaluations of Active Rules Within a Transaction», *Rapport de Recherche n°2816*, INRIA, Mars 1996.
- [263] F. FABRET, F. LLIRBAT, E. SIMON, «Eliminating Costly Redundant Computations from SQL Trigger Executions», *rapport de recherche*, INRIA, Octobre 1996.
- [264] F. LLIRBAT, D. TOMBROFF, E. SIMON, «Using Versions in Update Transactions», *Rapport de Recherche n°2940*, INRIA, Juillet 1996.
- [265] M. SKUBISZEWSKI, N. PORTEIX, «GC-consistent Cuts of Databases», *Rapport de Recherche n°2681*, INRIA, Avril 1996.

## Divers

- [266] A. TOMASIC, E. SIMON, «Improving Access to Scientific Data», Second DELOS MetaData Workshop, Bonn, Germany, Octobre 1996, Position paper.

## 8 Abstract

The general objective of the Rodin group is to design new techniques for improving the functionality and performance of database management. These techniques yield languages or algorithms which can be implemented in database system components or tools. The group pursues application-driven, system-oriented research with strong emphasis to prototyping and technology-transfer.

In 1996, we have concentrated our research activities on : discovery and retrieval of information in heterogeneous, autonomous data sources over global networks (Internet and Intranet); specification and optimization of active database applications; persistent object management in distributed multiuser fault-tolerant systems. In addition, we have terminated our activities on object-oriented query optimization and object-oriented data models.

Our research is driven by two main application domains: electronic marketplace and environmental data management. Our main industrial partners are Bull and O2 Technology with which technology transfer is happening. We are also leading the Dyade joint R&D venture between Bull and Inria.