

Rapport INRIA 1994 — Programme 1
Systèmes de Bases de Données

PROJET RODIN

3 mai 1995

PROJET RODIN

Systemes de Bases de Données

Localisation : *Rocquencourt*

Mots-clés : base de données (1), base de données active (1), contrôle de concurrence (1), gestion d'objets répartis (1), gestion de transaction (1), modèle de base de données (1), optimisation (1), parallélisme (1).

1 Composition de l'équipe

Responsable scientifique

Patrick Valduriez, directeur de recherche

Responsable permanent

Eric Simon, directeur de recherche

Secrétariat

Elisabeth Baqué, assistant-ingénieur

Personnel INRIA

Frank Charlemagne, ingénieur expert

Claude Darriumerlou, ingénieur expert

Olivier Gruber, ingénieur expert

Michael Novak, ingénieur expert, jusqu'au 31 Août

Marcin Skubiszewski, chargé de recherche, depuis le 1er octobre

Conseillers scientifiques

Marie-Jo Bellosta-Tourtier, maître de conférences associé,
université Paris 9

Georges Gardarin, professeur, université de Versailles

Mikal Ziane, maître de conférences, université Paris 5

Professeurs invités

Michael Franklin, université du Maryland, USA, 2 mois

Louïqa Raschid, université du Maryland, USA, 6 mois

Betty Salzberg, université North Eastern, USA, 1 mois

Chercheurs doctorants

Eric Amiel, boursier Inria, université de Versailles

Laurent Amsaleg, boursier Inria, université Paris 6

Laurent Daynès, boursier Inria, université de Versailles

Eric Dujardin, boursier MESR, université Paris 6

Françoise Fabret, chargée de travaux, CNAM

Daniela Florescu, boursière MESR, université de Versailles

Jean-Robert Gruser, boursier MESR, université de Versailles

François Llibat, boursier MESR, université de Versailles

Maja Matulovic, boursière MESR, université de Versailles,
depuis le 1er octobre

Zhaohui Tang, université de Versailles

Dimitri Tombroff, boursier Inria, université Paris 6

Mohamed Zaït, boursier Inria, université Paris 6, jusqu'au 30
septembre

Stagiaires

Camille Ben-Achour, DEA, université Paris 6

Bich-Lin Doan, DEA, université Paris 9

Maja Matulovic, DEA, université de Versailles

James Nkouankam, DEA, université Paris 6

2 Présentation du projet

L'objectif général du projet est de concevoir et d'expérimenter les technologies nécessaires aux systèmes de gestion de bases de données (SGBD). Ces systèmes devraient intégrer efficacement des capacités déclaratives (règles actives et déductives) et objets (types abstraits, méthodes associées et objets complexes) sur des architectures distribuées.

Le projet s'inscrit dans cet axe en menant quatre actions de recherche complémentaires :

- Extension de la flexibilité et de l'expressivité des modèles de données à objets. Cette action aborde des aspects communs au domaine des langages de programmation à objets.
- Aide à la spécification et à l'optimisation d'applications de bases de données actives. Cette action recouvre des aspects communs aux langages de programmation et au génie logiciel.
- Optimisation de requêtes à objets pour différents modèles d'exécution, en particulier, parallèles. Cette action emprunte au domaine de la recherche opérationnelle, en particulier, les techniques d'optimisation combinatoire.
- Gestion d'objets persistants dans un environnement réparti, multi-utilisateur et tolérant aux fautes. Cette action vise à exploiter les systèmes d'exploitation répartis à base de micro-noyau et le parallélisme afin d'améliorer les performances de la gestion d'objets.

Les domaines de recherche connexes sont appréhendés avec la spécificité "bases de données", c'est à dire, en supposant de grandes collections de données partagées par des utilisateurs concurrents. Afin de faire progresser l'état de l'art des SGBD, la stratégie du projet est la suivante :

- coopérer étroitement avec des partenaires industriels (Bull, 02 technology, Matra Marconi Space, CNET) pour mieux comprendre les besoins des applications réelles et dégager les problèmes de recherche difficiles posés aux technologies des SGBD,
- étudier séparément ces problèmes depuis leurs aspects fondamentaux jusqu'au prototypage de solutions à des fins de démonstration et d'expérimentation,
- intégrer nos résultats dans des prototypes plus complets en partenariat avec des industriels pour les confronter aux applications réelles.

Notre activité est très complémentaire de celle du projet Verso avec qui nous entretenons une collaboration étroite (séminaires et projets Esprit communs). Nous participons aussi à un projet national (PRC) avec le projet Sor.

En 1994, les faits suivants ont marqué l'activité du projet Rodin : le lancement d'une nouvelle action de recherche pour l'aide au développement d'applications de bases de données actives, guidée par des applications; la mise au point d'un nouveau prototype d'optimiseur (Flora) et sa démonstration dans une manifestation internationale; le renforcement de l'équipe par l'arrivée de Marcin Skubiszewski pour travailler sur la gestion d'objets répartis.

3 Actions de Recherche

3.1 Bases de données à objets

Cette action a pour objectif d'étendre l'expressivité et la flexibilité des modèles de données à objets supportés par les SGBD existants, tout en préservant la sûreté de typage des méthodes. Trois thèmes de travail ont été abordés: (i) le développement d'un modèle de bases de données à objets, (ii) le développement d'un mécanisme de gestion d'exceptions destiné à faciliter les mises à jour de schémas et (iii) la spécification d'algorithmes efficaces pour réaliser l'envoi de multi-méthodes.

3.1.1 Développement d'un modèle de données à objets

Participants : Eric Dujardin, Marie-Jo Bellosta, Maja Matulovic, Eric Simon

Nous avons défini un modèle à objets qui améliore flexibilité et expressivité. Ce modèle permet la représentation générale et homogène d'objets partagés et d'objets non partagés (sémantique de valeur). Les concepts d'héritage et de sous-typage sont dissociés, ce qui permet par exemple que la structure d'un type n'ait aucune influence sur la structure de ses sous-types. Ensuite, toutes les méthodes exprimables dans le modèle sont des multi-méthodes qui généralisent le schéma conventionnel des mono-méthodes présent dans la plupart des modèles de bases de données. La sélection du code associé à l'invocation d'une multi-méthode s'effectue en fonction du type de tous les arguments de l'invocation. Enfin, le modèle supporte la définition explicite d'exceptions aux règles de cohérence comportementale que doit satisfaire un schéma de bases de données. Ces règles ont pour vocation de garantir le typage statique sûr de toutes les méthodes définies dans le schéma. Une architecture modulaire d'un système à objets reposant sur ce modèle a été définie et une

grande partie du modèle a été implémentée au-dessus du système O2 [42].

3.1.2 Aide à l'évolution de schémas

Participants : Eric Amiel, Marie-Jo Bellosta, Eric Dujardin, Eric Simon

Pour assurer la sûreté de typage dans les systèmes à objets, des règles de sous-typage ont été définies. Lors de la spécialisation d'une méthode, on impose de spécialiser le type de retour de la méthode (*covariance*) et de généraliser le type des arguments ne participant pas à la liaison retardée (*contravariance*). On exige aussi qu'une méthode soit applicable à *tous* les sous-types de ses arguments. En pratique, ces règles sont trop contraignantes, surtout lors de certaines opérations de mise à jour du schéma qui se produisent inévitablement au cours de l'évolution de la base de données. Dans les systèmes existants, le respect des règles de sous-typage impose, dès qu'une exception à ces règles apparaît, de restructurer le graphe des classes et de réécrire certaines méthodes. A ces coûts de maintenance s'ajoute celui de la propagation de la mise à jour du schéma sur les objets de la base (évolution d'objets).

Notre approche, décrite dans les références [20], [6], [3], consiste à *tolérer* les exceptions aux règles de sous-typage tout en assurant la sûreté de typage par un contrôle de type à l'exécution. Nous introduisons une construction syntaxique de la forme "*CHECK ... ELSE ... END*" qui contrôle la sûreté d'une invocation de méthode ou d'une assignation. Les clauses CHECK sont automatiquement insérées dans le code de l'utilisateur autour des instructions non sûres afin : (i) d'avertir l'utilisateur de l'existence d'un problème de typage, d'un code de traitement de l'exception. Cette approche permet d'assurer la *robustesse* d'un schéma (il n'y a pas d'erreur de type à l'exécution) tout en offrant une grande facilité d'utilisation (les exceptions) et en minimisant les coûts de maintenance (restructurations de la hiérarchie des classes et réécriture des méthodes). Une première implémentation de ce mécanisme a été réalisée au sein du prototype de système à objets décrit ci-dessus [37].

3.1.3 Sélection efficace de multi-méthodes

Participants : Eric Amiel, Olivier Gruber, 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 d'une fonction générique. L'approche des tables d'envoi, utilisée par exemple en C++ pour la sélection de mono-méthodes, est la seule mise en oeuvre efficace qui garantisse de coût en temps constant (c.a.d, qui n'utilise pas de mécanisme de cache). Le problème est que dans le cas des multi-méthodes, les tables d'envoi occupent une place mémoire trop importante pour être utilisables. Nous avons proposé un algorithme de compression des tables d'envoi de multi-méthodes [21]. Cet algorithme est très efficace : son coût est au pire de l'ordre de $k^2 \times K$, où K est le nombre de types dans le schéma et k est le nombre maximum de surtypes 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. De plus, bien que n'étant pas optimal, l'algorithme possède un fort taux de compression.

3.2 Bases de données actives

De nombreux SGBD incorporent un langage de définition de déclencheurs (triggers, en anglais) au langage SQL. Un déclencheur est une règle de la forme (événement, condition, action) dont l'action est exécutée lorsqu'un événement se produit et qu'une condition sur la base de donnée est vérifiée. Les événements sont générés par les transactions qui s'exécutent sur la base de données. Cette action de recherche a pour objectif d'aider à la spécification et au développement d'applications de bases de données actives (c.a.d, avec déclencheurs).

3.2.1 Aide au développement d'applications actives

Participants : Eric Dujardin, Françoise Fabret, François Lirbat, Eric Simon

Ce travail part du constat empirique qu'il existe un large fossé entre les outils de conception d'applications et les systèmes de bases de données [7]. Les outils de conception existants ne permettent pas l'expression des règles de comportement dynamique des applications (e.g., contraintes,

règles de gestion). D'un autre côté, les systèmes de base de données actives offrent des langages de définition de déclencheurs trop difficiles à utiliser par les concepteurs d'applications.

Ceci nous a amenés, dans le cadre du projet inter-PRC Active-Design, à proposer le développement d'un système d'aide à la spécification de bases de données actives. Le système proposé repose sur une architecture à trois niveaux [35]: un niveau conceptuel, un niveau formel intermédiaire et un niveau exécutable correspondant aux langages offerts par les SGBD existants. Le niveau conceptuel, développé à l'Université de Versailles au sein de l'équipe de Mokrane Bouzeghoub, permet l'expression de règles de comportement dynamiques dans un modèle à objets dédié aux concepteurs d'applications. A ce niveau, des interfaces multimodales sont utilisables pour exprimer les spécifications des utilisateurs. Le niveau formel permet la traduction des spécifications conceptuelles dans un langage optimisable et compilable en code exécutable sur différents SGBD. Notre développement de ce langage formel s'appuie d'une part sur le modèle de données à objets décrit plus haut et d'autre part sur notre expérience acquise avec le développement du langage de règles actives A-RDL [15].

3.2.2 Evaluation incrémentielle de déclencheurs

Participants : Camille Ben-Achour, Françoise Fabret, Eric Simon

Les algorithmes d'évaluation incrémentielle de programmes de règles évitent de recalculer plusieurs fois une même expression dont les opérandes varient peu à chaque fois. L'idée consiste à mémoriser dans une anté-mémoire, le résultat du premier calcul et à le maintenir par calcul différentiel avant les demandes de calculs suivants. Un point crucial est de déterminer les expressions les plus rentables à mémoriser. Nous avons proposé une stratégie de mémorisation *globale* [4]: la décision de mémorisation d'une expression dans une règle dépend non seulement de la règle mais aussi de l'analyse, à l'aide d'heuristiques simples, des interactions possibles du reste du programme avec cette règle. Cette stratégie ainsi que les principales stratégies décrites dans la littérature (RETE, TREAT) ont été implémentées et comparées [33]. Les résultats obtenus montrent la supériorité de la stratégie globale mais aussi la nécessité d'affiner les heuristiques proposées initialement.

Parallèlement, nous avons étudié l'application de notre stratégie globale aux bases de données actives. Deux observations essentielles apparaissent. D'abord, les algorithmes incrémentiels existants sont surtout centrés sur l'optimisation de la récursion dans un programme de règles. Dans une application avec déclencheurs, il y a relativement peu de récursion. Par contre, l'exécution répétée d'un déclencheur peut être causée par la structure de la transaction déclenchante (e.g., présence de boucles). Ensuite, même si une application peut facilement avoir des centaines de déclencheurs, une transaction donnée n'en mettra en oeuvre qu'un petit sous-ensemble. Or, si on se borne à n'optimiser globalement que ce sous-ensemble de règles, on trouvera de meilleures opportunités d'optimisation pour chacune de ces règles. En effet, l'ensemble des interactions possibles entre règles est considérablement réduit et un plus grand nombre d'expressions invariantes peut être détecté. Ces observations nous ont conduits à définir un nouvel algorithme muni d'une stratégie de matérialisation adaptée qui permet d'optimiser par calcul différentiel les déclencheurs exécutés pour une transaction donnée [38].

3.2.3 Gestion de transactions et déclencheurs

Participants : François Llibat, Dimitri Tombroff, Eric Simon

Dans la plupart des systèmes, les déclencheurs sont exécutés en fin de transaction avant la validation de la transaction. Une classe importante de déclencheurs est celle dont l'action des déclencheurs ne modifie pas le contenu de la base de données. Cette classe inclut toutes les contraintes d'intégrité statiques mais aussi les règles de gestion qui filtrent les mises à jour de la base de données ou encore les règles destinées à alerter un utilisateur de l'occurrence d'une situation particulière. L'exécution de ces déclencheurs réduit le débit transactionnel pour deux raisons: (i) les verrous en lecture pris par les déclencheurs obligent des transactions écrivains à attendre (et inversement), et (ii) les verrous obtenus par une transaction ne sont libérés qu'après l'exécution des déclencheurs, 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 [46]. Ce protocole permet aux déclencheurs d'effectuer leurs lectures sans poser de verrous et donc sans bloquer d'autres transactions. La correction de ce protocole a été prouvée et l'étude de son implé-

mentation a débuté. Une excellente propriété du protocole est que son implémentation ne nécessite que de très légères modifications des algorithmes de contrôle d'accès concurrents multi-versions déjà implémentés par de nombreux SGBD.

3.3 Optimisation de requêtes

Cette action a trois objectifs complémentaires : (i) apporter des solutions à l'optimisation des SGBD à objets, (ii) valider les modèles de coût utilisés pour l'optimisation, et (iii) adapter les techniques d'optimisation à des modèles d'exécution parallèles.

3.3.1 Optimisation de requêtes à objets

Participants : Daniela Florescu, Georges Gardarin, Jean-Robert Gruser, Zhao-hui Tang, Patrick Valduriez, Mikal Ziane

Un optimiseur de requêtes relationnel est défini par: un *espace de recherche* qui décrit les plans d'exécution équivalents pour une requête donnée, un *modèle de coût* qui permet de prédire le temps d'exécution d'un plan, et une *stratégie de recherche* qui choisit un plan de coût minimal. Dans les systèmes à objets, l'optimisation de requêtes est compliquée par deux problèmes complémentaires: la plus grande richesse du langage (étendue de l'espace de recherche) et la richesse du modèle (problème de définition du modèle de coût). Pour simplifier, la plupart des optimiseurs de requêtes à objets utilisent des heuristiques.

Dans le projet Esprit IDEA, nous avons défini le langage Flora [9] [30] comme support intermédiaire pour l'optimisation de requêtes à objets. Flora offre un modèle à objets général et un langage algébrique qui accepte les fonctions utilisateurs. Nous avons conçu l'optimiseur Flora de manière modulaire [39] pour ne pas limiter l'espace de recherche et le modèle de coût. L'espace de recherche est défini par des règles d'équivalence qui permettent d'exprimer uniformément les transformations logiques et physiques des requêtes. Pour mieux comprendre le modèle de coût, nous avons commencé par définir un modèle d'exécution et un modèle de placement dans le cas centralisé [40].

Un module important de l'optimiseur est dédié à l'explication pour rendre compte des décisions quant aux transformations et méthodes d'accès choisies. Il offre aussi des services utiles pour la visualisation

graphique des plans et la sélection de solutions alternatives. Le principe consiste à s'appuyer sur le système de règles qui régit l'espace de recherche de l'optimiseur [26].

Nous avons réalisé un prototype d'optimiseur Flora en nous appuyant sur l'outil Ilog View pour l'interface graphique et le système O2 pour l'exécution des programmes Flora optimisés. Ce prototype a été démontré à la revue du projet IDEA en juin.

3.3.2 Validation d'un modèle de coût parallèle

Participants : Daniela Florescu, Patrick Valduriez, Mohamed Zaït

Dans le cadre du projet Esprit EDS, nous avons réalisé un optimiseur pour SGBD relationnels parallèles [14]. Le modèle d'exécution, le modèle de placement et le modèle de coût sur lesquels s'appuie l'optimiseur peut s'adapter aussi bien à un multiprocesseur à mémoire partagée qu'à mémoire distribuée [5].

La validation du modèle de coût est essentielle pour apprécier la précision de l'optimiseur dans le choix d'un plan d'exécution optimal. Nous avons entrepris la validation de l'optimiseur EDS en nous appuyant sur le prototype DBS3, le SGBD parallèle à mémoire partagée que nous avons réalisé en collaboration avec Bull. DBS3 est opérationnel sur les machines Encore Multimax et KSR. Pour ce faire, nous avons implémenté le banc d'essai standard AS3AP. La validation concerne les algorithmes de sélection et de jointure ainsi que l'exécution en "pipeline" des jointures [32] En comparant les temps d'exécution estimés avec les temps mesurés, nos expériences ont montré la précision de notre modèle de coût, les écarts restant très faibles.

3.3.3 Optimisation pour exécution parallèle

Participants : Jean-Robert Gruser, Mohamed Zaït, Mikal Ziane

L'optimisation pour exécution parallèle est fortement compliquée par le grand nombre de plans d'exécution possibles dans un contexte parallèle. Notamment, les caractéristiques de mémoire partagée ou distribuée influencent beaucoup les heuristiques d'optimisation.

Nous avons analysé systématiquement l'impact du parallélisme sur l'optimisation de requêtes [18]. Nous avons d'abord considéré la possibilité

d'utiliser un optimiseur classique (séquentiel) et de paralléliser le plan séquentiel optimal. Nos expériences ont montré qu'un optimiseur séquentiel donnait de bons résultats en mémoire partagée tant que les accès disques étaient négligeables. En revanche, un optimiseur parallèle (dont l'espace de recherche est constitué de plans parallèles) est indispensable pour un système à mémoire distribuée.

L'optimisation de requêtes, telle que nous l'avons abordée jusqu'à présent, est statique: les décisions de l'optimiseur sont toutes prises lors de la compilation. L'approche atteint ses limites lorsque le modèle d'exécution intègre objets et parallélisme. En particulier, la charge des noeuds et les tailles d'objets composés sont des informations difficiles à prédire sans faire d'hypothèses irréalistes. Nous abordons ce problème en étudiant l'incorporation de techniques d'optimisation dynamiques dans l'optimiseur Flora [41].

3.4 Gestion d'objets persistants

Cette action a pour objectif de faciliter la programmation d'applications complexes, multi-utilisateurs et manipulant des objets persistants dans des SGBD à objets répartis et tolérants aux fautes. Trois activités complémentaires ont été menées: (i) intégration d'un langage de programmation persistant dans le prototype EOS, (ii) proposition d'un algorithme de contrôle de concurrence pour SGBD à objets en architecture client-serveur et (iii) proposition d'une technique de ramasse-miettes pour SGBD à objets client-serveur.

3.4.1 Programmation d'Applications Persistantes dans EOS

Participants: Laurent Amsaleg, Laurent Daynès, Olivier Gruber, Patrick Valduriez

EOS (an Environment for Object-oriented Systems) est un environnement pour la construction d'applications ou de systèmes à objets répartis et multi-utilisateurs. Notre objectif est d'offrir des fonctionnalités de distribution, de persistance et de contrôle de concurrence de façon transparente et efficace. Nous avons précisé le cadre de coopération entre les divers éléments logiciels de l'environnement [13], à savoir le schéma, son compilateur et le gestionnaire d'objets. Il détaille aussi la mise en œuvre au dessus d'un micro-noyau des techniques d'adressage plat, de ver-

rouillage des objets, de distribution des traitements, de gestion mémoire et de réorganisation dynamique des objets sur disque.

Dans le cadre du projet Esprit BRA FIDE2, nous avons utilisé le langage Napier88, réalisé par l'Université de St Andrews, comme interface de programmation d'EOS. Un premier portage de ce langage au dessus du noyau d'EOS a été réalisé et démontré en septembre lors de la revue du projet FIDE2 à St Andrews.

Parallèlement, nous avons poursuivi notre coopération avec l'Institut Géographique National. Une version simplifiée d'un système d'information géographique [43] au dessus d'EOS a été également développé, affinant notre compréhension des problèmes liés à la gestion de ce type d'information.

3.4.2 Contrôle de Concurrence

Participants : Laurent Daynès, Olivier Gruber, Patrick Valduriez

Dans un SGBD à objets client-serveur, chaque client doit permettre à l'utilisateur d'exécuter simultanément des transactions pour manipuler les objets résidant dans sa mémoire. La plupart des systèmes n'autorisent qu'une transaction par processus client afin de limiter les surcoûts qu'introduirait un contrôle de concurrence interne au client. Pour obtenir plus d'une transaction sur sa machine, l'utilisateur doit alors exécuter plusieurs processus clients. Dans ce cas, le contrôle de concurrence entre les transactions d'une même machine cliente est implicitement effectué par le protocole de cohérence de cache. Cette solution simple entraîne une dégradation importante des performances à la fois sur les machines clientes et sur le serveur.

Nous proposons une solution consistant à gérer un unique cache d'objets par machine cliente, et à effectuer un contrôle local des accès concurrents [36, 23]. Nous avons comparé notre solution avec les deux seules autres solutions connues en réalisant trois gérants de transactions. Les mesures de performances ont montré la supériorité de notre solution [24]. Nous avons aussi généralisé notre solution afin de supporter un plus vaste échantillon de modèles de transactions [25]: transactions emboîtées, actions colorées, transactions dynamiquement restructurables, etc.

3.4.3 Ramasse-miettes

Participants : Laurent Amsaleg, Michael Franklin, Olivier Gruber

Un ramasse-miettes ou encore glaneur de cellules (GC) est une fonctionnalité majeure de nombreux langages de programmation. Un GC recycle automatiquement l'espace mémoire occupé par des objets devenus inaccessibles aux programmes, objets appelés *miettes*. Cette automatisation décharge le programmeur de la désallocation explicite d'objets qui est complexe et source d'erreurs.

Intégrer cette fonctionnalité dans un SGBD à objets est difficile car les techniques de GC ont été élaborées dans un cadre langage où un volume réduit de données est manipulé et où ni persistance des données ni résistance aux pannes ne sont offertes. Le premier problème est que le coût d'un GC est essentiellement proportionnel à la quantité d'objets qu'il doit examiner pour détecter les miettes. Le second problème est lié à la correction du GC. Un GC est dit incorrect lorsqu'il recycle par erreur des objets non miettes. L'atomicité des transactions, la résistance et la reprise sur pannes ainsi que l'environnement client-serveur rendent incorrectes les solutions traditionnelles [31].

Nous proposons un GC résolvant ces deux problèmes [34]. D'une part, il perturbe peu l'activité des clients et du serveur car il s'exécute de façon totalement asynchrone et parallèle. D'autre part, il s'intègre étroitement aux protocoles de résistance aux pannes sans pour autant les modifier en profondeur. Finalement, il prend en compte l'activité des clients sans introduire de surcoût de communication.

Pour démontrer et évaluer l'efficacité de notre solution, nous avons intégré notre GC au système de gestion d'objets Exodus, réalisé à l'université de Madison (Wisconsin, USA). Les premiers résultats montrent la supériorité de notre solution sur les rares solutions existantes.

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 partenaire

industriels sont Bull, O2 Technology, France Telecom (CNET) et Matra Marconi space.

4.1 ESPRIT R&D

L'équipe participe au projet Esprit3 IDEA (Bull, INRIA, ECRC, Politecnico di Milano, TXT, etc.) qui a commencé en juin 1992 pour quatre ans. Ce projet a 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 est 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 1994, nous avons réalisé et démontré un premier prototype de l'optimiseur Flora, qui est à la base d'un transfert de technologie en cours dans le produit ADS.

4.2 CNET

L'équipe collabore avec le CNET (Lannion) sur l'étude de la gestion d'objets persistants en suivant une approche couplage langage-gestionnaire d'objets. Cette collaboration s'inscrit dans le cadre d'un projet CNET concernant les nouveaux services du réseau intelligent qui doit supporter notamment le standard ISO CMIP (défini par un modèle à objets) pour l'administration de réseaux. Notre étude des différentes politiques de traitement de la persistance dans les systèmes objets avait conduit à la proposition de notre gestionnaire de schémas OMNIS comme une solution extensible. En 1994, nous avons réalisé un préprocesseur C++ pour OMNIS, utilisé par le CNET pour développer un prototype de SGBD pour l'administration de réseaux.

4.3 O2 Technology

Un protocole d'accord a été signé avec la société O2 Technology (Versailles) qui développe et commercialise le SGBD à objets O2. Cette coopération doit permettre de mieux définir nos thèmes de recherche par rapport à des besoins industriels, et de valider nos résultats de recherche par des transferts de technologie vers l'industrie. Pour O2 Technology, cette coopération donne accès à notre recherche poursuivie dans le domaine des SGBD pour influencer les versions futures du produit O2. Un transfert de notre technologie glaneur de cellules est en cours.

4.4 Matra Marconi Space

L'équipe a commencé une coopération avec la société Matra Marconi Space (Toulouse) dans le domaine des bases de données réparties et de l'optimisation de requêtes. Une première étude a permis de faire une proposition d'architecture de base de données fédérée et dupliquée pour le projet ENVISAT (gestion de bases de données satellite).

5 Actions nationales et internationales

5.1 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.

E. Simon anime le groupe de travail "Bases de Connaissances" du comité technique Bases de Données de l'AFCEP. Il contribue aussi à la mise en place d'un axe transversal environnement au sein de l'INRIA. Dans ce cadre, il a participé à plusieurs réunions de travail avec les chercheurs de l'action Masda (Rocquencourt) sur le thème de la définition de l'architecture d'un système de traitement d'images océanographiques (réponse à un appel d'offres du SHOM). Il a également rendu visite à divers centres de recherches publics et privés impliqués dans la gestion des ressources aquatiques continentales (Lyonnaise des Eaux, Cemagref, SEM Essonne, département d'hydrologie de l'Université d'Athènes). Ces contacts ont pour objectif (i) de dégager une problématique de recherche claire concernant la gestion des données aquatiques et (ii) d'isoler des échantillons d'application reflétant cette problématique.

Dans le cadre des projets inter-PRC, l'équipe participe au projet d'aide à la spécification et à la génération de bases de données actives. Les partenaires de ce projet sont le laboratoire PRISM (Univ. de Versailles), l'école des Mines de Paris, l'ESSEC et Bull. Ce projet est guidé par le développement d'une application de protection de l'environnement (aide à la prévention, détection et maîtrise des feux de forêts). L'équipe participe aussi avec le projet Sor au projet systèmes d'objets persistants répartis.

Un protocole d'accord a été signé avec l'Institut Géographique National avec qui nous coopérons dans le domaine de la gestion d'objets complexes. Cet accord nous permet d'utiliser une bases de données d'objets géographiques de 400 Mega-octets pour valider le prototype EOS.

5.2 Actions Internationales

5.2.1 Projet BRA FIDE2

Les deux équipes bases de données de l'INRIA (Rodin et Verso) participent au projet Basic Research Action Esprit FIDE2 lancé en juillet 1992 pour trois ans. Ce projet est coordonné par l'université de Glasgow (M. Atkinson) et a pour partenaires les universités de St Andrews (R. Morrison), Hambourg (J. Schmidt), Pise (A. Albano), l'IEI-CNR (F. Rabitti) et l'INRIA (P. Valduriez). L'objectif du projet est de spécifier une architecture pour la construction, la maintenance et l'exploitation de systèmes d'applications persistantes et distribuées, et de démontrer sa faisabilité par l'intégration de systèmes réalisés par plusieurs partenaires. La contribution de l'équipe Rodin est la définition de l'architecture du système en se basant sur EOS et son intégration avec les environnements de programmation persistants développés par nos partenaires écossais. En 1994, nous avons démontré l'intégration d'EOS avec le langage Napier88 de l'université de St Andrews.

5.2.2 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 :

- IBM, Almaden, Californie (Rakesh Agrawal, Jerry Kiernan, Mohamed Zaït).
- l'université de Stanford (Hector Garcia-Molina, Jennifer Widom).
- l'université de Madison, Wisconsin (Michael Carey, David DeWitt).
- NYU, New York (Dennis Shasha).
- l'université d'Alberta, Edmonton (Tamer Özsu).
- l'université de Darmstadt (Alex Buchmann).

- l'université de Rio de Janeiro (R. Lanzelotte) et l'université de Recife (D. Fonseca), avec qui nous avons un projet de coopération franco-brésilien.
- l'université de Glasgow (Malcolm Atkinson) ; L. Daynès y a été VSNA jusqu'en juin 1994.
- l'université de Jérusalem (C. Beeri, Y. Sagiv, E. Lozinskii); E. Simon est responsable de la partie française du contrat AFIRST entre l'INRIA et l'université de Jérusalem initialisé en 1993 pour une durée de deux ans.
- l'Asian Institute of Technology de Bangkok (Kanchana Kanchanasut).

6 Diffusion des résultats

6.1 Enseignement

P. Valduriez a donné un cours de DEA sur les bases de données réparties à Paris 6. E. Amiel a donné des travaux dirigés sur C++ à l'ENST. F. Fabret et D. Florescu enseignent les concepts et outils de programmation au CNAM de Versailles. O. Gruber est chargé de cours en MIAGE à Paris 9. L. Amsaleg y est chargé de travaux dirigés; il a aussi assuré un cours de bases de données réparties au CERICS, à l'ENSTB, l'INT, l'ESSI, et l'ENIC. E. Dujardin a assuré un cours UNIX à I2M, des TD sur le SGBD O2 à l'IUT d'Orsay et un cours sur les bases de données déductives au Mastère de l'université de Nice Sophia-Antipolis.

6.2 Participation à des conférences et colloques

P. Valduriez a donné plusieurs conférences invitées: université de East London, université de Twente (Hollande), laboratoires de NEC, JEIDA, et Mitsubishi (Tokyo), université de Tokyo. Il était aussi conférencier invité d'ADTI-94 (International Symposium on Advanced Database Technologies and their Integration) à Nara (Japon). Il a donné une conférence invitée sur les bases de données réparties au workshop de Rio de Janeiro sur les bases de données avancées. En France, P. Valduriez a donné un tutoriel sur les bases de données parallèles à la conférence BDA-94 et une conférence invitée sur les bases de données réparties aux journées d'étude de l'AFCEP sur la répartition et le parallélisme.

E. Simon a donné une conférence invitée sur les performances des bases de données actives au workshop de Rio de Janeiro sur les bases de données avancées. Il a été aussi conférencier invité au séminaire Dagstuhl (Allemagne) sur les bases de données actives. Il a donné un cours de deux jours sur les bases de données actives à l'Université de Darmstadt et à l'Université d'Athènes.

Mikal Ziane a présenté les activités du projet Rodin, en particulier celles concernant le parallélisme, au Workshop SGFI de Soleure (Suisse). Le SGFI représente la Suisse au sein de l'ERCIM et regroupe notamment l'EPFL de Lausanne et l'ETH de Zurich. Il a aussi donné une conférence sur les bases de données parallèles aux journées du laboratoire PRISM.

G. Gardarin a été conférencier invité de DEXA-94 (Int. Conf. on Database and Expert Systems Applications) à Athènes.

6.3 Organisation de colloques et de cours

E. Simon a été co-organisateur d'un cours européen COMETT de trois jours sur les bases de données actives et leurs applications industrielles. Il a aussi présidé le workshop international RIDE (Research Issues on Data Engineering) organisé en conjonction avec la conférence Data Engineering à Houston.

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

- Int. Conf. on Very Large Databases, Santiago, Chili, (G. Gardarin, P. Valduriez).
- Int. Conf. on Extending Database Technology, Cambridge, UK, (G. Gardarin, E. Simon, P. Valduriez).
- Int. Conf. on Applications of Databases, Vadstena, Suede (G. Gardarin, P. Valduriez).
- Int. Conf. on Parallel and Distributed Computing Systems, Las Vegas, USA (P. Valduriez).
- ACM Symp. on Applied Computing, USA (P. Valduriez).
- Int. Conf. on Management of Data, Bangalore, India (E. Simon).
- IEEE Int. Workshop on Research Issues on Data Engineering - Distributed Object Management, Taipei, 1995 (O. Gruber).
- Int. Conf. on Information and Knowledge Management, Baltimore, USA, (P. Valduriez).

- Int. Conf. on Parallel and Distributed Information Systems, Austin, USA (M. Ziane)
- Int. Conf. on Cooperative Information Systems, Toronto, Canada, (P. Valduriez).
- Journées Bases de Données Avancées, Clermont-Ferrand (P. Valduriez).
- Int. Conf. on Database and Expert Systems Applications, Athènes, (G. Gardarin).

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).
- VLDB Endowment (P. Valduriez).
- Journal of Data and Knowledge Engineering, North Holland (G. Gardarin).
- Ingénierie des Systèmes d'Information, Hermes (P. Valduriez, rédacteur en chef; G. Gardarin, E. Simon).
- Techniques et Sciences de l'Informatique, Hermes (P. Valduriez)

7 Publications

Livres et monographies

- [1] M. BOUZEGHOUB, G. GARDARIN, P. VALDURIEZ, *Objets: concepts, langages, bases de données, méthodes, interfaces*, Eyrolles, 1994.
- [2] T. ÖZSU, U. DAYAL, P. VALDURIEZ, *Distributed Object Management*, Morgan-Kaufmann, San Mateo, Californie, 1994.

Thèses

- [3] E. AMIEL, *Schémas Orientés-objet: Exceptions à la Cohérence Comportementale et Envoi de Multi-méthodes*, thèse de doctorat, université Paris 6, Juin 1994.

- [4] F. FABRET, *Optimisation du Calcul Incrémentiel Dans Les Langages de Règles pour Bases de Données*, thèse de doctorat, Université de Versailles, Décembre 1994.
- [5] M. ZAÏT, *Optimisation de Requête Relationnelles pour Exécution Parallèle*, thèse de doctorat, université Paris 6, Juin 1994.

Articles et chapitres de livre

- [6] E. AMIEL, M. BELLOSTA, E. DUJARDIN, E. SIMON, «Type Safe Relaxing of Schema Consistency Rules for Flexible Modelling in OODBs», *VLDB Journal*, A paraître.
- [7] M. BOUZEGHOUB, E. SIMON, «Integrity Specification and Enforcement in Databases», *Ingénierie des Systèmes d'Information 2*, 2, 1994.
- [8] B. FINANCE, G. GARDARIN, «A Rule Based Query Optimizer with Adaptable Search Strategies», *Data and Knowledge Engineering 3*, 1, 1994.
- [9] D. FLORESCU, J.-R. GRUSER, M. NOVAK, P. VALDURIEZ, M. ZIANE, «Design and Implementation of Flora, a Language for Object Algebra», *Information Science*, A paraître.
- [10] G. GARDARIN, B. FINANCE, P. FANKHAUSER, W. KLAS, «IRO-DB : A Distributed System Federating Object and Relational Databases», in : *Object-Oriented Multibase Systems : A Solution for Advanced Applications*, O. Bukhres et A. Elmagarmid (éd.), Springer-Verlag, 1994, ch. 1.
- [11] G. GARDARIN, «Convergence des Modèles Objets et Relationnels», *Ingénierie des Systèmes d'Information 2*, 3, 1994.
- [12] G. GARDARIN, «Object-Oriented Rule Languages and their Optimization», in : *Object-Oriented Database Systems*, T. Ozsü et T. Sellis (éd.), Springer-Verlag, 1994.
- [13] O. GRUBER, L. AMSALEG, L. DAYNÈS, P. VALDURIEZ, «An Object-oriented Foundation for Desktop Computing», *Distributed and Parallel Databases*, A paraître.
- [14] R. LANZELLOTTE, P. VALDURIEZ, M. ZAÏT, M. ZIANE, «Industrial-Strength Parallel Query Optimization: issues and lessons», *Information Systems*, A paraître.
- [15] E. SIMON, J. KIERNAN, «The A-RDL System», in : *Active Database Systems: Triggers and Rules for Advanced Database Processing*, J. Widom, S. Ceri, et U. Dayal (éd.), Morgan-Kaufmann, San Francisco, California, ch. 5, A paraître en 1995.

- [16] E. SIMON, « Adding Production Rules to Database Languages », *Information Systems*, 1994, A paraître.
- [17] P. VALDURIEZ, « Some Hints to Improve Writing of Technical Papers », *Ingénierie des Systèmes d'Information 2(3)*, 1994.
- [18] M. ZIANE, M. ZAÏT, H. H. QUANG, « Parallelism and Query Optimization », *Computer Systems Science and Engineering Journal*, A paraître.

Communications à des congrès, colloques, etc.

- [19] E. AMIEL, M. BELLOSTA, E. DUJARDIN, E. SIMON, « Aide à l'Evolution de Schémas dans les Bases de Données à Objets », in : *Journées Bases de Données Avancées*, INRIA, Clermont-Ferrand, Août 1994.
- [20] E. AMIEL, M. BELLOSTA, E. DUJARDIN, E. SIMON, « Supporting Exceptions to Behavioral Schema Consistency to Ease Schema Evolution in OODBMS », in : *Int. Conf. on Very Large Data Bases*, Morgan Kaufmann, Santiago, Septembre 1994.
- [21] E. AMIEL, O. GRUBER, E. SIMON, « Optimizing Multi-Method Dispatch Using Compressed Dispatch Tables », in : *Int. Conf. on OOPSLA*, ACM, Portland, Octobre 1994.
- [22] E. AMIEL, O. GRUBER, E. SIMON, « Sélection Efficace de Multiméthodes par Tables d'Envoi Compressées », in : *Journées Bases de Données Avancées*, INRIA, Clermont-Ferrand, Août 1994.
- [23] L. DAYNÈS, O. GRUBER, P. VALDURIEZ, « On the Cost of Lock Inheritance in Lock Managers supporting Nested Transactions », in : *Journées Bases de Données Avancées*, Clermont-Ferrand, France, Août 1994.
- [24] L. DAYNÈS, O. GRUBER, P. VALDURIEZ, « Locking in OODBMS Clients Supporting Multiple Nested Transactions », in : *Int. Conf. on Data Engineering*, Taipei (Taiwan), Mars 1995.
- [25] L. DAYNÈS, O. GRUBER, « Customizing Concurrency Controls using Graph of Locking Capabilities », in : *Int. Workshop on Persistent Object Systems*, Tarascon, France, Septembre 1994.
- [26] D. FLORESCU, P. VALDURIEZ, « Rule-based Query Processing in the IDEA System », in : *Int. Symp. on Advanced Database Technologies and Their Integration*, Nara, Japon, Octobre 1994. papier inviti.
- [27] G. GARDARIN, F. MACHUCA, P. PUCHERAL, « A Functional Execution Model to Evaluate Object-Oriented Queries », in : *Int. Conf. on Very Large Data Bases*, Morgan Kaufmann, Santiago, Septembre 1994. Poster paper.

- [28] G. GARDARIN, «Federating Object-Oriented and Relational Databases», *in: Int. Conf. on Databases and Expert Systems Applications*, Springer Verlag, Lecture Notes in Computer Sciences, Athènes, Grèce, Septembre 1994.
- [29] O. GRUBER, P. VALDURIEZ, «Towards Persistent Object Systems for Desktop Computing», *in: Int. Conf. on Cooperative Information Systems*, Toronto, Canada, Mai 1994.
- [30] M. NOVAK, G. GARDARIN, P. VALDURIEZ, «Flora: a Functional-Style Language for Object and Relational Algebra», *in: Int. Conf. on Databases and Expert Systems Applications*, Athènes, Grèce, Septembre 1994.
- [31] M. SHAPIRO, D. PLAINFOSSÉ, P. FERREIRA, L. AMSALEG, «Some Key Issues in the Design of Distributed Garbage Collection and References», *in: Seminar on Unifying Theory and Practice in Distributed Systems*, Dagstuhl (Allemagne), Septembre 1994.
- [32] M. ZAÏT, P. VALDURIEZ, D. FLORESCU, «On the Validation of a Parallel Query Optimizer», *in: Bases de Données Avancées*, Clermont-Ferrand, Août 1994.

Rapports de recherche et publications internes

- [33] C. B. ACHOUR, *Comparaison de Trois Algorithmes d'Evaluation de Programmes de Règles de Production: RETE, TREAT, COSMA*, Rapport de DEA, Université Paris 6, Septembre 1994.
- [34] L. AMSALEG, M. FRANKLIN, O. GRUBER, «Garbage Collecting a Client-Server OODBMS», *Rapport de Recherche n°2409*, INRIA, Novembre 1994.
- [35] M. BOUZEGHOUB, J. B. DE SA, E. DUJARDIN, F. FABRET, F. LLIRBAT, N. RAISS, E. SIMON, «ACTIVE-DESIGN: Aide à la spécification et à la Génération de Bases de Données Actives», *rapport de recherche*, Projet inter-PRC, Septembre 1994.
- [36] L. DAYNÈS, O. GRUBER, P. VALDURIEZ, «Lock Inheritance in Nested Transactions», *rapport de recherche n°FIDE/94/79*, projet Esprit BRA FIDE2, Janvier 1994.
- [37] B. DOAN, *Implantation d'un Mécanisme d'Exceptions Aux Règles de Cohérence Comportementale dans un Schéma de BDOO*, Rapport de DEA, Université de Dauphine, Septembre 1994.
- [38] F. FABRET, E. SIMON, «Optimizing Incremental Evaluation of Triggers Within a Transaction», *rapport de recherche*, Inria, octobre 1994.

- [39] D. FLORESCU, J. GRUSER, Z. TANG, P. VALDURIEZ, M. ZIANE, «The Flora Prototype», *Rapport Technique n°IDEA.DD.5N.012*, Projet Esprit IDEA, Juin 1994.
- [40] J. GRUSER, Z. TANG, M. ZIANE, «A Cost Model for Centralized OODBMS», *Rapport Technique n°IDEA.DD.5N.013*, Projet Esprit IDEA, Octobre 1994.
- [41] J. GRUSER, M. ZIANE, «Dynamic Optimization», *Rapport Technique n°IDEA.DD.5N.014*, Projet Esprit IDEA, Octobre 1994.
- [42] M. MATULOVIC, *Implantation du Modèle de Données IDLE*, Rapport de DEA, Université de Versailles, Septembre 1994.
- [43] J. NKOUANKAM, *Conception et Réalisation d'un Système d'Information Géographique sur une Plateforme Persistante*, Rapport de stage de dea, université Paris 6, Septembre 1994.
- [44] B. SALZBERG, D. TOMBROFF, «A Programming Tool to Support Long-Running Activities», *Rapport interne*, 1994.
- [45] D. SHASHA, F. LLIRBAT, E. SIMON, P. VALDURIEZ, «Chopping Up Transactions: Algorithms and Experimentation», *Rapport interne*, 1994, Soumis à publication.
- [46] D. TOMBROFF, F. LLIRBAT, E. SIMON, «Optimizing Database Trigger Execution Using Version-Based Concurrency Control», *Rapport interne*, 1994.

8 Abstract

The general objective of the Rodin group is to improve database system management by supporting rule-based and object-oriented capabilities and exploiting distributed system architectures. In 1994, we have carried out four complementary research activities: extension of object-oriented data models to gain flexibility and expressivity; specification and optimization of active database applications; object-oriented query optimization for distributed and parallel systems; persistent object management in distributed multiuser fault-tolerant systems.

The group pursues application-driven, system-oriented research with the following strategy: tight cooperation with industrial partners to better understand the hard problems posed to real applications; design and implementation of separate prototypes for experimentation and demonstration; and integration of our results in industrial-strength prototypes to ease technology-transfer.

Rapport d'activité INRIA 1994 — Annexe technique

We are strongly involved in the IDEA Esprit project (1992-1996) and the FIDE2 Basic Research Action (1992-1995). Our main industrial partners are Bull, O2 Technology, France Telecom (CNET) and Matra Marconi space.

Table des matières

1	Composition de l'équipe	1
2	Présentation du projet	2
3	Actions de Recherche	4
3.1	Bases de données à objets	4
3.1.1	Développement d'un modèle de données à objets	4
3.1.2	Aide à l'évolution de schémas	5
3.1.3	Sélection efficace de multi-méthodes	6
3.2	Bases de données actives	6
3.2.1	Aide au développement d'applications actives	6
3.2.2	Evaluation incrémentielle de déclencheurs	7
3.2.3	Gestion de transactions et déclencheurs	8
3.3	Optimisation de requêtes	9
3.3.1	Optimisation de requêtes à objets	9
3.3.2	Validation d'un modèle de coût parallèle	10
3.3.3	Optimisation pour exécution parallèle	10
3.4	Gestion d'objets persistants	11
3.4.1	Programmation d'Applications Persistantes dans EOS	11
3.4.2	Contrôle de Concurrence	12
3.4.3	Ramasse-miettes	13
4	Actions industrielles	13
4.1	ESPRIT R&D	14
4.2	CNET	14
4.3	O2 Technology	14
4.4	Matra Marconi Space	15

5	Actions nationales et internationales	15
5.1	Actions Nationales	15
5.2	Actions Internationales	16
5.2.1	Projet BRA FIDE2	16
5.2.2	Relations internationales	16
6	Diffusion des résultats	17
6.1	Enseignement	17
6.2	Participation à des conférences et colloques	17
6.3	Organisation de colloques et de cours	18
7	Publications	19
8	Abstract	23