
Projet LOCO

Programmation en LOGique avec COntraintes

Localisation : *Rocquencourt*

Mots-clés : programmation par contraintes, sémantique, implantation, compilation, satisfaction de contraintes, concurrence, programmation en logique, spécification formelle, débogage, Prolog, environnement de programmation, génie logiciel, grammaire attribuée, interprétation abstraite, langage de programmation, modèle d'exécution, négation, standardisation, parallélisme, traitement d'erreur, vérification de programme.

1 Composition de l'équipe

Responsables Scientifiques

Philippe Codognet, chargé de recherche, INRIA
Gérard Ferrand, professeur, Université d'Orléans

Responsable Permanent

Pierre Deransart, directeur de recherche, INRIA

Secrétaire

Josy Baron, partagée avec ATOLL, GENOME et AR

Personnel Orléans

Frédéric Benhamou, maître de Conférence, Université d'Orléans
Michel Bergère, maître de Conférence, Université d'Orléans
AbdelAli Ed-Dbali, maître de Conférence, Université d'Orléans
François Le Berre, professeur, Université d'Orléans

Chercheurs invités

Salvador Abreu, Assistant, Université d'Evora, Portugal, boursier HCM du 1.01.96 au 31.03.1996
José Siqueira, Boursier CNPq, Université Fédérale de Belo Horizonte, Brésil

Chercheur post-doctorant

Patrick Parot, VSN, Université de Belo Horizonte, Brésil, du 1.03.95 au 30.04.96

Chercheurs doctorants

Pascal Chambre, boursier INRIA
Yan Georget, Boursier DRET, INRIA, à partir du 1.10.1996
Laurent Granvilliers, boursier MENESR, Université d'Orléans
Arnaud Lallouet, ATER, Université d'Orléans
Bernard Malfon, professeur enseignement secondaire, Orléans
Sophie Renault, HCM, depuis le 1.05.95, Université de Rome
Jean-Hugues Réty, boursier INRIA
Alexandre Tessier, boursier MENESR, Université d'Orléans

Stagiaires

Arnaud Legrand, Stagiaire DEA, Université d'Orléans, de Mars 96 à Septembre 96
Nicolas Romero, Stagiaire DEA, Université d'Orléans, de Mars 96 à Septembre 96

Collaborateur Extérieur

Daniel Diaz, maître de Conférence, Université de Paris-I

2 Présentation du projet

2.1 Historique

Le projet LOCO, créé le 21 Octobre 96 à partir de l'avant projet LOCO, regroupe deux équipes géographiquement distinctes travaillant en étroite concertation: à l'INRIA-Rocquencourt et au laboratoire LIFO de l'université d'Orléans. La partie INRIA est originaire du projet ChLoE (Compilation, Logique et Environnement), créé il y a cinq ans. Le projet LOCO est issu de l'axe "Programmation en Logique" de ce projet avec un recentrage des thèmes autour de la programmation en logique avec contraintes.

2.2 Les axes de recherche

La Programmation en Logique avec Contraintes (PLC) connaît depuis quelques années un certain succès, tant au point de vue théorique et académique qu'au point de vue pratique et industriel. Elle allie en effet la simplicité d'écriture de programmes et la déclarativité de la Programmation en Logique (PL) à la puissance et l'efficacité du calcul sur des domaines spécifiques à l'aide d'outils particuliers et optimisés ("résolveurs de contraintes").

La PLC permet d'utiliser des domaines de calculs particuliers, tels les contraintes arithmétiques rationnelles, réelles ou même complexes, les formules booléennes, les domaines finis ou les structures à traits. L'intégration de ces calculs plus ou moins numériques dans un cadre logique se fait ainsi au moyen de *contraintes*, c'est à dire de relations représentant une certaine information portant sur un ensemble de variables et les liant ainsi entre elles. On peut en fait voir les contraintes comme un "interface" entre un découpage logique du programme (des clauses de Horn, c'est à dire un sous-ensemble des formules logiques du premier ordre) et un aspect calculatoire où se met en œuvre un (ou des) algorithme(s) de résolution particulier(s) sur un domaine calculable (contraintes). La PLC donne un sang neuf à la fois aux cadres théoriques et aux paradigmes anciens en PL (la programmation concurrente avec contraintes en étant l'exemple le plus frappant) et à la mise en œuvre de langages de programmation effectifs, témoin en est l'intérêt porté aux langages commerciaux de PLC. De nombreuses applications peuvent en effet s'exprimer simplement et efficacement dans le cadre PLC: problèmes combinatoires, ordonnancement, cutting-stock, simulation de circuit, diagnostic de panne, analyse financière, aide à la décision, etc. Celles-ci utilisent bien sûr différentes contraintes, domaines de calculs et algorithmes de résolution, mais elles peuvent toutes s'exprimer naturellement dans un cadre uniforme, la PLC, et être mises en œuvre efficacement dans les principaux langages de PLC existant actuellement. Ceci montre le large

champ d'application sur lequel s'est ouvert la PL en intégrant contraintes et résolveurs sur des domaines de calcul spécifiques.

Les problèmes liés à la PLC et son utilisation peuvent se regrouper en trois aspects principaux qui constituent les dominantes des axes de recherche du projet.

- Aspects langages.
- Aspects domaines de contraintes et algorithmes de résolution.
- Aspects environnement (documentation, validation, débogage).

3 Action de recherche

3.1 Aspects langages: Programmation Concurrente avec Contraintes

Participants : Pascal Chambre, Philippe Codognet, Pierre Deransart, Jean-Hugues Réty

Les langages logiques concurrents avec contraintes (CC) sont une généralisation de la PLC d'une part et des langages logiques concurrents d'autre part. Ils permettent de combiner les avantages des deux approches, c'est-à-dire d'une part la possibilité de raisonner (symboliquement) et calculer (numériquement) sur des domaines spécifiques (PLC) et d'autre part la possibilité d'avoir un contrôle dynamique des calculs dirigé par les données (concurrency) et non plus dirigé par les instructions du programme.

L'idée fondamentale des langages concurrents avec contraintes est l'utilisation de contraintes pour étendre les mécanismes de synchronisation et de contrôle des langages logiques concurrents. Ainsi différents agents (processus) communiquent et se synchronisent par l'intermédiaire d'un ensemble de contraintes global, où l'information s'accumule de manière monotone au fil du temps. Chaque agent peut soit ajouter une nouvelle contrainte (opération *Tell*), soit demander si une certaine contrainte est vraie dans (i.e. impliquée par) l'ensemble global (opération *Ask*). L'opération *Tell* correspond à l'ajout classique d'une contrainte en PLC. La synchronisation s'effectue par une opération *Ask bloquante* : si l'on ne peut décider si la contrainte est vraie ou fausse dans l'ensemble global de contraintes (i.e. il faut attendre d'avoir plus d'informations), l'agent est suspendu jusqu'à ce que d'autres agents ajoutent (*Tell*) les contraintes suffisantes pour décider.

Notre travail dans le domaine des langages concurrents avec contraintes se porte sur deux axes :

- Le développement de méthodes de preuves pour les langages CC, extension de travaux précédents sur les méthodes de validation des programmes logiques utilisant des techniques venant des grammaires attribuées.
- L'extension du cadre CC aux systèmes distribués, par la définition de nouvelles primitives de communication (par message) et d'un cadre sémantique adéquat.

3.1.1 CCC(;)

Sur le premier aspect l'effort a porté sur trois points principaux: la définition de méthodes de preuve de propriétés d'arbres attribués incomplets, la définition du paradigme CCC(;), et l'application de ces méthodes à l'analyse de propriétés de sûreté de programmes CCC(;). Ces travaux sont développés dans la thèse de Pascal Chambre et en partie synthétisés dans [178].

L'originalité des résultats obtenus repose sur une notion d'"assertion connectée". Supposant donnée une grammaire attribuée, un langage d'assertions sur les attributs et des dépendances entre les occurrences d'assertions dans les règles de la grammaire, les (occurrences d') assertions connectées dans un arbre donné sont celles qui ne dépendent (et ne sont pas) d'aucune assertion synthétisée d'une feuille indéfinie. La méthode mise au point assure que dans tout arbre partiel valide de la grammaire attribuée qui satisfait ses assertions héritées à la racine, les assertions connectées sont valides. On obtient

alors des conditions suffisantes (essentiellement décidables) pour que tout arbre valide ait au moins une feuille avec une assertion connectée.

Le paradigme CCC(;) est une spécialisation en forme clausale, du paradigme CC, nécessaire pour arriver à décrire des phénomènes de synchronisation dans les langages avec contraintes et bénéficier de l'approche clausale de la programmation en logique. Cette spécialisation ne détruit pas le pouvoir d'expression des CC's auxquels elle est pratiquement équivalente (au sens de la sémantique dénotationnelle). Le point virgule symbolise la séquentialité, codée dans le langage, des contraintes à satisfaire avant que soient lancés les processus descendants, en parallèle.

Enfin une sémantique de type "arbre de comportement", calquée sur la notion d'arbre de preuve développée pour la programmation en logique a été introduite. Elle permet de mettre en évidence de manière "naturelle" à la fois les propriétés déclaratives (typage, correction partielle, complétude) de solutions attendues, comme des propriétés de comportement (absence de "deadlock") et définir des méthodes de preuves pour ces propriétés. La dernière s'obtient en particulier en montrant, par application directe du résultat mentionné pour les grammaires attribuées, que tout arbre de comportement a au moins une feuille avec une assertion (supposée "débloquante") connectée valide.

3.1.2 CC distribué

Le paradigme CC classique est basé sur l'idée d'un ensemble de contraintes (*store*) global et est donc adapté aux architectures de type mémoire partagée mais pas aux systèmes distribués. Nous avons donc développé un cadre sémantique intégrant les notions de sites distribués, de *stores* locaux, de canaux de communication et de communication par message, dans le but d'étendre le cadre CC aux réseaux. Ce modèle, développé dans la thèse de Jean-Hugues Réty, se rapproche aussi d'autres formalismes plus classiques développés pour la concurrence, en particulier le π -calcul proposé par Milner.

3.2 Aspects domaines de contraintes et algorithmes de résolution

3.2.1 Contraintes sur les domaines finis

Participants : Philippe Codognet, Daniel Diaz, Yan Georget, Salvador Abreu

Parmi les différents domaines de calcul étudiés en PLC, les domaines finis semblent être les plus prometteurs, car ils sont très utiles dans de nombreuses applications industrielles comme par exemple les problèmes combinatoires, l'ordonnancement, cutting-stock, la simulation de circuits, le diagnostic, l'aide à la décision ou même les problèmes booléens.

Nous poursuivons nos recherches dans ce domaine en continuant le développement du système `clp(FD)` [170], qui utilise une approche "RISC" de décomposition des contraintes complexes en contraintes primitives de bas-niveau, à la fois simple et efficace. Le prototype `clp(FD)` implanté à l'INRIA est disponible par ftp anonyme, et est actuellement diffusé auprès d'environ 900 centres de recherche, universités (pour l'enseignement) et centres industriels (pour des usages de prototypage). Nos recherches actuelles portent sur le développement d'une nouvelle machine abstraite qui permettra une compilation plus rapide (la version actuelle compile le code Prolog et les contraintes sur les domaines finis en code C, qui est ensuite lui-même compilé), ainsi que la compilation vers le byte-code Java, pour une plus grande portabilité et une diffusion à l'intérieur d'applets sur internet.

Nous étudions aussi l'intégration de contraintes directement en Java, en dehors donc du cadre de la programmation en logique. L'idée est de fournir un solveur de contraintes modifiable et modulaire où l'utilisateur pourra définir ses propres contraintes. Un premier prototype a été développé par Salvador Abreu durant son séjour à l'INRIA.

Un autre axe de recherche que nous avons récemment entamé en collaboration avec l'université de Pise (Francesca Rossi, Ugo Montanari) concerne l'extension du cadre CLP pour le traitement de la non-monotonie et de la réactivité. En effet, de nombreuses applications nécessitent un certain degré

d'interaction, et donc de réaction, entre l'utilisateur et le système ou entre différentes parties du système, et il faut donc fournir des primitives pour un modèle sémantique et un modèle d'exécution pour gérer ces modifications incrémentalement et dynamiquement. Un premier résultat est un algorithme incrémental pour la rétraction de contrainte [179], dans le cadre des contraintes sur les domaines finis et plus particulièrement de l'approche RISC développée dans le système `clp(FD)`. Celui-ci nous servira bien évidemment de base pour aboutir à un langage intégrant des primitives interactives permettant la rétraction de contraintes de manière efficace.

3.2.2 Contraintes Numériques

Participants : Frédéric Benhamou, Laurent Granvilliers, Nicolas Romero, Arnaud Legrand

Nous nous sommes principalement intéressés aux problèmes suivants:

1. Résolution de contraintes hétérogènes [177, 176, 192, 191]
2. Résolution de contraintes numériques non-linéaires [175, 174, 169]

En ce qui concerne la résolution de contraintes hétérogènes, les motivations viennent de certaines remarques concernant les produits et prototypes existants et utilisés pour réaliser des applications. En effet, le processus de traitement du store à chaque étape de calcul (résolution) s'est notablement complexifié par rapport aux travaux fondateurs sur Prolog III ou CLP(R). Entre autres, les notions de coopération de solveurs de contraintes, de consistance locale et de manipulations symboliques sont désormais la base de la plupart des implantations réalistes. Pour rendre compte du comportement de ces processus de résolution, il semble donc nécessaire de proposer de nouvelles abstractions caractérisant les résultats du calcul.

C'est l'un des objectifs de nos travaux en cours où sont étudiés de tels comportements en termes de calculs de points fixes et en termes de consistance locale. Une implantation d'un méta-compilateur de CLP s'appuyant sur ces principes est également en cours de réalisation.

Notre second axe de recherche principal concerne la résolution de contraintes non-linéaires sur les nombres réels. Après avoir étudié les principes et le comportement expérimental des contraintes sur les intervalles telles qu'elles sont utilisées en CLP(BNR) ou en Prolog IV, nous nous sommes penchés sur des algorithmes spécialisés pour la résolution de contraintes polynômiales. Nos travaux les plus récents dans ce domaine concernent l'étude sémantique de langages tels que Newton et la combinaison d'opérateurs complexes et de techniques de calcul formel. La principale de ces techniques peut se résumer grossièrement à la production de contraintes redondantes par un calcul incomplet des combinaisons linéaires de polynômes qui permettent de calculer une base de Gröbner.

3.3 Aspects environnements

Une caractéristique importante de la PLC est de permettre une approche uniforme du langage et de son environnement. Et effet, les domaines de contraintes peuvent être utilisés à la fois pour décrire le programme et ses propriétés. L'aspect très "déclaratif" de la programmation logique avec contraintes permet de mettre en pratique des idées développées dans le cadre de la programmation en logique traditionnelle et sa réalisation "Prolog", qui n'avaient pu trouver d'applications réelles, faute d'être suffisamment déclaratifs.

Notre approche vise alors à surmonter deux difficultés majeures: à savoir, d'une part, comment consigner dans un document unique l'ensemble de l'expérience de développement et de maintenance acquise pour une application donnée (expérience résultant de sessions d'essais et de tentatives de validation); et d'autre part, offrir la possibilité de dominer la complexité des calculs développés dans des systèmes de contraintes hétérogènes avec des outils de mise au point adaptés. Deux projets plus particulièrement visent à explorer et proposer des solutions: le projet européen DiSciPI décrit en section 5.2.1, le projet décrit ci-dessous, ainsi que les travaux plus théoriques visant à enrichir ces environnements.

3.3.1 Hyper Programmation Logique

Participants : Pierre Deransart, Ali Ed-Dbali, José Siqueira, Patrick Parot

Le projet “HyperPro” développé en collaboration avec l’Université de Belo Horizonte au Brésil vise à intégrer la puissance de la documentation hypertexte avec les méthodes de développement de programmes avec contraintes mises au point dans notre équipe.

Le but est de développer un système expérimental dans lequel un programme (suite de clauses) est développé en même temps que les commentaires de manière interactive; suite à quoi le système permet de produire un texte documenté (introduction, clauses avec commentaires et références, et index). Le système est construit à partir du système Thot développé à l’INRIA (projet OPERA).

Le système est alors étendu pour se coordonner avec des méthodes de vérification, débogage et analyse de performances afin d’obtenir des outils de mise au point puissants et adaptés à la PLC.

Les premiers résultats sont exposés dans [181, 182].

3.3.2 Validation

Participants : Pascal Chambre, Pierre Deransart, Gérard Ferrand, Bernard Malfon

Outre les travaux concernant la validation de CCC(;) (section 3.1.1), nos études sur la formalisation de la sémantique déclarative des programmes logiques adaptée à une méthodologie de *preuve* de programmes a été poursuivie. Nous avons pu obtenir une formulation et des justifications claires et concises de nos résultats en privilégiant une approche inductive du problème, aussi bien pour la sémantique de Fitting que pour la sémantique dite “bien fondée”, bien connue dans la théorie des *bases de données déductives*.

Notons au passage que Sophie Renault et Bruno Dumant ont soutenu leur thèse en cours d’année sur des travaux détaillés dans les rapports d’activités des années précédentes [168, 166].

3.3.3 Modularité

Participants : Gérard Ferrand, Arnaud Lallouet, Bernard Malfon

Il est bien connu que les aspects de modularité sont encore embryonnaires en programmation en logique et constituent son point faible. La PLC risque dans un proche avenir de souffrir de la même faiblesse. En PLC, comme dans les autres paradigmes de programmation, des concepts de modularité sont donc nécessaires pour faciliter le développement et la maintenance d’applications de tailles raisonnables.

Nous avons étudié [184, 183, 167] une extension des techniques de preuve de propriétés de programmes logiques dans le cadre compositionnel. Cette approche permet en particulier de modéliser les systèmes existants (comme le système de modules de Gödel) et peut s’étendre à des systèmes beaucoup plus riches incluant négation et contraintes.

3.3.4 Diagnostic d’erreur

Participants : Alexandre Tessier, Michel Bergère, Pierre Deransart, Gérard Ferrand, François Le Berre, Bernard Malfon

Une des caractéristiques essentielles de la PLC est d’une part le caractère déclaratif de l’erreur (on peut “reconnaître” une portion erronée du programme hors de son contexte d’exécution), et la possibilité de s’aider, dans le diagnostic, d’assertions, décrivant des propriétés attendues.

Nous avons déjà adapté et étendu à la PLC l’approche suivie en programmation en logique traditionnelle pour reformuler la sémantique des programmes logiques en termes de “squelettes de preuve”. Nous avons commencé à utiliser ces résultats pour définir des méthodes de diagnostic déclaratif (recherche d’erreurs à partir de symptômes d’erreurs) qui étendent aux programmes logiques avec contraintes les méthodes existant déjà pour les programmes logiques.

Nous avons étudié les deux sortes de diagnostic liés aux deux sortes de symptômes : résultat faux et résultat manquant [186, 185, 172, 194]. En particulier pour le diagnostic associé à cette deuxième sorte de symptôme (appelé diagnostic d'insuffisance) nous avons non seulement étendu à la PLC des algorithmes de diagnostic d'insuffisance connus en PL mais aussi contribué à éclaircir les justifications de ces algorithmes. Ces éclaircissements devraient en particulier permettre d'étudier les possibilités d'optimisations de ce genre d'algorithme [189, 187, 188, 193].

Ces travaux sont essentiels pour le projet DiSciPl (section 5.2.1).

4 Action industrielles

4.1 PrologIA

Participants : Frédéric Benhamou

Collaboration avec la société PrologIA et participation à l'élaboration du produit commercial Prolog IV. Dans cette dernière année de développement du produit, cette collaboration s'est essentiellement portée sur la finalisation d'un certain nombre de choix de "design" pour le langage ainsi que sur une participation à l'élaboration des manuels de référence et d'utilisation.

4.2 Standard Prolog

Participants : Pierre Deransart, Ali Ed-Dbali

Concernant les travaux ISO sur la standardisation du langage Prolog, rappelons que le standard a été divisé en deux items: partie I (general core) et partie II (modules). La partie I est achevée depuis Juin 95, la partie II a été approuvée en Janvier 96 (SC22 N2046). La première partie est maintenant complétée par un livre de référence [165].

4.3 Rank Xerox, Grenoble

Participants : Philippe Codognet

Une collaboration avec le centre de recherche de Rank Xerox à Grenoble est en train de se développer autour du thème "Contraintes et Coordination", c'est à dire de l'utilisation de techniques de résolution de contraintes pour la synchronisation d'agents distribués. Une proposition Esprit LTR (CONCORDE) a été faite sur ce sujet (Juin 96) mais elle n'a pas été retenue. Nous comptons refaire une proposition sur ce sujet en 1997.

5 Actions nationales et internationales

5.1 Activités nationales

Philippe Codognet, Pierre Deransart et Gérard Ferrand ont fait partie du comité de programme de JFPLC'96.

Pierre Deransart est Président élu de l'AFPL (Association Française pour la Programmation en Logique). L'INRIA apporte son soutien logistique en assurant une partie du secrétariat administratif (Ghislaine Le Corre jusqu'en Juillet).

G. FERRAND est Directeur du LIFO (Laboratoire d'Informatique Fondamentale d'Orléans, EA1637).

5.2 Projets internationaux

5.2.1 Projet Européen DiSCiPI

Participants : Pierre Deransart, Frédéric Benhamou, Philippe Codognet, Gérard Ferrand, Laurent Granvilliers, Alexandre Tessier

Le projet Esprit (reactive LTR, no 22523) "Debuging System for Constraint Programming (DiSCiPI)" a commencé le 1^{er} octobre 1996. Sa durée est de 2 ans et demi. Il regroupe 4 partenaires industriels (COSYTEC, PrologIA, ICON, OM Partners) et 4 partenaires académiques (INRIA-Rocquencourt avec l'Université d'Orléans, au sein de LOCO, UPM-Madrid, Université de Linköping, Université de Bristol).

Le but du projet est la définition de méthodes et d'outils de mise au point pour la programmation avec contraintes, ainsi que leur insertion dans les produits commerciaux CHIP V5 et Prolog IV.

L'approche proposée repose sur l'intégration de différentes techniques de validation développées pour la programmation en logique (débogage déclaratif, analyse et génération par interprétation abstraite, les preuves par méthodes assertionnelles et la visualisation graphique de l'espace de recherche), ainsi qu'une vision générale abstraite des solveurs de contraintes.

5.2.2 European Network of Excellence in Computational Logic et Pilot Task Force

Participants : Pierre Deransart

Pierre Deransart a été membre élu du conseil exécutif du NOE Compulog II. Il est également responsable du nœud "programmation en logique à Rocquencourt".

Il assure avec Jan Komorowski de l'Université de Trondheim (Norvège) l'organisation d'une "Pilot Task Force" réunissant les nœuds industriels et académiques du réseau dont l'objectif est de réaliser un état de l'art et des applications industrielles pour aboutir à des propositions de projets dans le domaine des environnements pour la programmation en logique. Un premier succès réside dans l'obtention du projet européen DiSCiPI (cf. ci-dessus).

5.2.3 Portugal

Participants : Pierre Deransart, Philippe Codognet, Gérard Ferrand

Le nouveau réseau formation-recherche franco-portugais a été mis en place en février 96 avec les universités de Lisbonne et Braga, sur le thème programmation en logique avec contraintes, preuves et langue naturelle.

Il réunit les projets LOCO (Pierre Deransart, coordinateur) et ATOLL à l'INRIA, l'université d'Orléans (Gérard Ferrand et Jean-Claude Bassano) et l'IRIT (Patrick Saint-Dizier), avec les universités de Lisbonne (UNL, Luis Moniz Pereira, coordinateur) et Braga (Esgalhado Valencia). Il privilégie les thèses en cotutelle.

5.2.4 Europe de l'est et Russie

Participants : Philippe Codognet, Frédéric Benhamou

Le projet INTAS Call93, Ref.No. INTAS-93-2758 "Improving Efficiency of logic programs" a commencé le 1.12.94 et dure jusqu'au 1.12.96. Il regroupe en particulier les Universités de Leuven, Bristol, Rome, l'INRIA, l'académie des Sciences de Moscou et Saint Petersburg et l'Université d'Irkutsk en Russie. Philippe Codognet participe pour l'INRIA dans cette activité.

Frédéric Benhamou est impliqué dans l'élaboration d'un projet de recherche commun avec l'institut d'Intelligence Artificielle de l'Académie des Sciences de Novosibirsk sur la programmation par contraintes.

L'équipe d'Intelligence Artificielle, dirigée par Tania Yakhno, développe depuis de nombreuses années une série de projets autour de solveurs de contraintes numériques. L'une de leurs réalisations, Unicalc, vient d'ailleurs de remporter un marché auprès d'une grande entreprise française. Dans le cadre de l'Institut Liapounov, liant l'Université de Moscou et l'Inria, nous sommes en train de mettre au point un projet de recherche commun sur un certain nombre de thèmes, parmi lesquels:

- résolution et optimisation de contraintes numériques non-linéaires
- problèmes combinatoires
- applications de la programmation par contraintes

L'équipe de Novosibirsk est également engagée dans un projet commun avec IC park (Imperial College, Londres).

5.2.5 Brésil

Participants : Pierre Deransart, Ali Ed-Dbali, José Siqueira, Patrick Parot

Pierre Deransart est responsable pour l'INRIA des relations scientifiques avec le Brésil.

Nous avons un projet en cours avec l'Université Fédérale de Minas Gerais, décrit en section 3.3.1 (Hyper Programming Logique). Il est financé partiellement par les crédits de coopération franco-brésilienne (MAE), le MEN brésilien (programme RHAE) et le CNPq.

5.2.6 Etats Unis, contrat NSF

Participants : Pierre Deransart, Philippe Codognet, Gérard Ferrand, Arnaud Lallouet, Bernard Malfon

Projet avec ATOLL (A.HUI BON HOA) et l'université de Pennsylvanie (D. MILLER), sur le thème: "Structuring of Proof Search in the Logic Programming Paradigm", commencé en mai 95 pour 3 ans. Un séminaire commun a eu lieu à Philadelphie en décembre 95 portant sur les langages logiques et la modularité.

5.3 Comités de programme de conférences internationales

Frédéric Benhamou a fait partie du comité de programme de la "second Andrei Ershov conference on Informatics", Novosibirsk, Russie, juin.

Philippe Codognet a été membre des comités de programme de CP'96 (2nd International conference on principle and practice of Constraint Programming), Boston, USA, août et JICSLP'96 (3rd Joint International Conference and Symposium on Logic Programming), Bonn, Allemagne, septembre.

Philippe Codognet est membre du comité de rédaction de la revue "CONSTRAINTS" (Kluwer Academic Publishers).

Pierre Deransart a fait partie des comités de programme de PAP'96 (Practical Prolog Applications), Londres, GB, avril, ALP'96 (Algebraic and Logic Programming), Aix la Chappelle, Allemagne, septembre, session Posters at JICSLP'96, Bonn, Allemagne, septembre, laquelle a donné lieu à un processus de sélection indépendant.

6 Diffusion des résultats

6.1 Actions d'enseignement

F. Benhamou, Ph. Codognet, P. Deransart et G. Ferrand participent à la formation doctorale en Informatique de l'Université d'Orléans et en particulier au DEA.

F. Benhamou est également intervenu dans le cadre du DESS IMOI (Ingénierie Mathématique et Outils Informatiques) de l'Université d'Orléans pour un cours sur le langage Prolog III.

Ph. Codognet a donné en Avril/Mai 1996 un cours sur la Programmation par Contraintes au programme de formation Ph.D. de l'Université de Pise, Italie.

Ph. Codognet participe avec F. Fages au cours de Programmation Logique avec Contraintes du DEA SPP de l'université de Paris VII pour l'année 1996/97.

6.2 Participation à des colloques

Des membres de l'équipe ont participé à des conférences et workshops; on se reportera à la bibliographie pour en avoir la liste.

6.3 Conférences invitées, tutoriels, cours, etc.

Frédéric Benhamou co-édite avec Pascal Van Hentenryck un numéro spécial de la revue CONSTRAINTS sur les contraintes d'intervalle. Il a également été invité au séminaire Contraintes de l'école normale supérieure de Paris.

A l'occasion de la "second Andrei Ershov conference on Informatics", Frédéric Benhamou a donné deux séminaires à l'institut d'Intelligence Artificielle de l'académie des Sciences de Novosibirsk sur la programmation par contraintes. Il a été également invité à la conférence IMACS ACA'96

Ph. Codognet a donné des séminaires à l'université de Pise, au C.W.I. (Amsterdam), à l'I.M.A.G. de Grenoble, et à la faculté polytechnique de Mons à l'invitation de la société Belge de recherche opérationnelle.

Ph. Codognet a été invité au workshop "Strategic Directions in Computing Research" organisé par l'ACM en Juin 1996 au MIT, Boston, USA, dans le groupe de travail "Constraint Programming".

Ph. Codognet édite un numéro spécial de la revue "Science of Computer Programming" sur la programmation concurrente avec contraintes.

Pierre Deransart a donné un tutoriel à PAP'96 sur "standard Prolog" à l'occasion du lancement du livre [165].

Gérard Ferrand a été invité au workshop "Program Correctness: Abstract Interpretation v.s. classical Verification Method" à Venise, 10-12 juin 1996.

Ph. Codognet a été rapporteur des thèses de M. Jampel (City University, Londres), E. Morel (Université de Grenoble) et L. Ciortuz (université de Lille).

P. Deransart a été rapporteur des thèses de D. Binks (Université de Bristol) et M. NGomo (Université de Rouen).

6.4 Diffusion de logiciels

Le système de programmation par contraintes `clp` (FD) et le compilateur Prolog `wamcc` sont diffusés par FTP. Environ 900 utilisateurs académiques ou industriels ont pris le premier logiciel, et 1200 le second. On peut citer parmi ceux-ci le CENA, Dassault-Aviation, Thomson-CSF, la NASA, SICS, DFKI.

7 Publications

Livres et monographies

- [165] P. DERANSART, A. ED-DBALI, L. CERVONI, *Prolog, The Standard; Reference Manual*, Springer Verlag, Avril 1996.

Thèses

- [166] B. DUMANT, *Analyses statiques en Programmation Logique avec Contraintes*, thèse de doctorat, Université de Paris 7, Mai 1996.
- [167] A. LALLOUET, *Modularité, validation et parallélisme de données en programmation logique*, thèse de doctorat, Université d'Orléans, 1996.
- [168] S. RENAULT, *Vérification de programmes normaux*, thèse de doctorat, Université d'Orléans, Juin 1996.

Articles et chapitres de livre

- [169] F. BENHAMOU, L. GRANVILLIERS, «A Fixpoint Approximate for Cooperating Numerical Solvers», *Reliable Computing, Proceedings of INTERVAL'96*, 1996, (Supplément).
- [170] P. CODOGNET, D. DIAZ, «Compiling Constraint in clp(FD)», *Journal of Logic Programming* 27 (3), 1996.
- [171] P. CODOGNET, D. DIAZ, «A Simple and Efficient Boolean Solver for Constraint Logic Programming», *Journal of Automated Reasoning* 17, 1996.
- [172] G. FERRAND, A. TESSIER, «Declarative Debugging», *The Newsletter of the European Network in Computational Logic*, Décembre 1996, COMPULOG NET.
- [173] P. V. HENTENRYCK, V. SARASWAT, P. CODOGNET, ET AL., «Constraint Programming», *ACM Computing Surveys*, Décembre 1996.

Communications à des congrès, colloques, etc.

- [174] F. BENHAMOU, L. GRANVILLIERS, «Application of Gröbner Bases for Numerical Constraint Solving», in: *Proceedings of IMACS ACA'96*, Linz, Austria, 1996.
- [175] F. BENHAMOU, L. GRANVILLIERS, «Combining Local Consistency, Symbolic Rewriting and Interval Methods», in: *Proceedings of AISMC-3, LNCS, 1138*, Springer-Verlag, p. 144–159, Steyr, Austria, 1996.
- [176] F. BENHAMOU, «An Overview of Prolog IV», in: *9th Symposium and Exhibition on Industrial Applications of Prolog (INAP'96)*, Tokyo, Japan, 1996.
- [177] F. BENHAMOU, «Heterogeneous Constraint Solving», in: *Proceedings of the fifth International Conference on Algebraic and Logic Programming (ALP'96), LNCS 1139*, Springer-Verlag, p. 62–76, Aachen, Germany, 1996.
- [178] P. CHAMBRE, P. DERANSART, J. MAŁUSZYŃSKI, «A Proof Method for Safety Properties of Clausal Concurrent Constraint Programs», in: *JICSLP'96 post conference workshop on Verification and Analysis of Logic Programs*, F. S. de Boer, M. Gabbriellini (éd.), TR-96-31, Université de Pise, DCS, Septembre 1996.
- [179] P. CODOGNET, D. DIAZ, F. ROSSI, «Constraint Retraction in FD», in: *16th Conference on Foundations of Software Technology and Theoretical Computer Science*, V. Chandru (éd.), Spinger Verlag, Hyderabad, Inde, Décembre 1996.
- [180] P. CODOGNET, «Finite Domains in CLP (Tutorial)», in: *9th Symposium and Exhibition on Industrial Applications of Prolog (INAP'96)*, Tokyo, Japan, 1996.

- [181] P. DERANSART, R. DA SILVA BIGONHA, P. PAROT, M. A. DA SILVA BIGONHA, J. DE SIQUEIRA, «A Literate Logic Programming System», in: *I^eir^o Simpósio Brasileiro de Linguagens de Programação, SBLP'96*, R. S. Bigonha (éd.), ANAIS, Belo Horizonte, MG, Brazil, Septembre 1996.
- [182] P. DERANSART, R. DA SILVA BIGONHA, P. PAROT, M. A. DA SILVA BIGONHA, J. DE SIQUEIRA, «A Hypertext Based Environment to Write Literate Programs», in: *Proceedings of the Poster Session at JICSLP'96*, N. Fuchs, U. Geske (éd.), 296, GMD- Forschungszentrum, Informationstechnik GMBH, Berlin, Septembre 1996.
- [183] A. LALLOUET, L. MARTIN, «Compositional validation in Inductive Logic Programming», in: *International Symposium on Methodologies for Intelligent Systems*, Zakopane, Poland, 9-13 Juin 1996. Poster session.
- [184] A. LALLOUET, «Une notion de composant logiciel en Programmation Logique», in: *Journées Francophones de la Programmation Logique et de la Programmation par Contraintes, JFPLC*, J.-L. Imbert (éd.), HERMES, p. 83–98, Clermont-Ferrand, France, 5-7 Juin 1996.
- [185] F. LE BERRE, A. TESSIER, «Declarative Incorrectness Diagnosis in Constraint Logic Programming», in: *Joint Conference on Declarative Programming, GULP/PRODE'96*, P. Lucio, M. Martelli, M. Navarro (éd.), p. 379–391, 1996.
- [186] B. MALFON, «Quelques modèles quadrivalués du complété, application à des notions déclaratives de symptôme et d'erreur», in: *Journées Francophones de la Programmation Logique et de la Programmation par Contraintes, JFPLC*, J.-L. Imbert (éd.), HERMES, p. 99–113, Clermont-Ferrand, France, 5-7 Juin 1996.
- [187] A. TESSIER, «Declarative Debugging in Constraint Logic Programming», in: *International Conference on Algebraic and Logic Programming (Poster session)*, M. Hanus (éd.), Technical Report 96/9, Aachener Informatik – Berichte, p. 38–49, 1996.
- [188] A. TESSIER, «Declarative Debugging in Constraint Logic Programming», in: *Asian Computing Science Conference*, J. Jaffar (éd.), *Lecture Notes in Computer Science, 1023*, Springer-Verlag, 1996.
- [189] A. TESSIER, «Diagnostic déclaratif d'insuffisance en programmation logique avec contraintes», in: *Journées Francophones de Programmation Logique et Programmation par Contraintes*, J.-L. Imbert (éd.), HERMES, p. 65–82, 1996.
- [190] A. TESSIER, «Une caractérisation des arbres SLD en programmation logique avec contraintes», in: *Pôle Contraintes et Programmation Logique*, Journées du GDR Programmation du CNRS, 1996.

Rapports de recherche et publications internes

- [191] A. LEGRAND, *Extension de la WAM aux systèmes de contraintes hétérogènes*, Mémoire, Université d'Orléans, 1996.
- [192] N. ROMERO, *Etude d'un meta-compileur de langages CLP*, Mémoire, Université d'Orléans, 1996.
- [193] A. TESSIER, «Declarative Debugging in Constraint Logic Programming: the Cover Relation», *rapport de recherche n°96/09*, LIFO, Université d'Orléans, 1996.
- [194] A. TESSIER, «Diagnostic Déclaratif d'Insuffisance en Programmation Logique avec Contraintes», *rapport de recherche n°96/04*, LIFO, Université d'Orléans, 1996.

8 Abstract

For the year 1996, the LOCO project can be abstracted with two main facts :

1. LOCO project creation

The 1996 October 21st, LOCO switched from action to project status. Note that LOCO is a common project between INRIA-Rocquencourt and Orléans University. LOCO's research follows three directions: constraints resolution (finite domain, constraints on real intervals), languages aspects (particularly concurrent languages with constraints) and programming environment (debugging of logic programs with constraints mainly).

2. DiSCiPl European project creation

The Esprit project (reactive LTR, no 22523) "Debuging System for Constraint Programming (DiSCiPl)" begun the 1st of October 1996. It's duration will be 2 years and half. It contains 4 industrial partners (COSYTEC, PrologIA, ICON, OM Partners) and 4 academic partners (INRIA-Rocquencourt with Orléans University inside the LOCO team,, UPM-Madrid, Linköping University, Bristol University).

