

# *Projet LOCO*

*Programmation en LOGique avec COntraintes*

*Rocquencourt*

THÈME 2A

*R* *apport*  
*d'Activité*

1999



## Table des matières

<b>1</b>	<b>Composition de l'équipe</b>	<b>3</b>
<b>2</b>	<b>Fondements scientifiques</b>	<b>5</b>
2.1	Aspects langages . . . . .	5
2.2	Aspects domaines de contraintes et algorithmes de résolution . . . . .	5
2.3	Aspects environnements . . . . .	6
<b>3</b>	<b>Domaines d'applications</b>	<b>8</b>
<b>4</b>	<b>Logiciels</b>	<b>8</b>
4.1	GNU Prolog . . . . .	8
4.2	CLP(FD,S) . . . . .	8
4.3	DecLIC . . . . .	9
4.4	ISO Prolog . . . . .	9
<b>5</b>	<b>Résultats nouveaux</b>	<b>9</b>
5.1	Aspects langages . . . . .	9
5.1.1	Modélisation des solveurs en terme de règles . . . . .	9
5.1.2	Typage pour la PLC . . . . .	10
5.1.3	Programmation par contraintes et systèmes multi-agents . . . . .	10
5.1.4	Contraintes et conception de mondes virtuels . . . . .	11
5.1.5	Système GNU Prolog . . . . .	11
5.1.6	Placement de caméras . . . . .	12
5.1.7	Propagation de contraintes parallèle . . . . .	12
5.1.8	Étude du parallélisme de données en Programmation en Logique . . . . .	12
5.1.9	Arithmétique d'intervalles . . . . .	13
5.2	Aspects domaines de contraintes et algorithmes de résolution . . . . .	13
5.2.1	Rétraction de contraintes et contraintes floues . . . . .	13
5.2.2	Extensions de la box consistance . . . . .	14
5.2.3	Résolveurs symboliques/numériques . . . . .	14
5.2.4	Analyse par intervalles de systèmes linéaires . . . . .	15
5.2.5	Optimisation locale/globale . . . . .	15
5.2.6	Contraintes vectorielles: le domaine $X^*$ . . . . .	16
5.3	Aspects environnements . . . . .	17
5.3.1	Diagnostic déclaratif . . . . .	17
5.3.2	Simplification de contraintes sur les domaines finis . . . . .	17
5.3.3	Débogage avec S-box . . . . .	17
5.3.4	Interface graphique pour le débogage . . . . .	18
5.3.5	ISO Prolog . . . . .	18
5.3.6	Indexation de pages WEB . . . . .	19

---

<b>6 Contrats industriels (nationaux, européens et internationaux)</b>	<b>19</b>
6.1 PrologIA . . . . .	19
6.2 COSYTEC . . . . .	20
<b>7 Actions régionales, nationales et internationales</b>	<b>20</b>
7.1 Actions nationales . . . . .	20
7.2 Actions européennes . . . . .	20
7.2.1 Projet Esprit DiSCiPl . . . . .	20
7.2.2 European Network of Excellence in Computational Logic . . . . .	21
7.2.3 Europe de l'est et Russie . . . . .	21
7.2.4 ERCIM working group on Constraints . . . . .	21
7.2.5 Portugal . . . . .	21
7.3 Relations internationales . . . . .	22
7.3.1 Brésil . . . . .	22
7.4 Comités de programme de conférences internationales . . . . .	22
<b>8 Diffusion de résultats</b>	<b>22</b>
8.1 Animation de la communauté scientifique . . . . .	22
8.2 Enseignement universitaire . . . . .	23
8.3 Participation à des colloques, séminaire, invitations . . . . .	23
<b>9 Bibliographie</b>	<b>24</b>

# 1 Composition de l'équipe

## Responsables scientifiques

Philippe Codognet [professeur, Université de Paris 6]

Gérard Ferrand [professeur, Université d'Orléans]

## Responsable permanent

Pierre Deransart [DR, INRIA]

## Secrétariat

Josy Baron [AST, partagée avec ATOLL]

## Personnel INRIA

François Fages [DR, à l'INRIA depuis le 1.10]

## Collaborateurs extérieurs

Frédéric Benhamou [professeur, Université de Nantes]

Michel Bergère [maître de Conférence, Université d'Orléans]

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

AbdelAli Ed-Dbali [maître de Conférence, Université d'Orléans]

Laurent Granvilliers [ATER à l'Université d'Orléans du 01.01 au 31.08; maître de Conférence à l'Université de Nantes à partir du 01.09]

Arnaud Lallouet [maître de Conférence, Université d'Orléans]

Éric Languéno [maître de Conférence, Université de Nantes]

Bernard Malfon [PRAG, IUFM Orléans-Tours, Université d'Orléans]

Alexandre Tessier [maître de Conférence, IUFM Orléans-Tours, Université d'Orléans]

## Chercheur invité

Jonathan Hodgson [professeur, Saint Joseph's University, Philadelphia, USA, jusqu'au 31.07]

**Chercheurs post-doctorants**

Jean-Marie Geffroy [boursier INRIA, depuis le 1.10]

Frédéric Goualard [bourse MENRT, Université d'Orléans, du 01.01 au 31.08; ATER à l'Université de Nantes à partir du 01.09]

Jan-Georg Smaus [boursier ERCIM, depuis le 1.10]

**Ingénieur expert**

Christophe Aillaud [INRIA, jusqu'au 31.03]

**Doctorants**

Julien Arsouze [ bourse MENRT à l'Université d'Orléans à partir du 01.09]

Lucas Bordeaux [ bourse MENRT à l'Université de Nantes à partir du 01.09]

Martine Ceberio [ bourse MENRT à l'Université de Nantes à partir du 01.09]

Marc Christie [bourse « ville de Nantes » à l'Université de Nantes]

Sorin Craciunescu [AMX, à l'INRIA depuis le 1.10]

Yan Georget [bourse DRET, INRIA, jusqu'au 31.08]

Luc Hernandez [bourse MENRT, université de Paris 6]

Franck Jardillier [doctorant à l'Université de Nantes à partir du 01.09]

Willy Lesaint [ bourse MENRT à l'Université d'Orléans à partir du 01.09]

Nadine Richard [bourse INRIA]

Nicolas Romero [bourse MENRT, Université d'Orléans]

Sylvain Soliman [IA Recherche, à l'INRIA depuis le 1.10]

**Stagiaires**

Omar Badda [stagiaire DEA, Université d'Orléans, du 1.04 au 30.09]

Olivier Ballereau [stagiaire DEA, Université d'Orléans, du 1.04 au 01.09]

Saïd Bekraoui [stagiaire DEA, Université d'Orléans, du 1.04 au 01.09]

## 2 Fondements scientifiques

### 2.1 Aspects langages

**Participants** : Julien Arsouze, Frédéric Benhamou, Michel Bergère, Lucas Bordeaux, Marc Christie, Philippe Codognet, Marco Comini, Pierre Deransart, Daniel Diaz, Gerard Ferrand, Frédéric Goualard, Laurent Granvilliers, Arnaud Lallouet, Willy Lesaint, Nicolas Romero, Nadine Richard, Jan-Georg Smaus, Alexandre Tessier.

**Mots clés** : programmation par contrainte, sémantique, implantation, compilation, satisfaction de contrainte, concurrence, programmation en logique, langage de programmation, modèle d'exécution, parallélisme de données, monde virtuel, agent intelligent, VRML, typage.

L'approche "langage" des contraintes peut se faire de deux manières opposées: partir d'un langage étendu avec des contraintes, ou partir de contraintes et rechercher le langage le mieux adapté pour les manipuler.

Plusieurs systèmes sont développés au sein de l'équipe, qui concernent différents domaines de contraintes, et illustrent ces différentes approches: GNU Prolog (domaines finis, Prolog étendu), Cosinus (contraintes numériques et langage facilitant la description de stratégies et coopération de solveurs), OpAc (bibliothèque C++ résolution de contraintes par propagation de domaines),  $clp(FD,S)$  (contraintes valuées dans demi anneaux, Prolog étendu), DP-log (manipulation de données vectorielles).

Notre équipe étudie plus particulièrement la programmation en logique avec contraintes (PLC) où les contraintes peuvent être structurées par des relations. La PLC a une sémantique déclarative logique (arbres de preuve) et une sémantique opérationnelle qui repose sur l'espace de recherche des solutions représenté par les "arbres SLD". Cette dualité permet la conception d'outils d'analyse (statique et dynamique) particulièrement puissants.

### 2.2 Aspects domaines de contraintes et algorithmes de résolution

**Participants** : Julien Arsouze, Olivier Ballereau, Martine Ceberio, Philippe Codognet, Daniel Diaz, Yan Georget, Frédéric Benhamou, Frédéric Goualard, Laurent Granvilliers, Arnaud Lallouet, Nicolas Romero.

**Mots clés** : programmation par contrainte, sémantique, implantation, compilation, satisfaction de contrainte, contrainte non-monotone, contrainte floue, modèle d'exécution, coopération de solveurs.

Parmi les différents domaines de contraintes, 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*<sup>1</sup>, la simulation de circuits, le diagnostic, l'aide à la décision ou même les problèmes booléens.

Certains autres domaines d'application, encore peu développés, imposent une modélisation, et donc un traitement, de contraintes non-linéaires sur des domaines continus (typiquement sur

---

1. Problèmes de découpe.

les réels). Parmi ces domaines, on peut citer l'économie, la chimie, l'ingénierie, la robotique ou la géométrie (synthèse d'image). Nous travaillons également sur les contraintes d'intervalles, à la base de méthodes qui semblent adaptées à ces types de problèmes.

Nos travaux se poursuivent sur les axes suivants:

- non-monotonie : définition et implantation d'une primitive incrémentale de rétraction de contrainte;
- contraintes floues : cadre général permettant d'associer un degré de possibilité à la satisfaction d'une contrainte (au lieu d'une simple valeur booléenne oui/non).
- contraintes d'intervalle : systèmes et langages pour la résolution et l'optimisation de contraintes non-linéaires sur les réels;
- coopération de solveurs.

### 2.3 Aspects environnements

**Participants** : Christophe Aillaud, Frédéric Benhamou, Michel Bergère, Saïd Bekraoui, Philippe Codognot, Pierre Deransart, Daniel Diaz, Ali Ed-Dbali, Gérard Ferrand, Frédéric Goualard, Jonathan Hogdson, Willy Lesaint, Bernard Malfon, Alexandre Tessier

**Mots clés** : spécification formelle, débogage, Prolog, contraintes, environnement de programmation, grammaire attribuée, interprétation abstraite, validation, débogage de performance, vérification de programme, simplification de contrainte.

Une caractéristique importante de la PLC est de permettre une approche uniforme du langage et de son environnement. En 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 qui n'avaient pu trouver d'applications réelles, le langage Prolog n'étant pas lui-même suffisamment déclaratif.

Par ailleurs les contraintes soulèvent des problèmes de débogage originaux qui impliquent différents aspects: correction, propagation et performance. Ces aspects peuvent être liés dans la mesure où, par exemple, une incomplétude du programme peut entraîner une recherche de solution dans un espace non borné, avec une complexité inattendue. Les méthodes de débogage traditionnelles se révèlent inefficaces car on doit tenir compte à la fois du non déterminisme dans la recherche de solution comme des méthodes de calcul particulières utilisées pour la résolution de contraintes.

La figure 1 montre les éléments fondamentaux sur lesquels reposent les méthodes de débogage utilisées ainsi que leur imbrication:

- *L'analyse statique* destinée à établir la correction du programme (débogage de correction) hors exécution, basée sur des méthodes de preuve utilisant l'interprétation abstraite



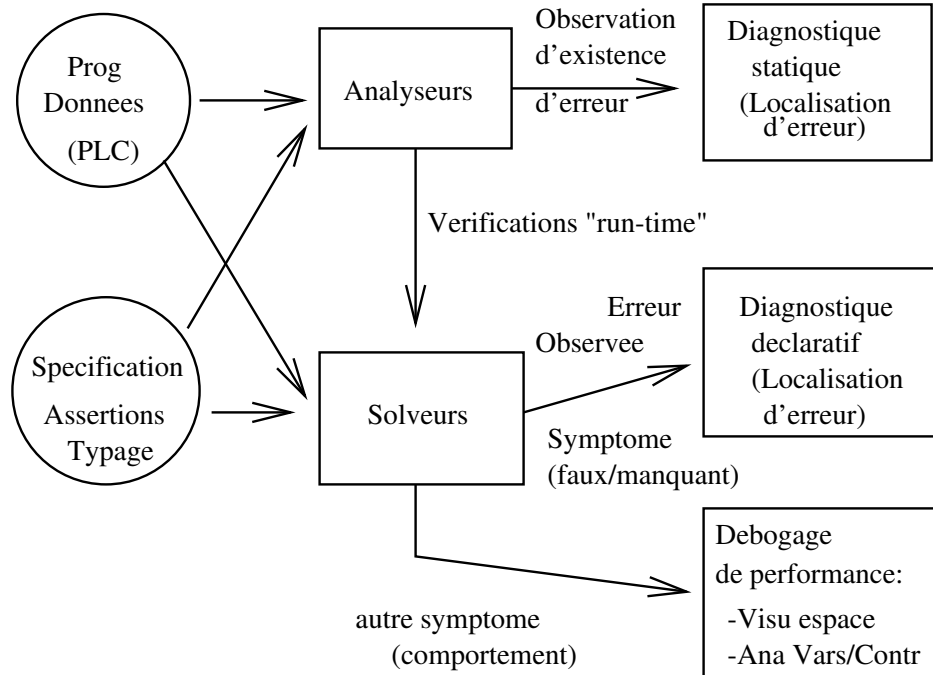


FIG. 1 – Différents aspects du débogage

[BDD<sup>+</sup>97] et ou les annotations [DM93] (dont les “types directionnels” sont une application [BM96]).

- *L’analyse d’une preuve*, là aussi destinée à établir la correction du programme (correction partielle et complétude), mais à partir d’une exécution. Le fait de connaître les détails d’un calcul (résolution et propagation) permet une localisation plus précise des erreurs. L’approche se fonde sur les travaux sur le “débogage déclaratif” [FT97], mais aussi sur les sémantiques de la résolution de contraintes [Ben97]. Un aspect important du débogage est le problème de représentation d’ensembles de contraintes. Ce problème, qui a déjà fait

- 
- [BDD<sup>+</sup>97] F. BUENO, P. DERANSART, W. DRABENT, G. FERRAND, M. HERMENEGILDO, J. MALUSZYNSKI, G. PUEBLA, «On the Role of Semantic Approximations in Validation and Diagnosis of Constraint Logic Programs», in: *Proc. of the 3rd. Int’l Workshop on Automated Debugging-AADEBUG’97*, U. of Linköping Press, p. 155–170, Linköping, Sweden, mai 1997, <ftp://clip.dia.fi.upm.es/pub/papers/DIRECTORY.html>.
- [DM93] P. DERANSART, J. MALUSZYŃSKI, *Grammatical View of Logic Programming*, The MIT Press, 1993.
- [BM96] J. BOYE, J. MALUSZYŃSKI, «Directional Types and the Annotation Method», *Journal of Logic Programming*, 1996, Aussi RR INRIA 2471, 1995.
- [FT97] G. FERRAND, A. TESSIER, «Positive and Negative Diagnosis for Constraint Logic Programs in terms of proof skeletons», in: *International Workshop on Automated Debugging*, U. of Linköping Press, Linköping, Sweden, mai 1997, <ftp://clip.dia.fi.upm.es/pub/papers/DIRECTORY.html>.
- [Ben97] F. BENHAMOU, «Contribution à l’étude de la résolution de contraintes», *habilitation*, LIFO, Université d’Orléans, janvier 1997.

l'objet de nombreux travaux en CLP( $\mathbb{R}$ ), a encore été très peu étudié pour CLP(FD).

- *L'analyse de l'espace de recherche et des contraintes.* La complexité des problèmes traités par contraintes nécessite des techniques donnant une vision pratique à la fois de l'espace de recherche et de l'utilisation des contraintes. Des outils adaptés doivent être développés, aussi généraux que possible. C'est le problème le plus difficile à résoudre afin de rendre la technologie P(L)C accessible. C'est un domaine extrêmement ouvert.

### 3 Domaines d'applications

De nombreuses applications peuvent s'exprimer simplement et efficacement dans le cadre PLC: problèmes combinatoires, ordonnancement, *cutting-stock*, simulation de circuits, diagnostic de panne, analyse financière, aide à la décision, etc... et ceci dans des domaines aussi variés que la santé, les télécommunications, les transports, le commerce, l'environnement, ou les domaines artistiques ...

Les applications 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.

## 4 Logiciels

### 4.1 GNU Prolog

- **Système:** GNU Prolog
- **Correspondant:** Daniel Diaz
- **URL:** <http://www.gnu.org/software/prolog>
- **Description:** GNU Prolog est un compilateur qui traduit du Prolog (conforme à la norme ISO, plus beaucoup d'extensions) et des contraintes sur les domaines finis en code natif. GNU Prolog est la version la plus récente des logiciels précédemment développés dans le projet sous les noms de CLP(FD) puis Calypso. La version courante de GNU Prolog est disponible sur les plateformes suivantes: sunos/sparc, solaris/sparc et linux/ix86, un portage pour Windows95/98/NT est en cours de développement. Ces versions sont distribuées par ftp sur le site de l'INRIA, miroir du site GNU. Une version pour MacOS/powerPC a été développée en dehors du projet.

### 4.2 CLP(FD,S)

- **système:** ,
- **correspondant:** Yan Georget,
- **URL:** [http://loco.inria.fr/~georget/clp\\_fds/clp\\_fds.html](http://loco.inria.fr/~georget/clp_fds/clp_fds.html)

- **description:** langage de PLC permettant la résolution de contraintes valuées dans des demi-anneaux.

### 4.3 DecLIC

- **Système:** DecLIC
- **Correspondant:** Frédéric Goualard
- **URL:** <http://www.sciences.univ-nantes.fr/info/perso/permanents/goualard/Research/software.html>
- **Description:** DecLIC est une extension de `clp(fd)` permettant la résolution de systèmes de contraintes réelles (utilisation de la hull consistance et de la box consistance).

### 4.4 ISO Prolog

- **Système:** ISO Prolog (pages WEB interactives)
- **Correspondants:** Jonathan Hodgson, Saint Joseph’s University, Philadelphia et AbdelAli Ed-Dbali.
- **URL:** <http://loco.inria.fr/~hodgson/prolog>
- **Description:** Une collection de pages WEB avec des batteries de tests “on-line” (voir section 5.3.5).

## 5 Résultats nouveaux

### 5.1 Aspects langages

#### 5.1.1 Modélisation des solveurs en terme de règles

**Participants :** Julien Arsouze, Michel Bergère, Lucas Bordeaux, Gérard Ferrand, Arnaud Lallouet, Willy Lesaint, Alexandre Tessier.

La résolution d’un système de contraintes (CSP) est présentée classiquement en termes d’itération chaotique d’opérateurs contractants. Cette présentation permet de rendre compte de divers aspects opérationnels tels que les algorithmes utilisés pour obtenir une certaine consistance locale. Une certaine indépendance par rapport à la stratégie d’ordonnancement des calculs permet de penser qu’un calcul peut être défini déclarativement par un système de règles adéquat.

Nous avons commencé cette étude lors du stage de DEA de Lucas Bordeaux [34] en étudiant différents formalismes de la PLC et de la résolution de contraintes. Nous nous sommes particulièrement intéressés à la modélisation de l’unification en termes d’itérations chaotiques. Une vision trop simple en terme d’opérateurs de réduction de domaines ne conduit qu’à une approximation du résultat obtenu par l’algorithme classique. Ceci est dû à la perte de cohérence causée par l’absence de prise en compte des occurrences multiples des variables. Nous

avons proposé une vision basée sur la réduction simultanée du domaine de toutes les variables qui permet de retrouver la notion classique de substitution.

Nous sommes actuellement en train d'étendre ce formalisme à l'occasion du début de la thèse de Julien Arsouze.

### 5.1.2 Typage pour la PLC

**Participants** : Christophe Aillaud, Marco Comini, Pierre Deransart, Jan-Georg Smaus.

Dans le cadre du projet DiSCiPl, une nouvelle approche du typage en PLC a été définie, baptisée "soft typing". Il s'agit de pouvoir combiner le fait que la PLC possède un typage prescriptif inhérent aux domaines de contraintes utilisés, avec une approche descriptive, basée sur la génération automatique de type en PLC telle que développée à l'université de Linköping. Cette étude se poursuit en tentant de rapprocher les méthodes traditionnelles basées sur un typage prescriptif et celles de génération de type utilisée dans le typage descriptif.

Par ailleurs, nous avons poursuivi l'importation du système de diagnostic d'erreurs développé à Linköping en l'intégrant à l'environnement de débogage mis au point pour Calypso<sup>2</sup>. Les idées développées pour l'interface graphique ont été reprises par le partenaire Suédois. Ce travail devrait se poursuivre avec M. Comini. Cette étude a montré l'importance que revêtait la qualité de l'interface pour rendre utilisables en pratique les résultats des analyses de type.

### 5.1.3 Programmation par contraintes et systèmes multi-agents

**Participants** : Philippe Codognet, Nicolas Romero.

Nous cherchons à explorer la relation entre la programmation "orientée agents" et la programmation par contraintes, avec trois perspectives :

- étendre les systèmes de résolution de contraintes existants (permettre par exemple la résolution distribuée),
- utiliser les techniques de coordination/coopération issues des systèmes multiagents pour faire coopérer des solveurs,
- profiter des avantages de la programmation par contraintes (notamment la déclarativité et la gestion d'informations partielles) dans le cadre multiagent.

Dans cette optique, nous avons défini une extension des cc introduits par Saraswat. Notre approche est basée sur l'éclatement du store de contraintes en sites distincts, chacun possédant un store local et un ensemble de processus s'y exécutant. Les sites peuvent se déplacer, le site destination possédant la capacité de filtrer les migrants pour n'accepter que ceux répondant à certaines conditions. Ceci permet la distribution réelle et dynamique du code et des données. D'autre part, nous voulons étendre les possibilités de coordination des cc car la synchronisation temporelle, basée sur le découpage du temps par la répétition d'un signal, est difficilement réalisable dans les cc traditionnels. La coordination par l'imposition de contraintes et l'opération

---

2. Calypso est la version préliminaire de GNU Prolog.

*ask* sont en effet insuffisantes. D'une part, le store étant monotone, on ne peut pas modéliser la répétition d'un signal par de multiples ajouts d'une contrainte donnée dans le store. D'autre part, le fait qu'un processus n'ait pas à savoir quand ni pourquoi il va être interrompu exclut l'utilisation du *ask*. Dans la lignée des langages synchrones, nous avons donc défini des primitives de synchronisation des sites (suspension, déclenchement, et arrêt) et rendu la sémantique du langage obtenu sensible au temps.

#### 5.1.4 Contraintes et conception de mondes virtuels

**Participants :** Philippe Codognet, Nadine Richard.

Le projet InViWo (*Intuitive Virtual Worlds*), qui fait l'objet de la thèse de Nadine Richard, a pour objectif de concevoir un outil intuitif de création de mondes virtuels, et plus particulièrement de proposer un langage de description de comportements d'agents vivants dans des environnements virtuels en 3D. Le projet se composera à terme de trois parties distinctes :

- une bibliothèque modulaire permettant de programmer et d'exécuter des comportements d'agents virtuels. L'exécution d'un monde virtuel doit tenir compte des éventuelles contraintes (locales ou globales) auxquelles peuvent être soumis les agents qui le composent. Cette bibliothèque est destinée aux programmeurs uniquement ;
- un langage déclaratif de haut niveau pour la manipulation de comportements d'agents, inspiré de TCC (Timed Concurrent Constraint) et destiné aux non-programmeurs ;
- une interface graphique de création par composants d'agents et de mondes virtuels.

La bibliothèque de programmation et d'exécution des agents est en cours de réalisation. Les modèles d'agent et de gestion des comportements implémentés permettent d'ores et déjà une conception de haut-niveau d'agents virtuels. Le langage de script s'inspirera de ces modèles pour rendre cette conception plus accessible aux non-programmeurs.

#### 5.1.5 Système GNU Prolog

**Participants :** Daniel Diaz, Jean-Marie Geffroy.

Notre travail s'est concentré sur la finition du système **GNU Prolog**. Diverses extensions permettent désormais d'utiliser **GNU Prolog** pour le développement d'applications "professionnelles" (avec de la gestion d'erreurs, des interfaces graphiques). En ce qui concerne la couche "contraintes sur les domaines finis" nous avons amélioré le résolveur de contraintes et ajouté de nouvelles contraintes prédéfinies. Documentation et un portage pour Windows95/98/NT est en cours de développement avec Jean-Marie Geffroy.

**GNU Prolog** a été diffusé actuellement à plus de 2500 exemplaires depuis sa mise à disposition en avril 1999.

### 5.1.6 Placement de caméras

**Participants** : Frédéric Benhamou, Éric Languéno, Frédéric Goualard, Marc Christie.

Nous nous intéressons depuis l'année dernière au problème du placement automatique de caméras dans un environnement virtuel. Nous avons défini l'année dernière un algorithme permettant de gérer des contraintes temporelles (systèmes de contraintes réelles non-linéaires avec une variable quantifiée). Ces travaux ont été présentés cette année à la conférence PSI'99 [17]. Un langage de haut niveau pour la déclaration de contraintes graphiques, ainsi qu'un prototype de modeleur déclaratif pour la spécification de mouvements de caméras en utilisant le vocabulaire cinématographique ont été implémentés au-dessus d'OpAC. Afin d'accélérer les calculs, nous avons intégré cette année des notions de contraintes redondantes (contraintes induites par les contraintes explicitement données par l'utilisateur, du fait de limitations physiques ou autres) ainsi que de repères locaux. Ces travaux seront présentés lors des douzièmes journées de l'AFIG (Association Française de l'Informatique Graphique) [40].

La création de *plugins* pour Maya de la société *Alias Wavefront* (un des leaders mondiaux dans le domaine des logiciels de synthèse d'image professionnels) pour permettre l'utilisation aisée par des graphistes des algorithmes issus de nos recherches est actuellement à l'étude, cette société nous ayant offert une version de ce logiciel à titre gracieux.

### 5.1.7 Propagation de contraintes parallèle

**Participants** : Laurent Granvilliers, Gaétan Hains [Professeur, Université d'Orléans].

Nous nous sommes intéressés à la parallélisation d'algorithmes de propagation de contraintes utilisant les calculs d'intervalles. Ce travail est basé sur l'algorithme séquentiel BC3 de calcul de la box consistence. Nous avons montré comment modifier la stratégie de propagation en créant des calculs indépendants afin de profiter au maximum d'une machine parallèle. L'étude de faisabilité dans le modèle *Bulk-Synchronous-Programming* a montré la convergence de l'algorithme parallèle et prévu une accélération en moyenne en fonction des types de machines parallèles et de leurs performances. Une implantation, portée sur une Sun bi-processeur et une Cray-T3E à 512 processeurs, a confirmé cette étude. Un ensemble de problèmes standards ont été résolus avec succès et nous avons pu observer une accélération constante en fonction du nombre de processeurs. Les premiers résultats ont été présentés l'année dernière lors de la première rencontre du groupe de travail LODEC du GDR ALP. Ceux de cette année le seront prochainement [37] et font l'objet d'un article soumis au journal *Information Processing Letters*.

### 5.1.8 Étude du parallélisme de données en Programmation en Logique

**Participant** : Arnaud Lallouet.

Le parallélisme de données est un des meilleurs moyens permettant d'exploiter la puissance des machines massivement parallèles. Il n'a pourtant fait l'objet que de peu de travaux dans le domaine de la Programmation en Logique. Ceci tient au fait que celle-ci possède une interprétation parallèle naturelle en termes de parallélisme de tâches. Nous avons proposé une

approche complémentaire qui consiste à manipuler directement des données vectorielles au sein d'un programme logique : l'interprétation parallèle naturelle se fait en termes de parallélisme de données.

Notre langage, appelé DP-LOG, possède deux caractéristiques originales visant à exploiter ce type de parallélisme :

1. les données sont étendues de façon spatiale et distribuées sur l'architecture-cible ;
2. la communication se fait à l'aide d'une primitive qui permet de délocaliser la suite d'une preuve en un autre lieu.

Ces deux aspects sont harmonieusement intégrés au sein du langage logique et ont une lecture déclarative, un des avantages bien connus de la Programmation en Logique. Un autre aspect qui nous paraît important à signaler est l'expressivité de ce langage qui apporte des caractéristiques nouvelles par rapport à Prolog concernant le traitement des tableaux. Par son traitement natif et naturel des vecteurs, DP-LOG permet d'exprimer les algorithmes manipulant des tableaux de manière simple, claire et concise. Ces résultats sont rassemblés dans [12].

### 5.1.9 Arithmétique d'intervalles

**Participant** : Frédéric Goualard.

Les outils de résolution de contraintes (DecLIC, OpAC, Cosinus) implémentés par Frédéric Benhamou et son équipe utilisent l'arithmétique des intervalles pour manipuler rigoureusement des quantités réelles non représentables en machine et pour raisonner sur des ensembles de valeurs réelles. La qualité de ces outils (fiabilité/efficacité) est donc fortement dépendante de celle de la librairie d'arithmétique des intervalles (AI) utilisée. Afin de maximiser tant la fiabilité que l'efficacité de nos systèmes, nous avons défini et implémenté cette année JAIL (*Just Another Interval Library*), une librairie C++ de calcul sur les intervalles réels. Grâce à l'utilisation de la méthode des *traits* de Myers, JAIL est une librairie polymorphe statique (les bornes des intervalles peuvent être tant des `doubles` que des nombres réels en précision arbitraire). Les premiers tests ont montré que JAIL se compare très favorablement avec les autres bibliothèques d'AI disponibles telles que Profil/BIAS et Intlib, tout en offrant, contrairement à ces bibliothèques, le paramétrage du type des bornes des intervalles. JAIL sera prochainement disponible sur le WEB.

## 5.2 Aspects domaines de contraintes et algorithmes de résolution

### 5.2.1 Rétraction de contraintes et contraintes floues

**Participants** : Philippe Codognet, Daniel Diaz, Yan Georget.

Le but de ce travail, entrepris fin 1997, est d'étendre le paradigme de la Programmation Logique avec Contraintes afin de résoudre des problèmes sur-contraints, des problèmes flous et de faire du raisonnement hypothétique.

Nous nous sommes basés sur des travaux théoriques développés à Pise par Stefano Bistarelli, Francesca Rossi et Ugo Montanari. Le cadre qu'ils proposent est celui des contraintes valuées

dans des demi-anneaux (les contraintes deviennent donc des fonctions associant à un  $n$ -uplet un élément d'un demi-anneau).

Nos travaux ont été théoriques (thèse de Yan Georget [3]):

- étude de la consistance d'arc,
- consistance d'arc approchée,
- expressions de contraintes globales par des contraintes valuées dans des ensembles.

Mais aussi pratiques (cf. section 4.2):

- implémentation du système (distribué par `http`), ce langage générique de Programmation Logique avec Contraintes permet la résolution efficace de problèmes où les contraintes sont valuées dans un demi-anneau spécifiable par l'utilisateur.
- résolution de nombreux problèmes (tomographie, démonstration automatique, emplois du temps, ...).

### 5.2.2 Extensions de la box consistance

**Participants** : Frédéric Benhamou, Frédéric Goualard, Laurent Granvilliers.

Nous avons dégagé trois problèmes liés au calcul de la *box consistance* : convergence lente dans la recherche locale des zéros d'une contrainte, calculs inutiles dans le cas de contraintes redondantes, coût de la méthode de Newton combinée avec une recherche binaire.

Dans [27], nous avons défini une nouvelle notion de consistance appelée *box<sub>φ</sub> consistance* et un nouvel algorithme `BCφ` de calcul de la *box consistance*. L'idée est d'affaiblir la précision des calculs locaux pour éviter des convergences lentes et de renforcer l'utilisation des contraintes les plus utiles pour la contraction des domaines. Une stratégie coopérative permet le calcul du point-fixe, *i.e.* la *box consistance*. Une implémentation a été faite dans le prototype `Cosinus`; les résultats expérimentaux sont probants.

Dans [16], `HC4`, un nouvel algorithme de calcul de la *hull consistance*, a été proposé. Un double parcours de l'arbre correspondant à l'expression d'une contrainte permet de contracter tous les domaines des variables associées. C'est une alternative à la décomposition des contraintes moins coûteuse en mémoire et plus rapide grâce à un partage des calculs et un allègement de la propagation. Cet algorithme est aussi utilisé pour calculer la *box consistance* dans le cas d'occurrences simples de variables. Il en résulte un nouvel algorithme de calcul de la *box consistance*, `BC4`, appliquant alternativement `HC4` et l'algorithme standard `BC3` pour les occurrences multiples de variables. Une implantation a été réalisée dans `Cosinus` et démontre la supériorité de `BC4` sur `BC3` tant en précision qu'en temps de calcul.

### 5.2.3 Résolveurs symboliques/numériques

**Participant** : Laurent Granvilliers.



Nous avons porté une attention particulière aux algorithmes symboliques/numériques pour agir sur le temps et la précision des calculs d'intervalles. L'exemple le plus frappant est l'utilisation des bases de Gröbner pour pré-traiter des systèmes polynomiaux. Cette année, nous avons étudié les différentes formes d'expressions des contraintes. D'une part, nous avons implanté dans *Cosinus* la *nested form* et développé une nouvelle forme hybride dans le but de supprimer un maximum d'occurrences de variables : outre une expression plus simple, partiellement factorisée comme le fait la *nested form*, elle permet l'inversion des contraintes comme alternative à la *box consistence*. Elle est particulièrement utile pour les problèmes de manipulation de bras de robots, généralement exprimés comme des systèmes quasi polynomiaux. L'amélioration qui en résulte est significative. Une discussion de ces problèmes a été faite dans le cadre de [38]. D'autre part, nous avons implanté dans *Cosinus* la forme de Taylor utilisée pour linéariser un système d'équations non-linéaires. Le système linéaire obtenu est ensuite préconditionné et traité par la méthode de Gauss-Seidel. Le meilleur algorithme implémenté dans *Cosinus* — combinant la *hull consistence*, la *box consistence* et la méthode de Gauss-Seidel sur les formes intervalles précédemment citées — est de cinq à dix fois plus efficace que l'algorithme de résolution de *Numerica*, le système de référence du domaine.

Le langage de stratégies de résolution de CSPs numériques a été présenté dans [28]. Il est en cours de révision pour permettre une expression intuitive de problèmes tels que l'estimation à erreurs bornées qui se pose en termes de sommes d'exponentielles et admet en général un pavé de solutions.

#### 5.2.4 Analyse par intervalles de systèmes linéaires

**Participants :** Laurent Granvilliers, Julien Arsouze.

Les méthodes dites classiques d'analyse par intervalles font une utilisation intensive de techniques de résolution de systèmes linéaires ; en particulier les systèmes non-linéaires sont au préalable linéarisés par des développements de Taylor.

Deux études ont été réalisées : l'arithmétique affine comme alternative à l'arithmétique standard ; les stratégies de propagation d'intervalles comme implémentation des méthodes itératives du type Gauss-Seidel. Les points éclaircis sont les suivants : l'arithmétique affine se révèle moins précise dans certains cas pourtant cruciaux pour le problème qui nous intéresse ; la stratégie standard est la mieux adaptée aux systèmes préconditionnés. Une implémentation d'un prototype en C++ a confirmé ces conclusions pour un ensemble de systèmes linéaires générés aléatoirement.

#### 5.2.5 Optimisation locale/globale

**Participants :** Frédéric Benhamou, Martine Ceberio.

Dans de nombreux cas, la recherche des solutions d'un système de contraintes réelles ne constitue qu'un des aspects de la résolution d'un problème. En effet, l'utilisateur souhaite souvent obtenir une solution « optimale », cet optimum étant exprimé comme le maximum (ou le minimum) d'une fonction objectif. Nous nous sommes intéressés à la recherche d'algorithmes efficaces d'optimisation globale sous contraintes dans le cadre des contraintes d'intervalles.

L'optimisation sous contraintes dans le cadre qui nous intéresse fait appel à des techniques de préconditionnement de matrices d'intervalles, bien étudiées pour le cas de matrices carrées, mais pas, à notre connaissance, pour celui de matrices non-carrées. Or, l'introduction de contraintes redondantes visant à accélérer les calculs crée souvent des systèmes non-carrés. Aussi, nous avons recherché des méthodes permettant d'obtenir un système carré à partir d'un système qui ne l'est pas. À l'issue de son stage [35], Martine Ceberio a proposé deux solutions à ce problème : la première est basée sur la multiplication par la transposée du système manipulé et la seconde sur l'agrégation de contraintes. Ces travaux restent cependant à valider expérimentalement. Martine Ceberio étudie la possibilité de rendre triangulaire carré un système non carré afin d'obtenir directement les solutions.

Dans le cadre de nos recherches sur le placement de caméras, nous nous sommes aussi intéressés à des problèmes d'optimisation locale : étant donné un point d'intérêt et une caméra filmant ce point, on recherche les plus petites modifications des paramètres définissant l'orientation de la caméra qui lui permettent de continuer à filmer le point d'intérêt après avoir été déplacée par l'utilisateur. La recherche d'une solution optimale (locale ou globale) a fait l'objet d'une partie du stage de DEA de Martine Ceberio. Les résultats obtenus, basés sur une utilisation de la méthode de Newton sur les intervalles pour la recherche de points vérifiant la condition nécessaire de minimalité du premier ordre, ont été utilisés par Olivier Heulot, stagiaire de DEA à l'université de Nantes, pour le placement de caméras réactives.

### 5.2.6 Contraintes vectorielles : le domaine $X^*$

**Participants :** Arnaud Lallouet, Gaétan Hains, Olivier Ballereau.

L'extension de nos travaux concernant le parallélisme de données en programmation en Logique nous a conduit à étendre ce formalisme aux contraintes [29, 39]. Nous proposons tout d'abord d'étendre un domaine de calcul scalaire  $X$  vers le domaine  $X^*$  des vecteurs sur  $X$ . Nous proposons de définir des contraintes de deux types sur les vecteurs ainsi obtenus : des contraintes vectorielles manipulant un vecteur comme un tout et des contraintes scalaires qui sont locales aux éléments des vecteurs. Les contraintes vectorielles peuvent être construites par réplication point à point de contraintes scalaires ou comporter des composantes non locales. Ce dernier type de contraintes nous permet de définir la communication entre les différents processeurs de façon déclarative : la communication est une égalité entre deux éléments non locaux de deux vecteurs.

La manière avec laquelle cette réalité est traitée définit un solveur pour les contraintes vectorielles. Nous définissons trois solveurs principaux : CentralSolv qui est complet mais nécessite une mémoire partagée, CopySolv qui est un solveur distribué complet d'une grande complexité en termes de communications et MatchSolv qui implémente l'égalité de manière incomplète. Ce dernier permet de retrouver la vision classique d'une communication en tant qu'évènement.

Une implantation sous forme d'une bibliothèque C++, GLaSS, a été commencée lors du stage de DEA d'Olivier Ballereau [32]. A cette occasion, un nouveau solveur complet mais néanmoins efficace a été défini : ComSolv.

## 5.3 Aspects environnements

### 5.3.1 Diagnostic déclaratif

**Participants :** Gérard Ferrand, Alexandre Tessier, Daniel Diaz, Christophe Aillaud, Pierre Deransart.

Suites à nos précédents travaux sur la sémantique des programmes logiques avec contraintes, présentée en terme de sémantique positive et sémantique négative [FT98], les travaux sur le diagnostic déclaratif d'erreur ont été approfondis.

Une nouvelle approche duale du diagnostic pour le cas des réponses fausses (*diagnostic positif*) et celui des réponses manquantes (*diagnostic négatif*) a été établie. L'avantage de l'approche est le cadre uniforme et simplifié des algorithmes de recherche d'erreur. Ainsi on a un cadre théorique complet pour le diagnostic déclaratif d'erreur pour tout système de PLC.

L'année a principalement été consacrée à l'implantation d'un outil de diagnostic déclaratif pour le langage Calypso (version primitive de GNU Prolog). La partie concernant le diagnostic positif a été présentée à la revue finale du projet ESPRIT DiSCiPl [13].

### 5.3.2 Simplification de contraintes sur les domaines finis

**Participants :** Saïd Bekraoui, Gérard Ferrand, Willy Lesaint, Alexandre Tessier.

Lors du diagnostic déclaratif d'erreur des programmes logiques avec contraintes, des ensembles de contraintes peuvent être présentés à l'utilisateur. Ces ensembles peuvent être éventuellement gros et illisibles. Par conséquent, nous avons étudié le *problème de visualisation* d'un ensemble de contraintes à travers deux stages de DEA [41, 33].

L'un des stages s'est intéressé aux aspects théoriques de la *simplification* d'un ensemble de contraintes, l'autre s'est intéressé à une implantation.

Cette nouvelle voie de recherche est très prometteuse. En effet le *problème de présentation* d'un ensemble de contraintes sous une forme équivalente et lisible se pose déjà dans l'implantation de système de PLC pour la présentation des réponses. De plus ces méthodes peuvent être utiles pour de nombreuses techniques de mise au point de programme. Enfin le problème d'élimination de contraintes redondantes est similaire au problème d'*entailment* dont on connaît l'importance en programmation par contraintes. De nombreuses recherches sur ce thème ont été réalisées dans d'autres équipes pour le cas des domaines continus ( $\mathbb{R}$ ). Pour les domaines discrets ( $\mathbb{N}$ , domaines finis) le problème reste aujourd'hui ouvert.

### 5.3.3 Débogage avec S-box

**Participants :** Frédéric Benhamou, Frédéric Goulard.

Dans le cadre du projet ESPRIT DiSCiPl, nous avons défini l'année dernière une méthode de débogage en programmation par contraintes s'appuyant sur la notion de *S-box* pour hiéar-

---

[FT98] G. FERRAND, A. TESSIER, «Correction et complétude des sémantiques PLC revisitée par (co-) induction», in: *Journées Francophones de Programmation Logique et Programmation par Contraintes*, O. Ridoux (éd.), Hermès, p. 19–38, mai 1998.

chiser le *store* de contraintes et permettre ainsi son observation par le programmeur [26]. Nous avons finalisé cette année un prototype de débogueur basé sur cette méthode et présenté à la revue finale du projet. Ce logiciel est par ailleurs régulièrement présenté. Il est adapté pour CLP(FD)<sup>3</sup>. L'ensemble des travaux réalisés est présenté dans [8].

### 5.3.4 Interface graphique pour le débogage

**Participants** : Pierre Deransart, Christophe Aillaud.

Dans le cadre du projet ESPRIT DiSCiP1, nous avons commencé à élaborer un interface graphique pour un environnement de débogage qui intègre les différentes fonctionnalités présentées dans la figure 1. Les différents composants sont:

- **Diagnostiqueur statique**: le logiciel d'analyse de type et le système de diagnostic qui lui correspond (cf. section 5.1.2).
- **Diagnostiqueur déclaratif**: le logiciel de débogage déclaratif décrit en section 5.3.1.
- **Visualisation d'abstraction d'arbres**: le logiciel d'abstraction et de visualisation d'arbres de recherche développé avec C. Aillaud [4, 14, 15].

Cette réalisation a montré l'importance de la qualité de l'interface graphique pour rendre effectives les méthodes de débogage. L'intégration avec le système décrit en section 5.3.3 reste à accomplir afin d'avoir l'ensemble des fonctionnalités souhaitables accessibles dans un même environnement.

Le logiciel, qui préfigure ce que pourrait être un véritable environnement de débogage pour la PLC, est régulièrement présenté.

### 5.3.5 ISO Prolog

**Participants** : Jonathan Hogdson, AbdelAli Ed-Dbali, Pierre Deransart.

Une suite de pages WEB, interactives, sur la norme ISO-Prolog a été créée sur l'URL du projet. Elles contiennent, pour l'essentiel:

- Des explications du système d'exceptions de la norme.
- Des explications du système d'entrée/sortie de la norme, ainsi que la possibilité de tester les "write options"<sup>4</sup>.
- Des accès à la spécification formelle de Pierre Deransart et Ali Ed-Dbali telle qu'elle a été publiée dans le standard<sup>[DEDC96]</sup>, avec la possibilité d'y soumettre des exemples.

---

3. CLP(FD) est la version initiale de GNU Prolog.

4. Les "write options" sont des arguments du prédicat système `write_term/3` de Prolog standard et autres prédicats similaires. Elles définissent la manière d'afficher des termes.

---

[DEDC96] P. DERANSART, A. ED-DBALI, L. CERVONI, *Prolog, The Standard; Reference Manual*, Springer Verlag, 1996.

- Une suite de vérification de conformité pour toute nouvelle implantation de Prolog.
- Des “exemples vifs” en forme de logiciel “cgi-bin” écrit en GNU Prolog, et utilisant la librairie PiLLoW<sup>[HCV95]</sup> (UPM, Madrid) pour illustrer les points délicats du standard.
- Une documentation explicative abondante sur le standard Prolog.

Ces pages ont pour but de contribuer à faire connaître le standard et aider à la vérification de conformité des logiciels se recommandant du standard [10, 11].

### 5.3.6 Indexation de pages WEB

**Participants** : Jonathan Hogdson, François Role [Projet ATOLL].

L’objectif de notre recherche est de voir comment l’utilisation de techniques d’analyse de langue naturelle permet de raffiner la sélection de pages WEB dans les moteurs de recherche. Nous avons donc procédé en deux étapes principales:

- *Choix de la zone de sélection.* Dans le but de vérifier la possibilité de se limiter aux en-têtes et aux références, nous avons analysé leurs mot-clefs et procédé à une analyse comparative avec le corps des mêmes pages (étude portant sur une centaine de pages relatives à la programmation en logique).
- *Analyse de texte HTML.* Pour cette étude il a été nécessaire de réaliser un analyseur syntaxique de textes HTML conduisant à une analyse “approximative” du texte naturel (en tenant compte du contexte connu a priori des pages “programmation en logique”).

Nous avons été également conduits à nous intéresser à la construction de serveurs écrits en Prolog et qui pourraient être utilisées pour l’analyse et la maintenance de données structurées. Plusieurs articles sont en préparation.

## 6 Contrats industriels (nationaux, européens et internationaux)

### 6.1 PrologIA

**Participants** : Pierre Deransart, Christophe Aillaud, Frédéric Benhamou, AbdelAli Ed-Dbali, Frédéric Goualard, Alexandre Tessier.

Collaboration avec la société PrologIA autour du produit commercial Prolog IV et, dans le cadre du projet DiSCiPl, amélioration de l’environnement de débogage: vérification de propriétés statiques.

---

[HCV95] M. HERMENEGILDO, D. CABEZA, S. VARMA, «The PilloW Programming Library», *Logiciel*, UPM, Madrid, 1995, <http://www.clip.dia.fi.upm.es/miscdocs/pillow/pillow.html>.

## 6.2 COSYTEC

**Participants** : Pierre Deransart, Christophe Aillaud, Frédéric Benhamou, AbdelAli Ed-Dbali, Frédéric Goualard, Alexandre Tessier.

Collaboration dans le cadre du projet DiSCiPl pour l'amélioration de l'environnement de débogage de CHIP-V5: typage et abstraction d'arbres de recherche.

## 7 Actions régionales, nationales et internationales

### 7.1 Actions nationales

Le projet participe au GDR ALP, projet "PLC" coordonné par P. Codognet et F. Fages, et projet "LODEC" coordonné par G. Hains et S. Anantharaman (Université d'Orléans).

### 7.2 Actions européennes

#### 7.2.1 Projet Esprit DiSCiPl

**Participants** : Pierre Deransart, Christophe Aillaud, Frédéric Benhamou, Michel Bergère, Philippe Codognet, Daniel Diaz, Gérard Ferrand, Frédéric Goualard, Laurent Granvilliers, Alexandre Tessier.

Le projet Esprit (reactive LTR, no 22523) « *Debugging System for Constraint Programming (DiSCiPl)* » a commencé le 1<sup>er</sup> octobre 1996 et s'est terminé le 30 juin 1999. Il regroupait 4 partenaires industriels (COSYTEC, PrologIA, ICON (Italie), OM Partners (Belgique)) et 4 partenaires académiques (INRIA-Rocquencourt avec l'Université d'Orléans, au sein de LOCO, en association avec ERCIM, UPM-Madrid, Université de Linköping, Université de Bristol).

Le projet a permis de réaliser une étape substantielle dans la définition de méthodes et d'outils de mise au point pour la programmation avec contraintes. Il a abouti à la réalisation d'une série de prototypes et à l'amélioration des plateformes industrielles de PrologIA et Cosytec par l'insertion dans les produits commerciaux CHIP V5 et Prolog IV d'outils de débogage.

On trouvera dans la bibliographie les différents documents produits par le projet au cours de cette dernière période (publications et documents internes [36]).

L'approche utilisée a reposé 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. Les prototypes développés par le partenaire INRIA sont décrits en sections 5.3.1, 5.3.3 et 5.3.4.

La fin du projet a été marquée par l'organisation de la revue finale les 20 et 21 mai, sous la forme d'une manifestation exceptionnelle organisée en deux temps: une journée industrielle (Hôtel Méridien Montparnasse, Paris) et un séminaire ouvert, organisé avec l'AFPLC, et qui s'est tenu à l'INRIA. La journée industrielle a attiré une soixantaine de personnes.

Nos prototypes ont fait l'objet de plusieurs présentations et l'ensemble des résultats sont en cours de publication sous forme d'un livre [2] qui contiendra, entre autres, des contributions

de P. Deransart (éditeur principal), C. Aillaud, F. Benhamou, G. Ferrand, F. Goualard et A. Tessier.

P. Deransart a organisé plusieurs séminaires de dissémination destinés à discuter avec d'autres partenaires potentiels l'évolution du domaine: à PrologIA le 22 mars, à l'INRIA-Rocquencourt les 26 et 27 avril, à l'INRIA-Rennes le 21 juin et à l'ENM Nantes le 14 décembre.

L'ensemble des "délivrables" sont publiques et accessibles à l'URL: <http://discipl.inria.fr>

### 7.2.2 European Network of Excellence in Computational Logic

**Participants :** Philippe Codognet, Pierre Deransart.

Pierre Deransart est responsable du nœud de Rocquencourt « programmation en logique avec contraintes ». URL: <http://www.compulog.org>

### 7.2.3 Europe de l'est et Russie

**Participants :** Philippe Codognet, Frédéric Benhamou.

Projet de l'institut Liapunov, concernant une collaboration entre l'Institut d'Intelligence Artificielle de l'académie des sciences de Novossibirsk, l'Université de Nantes, l'INRIA et l'Université d'Orléans. Cette collaboration porte essentiellement sur des aspects langages et résolution de contraintes et devrait permettre des échanges de chercheurs et d'étudiants.

### 7.2.4 ERCIM working group on Constraints

Philippe Codognet a co-fondé le *ERCIM working group on Constraints* avec Krzysztof Apt (CWI, Amsterdam), qui en est le coordinateur actuel. Ce réseau regroupe des équipes à l'INRIA, CWI, CNR (Italie), GMD (Allemagne), les universités de Prague et Brno (République Tchèque) et de Madrid (Espagne).

### 7.2.5 Portugal

**Participants :** Pierre Deransart, Philippe Codognet, Gérard Ferrand, Frédéric Benhamou.

Le réseau formation-recherche franco-portugais a été mis en place en février 96 pour trois ans 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, Jean-Claude Bassano et Sylvie Guilloré) et l'IRIT (Patrick Saint-Dizier et Henri Prade), avec les universités de Lisbonne (UNL, Luis Moniz Pereira, coordinateur) et Braga (Esgalhado Valencia). Il privilégie les thèses en co-tutelle.

Un atelier franco-portugais a été co-organisé par P. Deransart et Sylvie Guilloré à Orléans.

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

## 7.3 Relations internationales

### 7.3.1 Brésil

**Participants** : Mariza Andrade da Silva Bigonha, Roberto da Silva Bigonha, Pierre Deransart, AbdelAli Ed-Dbali, Flávia Peligrinelli Ribeiro, Khaled El Qorchi, José de Siqueira.

Notre projet avec l'Université Fédérale de Minas Gerais, à Belo Horizonte s'est terminé avec le parachèvement du prototype "basic HyperPro" (documentation hypertexte de PLC) et la parution d'un article [23].

Pierre Deransart est responsable pour l'INRIA des relations scientifiques avec le Brésil. À ce titre, il a lancé avec Marie-Christine Imbert et le CNPq le second appel d'offre INRIA/CNPq en décembre. Il a co-organisé avec Marie-Christine Imbert et les correspondants brésiliens la visite officielle de l'INRIA au CNPq du 2 au 14 mai.

## 7.4 Comités de programme de conférences internationales

Frédéric Benhamou a fait partie des comités de programme des conférences PSI'99 (*Perspectives of Systems Informatics*, Akademgorodok, Russie) et JFPLC'99 (*Journées Francophones de Programmation Logique et Programmation par Contraintes*, Lyon, France).

Philippe Codognet a fait partie du comité de programme de JFPLC'99, ICLP'99 et EPIA99. Il est également membre du comité de rédaction de la revue *CONSTRAINTS* (Kluwer Academic Publishers).

Pierre Deransart a fait partie du comité de programme de PACLP'99 (*The Practical Application of Constraint Technologies and Logic Programming*).

Gérard Ferrand est membre du comité de programme de la Stream Theory and extensions of logic programming de CL'2000 (Computational Logic).

Alexandre Tessier a été membre du comité de programme de JFPLC'99 (*Journées Francophones de Programmation Logique et Programmation par Contraintes*, Lyon, France).

## 8 Diffusion de résultats

### 8.1 Animation de la communauté scientifique

Frédéric Benhamou est directeur de l'IRIN (Institut de Recherche en Informatique de Nantes) et membre du conseil d'administration de l'AFPLC.

Philippe Codognet est membre du comité exécutif de l'ALP (Association for Logic Programming) et du conseil d'administration de l'AFPLC.

Pierre Deransart est Secrétaire Général de l'AFPLC (Association Française pour la Programmation en Logique et la programmation par Contraintes) depuis le 3 juin. L'AFPLC regroupe les chercheurs francophones travaillant sur la programmation en logique et avec



contraintes. L'AFPLC est le représentant en France de l'ALP (*Association for Logic Programming*). Elle organise un séminaire national et une conférence «JFPLC» (Journées Francophones pour la Programmation en Logique et avec Contraintes) et contribue ainsi à la promotion du domaine et l'animation de la communauté scientifique.

Gérard Ferrand est directeur du LIFO (Laboratoire d'Informatique Fondamentale d'Orléans, EA1637).

## 8.2 Enseignement universitaire

Frédéric Benhamou, Pierre Deransart et Gérard Ferrand participent à la formation doctorale en Informatique de l'Université d'Orléans et en particulier au DEA.

Frédéric Benhamou participe à la formation doctorale en Informatique de l'Université de Nantes. Il a été rapporteur des thèses de doctorat de Stéphane N'Dong, Université de Marseille, Sarah Mallet, IRISA, Yann Georget, Polytechnique Paris, François Delobel, Université de Nice et Patricia Mainguet, Université de Clermont-Ferrand, et rapporteur de la thèse d'habilitation à diriger des recherches de Luc Jaulin, Université d'Orsay.

Philippe Codognet participe à la formation doctorale de l'université de Paris 6 (DEA IARFA, DESS GLA) et de Paris 7 (DEA SPP). Il a été rapporteur des thèses de Frédéric Meloni (Ecole Polytechnique) et Stéphane Belot (Université d'Evry).

Gérard Ferrand a été rapporteur de l'Habilitation à Diriger des Recherches de Andreas Herzig (Toulouse, janvier 1999).

Alexandre Tessier participe à la formation doctorale en Informatique à l'Université d'Orléans et en particulier au DEA et au DESS.

Frédéric Benhamou, Michel Bergère, Pierre Deransart, Gérard Ferrand, Laurent Granvilliers, Arnaud Lallouet et Alexandre Tessier ont encadré ou co-encadré des stages de maîtrise et de DEA.

## 8.3 Participation à des colloques, séminaire, invitations

Frédéric Benhamou a été invité à faire un exposé lors du *workshop* ERCIM/CompulogNet sur la programmation par contraintes organisé à Paphos, Chypre, en octobre 1999.

Philippe Codognet a fait un tutoriel invité lors du même *workshop*.

Pierre Deransart a été conférencier invité à PACLP'99 (Londres, avril) et Benelog'99 (Maastricht, novembre).

Jonathan Hogdson a été invité à l'université d'Orléans, et à l'UPM, Madrid, pour y faire des séminaires.

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

## 9 Bibliographie

### Livres et monographies

- [1] S. ANANTHARAMAN, F. BENHAMOU (éditeurs), *Numéro thématique Formalismes Logiques, Contraintes et Applications*, 18, 2, Techniques et Sciences Informatiques, Hermès, 1999.
- [2] P. DERANSART, J. MALUSZYŃSKI, M. HERMENEGILDO (éditeurs), *Analysis and Visualization Tools for Constraint Programming*, Springer Verlag, à paraître.

### Thèses et habilitations à diriger des recherches

- [3] Y. GEORGET, *Extensions réactives de la programmation par contraintes*, thèse de doctorat, Doctorat de l'Ecole Polytechnique, octobre 1999.

### Articles et chapitres de livre

- [4] C. AILLAUD, P. DERANSART, «Towards a Language for CLP Choice-tree Visualization and Abstraction», à paraître dans [2].
- [5] F. BENHAMOU, «Interval Constraints», *in: Encyclopedia of Optimization*, Kluwer Academic Publishers, 1999, à paraître.
- [6] F. FAGES, P. RUET, S. SOLIMAN, «Linear concurrent constraint programming: operational and phase semantics», *Information and Computation*, à paraître.
- [7] Y. GEORGET, P. CODOGNET, F. ROSSI, «Constraint Retraction in CLP(FD): Formal Framework and Performance Results», *Constraints, an international journal* 4, 1, 1999, p. 5–42.
- [8] F. GOUALARD, F. BENHAMOU, «Debugging Constraint Programs by Store Inspection», à paraître dans [2].
- [9] F. GOUALARD, F. BENHAMOU, L. GRANVILLIERS, «An Extension of the WAM for Hybrid Interval Solvers», *The Journal of Functional and Logic Programming. The MIT Press 1999*, 4, avril 1999, Special issue on Parallelism and Implementation Technology for Constraint Logic Programming, Vítor Santos Costa, Enrico Pontelli and Gopal Gupta (eds).
- [10] J. HOGDSON, «Report on the Open Meeting of WG17 at PACLP'99», *ALP Newsletter* 12, 2, mai 1999.
- [11] J. HOGDSON, «A Response to Roberto Bragnara», *ALP Newsletter* 12, 1, février 1999.
- [12] A. LALLOUET, «DP-LOG: un langage de programmation logique data-parallèle», *Technique et Science Informatiques* 18, 2, février 1999, p. 233–261.
- [13] A. TESSIER, G. FERRAND, «Declarative Diagnosis in the CLP Scheme», à paraître dans [2].

**Communications à des congrès, colloques, etc.**

- [14] C. AILLAUD, P. DERANSART, «CLP performance analysis: Search-tree Visualisation and Abstraction», *in* : *CP-AI workshop*, M. Milano (éditeur), Ferrara, Italie, 25-26 Février 1999.
- [15] C. AILLAUD, P. DERANSART, «Vers un langage de spécification d'abstraction d'arbre de choix en PLC», *in* : *Actes des 8<sup>e</sup> journées francophones de programmation logique et programmation par contraintes*, Hermès, p. 249–264, Lyon, France, 1999. ISBN 2-7462-0025-2.
- [16] F. BENHAMOU, F. GOULARD, L. GRANVILLIERS, J.-F. PUGET, «Revising hull and box consistency», *in* : *proceedings of the International Conference on Logic Programming (ICLP'99)*, The MIT Press, ICLP'99, Las Cruces, New Mexico, 1999.
- [17] F. BENHAMOU, F. GOULARD, E. LANGUÉNOU, M. CHRISTIE, «An Algorithm to Compute Inner Approximations for Interval Constraints», *in* : *Proceeding of the Andrei Ershov Third International Conference on "Perspectives of System Informatics" (PSI'99), Lecture Notes in Computer Science*, Springer Verlag, Novosibirsk, Akademgorodok, Russia, 1999.
- [18] F. BENHAMOU, L. GRANVILLIERS, «Interval Constraints: Some Results and Perspectives», *in* : *Proceedings of ERCIM Workshop on Constraints, LNAI*, Springer Verlag, Paphos, Cyprus, 1999. Invited paper.
- [19] P. CODOGNET, «Animating Autonomous Agents in Shared Virtual Worlds», *in* : *Proceedings of DMS99, IEEE International Conference on Distributed Multimedia Systems*, Aizu, Japan, 1999. abstract présenté à SIGGRAPH99, session sketches and applications.
- [20] P. CODOGNET, «Declarative Behaviors for Virtual Creatures», *in* : *Proceedings of ICAT99, 9th International Conference on Artificial Reality and Telepresence*, Tokyo, Japan, 1999.
- [21] P. CODOGNET, «An Historical Account of Indexical Images: from Ancient Art to the Web», *in* : *Proceedings of VL99, IEEE International Conference on Visual Languages*, Tokyo, Japan, 1999.
- [22] J. CRUZ, P. BARAHONA, F. BENHAMOU, «Integrating Deep Biomedical Models into Medical Decision Support Systems: an Interval Constraint Approach», *in* : *Proceedings of the Seventh biennial meetings of the AIME and ESMDM societies (AIMDM'99), Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making, Lecture Notes in Artificial Intelligence*, Springer Verlag, Aalborg, Denmark, juin 1999.
- [23] M. A. DA SILVA BIGONHA, A. ED-DBALI, F. P. RIBEIRO, R. DA SILVA BIGONHA, P. DERANSART, J. DE SIQUEIRA, «Projection of HyperPro Document», *in* : *III Simósio brasileiro de linguagens de programação (IISBLP'99)*, Porto Alegre, Brésil, 5-7 mai 1999.
- [24] P. DERANSART, «The DiSCiPl Debugging Methodology», *in* : *PACLP'99*, London, 19-21 avril 1999.
- [25] F. GOULARD, F. BENHAMOU, «Un outil de débogage du store pour la programmation par contraintes», *in* : *Actes des 8<sup>e</sup> journées francophones de programmation logique et programmation par contraintes*, Hermès, p. 281–296, 1999.
- [26] F. GOULARD, F. BENHAMOU, «A Visualization Tool for Constraint Program Debugging», *in* : *Proceedings of the 14th IEEE International Conference on Automated Software Engineering (ASE'99)*, IEEE Computer Society, p. 110–117, Cocoa Beach, Florida, 1999.

- [27] L. GRANVILLIERS, F. GOULARD, F. BENHAMOU, «Box Consistency through Weak Box Consistency», *in: Proceedings of the eleventh IEEE International Conference on Tools with Artificial Intelligence*, IEEE Computer Society, p. 373–380, Chicago, IL, USA, 1999. ISBN 0-7695-0456-6.
- [28] L. GRANVILLIERS, «Stratégies de résolution de CSP numériques», *in: Actes des 8<sup>e</sup> journées Francophones de la Programmation Logique et de la Programmation par Contraintes (JFPLC'99)*, Hermès, p. 221–237, Lyon, France, 1999. ISBN 2-7462-0025-2.
- [29] A. LALLOUET, G. HAINS, «Programmation par Contraintes Parallèle: l'Approche CLP( $X^*$ )», *in: Actes des 8<sup>e</sup> journées Francophones de la Programmation Logique et de la Programmation par Contraintes (JFPLC'99)*, F. Fages (éditeur), Hermès, p. 107–122, Lyon, France, juin 1999.
- [30] N. RICHARD, P. CODOGNET, A. GRUMBACH, «The InViWo virtual agents», *in: Short papers proceedings of Eurographics'99*, Milan, Italie, 1999.

## Rapports de recherche et publications internes

- [31] J. ARSOUZE, «Analyse par intervalles de systèmes linéaires», *Mémoire de dea*, LIFO, University of Orléans, septembre 1999.
- [32] O. BALLEREAU, «GLaSS : un système ouvert pour la résolution de contraintes. Etude du domaine des vecteurs», *Mémoire de dea*, LIFO, Laboratoire d'Informatique Fondamentale d'Orléans, Université d'Orléans, BP 6759, F-45067 Orléans Cedex 2, septembre 1999.
- [33] S. BEKRAOUI, «Implantation des algorithmes de simplification de contraintes arithmétiques linéaires sur les domaines finis», *Mémoire de dea*, LIFO, University of Orléans, septembre 1999.
- [34] L. BORDEAUX, «Modélisation des solveurs de contraintes», *Mémoire de dea*, LIFO, Laboratoire d'Informatique Fondamentale d'Orléans, Université d'Orléans, BP 6759, F-45067 Orléans Cedex 2, septembre 1999.
- [35] M. CEBERIO, «Techniques de préconditionnement et de résolution pour l'optimisation globale avec contraintes», *Mémoire de dea*, Institut de Recherche en Informatique de Nantes, Université de Nantes, septembre 1999.
- [36] P. DERANSART, B. L. DANTEC, «Fifth Management Report», *Management report n° MR5*, INRIA-Rocquencourt, Juin 1999, Couvre la dernière période de 9 mois; contient un résumé des travaux et des résultats de tous les partenaires; il inclue également les derniers cost-statments.
- [37] L. GRANVILLIERS, G. HAINS, «Stratégies parallèles de propagation d'intervalles : résultats et perspectives», *rapport de recherche*, Deuxième rencontre du groupe de travail LODEC du GDR ALP, Paris, France, 1999.
- [38] L. GRANVILLIERS, «Simplification de systèmes non-linéaires pour le calcul d'intervalles», *rapport de recherche*, École de jeunes chercheurs en algorithmique et calcul formel, Bordeaux, France, 1999.
- [39] A. LALLOUET, G. HAINS, «Parallel Constraint Programming over BSP Vectors», *Research Report n° RR-LIFO-99-11*, LIFO, Laboratoire d'Informatique Fondamentale d'Orléans, Université d'Orléans, BP 6759, F-45067 Orléans Cedex 2, mai 1999.
- [40] É. LANGUÉNOU, M. CHRISTIE, «Contraintes redondantes, sur-contraintes et repères attachés», *rapport de recherche*, Actes des douzièmes journées de l'Association Française de l'Informatique Graphique (AFIG'99), 1999.

- [41] W. LESANT, « Simplification de contraintes dans les domaines finis », *Mémoire de dea*, LIFO, University of Orléans, septembre 1999.