



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team Contraintes

Constraint Programming

Rocquencourt

THEME SYM

Activity
R *eport*

2004

Table of contents

1. Team	1
2. Overall Objectives	1
3. Scientific Foundations	2
3.1. Concurrent constraint programming	2
3.2. Constraint solvers	2
3.3. Constraint programming environments	3
3.4. Systems biology	3
4. Application Domains	3
4.1. Combinatorial optimization problems	3
4.2. In-silico cell	4
5. Software	4
5.1. GNU-Prolog	4
5.2. BIOCHAM	4
5.3. TCLP	5
5.4. CLPGUI	5
5.5. Codeine	5
5.6. Adaptive search library	5
6. New Results	6
6.1. SiLCC implementation	6
6.2. Contextual logic programming	6
6.3. Prescriptive type systems in quasi-lattices	7
6.4. Global constraints	7
6.5. Generic trace format	7
6.6. Visualization for constraint programming	7
6.7. Multi-user distributed resource allocation problems	8
6.8. The Biochemical Abstract Machine BIOCHAM	8
6.9. Continuous dynamics in BIOCHAM 2	9
6.10. Learning bio-chemical reactions from temporal logic properties in BIOCHAM	9
7. Contracts and Grants with Industry	9
7.1. COSYTEC	9
7.2. ILOG	9
7.3. TOTAL	10
8. Other Grants and Activities	10
8.1. National contracts	10
8.2. European contracts	10
8.3. Invitations	10
9. Dissemination	10
9.1. Teaching	10
9.2. Leadership within scientific community	11
10. Bibliography	11

1. Team

Head of project-team

François Fages [DR, INRIA]

Vice-head of project-team

Sylvain Soliman [CR, INRIA]

Administrative assistant

Emmanuelle Grousset [up to 1/7/4]

Nadia Mesrar [from 1/7/4]

Staff member

Pierre Deransart [DR, INRIA]

Daniel de Rauglaudre [IR, INRIA, from 22/1/4]

Post-Doctoral fellow

Laurence Calzone [INRIA, ARC CPBIO, since 16/8/4]

Project technical staff

Guillaume Arnaud [IE, INRIA, up to 14/5/4]

Research scientist (partner)

Daniel Diaz [Associate Professor, University of Paris 1]

Ph.D. students

Nathalie Chabrier-Rivier [INRIA scholarship]

Emmanuel Coquery [ATER CNAM]

Sorin Craciunescu [ATER University of Paris 12, up to 1/4/4]

Rémy Haemmerlé [INRIA scholarship]

Ludovic Langevine [INRIA scholarship]

Aurélien Strobbe [CIFRE TOTAL, since 1/6/4]

Student interns

Olivier Bouissou [Ecole Centrale de Paris et Univ. de Berlin, from 22/3/4 to 17/9/4]

Rajiv Dinesh [IIT Dehli, from 16/5/4 to 27/7/4]

Lucie Gentils [DEA SPL, from 5/4/4 to 1/10/4]

Jean-Michel Leconte [ENIB, from 5/7/4 to 27/8/4]

Martin Pernollet [EFREI, from 1/3/4 to 30/11/4, shared with Complex team]

Rajagopalan Shrivaths [IIT Dehli, from 16/5/4 to 27/7/4]

2. Overall Objectives

Constraint Logic Programming supports a great ambition for programming: the one of making of programming essentially a modeling task, with equations, constraints and logical formulas.

Constraint Programming is a field born during the mid 80s from Logic Programming, Linear Programming coming from Operations Research, and Constraint Propagation techniques coming from Artificial Intelligence. Its foundation is the use of relations on mathematical variables to compute with partial information. The successes of Constraint Programming for solving combinatorial optimization problems in industry or commerce are related to the bringing of new local consistency techniques and of *declarative languages* which allow control on the mixing of heterogeneous resolution techniques: numerical, symbolic, deductive and heuristic.

The "Contraintes" group investigates the logical foundations, design, implementation, programming environments and applications of constraint programming languages. The study of Concurrent Constraint languages is a core aspect of the project as they provide a conceptual framework for analyzing different issues of constraint programming, like constraint resolution techniques, concurrent modeling, reactive applications, etc.

The main application domains investigated are combinatorial optimization problems and bio-informatics. In bio-informatics, our objective is not to work on structural biology problems, which has been the main trend up to now, but to attack the great challenge of systems biology, namely to model the function, activity and interaction of molecular systems in living cells, with logic programming concepts and constraint programming and program verification technologies.

3. Scientific Foundations

3.1. Concurrent constraint programming

The class of Concurrent Constraint programming languages (CC) was introduced a decade ago by Vijay Saraswat as a unifying framework for constraint logic programming and concurrent logic programming. The CC paradigm constitutes a representative abstraction of constraint programming languages, and thus allows a fine grained study of their fundamental properties.

CC generalizes the Constraint Logic Programming framework (CLP) by introducing a synchronization primitive, based on constraint entailment. It is a model of concurrent computation, where agents communicate through a shared store, represented by a constraint, which expresses some *partial information* on the values of the variables involved in the computation. The variables play the role of transmissible dynamically created communication channels.

One of the big successes of CC has been the simple and elegant reconstruction of finite domain constraint solvers, and the cooperation of several models to solve a single combinatorial problem. On the other hand, to use CC for programming reactive applications forces one to abandon the hypothesis of monotonic evolution of the constraint store; this is a strong motivation for new extensions of CC languages.

There are strong completeness theorems relating the execution of a CLP program and its translation in classical logic, which provide smooth reasoning techniques for such programs. However these theorems are broken by the synchronization operation of CC. Looking for a logical semantics of CC programs in the general paradigm of logic programming, *program = formula*, *execution = proof search*, leads to a translation in Jean-Yves Girard's linear logic. This allows the recovery of some completeness results about successes and stores; even suspensions may be characterized with the non-commutative logic of Ruet and Abrusci.

It is thus possible to address important issues for Constraint Programming:

- verifying CC programs;
- combining CLP and state-based programming;
- dealing with local search inside a global constraint solving procedure.

The last two cases rely on a natural extension of CC languages, called Linear Concurrent Constraint languages (LCC), which simply replaces constraint systems built onto classical logic by constraint systems built onto linear logic. This allows us to represent state changes thanks to the consumption of resources during the synchronization action, modeled by the linear implication.

3.2. Constraint solvers

Our domains of application use quite different constraint systems:

- finite domains (bounded natural numbers): primitive constraint of some finite domain membership, numerical, symbolic, higher order and global constraints;
- reals: Simplex algorithm for linear constraints and interval methods otherwise;
- terms: subtyping constraints and ontologies.

The project works on both the constraint resolution methods and their cooperation. The main focus is the efficiency of constraint propagation methods, and their combination with local search methods.

3.3. Constraint programming environments

Handling hundreds of thousands of heterogeneous constraints on as many variables is impossible without tools specially designed to show the different views of the execution.

The DiSciPI ESPRIT project showed the use of declarative diagnosis methods based on the logical semantics of programs for interactive debugging systems, and developed tools to visualize the execution of CLP programs: showing the effect of propagation, the shape of the search space (detection of symmetries), the impact of heuristics, etc. We have pursued this research with our partners of the RNTL project OADymPPaC. The main results of this project are a generic trace format for constraint programming languages and a large investigation of visualization tools.

Semantics based debugging, static and dynamic typing, and more generally validation of CC programs are also studied in the project. Concerning typing issues in constraint programming, the usual coercions between constraint domains (e.g. between booleans and integers, lists and terms) offer a particularly challenging task to type systems design.

3.4. Systems biology

Systems biology is a cross-disciplinary domain involving biology, computer science, logics, mathematics, and physics to elucidate the high-level functions of the cell from their biochemical bases at the molecular level.

At the end of the Nineties, research in Bioinformatics evolved, passing from the analysis of the genomic sequence to the analysis of various data produced in mass by *post-genomic* technologies (expression of RNA and proteins, protein-protein interactions, etc). The complexity of the systems concerned requires a large research effort to develop the symbolic notation of biological processes and data.

In order to scale-up and get over the complexity walls to reason about biological systems, there is a general feeling that biology needs to be partly reinvented, and that, beyond providing tools to biologists, computer science has much to offer in terms of concepts and methods.

4. Application Domains

4.1. Combinatorial optimization problems

The number and economic impact of combinatorial optimization problems found in the industrial world are constantly increasing. They cover:

- resource allocation;
- placement;
- scheduling;
- parallelization;
- transport;
- etc.

The last forty years have brought many improvements in Operations Research resolution techniques. In this framework, Constraint Programming can be seen as providing, on the one hand, local coherence techniques that can be applied to various numerical or symbolic constraints, and on the other hand, declarative languages. This last point is crucial for quickly developing complex combinations of algorithms, which is not possible without a language with a high level of abstraction. It allowed for better results, for instance in scheduling problems, than traditional methods, and is promised to an even better future when thinking about cooperation of global resolution and local propagation techniques.

The project builds upon its knowledge of CC languages, constraint solvers and their implementation to work in these directions. The LCC languages offer a framework for theoretical analysis; and the soft constraints framework gives a new way to tackle over-constrained problems. The work on programming environments helps to integrate the Constraint Programming tools into this application domain.

4.2. In-silico cell

In 2002, we have started a Collaborative Research Initiative ARC CPBIO on “Process Calculi and Biology of Molecular Networks” [5] which is ending this year. By working on well understood biological models, we sought:

- to identify in the family of competitive models coming from the Theory of Concurrency (Pi-calculus, Join-calculus and their derivatives) and from Logic Programming (Constraint Logic Programming, Concurrent Constraint languages and their extensions to discrete and continuous time, TCC, HCC), the ingredients of a language for the modular and multi-scale representation of biological processes;
- to provide a series of examples of biomolecular processes transcribed in formal languages, and a set of biological questions of interest about these models;
- to design and apply to these examples formal computational reasoning tools for the simulation, the analysis and the querying of the models.

This work lead us to the design and implementation of the Biochemical Abstract Machine BIOCHAM that has the unique feature of providing formal languages corresponding to different levels of abstraction, for, on the one hand, modeling bio-molecular interaction diagrams with reaction rules, and on the other hand, modeling the biological properties of the system in temporal logic.

This double formalization of both the model and the properties of the biological system at hand opens several new research avenues. Our research continues by a large involvement in two European Projects of the 6th PCRD. First the STREP **APrIL II** where the main objective is to apply probabilistic inductive logic programming techniques to bioinformatics applications. Our main focus is on the networks and pathways of the cell, and the semi-automatic completion of models from observed temporal properties of the system. Second, the Network of Excellence **REVERSE**, in which we focus, amongst other themes, on bio-informatics as a field of application of the new Semantic Web technologies based on rules and constraints [25][12]. In this context we intend to use the biological knowledge stored in online ontologies like **GO** in order to improve different tasks of the modellers:

- re-use and composition of models;
- semi-automatic correction/completion of models.

5. Software

5.1. GNU-Prolog

Participants: Daniel Diaz, Rémy Haemmerlé.

GNU Prolog is a free Prolog compiler with constraint solving over finite domains developed by Daniel Diaz. GNU Prolog accepts Prolog extended with primitives for constraint programming and produces native binaries (like gcc does from a C source). The Prolog part conforms to the ISO standard for Prolog with many practical extensions (global variables, OS interface, sockets,...). GNU Prolog also includes an efficient constraint solver over Finite Domains (FD), giving the user the combined power of constraint programming and the declarativity of logic programming.

An experimental concurrent extension called GNU-Prolog-RH is developed by Rémy Hæmmerlé [35]. This version is an extension of GNU-Prolog with attributed variables, coroutining and constraint logic programming over reals. This version greatly extends the expressive and modeling power of GNU-Prolog. It is also used to prototype the design and implementation of our new SiLCC language.

5.2. BIOCHAM

Participants: Nathalie Chabrier-Rivier, François Fages, Sylvain Soliman.

The Biochemical Abstract Machine **BIOCHAM** provides precise semantics to bio-molecular interaction maps at two abstraction levels: the quantitative level of molecular concentrations, and the qualitative level of Boolean values. Based on these formal semantics, BIOCHAM offers:

- a compositional rule-based language for modeling biochemical systems, allowing patterns and kinetic expressions when numerical data are available;
- a numerical simulator and a non-deterministic boolean simulator;
- a powerful query language based on temporal logic (CTL for qualitative models and LTL with constraints for quantitative models) for expressing biological queries such as reachability, checkpoints, oscillations or stability;
- a machine learning system to infer interaction rules from observed temporal properties.

BIOCHAM is fully implemented in Prolog and interfaced to the state-of-the-art symbolic model checker NuSMV for evaluating boolean CTL queries in large models over several hundreds of variables. BIOCHAM models can be imported from, and exported to, the standard Systems Biology Markup Language SBML. A repository of computational models of biological systems called **CMBSlib** [23] aimed at comparing models as well as formalisms, has been developed in collaboration with our partners of the **ARC CPBIO**.

5.3. TCLP

Participant: Emmanuel Coquery.

TCLP is a prescriptive type system for Constraint Logic Programming, currently: ISO-Prolog, GNU-Prolog, SICStus Prolog and the constraint programming libraries of SICStus Prolog. The flexibility of type checking in TCLP is due to three kinds of polymorphism: parametric polymorphism (e.g. $list(A)$), subtyping (e.g. $list(A) < term$) and overloading (e.g. $-:num*num \rightarrow num$ and $-:A*B \rightarrow pair(A,B)$). No type declaration are required, thanks to the type inference algorithm for predicates and to a default *term* type for function symbols.

5.4. CLPGUI

Participant: François Fages.

CLPGUI is a generic graphical user interface written in Java for constraint logic programming. It is currently available for GNU-Prolog and Sicstus Prolog. CLPGUI has been developed both for teaching purposes and for debugging complex programs. The graphical user interface is composed of several windows: one main console and several dynamic 2D and 3D viewers of the search tree and of finite domain variables. With CLPGUI it is possible to execute incrementally any goal, backtrack or recompute any state represented as a node in the search tree. The level of granularity of the search tree is defined by annotations in the CLP program.

5.5. Codeine

Participants: Ludovic Langevine, Pierre Deransart.

Codeine Codeine is an event-oriented tracer which extends the tracer of GNU-Prolog (version 1.2.16). It is pattern driven and allows to dynamically modify the produced trace through a dialog between a debugging tools and the tracer. The execution is sliced into elementary steps and, for each step, Codeine makes a lot of execution data available. It is distributed under the GPL licence[36].

5.6. Adaptive search library

Participant: Daniel Diaz.

This is a C **library** to solve constraint satisfaction problems by the adaptive local search method. The current release is limited to permutation problems. More precisely, all n variables have the same domain $x_1 \dots x_n$ and

are subject to an implicit all-different constraint. Several problems fall into this category and some examples are provided with the library.

6. New Results

6.1. SiLCC implementation

Participants: François Fages, Rémy Haemmerlé, Daniel de Rauglaudre, Sylvain Soliman.

We are developing SiLCC [20], an imperative and concurrent constraint programming language based on a single paradigm: the one of Vijay Saraswat's concurrent constraint programming extended with constraint systems based on Jean-Yves Girard's Linear Logic. In the late 90's we developed the theory of this extension and we have now begun its implementation.

From a constraint programming point of view, the unique combination of constraint programming with imperative features opens many new possibilities, among which:

- the capability of programming constraint solvers in the language, making them extensible by the user,
- making a fully bootstrapped implementation of a constraint programming language (for the first time since Prolog)
- combining constraint reasoning with state change
- proving program correctness using Linear Logic.

Our current implementation of SiLCC uses GNU-Prolog-RH [35] as temporary kernel language, on top of which a module system and the first bootstrapping libraries have been developed [26]. The objective is to define a small kernel language as an instance of LCC over a simple constraint domain of labelled graphs, on top of which the complete SiLCC language will be built by bootstrapping. Bootstrapping is a fundamental step for getting over the current limits concerning extensibility, robustness, teaching and even efficiency of today constraint programming tools.

6.2. Contextual logic programming

Participant: Daniel Diaz.

While a program-structuring feature is required for a production programming language, the current proposals for the inclusion of modules in the ISO Prolog standard are not very consensual. We have thus investigated an alternative solution based on Contextual Logic Programming (CxLP). Informally, the main point of CxLP is that programs are structured as sets of predicates (*units*) which can be dynamically combined in an execution attribute called a *context*. Goals are seen just as in regular Prolog, except for the fact that the matching predicates are to be located in all the units which make up the current context. We extended CxLP to attach *arguments* to units: these serve the dual purpose of acting as "unit-global" variables and as state placeholders in actual contexts. CxLP clearly carries a higher overhead than regular Prolog, as the context must be searched at run-time for the unit that defines a goal's predicate, a process which requires at least one extra indirection compared to straight Prolog; this kind of situation has become more usual and less of a performance issue in recent systems, in Object-Oriented and even in procedural languages, for instance as a result of using dynamically-loaded shared libraries. Last year we built a first prototype implementation of a Contextual Logic Programming language inside GNU-Prolog. This prototype has surprisingly good performance, considering the lack of optimizations. Contextual Constraint Logic Programming is a powerful paradigm in which to design and implement Organizational Information Systems, particularly when integrated with the ISCO/ISTO mediator framework [10][11]. The implementation has been improved and is actively used in a real-world setting in the Universidade de Evora's second generation web-based academic information system.

6.3. Prescriptive type systems in quasi-lattices

Participants: Emmanuel Coquery, François Fages.

In his thesis [2], Emmanuel Coquery has shown the decidability and NP-completeness of the satisfiability problem for non-structural subtyping constraints in quasi-lattices, and has implemented a complete prescriptive type system for constraint logic programs, named TCLP.

These results are currently applied to the typing of XML documents and semantic web languages [28][27] and to ontological reasoning [12].

6.4. Global constraints

Participants: Rajiv Dinesh, François Fages, Jean-Michel Leconte, Aurélie Strobbe.

We have designed a new global constraint for cutset problems [6]. A cutset in a directed graph $G = (V, E)$ is a set of vertices that cuts all cycles in G . Finding a cutset of minimum cardinality is NP-hard. There exist several approximate algorithms and exact algorithms, most of them using graph reduction techniques. The cutset constraint we propose is a boolean constraint over variables associated to the vertices of a given graph, that states that the subgraph restricted to the vertices having their boolean variable set to true is acyclic. We propose a filtering algorithm based on graph contraction operations and inference of simple boolean constraints, that has a linear time complexity in $O(|E| + |V|)$. The efficiency of the cutset constraint combined with heuristics based on graph properties is shown on benchmarks of the literature for pure minimum cutset problems, and on an application to log-based reconciliation problems where the global cutset constraint is mixed with other boolean constraints.

A constraint programming approach to cutwidth problems, that are central problems in model checking and variable ordering heuristics, has been investigated with less success in [29].

Another new global constraint mixing flow constraints with concentration constraints is emerging from our work with TOTAL concerning the optimisation of crude oil blending.

6.5. Generic trace format

Participants: Pierre Deransart, François Fages, Ludovic Langevine.

One main result of the RNTL **OADymPPaC** project is the finalisation of a *generic trace format* for constraint programming, based on an abstract semantics of finite domain solvers [9]. This generic trace enables debugging tools to be defined almost independently from finite domain solvers, and conversely, tracers to be built independently from these tools [16].

The trace syntax is represented using an XML DTD, called “gentra4cp.dtd” and described in [34]. A compliant trace is encoded in an XML format according to this DTD and follows the described semantics.

The generic trace format has been implemented as an extension of the GNU-prolog tracer, allowing to trace constraint search-tree and propagation [22][21].

The generic trace format is now used in several solvers. It should be extended to handle new constraint domains. Therefore we are organizing its distribution and evolution through a sourceforge **tra4cp project**.

6.6. Visualization for constraint programming

Participants: Guillaume Arnaud, Pierre Deransart, François Fages, Ludovic Langevine, Sylvain Soliman.

A complete description of the **CLPGUI** tool has been published in [8] including:

- its generic architecture for visualizing and controlling the execution of a constraint program,
- the representation of large search trees and domain filtering with 2D and 3D views,
- the capability to create application-oriented interfaces with two examples of placement problems, one in augmented virtual reality.

A new platform called **PAVOT** has also been designed, allowing an easier and modular integration of several visualizers with several solvers, thanks to the generic trace format described in the previous section. The CLPGUI 2D and 3D search tree viewers have been extended to show the volume of propagation events at each node. The originality lies in the variety of parameters which allow a general analysis of constraint propagation events as well.

We also contributed to the adaptation of the CP-INFOVIS Analysis Toolkit to handle the generic trace format [37], based on the INFOVIS toolkit developed by J.-D. Fekete (INRIA-Futurs) and M. Ghoniem (EMN-Nantes) **CP-INFOVIS**. This opens the possibility to analyse constraint program execution with adjacency matrix and plotted graphs.

6.7. Multi-user distributed resource allocation problems

Participants: François Fages, Martin Pernollet, Daniel de Rauglaudre.

In collaboration with Evelyne Lutton (COMPLEX Team).

The problem of office assignment on the INRIA Rocquencourt campus can be considered as a very complex constraint satisfaction problem: the demand of research teams exceeds the actual resource, and in the same time the constraints and preferences of each team are difficult to represent and tune within standard constraint satisfaction software. The adaptive search library described above has been used to find solutions which maximize the overall preferences, given preference criteria for each user. Evolutionary techniques have been used to infer the preferences of the user from their notations of the solutions proposed by the system.

We have experimented in 2003 the scheme of a multi-user interactive evolutionary approach for the management of relative weights of user preferences. This work has been continued in 2004 [31], in order to produce a prototype for real size testing.

6.8. The Biochemical Abstract Machine BIOCHAM

Participants: Nathalie Chabrier-Rivier, Rajiv Dinesh, François Fages, Sylvain Soliman.

The biochemical abstract machine **BIOCHAM** [33][7] is a software environment to represent and analyze protein-protein and protein-DNA interaction networks. The expressiveness of the BIOCHAM modeling language has been illustrated by providing a formal counterpart of Kohn's map of the mammalian cell cycle control [5], and by importing many other models written in other formalisms, from the Web [23]. This effectively turns an otherwise static knowledge into a discrete transition system incorporating a qualitative description of the dynamics. Our proposal to use the Computation Tree Logic CTL as a query language for querying the biological properties of the system has been integrated in BIOCHAM and validated with many examples with respect to:

- the capability to express biological properties of the system in a completely formal way [14][5],
- the efficiency of symbolic model checking tools in this context [5][17],
- the effect of decomposing reactions [15].

6.9. Continuous dynamics in BIOCHAM 2

Participants: Laurence Calzone, Nathalie Chabrier-Rivier, François Fages, Rajagopalan Shrivaths, Sylvain Soliman.

Since version 2.0 we have extended BIOCHAM to support numerical models based on (highly non-linear) Ordinary Differential Equations (ODE) [32]. BIOCHAM now provides different simulation methods for such models, including an implementation in Prolog of Rosenbrock's method for stiff systems, but also provides a Prolog implementation of constraint-based LTL model-checking for numerical traces, simplifying [18].

The generalization of this approach to probabilistic models using probabilistic model checking is also under investigation in the APRIL II European project. The idea is to uniformly support different abstraction levels to reason about complex bio-molecular interaction networks in BIOCHAM, namely three abstraction levels [19]:

1. population of molecules,
2. concentrations,
3. presence/absence of molecules.

6.10. Learning bio-chemical reactions from temporal logic properties in BIOCHAM

Participants: Laurence Calzone, Nathalie Chabrier-Rivier, François Fages, Lucie Gentils, Sylvain Soliman.

The formalization of bio-molecular interactions in BIOCHAM syntax, and the specification of the observed behavior of the system in temporal logic (CTL or LTL) make it possible to develop machine-learning algorithms to automatically correct or complete existing models. It is worth noting that structural learning from temporal logic properties is quite new, both from the machine learning perspective and from the systems biology perspective.

In the framework of the APRIL II STREP we first applied state-of-the-art inductive logic programming tools to simple reachability properties. We then developed a more powerful ad-hoc enumerative approach to the learning of new rules from general temporal logic properties, where the objective is to complete a boolean model w.r.t. a CTL specification [30].

In the quantitative case too, the same strategy can be applied. Since a constraint-based LTL model-checking algorithm has been implemented in BIOCHAM, an enumerative approach similar to the qualitative case can be applied to the estimation of parameters w.r.t. an LTL specification. This provides already an interesting speedup compared to the manual trial and error method used by biologists/modelers.

In both of these cases, the results are quite promising and a more thorough study is under work.

7. Contracts and Grants with Industry

7.1. COSYTEC

Collaboration within the RNTL project **OADymPPaC** (Nov. 2000 - May 2004)

Technology transfer of the generic trace format for the CHIP product.

Technology transfer of the CLPGUI software for the dynamic visualization of CHIP program execution.

Technology transfer of visualization techniques through the adaptation of INFOVIS toolkit.

Contribution to the seminar of the CHIP user's Club.

7.2. ILOG

Collaboration within the RNTL pre-competitive project Manifico (Feb. 2003 - Feb. 2006, 150 Keuros) on non intrusive metacompilation of matching with constraints in rule based languages

Collaboration within the RNTL project **OADymPPaC**, evaluation of Ilog Discovery visualization component in the framework of constraint programming

7.3. TOTAL

Collaboration with TOTAL on a constraint programming approach to the optimization of crude oil blending and the Thesis of Aurélie Strobbe under a CIFRE contract.

8. Other Grants and Activities

8.1. National contracts

- RNTL project MANIFICO (Sep. 2003-2006) on the compilation of rules and constraints. with LORIA PROTHEO and ILOG coord.
- INRIA cooperative research initiative **ARC CPBIO** (2002-2004) on “Process Calculi and Molecular Networks Biology”, with LORIA MODBIO, CNRS PPS lab. of University Paris 7 and Genoscope Evry, coord. F. Fages INRIA Rocquencourt.
- RNTL project **OADymPPaC** (Nov. 2000 - May 2004) “Tools for Dynamic Analysis and Debugging of Constraint Programs” with IRISA LANDES/INSA, INRIA-Futurs, Ecole des Mines de Nantes, University of Orléans, COSYTEC and ILOG, coord. P. Deransart INRIA Rocquencourt.
- CIFRE thesis and industrial contract with TOTAL (July 2004- July 2007) on the optimization of petroleum processes by constraint programming.

8.2. European contracts

- 6th PCRD STREP **APRIL II** “Applications of probabilistic inductive logic programming”, coord. Prof. L. de Raedt, University of Freiburg.
- 6th PCRD Network of Excellence **REVERSE** “Reasoning with rules and semantics”, coord. Prof. F. Bry, Ludwig Maximilian’s University in Munich.
- 5th PCRD Network of Excellence **COLOGNET** “Computational logic network”, area of constraint logic programming coord. F. Rossi, University of Padova.
- ERCIM Working Group on **Constraints**, coord. F. Fages, INRIA Rocquencourt.

8.3. Invitations

Prof. Luc de Raedt from the University of Freiburg in Germany and Prof. Jan Maluzinski from the University of Linköping in Sweden have been invited for short visits.

9. Dissemination

9.1. Teaching

Contraintes is affiliated to the Doctoral school EDITE of the University of Paris 6.

All Ph.D. students and some members of Contraintes teach in the first or second cycles of Universities or Engineering schools. Our involvement in third cycle cursus is the following:

- Master Parisien de Recherche en Informatique (MPRI) lecture on *Programmation par Contraintes* by Sylvain Soliman and François Fages (15h).
- MPRI lecture on *Bio-informatique formelle* by François Fages and Laurence Calzone (15h).
- **DEA SPL** “Sémantique, Preuve et Langages”, ENS, X, University Paris 6, 7, 11: course of **programmation par contraintes**, François Fages 10h, Sylvain Soliman 10h.
- DEA Informatique, University of Orléans: François Fages 3h.
- DESS Informatique, University Paris Sorbonne: Daniel Diaz 30h.

9.2. Leadership within scientific community

- Pierre Deransart is the General Secretary, past Chairman, of the “Association Française pour la Programmation en Logique et la programmation par Contraintes” **AFPLC**, and member of the Steering Committee of **PPDP**. He is in charge of international relationships of INRIA with Brazil and Portugal. He is the manager of the OADymPPaC project.
- François Fages is the Chairman of the ERCIM Working Group on **Constraints**, vice-Chairman, past Chairman of the “Association Française pour la Programmation par Contraintes” **AFPLC**, member past-Chairman of the Steering Committee of the ACM International Conference on Principles and Practice of Declarative Programming, **PPDP**, a member of the Scientific Council of the French-Russian **Liapunov Institute**, and a member of the Editorial Board of **RAIRO Operations Research**.
- Sylvain Soliman is the Secretary of the ERCIM Working Group on **Constraints**.

10. Bibliography

Books and Monographs

- [1] K. APT, F. FAGES, F. ROSSI, P. SZEREDI, J. VANCZA. *Recent Advances in Constraints, selected papers from the Joint ERCIM/Colognet Workshop CSCLP'03*, LNAI, n° 3010, Springer-Verlag, 2004.

Doctoral dissertations and Habilitation theses

- [2] E. COQUERY. *Typage et programmation en logique avec contraintes*, Ph. D. Thesis, Université Paris 6 - Pierre et Marie Curie, December 2004.
- [3] S. CRACIUNESCU. *Preuves d'équivalence de programmes logiques*, Ph. D. Thesis, Ecole Polytechnique, March 2004.
- [4] L. LANGEVINE. *Observation de programmes avec contraintes : une sémantique générique et une architecture d'analyse dynamique*, Ph. D. Thesis, Institut national des sciences appliquées de Rennes, December 2004.

Articles in referred journals and book chapters

- [5] N. CHABRIER-RIVIER, M. CHIAVERINI, V. DANOS, F. FAGES, V. SCHÄCHTER. *Modeling and querying biochemical interaction networks*, in "Theoretical Computer Science", vol. 325, n° 1, September 2004, p. 25–44.
- [6] F. FAGES, A. LAL. *A Constraint Programming Approach to Cutset Problems*, in "Journal of Computation and Operations Research", to appear, 2004.
- [7] F. FAGES, S. SOLIMAN, N. CHABRIER-RIVIER. *Modelling and Querying Interaction Networks in the Biochemical Abstract Machine BIOCHAM*, in "Journal of Biological Physics and Chemistry", vol. 4, n° 2, October 2004, p. 64–73.
- [8] F. FAGES, S. SOLIMAN, R. COOLEN. *CLPGUI: a Generic Graphical User Interface for Constraint Logic Programming*, in "Journal of Constraints, Special Issue on User-Interaction in Constraint Satisfaction", vol. 9, n° 4, October 2004, p. 241–262, <http://dx.doi.org/10.1023/B:CONS.0000049203.53383.c1>.

- [9] L. LANGEVINE, P. DERANSART, M. DUCASSÉ. *A Generic Trace Schema for the Portability of CP(FD) Debugging Tools*, in "Recent Advances in Constraints", K. R. APT, F. FAGES, F. ROSSI, P. SZEREDI, J. VANCZA (editors)., Lecture Notes in Artificial Intelligence, Selected papers of the Joint ERCIM/CoLogNET International Workshop on Constraint Solving and Constraint Logic Programming, n° 3010, Springer Verlag, 2004, p. 171–195.

Publications in Conferences and Workshops

- [10] S. ABREU, D. DIAZ. *Contexts for Organizational Information System Design and Implementation*, in "Proceedings of International Conference on Enterprise Information Systems ICEIS'04, Porto, Portugal", 2004.
- [11] S. ABREU, D. DIAZ, V. NOGUEIRA. *Organizational Information Systems Design and Development with Contextual Constraint Logic Programming*, in "Proceedings of International Conference of European University Information Systems EUNIS'04, Ljubljana, Slovenia", 2004.
- [12] R. BACKOFEN, M. BADEA, P. BARAHONA, L. BADEA, F. BRY, G. DAWELBAIT, A. DOMS, F. FAGES, C. GOBLE, A. HENSCHER, A. HOTARAN, B. HUANG, L. KRIPPAHL, P. LAMBRIX, W. NUTT, M. SCHROEDER, S. SOLIMAN, S. WILL. *Poster: Towards a semantic web for bioinformatics*, in "Proceedings of Bioinformatics 2004, Linköping, Sweden", SocBIN - Society for Bioinformatics in the Nordic countries, June 2004.
- [13] N. CHABRIER-RIVIER, F. FAGES, S. SOLIMAN. *Poster: The Biochemical Abstract Machine BIOCHAM*, in "Proceedings of the 5th International Conference on Systems Biology ICSB'04", October 2004.
- [14] N. CHABRIER-RIVIER, F. FAGES, S. SOLIMAN. *The Biochemical Abstract Machine BIOCHAM*, in "Proceedings of the Evry Spring School on Modelling and Simulation of Biological Processes in the Context of Genomics", P. AMAR, J.-P. CORNET, F. KÉPÈS, V. NORRIS (editors)., Platypus Press, 2004, p. 35–49.
- [15] N. CHABRIER-RIVIER, F. FAGES, S. SOLIMAN. *The Biochemical Abstract Machine BIOCHAM*, in "CMSB'04: Proceedings of the second Workshop on Computational Methods in Systems Biology", V. DANOS, V. SCHÄCHTER (editors)., Lecture Notes in Bioinformatics, Springer-Verlag, 2004.
- [16] P. DERANSART. *Poster: Main Results of the OADymPPaC Project*, in "Proceedings of the International Logic Programming Conference ICLP'04", Lecture Notes in Computer Science, n° 3132, Springer-Verlag, 2004, p. 456–457.
- [17] F. FAGES. *Automated reasoning tools for molecular biology (invited talk)*, in "Proceedings of the 27th Annual German Conference on AI, KI'04, Ulm, Germany", Lecture Notes in Artificial Intelligence, vol. 3238, Springer-Verlag, September 2004.
- [18] F. FAGES. *Constraint-based Model Checking of Non-deterministic Hybrid Systems: A First Experiment in Systems Biology (invited talk)*, in "Proc. Workshop on Constraint Programming and Constraints for Verification CP+CV'04 of ETAPS'04, Barcelona", March 2004.
- [19] F. FAGES. *From Syntax to Semantics in Systems Biology - Towards Automated Reasoning Tools*, in "Converging Sciences, Trento, Italy", December 2004.

- [20] F. FAGES. *SiLCC a Concurrent Constraint Programming Language Based on Linear Logic*, in "Franco-Japanese Workshop on Constraint Programming, Tokyo, Japan", October 2004.
- [21] L. LANGEVINE, M. DUCASSÉ. *Poster: A Tracer Driver to Enable Debugging, Monitoring and Visualization of CLP Executions from a single Tracer*, in "Proceedings of the International Logic Programming Conference ICLP'04", Lecture Notes in Computer Science, n° 3132, Springer-Verlag, 2004, p. 462–463.
- [22] L. LANGEVINE, M. DUCASSÉ. *Un pilote de traceur pour la PLC. Déboguer, auditer et visualiser une exécution avec un même traceur*, in "Programmation en logique avec contraintes, proceedings of JFPLC 2004, Angers (France)", F. MESNARD (editor)., Hermès, June 2004, p. 19–36.
- [23] S. SOLIMAN, F. FAGES. *CMBSlib: a Library for Comparing Formalisms and Models of Biological Systems*, in "CMSB'04: Proceedings of the second Workshop on Computational Methods in Systems Biology", V. DANOS, V. SCHÄCHTER (editors)., Lecture Notes in BioInformatics, Springer-Verlag, 2004.
- [24] S. SOLIMAN. *Pi-calcul et LCC, une odysée de l'espace*, in "Programmation en logique avec contraintes (Actes de JFPLC 2004), Angers (France)", F. MESNARD (editor)., Hermès, June 2004, p. 201–218.

Internal Reports

- [25] R. BACKOFEN, M. BADEA, A. BURGER, F. FAGES, P. LAMBRIX, W. NUTT, M. SCHROEDER, S. SOLIMAN, S. WILL. *State-of-the-art in Bioinformatics*, deliverable, n° A2-D1, Dresden, 2004, <http://idefix.pms.ifi.lmu.de:8080/reverse/index.html#REWERSE-DEL-2004-A2-D1>.
- [26] O. BOUISSOU. *A CHR library for SiLCC*, Rapport de stage, Diplomathesis Univ. Berlin, INRIA, November 2004.
- [27] H. CIRSTEAN, E. COQUERY, W. DRABENT, F. FAGES, C. KIRCHNER, J. MALUSZYNSKI, B. WACK. *Reasoning on the Web with Rules and Semantics*, deliverable, n° I3-D1, Linköping, 2004, <http://idefix.pms.ifi.lmu.de:8080/reverse/index.html#REWERSE-DEL-2004-I3-D1>.
- [28] H. CIRSTEAN, E. COQUERY, W. DRABENT, F. FAGES, C. KIRCHNER, J. MALUSZYNSKI, B. WACK. *Types for Web Rule Languages: a preliminary study*, deliverable, n° I3-D2, LORIA, 2004, <http://idefix.pms.ifi.lmu.de:8080/reverse/index.html#REWERSE-DEL-2004-I3-D2>.
- [29] R. DINESH. *Use of Cutwidth based techniques to improve BIOCHAM*, Rapport de stage de l'IIT Dehli, INRIA, July 2004.
- [30] L. GENTILS. *Apprentissage de modèle biochimique par Programmation Logique Inductive*, Rapport de stage de DEA Sémantique, Preuves et Programmation, INRIA, September 2004.
- [31] M. PERNOLLET. *CONSENSUS: aide à la décision de groupe*, Rapport de stage de fin d'étude, EFREI, November 2004.
- [32] R. SHRIVATHS. *BIOCHAM 2*, Rapport de stage de l'IIT Dehli, INRIA, July 2004.

Miscellaneous

- [33] N. CHABRIER, F. FAGES, S. SOLIMAN. *BIOCHAM's user manual*, 2003–2004, <http://contraintes.inria.fr/BIOCHAM/>.
- [34] P. DERANSART, E. AL.. *Generic Trace Format for Constraint Programming, Version 2.1*, Mai 2004, <http://contraintes.inria.fr/OADymPPaC/Public/delivrables.html>.
- [35] R. HÆMMERLÉ. *GNU-Prolog-RH*, 2004, <http://contraintes.inria.fr/~haemmerl/gprolog-rh/>.
- [36] L. LANGEVINE. *Codeine, a propagation tracer for GNU-Prolog*, 2004, <http://contraintes.inria.fr/~langevin/codeine/>.
- [37] R. MARTIN, J. FEKETE, P. DERANSART. *INFOVIS Analysis Toolkit for GENTR4CP*, 2004, <http://contraintes.inria.fr/OADymPPaC/Public/INFOVIS/index.html>.
- [38] É. PARIZOT, S. SOLIMAN, F. FAGES. *La programmation par contraintes*, February 2004, http://interstices.info/display.jsp?id=c_5782&q=cid%3Djalios_5001, Dossier algorithmique du site web de vulgarisation interstices.info.
- [39] S. SPREEUWENBERG, P. ALBERT, F. FAGES, A. LONJON, J. VANTHIENEN. *Manifeste pour les règles métiers: les principes de l'indépendance des règles*, Business Rules Group, 2004, <http://www.businessrulesgroup.org/brmanifesto/BRManifesteFr.pdf>.