



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team Contraintes

Constraint Programming

Rocquencourt

THEME SYM

Activity
R *eport*

2005

Table of contents

1. Team	1
2. Overall Objectives	1
2.1. 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	5
5.1. BIOCHAM	5
5.2. TCLP	5
5.3. GNU-Prolog	5
5.4. CLPGUI	6
5.5. Tra4CP	6
5.6. Adaptive search library	6
6. New Results	6
6.1. Implementation of SiLCC	6
6.2. Module System for SiLCC	7
6.3. Type systems for constraint logic languages	7
6.4. Global constraint of scheduling with concentration constraints	7
6.5. Multi-user distributed resource allocation problems	8
6.6. Continuous dynamics in BIOCHAM	8
6.7. Stochastic dynamics in BIOCHAM	9
6.8. Learning reaction rules in BIOCHAM	9
6.9. Coupled model of the cell cycle and the circadian cycle in mammalian cells	10
6.10. Space in BIOCHAM	10
6.11. Protein dynamics in growth of fission yeast cells	10
7. Contracts and Grants with Industry	11
7.1. ILOG	11
7.2. TOTAL	11
8. Other Grants and Activities	11
8.1. National contracts	11
8.2. European contracts	11
8.3. Invitations	11
9. Dissemination	11
9.1. Teaching	11
9.2. Leadership within scientific community	12
10. Bibliography	12

1. Team

Head of project-team

François Fages [DR, INRIA]

Vice-head of project-team

Sylvain Soliman [CR, INRIA]

Administrative assistant

Nadia Mesrar [Assistant, INRIA]

Staff member

Pierre Deransart [DR, INRIA]

Daniel de Rauglaudre [IR, INRIA]

Post-Doctoral fellow

Laurence Calzone [INRIA]

Research scientist (partner)

Emmanuel Coquery [Associate Professor, University of Lyon 1]

Daniel Diaz [Associate Professor, University of Paris 1]

Ph.D. students

Nathalie Chabrier-Rivier [INRIA scholarship]

Rémy Haemmerlé [INRIA scholarship]

Aurélie Strobbe [CIFRE TOTAL]

Student interns

Sakina Ayata [MMFAI, ENS Paris]

Loïc Fosse [EPITA]

Rahul Narain [IIT, New Delhi]

Neha Mittal [IIT, New Delhi]

Sylvain Sécherre [ENSTA, DGA]

Sadhika Sharma [IIT, New Delhi]

2. Overall Objectives

2.1. 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 bioinformatics. In bioinformatics, 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 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,

$$\begin{aligned} \text{program} &= \text{logical formula,} \\ \text{execution} &= \text{proof search,} \end{aligned}$$

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;
- temporal constraints: CTL and LTL formulae, either propositional or with numerical constraints.

The project works on constraint resolution methods and their cooperation. The main focus is the efficiency of constraint propagation methods, the design of global constraints and the combination of constraint propagation 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. We study the use of several software analysis tools to help to follow and understand what is going on during constraint resolution. Our approach is based on two elements: a generic XML based trace format for constraints (Gentra4CP) to keep and store execution traces, and the use of different and original kinds of man-machine-interface analysis techniques, mainly based on the visualization of large graphs and trees.

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 post-genomic interaction networks (expression of RNA and proteins, protein-protein interactions, etc). The complexity of these networks requires a large research effort to develop symbolic notation and analysis tools for 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 beyond providing tools to biologists, computer science has much to offer in terms of concepts and methods.

We are interested in the modeling and analysis of complex molecular processes in the cell, at different levels of abstraction, qualitative and quantitative. The most original aspect of our research can be summarized by the following identifications:

$$\begin{aligned} \text{biological property} &= \text{temporal logic formula,} \\ \text{biological validation} &= \text{model-checking.} \end{aligned}$$

Our main research axis is thus to apply logic programming concepts and circuit design technologies to cell molecular biology.

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;
- planing;
- transport;
- etc.

The last forty years have brought many improvements in Operations Research resolution techniques. In this context, Constraint Programming can be seen as providing, on the one hand, local consistency 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 the cooperation of global resolution, local consistency techniques and search methods.

The project builds upon its knowledge of CC languages, constraint solvers and their implementation to work in these directions. The LCC paradigm offers at the same time a theoretical framework for analysis, and a valuable guide for practical language design and implementation. The work on programming environments helps to integrate the Constraint Programming tools into this application domain.

4.2. In-silico cell

In 2002, we started a Collaborative Research Initiative ARC CPBIO on “Process Calculi and Biology of Molecular Networks”. By working on well understood biological models, we sought:

- to identify in the family of competitive models coming from the Theory of Concurrency¹ and from Logic Programming², 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 led us to the design and implementation of the Biochemical Abstract Machine BIOCHAM that has the unique feature of providing formal languages corresponding to different qualitative and quantitative levels of abstraction, for, on the one hand, modeling biomolecular 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 biological properties of the system at hand opens several new research avenues on the design and systematic validation of biological models.

We participate 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 regulatory and metabolic networks in the cell, and the semi-automatic completion or revision of models from observed temporal properties of the system [6], [7], [8]. Second, the Network of Excellence **REVERSE**, in which we focus, among other themes, on bioinformatics as a field of application of the new Semantic Web technologies based on rules and constraints [24], [23]. In this context we intend to use the biological knowledge stored in online ontologies like **GO** in order to improve the re-use and composition of models, as well as the semi-automatic correction/completion of models.

¹ π -calculus, Join-calculus and their derivatives

²Constraint Logic Programming, Concurrent Constraint languages and their extensions to discrete and continuous time, TCC, HCC

5. Software

5.1. BIOCHAM

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

The Biochemical Abstract Machine **BIOCHAM** is a modeling and validation environment for molecular systems biology. BIOCHAM provides precise semantics to biomolecular interaction maps at three abstraction levels:

1. the boolean semantics of molecules presence and absence,
2. the continuous semantics of molecular concentrations,
3. the stochastic semantics of molecule populations.

Based on this formal framework, BIOCHAM offers:

- a compositional rule-based language for modeling biochemical systems, allowing patterns and kinetic expressions when numerical data are available;
- numerical and boolean simulators;
- a temporal logic language (CTL for qualitative models and LTL with numerical constraints for quantitative models) for formalizing biological properties such as reachability, checkpoints, oscillations or stability, and checking them automatically with model-checking techniques;
- automatic search procedures to infer interaction rules and parameter values from known temporal properties of the system.

BIOCHAM is fully implemented in GNU-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**.

5.2. 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. TCLP can also type check constraint solvers written in the *Constraint Handling Rules* (CHR) language.

The flexibility of the TCLP type system is obtained by the combination of 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 a type inference algorithm for predicates and to a default *term* type for function symbols.

5.3. 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 version, called GNU-Prolog-RH, is developed as an extension introducing concurrency, attributed variables, and constraint solving over the 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.

5.4. CLPGUI

Participant: François Fages.

CLPGUI is a generic graphical user interface written in Java for constraint logic programming. It is 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. Tra4CP

Participant: Pierre Deransart.

Following the former OADymPPaC project [21], we have set up with J.-D Fekete a new open source project called **Tra4CP** (Traces for Constraint Programming). The objective of this *sourceforge project* project is to build a repository of constraint resolution traces and visualization tools. The purpose is to disseminate the results of the former **OADymPPaC** project and to facilitate contacts and working exchanges between researchers in the field of complex problem solving using constraint programming.

5.6. Adaptive search library

Participant: Daniel Diaz.

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

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

We are developing SiLCC 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 [20] uses GNU-Prolog-RH [28] as temporary kernel language, on top of which a module system and the first bootstrapping libraries have been developed. 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 of today constraint programming tools, concerning their extensibility, robustness, and teaching.

6.2. Module System for SiLCC

Participants: François Fages, Rémy Haemmerlé, Sylvain Soliman.

In the context of the implementation of SiLCC, we are working on a module system for concurrent constraint logic programming languages. Our first contribution has been the design of a secure module system for constraint logic programs [19], [18]. Although this system can be seen as a simple layer of syntactic sugar, it does provide a discipline for naming predicates and ensuring the protection of the code, making possible the development of libraries and facilitating the safe re-use of existing code. Furthermore we provide this module system with logical and operational semantics and show how the notion of closures inspired from LCC can replace advantageously meta programming predicates in the context of modular programming [30], [17]. This system is implemented in the current version of SiLCC.

We are currently developing an extension of this preliminary module system in which modules are first class objects. In this system, primitive operators of LCC allow us to natively express notions of environment and hiding, and to unify modules, functors and closures as a particular case of LCC agents.

6.3. Type systems for constraint logic languages

Participants: Emmanuel Coquery, François Fages, Rahul Narain.

The *Constraint Handling Rules* (CHR) are an untyped rewrite rule language, initially designed for implementing constraint solvers. In order to benefit from type programming for the development of larger programs, we have designed a generic type system for CHR [9], [10]. CHR being a high-level extension of a host language, such as Prolog or Java, the type system is parameterized by the type system of the host language. The consistency of the type system has been shown for the operational semantics of CHR, as well as for the extended execution model CLP+CHR, for the case where the host language is a constraint logic programming language. This type system has been implemented as an extension of our type checker TCLP which is itself implemented in CLP+CHR. It has been successfully evaluated on twelve CHR solvers and programs, including TCLP itself [11].

This work addresses also some typing issues concerning the semantic web languages investigated in the Reverse network of excellence. Xcerpt is a Web query language based on rules and non-deterministic pattern matching for semi-structured XML data. In collaboration with Sacha Berger from the University of Munich, Włodzimierz Drabent from the Polish Academy of Sciences and Artur Wilk from Linköping University, we propose a set of typing rules for Xcerpt [5]. The rules define a descriptive type system in the sense that the typing of a program is an approximation of its semantics. We provide a formal semantics for a fragment of Xcerpt and a soundness theorem for the type system. This theorem can also be viewed as a subject reduction theorem reduced to one rule, which means that the type system can also be considered as prescriptive. We are currently working on extensions of the type system to deal with rule chaining and some new constructions of Xcerpt.

6.4. Global constraint of scheduling with concentration constraints

Participants: François Fages, Sadhika Sharma, Aurélie Strobbe.

In collaboration with TOTAL, we study an optimization problem of crude oil blending in refineries. The problem is to find a plan of transfer and mixing of crude oil, from the boats to the distillation units through storage and mixing tanks, under various constraints of product concentrations, continuous distillation, schedule, and optimization criteria. This NP-hard problem is the combination of a planning problem (for choosing the transfer tasks and quantities) with a scheduling problem.

Previous approaches based on Mixed-Integer Linear Programming (MILP) rely on a discretization of time that may give infeasible or suboptimal solutions. On the other hand, a constraint programming approach is appealing for its capability to treat efficiently the scheduling part of the problem without time discretization. We have developed such a solution with ILOG tools, by introducing a global constraint of scheduling with concentration constraints, and using heuristics for the planning part of the problem.

Because of the importance of the heuristics part on real-size data, we are also considering a constraint-based local search method for this problem.

6.5. Multi-user distributed resource allocation problems

Participants: François Fages, Loïc Fosse, Sylvain Sécherre.

In collaboration with Evelyne Lutton (COMPLEX Team).

The problem of office assignment on the INRIA Rocquencourt campus can be considered as a difficult 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 define and deal with. Since 2003, in collaboration with COMPLEX team, we have developed a multi-user interactive optimization approach to this problem, based on constraint-based local search for computing solutions, and on genetic algorithms for learning the user preferences from their notations of the proposed solutions.

This year, we have shown that the user preference learning problem could be solved exactly with a linear program, under the assumption that the scores given by the user are a linear combination of criteria. The constraint error functions used in the adaptive search procedure have also been rewritten and a new prototype has been completely re-implemented in SCILAB, showing the capability of the system to propose interesting solutions [31]. This prototype is currently rewritten more efficiently in C++ in the Complex team.

6.6. Continuous dynamics in BIOCHAM

Participants: Laurence Calzone, François Fages, Neha Mittal, Sylvain Soliman.

The Biochemical Abstract Machine **BIOCHAM** [26] is a software environment to represent and analyze protein-protein and protein-DNA interaction networks. Based initially on a boolean view of presence/absence of compounds, with corresponding model-checking techniques for very large networks of proteins, the semantics of BIOCHAM has been enriched with the ability to take into account the kinetics of interactions and the continuous dynamics of the system.

Classical numerical integration methods (Runge-Kutta and Rosenbrock's implicit method for stiff systems) were implemented in GNU-Prolog to deal with the ordinary differential equations (ODEs) derived from the kinetics of the reactions. With values for the initial concentrations BIOCHAM thus proposes a deterministic time series (with fixed or variable step size) describing the evolution of the system. Taking this time series as a linear Kripke structure allows us to use model-checking techniques for LTL queries [7]. An ad-hoc model-checker was incorporated in BIOCHAM, once again implemented in GNU-Prolog. It enables the automatic verification of quantifier-free first-order LTL formulae (about concentrations and their derivatives) but also of some specific quantified first-order formulae, such as periods of oscillations for instance [4].

Current work involves:

- using this model-checking features for learning parameter values, as explained in section 6.9;
- identifying precisely which fragment of first-order LTL is wanted;
- formalizing, as an abstraction, the relationship between the continuous and boolean semantics;
- defining more precisely the relationship with the stochastic level of section 6.7;
- using bifurcation theory to gain information about possible parameter values for certain behaviors of the system, and elucidating how to connect LTL with "types of bifurcations".

6.7. Stochastic dynamics in BIOCHAM

Participants: François Fages, Loïc Fosse, Sylvain Soliman.

Following the same ideas as above, a third abstraction level has been added to BIOCHAM for specifically with low numbers of molecules. The population semantics associates to each compound its *number of molecules* in the system, and to rules a continuous time Markov chain [29].

Probabilistic model-checking was considered to query the properties of such a system, but even state-of-the-art PCTL model-checkers like PRISM did not scale up to treat simple BIOCHAM models. Stochastic simulation techniques, *a la* Gillespie, were then used to provide, once again, linear Kripke structures on which LTL model-checking can be handled as in section 6.6. Since we now have a stochastic simulation engine, the results of the model-checking are also probabilistic, and the result is thus the probability, estimated by counting, that an LTL formula is true [4], [3].

6.8. Learning reaction rules in BIOCHAM

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

The formalization of biomolecular 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 of reaction rules 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. The objective was to complete a boolean BIOCHAM model to satisfy a CTL specification. We developed a more powerful search algorithm for learning new reaction rules from general temporal logic properties. This algorithm evolved into a theory revision algorithm, taking into account that ACTL and ECTL formulae (i.e. those where the only path operator is respectively *A* and *E*) can guide the search for the addition or deletion of rules [6], [8].

In the quantitative case, the same strategy can be applied to learn kinetic parameter values. Once the biological properties of a system under various initial conditions, are described both qualitatively and quantitatively with LTL formulae and constraints, an enumerative algorithm can be applied to estimate parameter values by model-checking [7], [4], [3]. This provides an interesting tool to the biologist/modeler who has to proceed in part by trial and error.

Current work concerns:

- inferring the types of molecules (kinases, phosphatases,...) involved in a model, and guiding the search of rules additions and deletions using this information;
- using information obtained from a bifurcation diagram (see section 6.6);
- defining a measure of how well an LTL specification is satisfied, in order to be able to use gradient methods to speed up the search.

6.9. Coupled model of the cell cycle and the circadian cycle in mammalian cells

Participants: Laurence Calzone, Sylvain Soliman.

Recent advances in cancer chronotherapy techniques support the evidence that there exist some links between the cell and the circadian cycles. Both cycles have been successfully modeled, however, as of today, there are no precise models describing the coupling of the two cycles. One purpose of a coupled model is to better understand how to efficiently target malignant cells depending on the phase of the day.

For this study, we interact with the group of A. Goldbeter from ULB and with J. Clairambault of the BANG team. Our model is built from two models, a model of the circadian cycle from Leloup and Goldbeter and a generic model of the cell cycle model focusing on a crucial event, the entry into mitosis. Currently, the model is focused on the coupling through the WEE1 kinase. This protein plays an important role in the regulation of the activity of MPF, a complex crucial in cell division. The WEE1 kinase seems to be regulated positively by CLOCK/BMAL1 and negatively by CRY, these three genes being well-known circadian core genes.

The BIOCHAM feature for learning parameter values under period constraints was introduced for this study. This allowed us to compute the domain of entrainment of the cell cycle by the circadian cycle. This domain mainly depends on the strength of the WEE1 kinase, regulated by circadian genes, on the activity of MPF. We now intend to incorporate other links, based on evidence from the literature: through the protein c-MYC for instance; and to make the generic cell cycle model more complete in order to see the effect of these other links.

6.10. Space in BIOCHAM

Participants: Sakina Ayata, Laurence Calzone, François Fages, Sylvain Soliman.

Until now, BIOCHAM did not consider the spatial distribution of the molecules. During simulations, all molecules were considered as being either homogeneously mixed together or implicitly localized so that reactions occur like in a molecular soup. But in the cells, molecules are localized and do not necessarily have the same activity in different locations. Moreover transport phenomena induce reaction delays and concentration changes, which may impact the behavior of the system.

A purely symbolic notion of locations has been introduced in BIOCHAM in order to model spatial phenomena [22]. Several cell compartments (such as nucleus, cytoplasm, membrane, etc.) but also grid of cells can be modeled using locations. Formally, transport rules are a special case of reaction rules transforming a molecule in one compartment to the same molecule in another compartment. When the compartment volumes differ from each other, the volume ratios are taken into account in the kinetic models.

An example of transport rules in the p53/mdm2 model and an example of cell to cell signalling have been developed with this version of BIOCHAM. More efficient techniques are under investigation to deal efficiently with large number of locations.

6.11. Protein dynamics in growth of fission yeast cells

Participant: Laurence Calzone.

In collaboration with Michel Kerszberg, from Paris VI University (UMR CNRS 7138), we are currently developing a model of the dynamics of proteins responsible for the polarity in fission yeast cells (*schizosaccharomyces pombe*). The model is both spatial (2D) and stochastic, and thus complements nicely the other type of modeling described in the previous section. Using the *ctrl-dev* software tool of M. Kerszberg, we try to understand the mechanics of growth, polarity and septation (division) of fission yeast cells.

Building on results coming mainly from mutations, we concentrate on two proteins that seem to play a major role in the process. The first one, *tea1p*, present in the cytoplasm binds to and moves along microtubules until it is deposited to the cell cortex by binding to a membrane protein, *mod5p*. Once in complex, *tea1p* is able to recruit polarity factors orienting the direction of growth in the cell.

We hope to explain, thanks to this model, some of the principles of growth in yeast cells. Another interesting aspect of this analysis is to measure the importance and differences of stochastic modeling as opposed to

continuous modeling. We also hope to gain from such a spatial model a better insight into the needed features for a richer notion of locations in BIOCHAM.

7. Contracts and Grants with Industry

7.1. 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

7.2. 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

- ACI IMPbio VICANNE (2004-2007) Modélisation dynamique et simulation des systèmes biologiques.
- RNTL project MANIFICO (Sep. 2003-2006) on the compilation of rules and constraints. with LORIA PROTHEO and ILOG coord.
- 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.
- ERCIM Working Group on **Constraints**, coord. F. Fages, INRIA Rocquencourt.

8.3. Invitations

Prof. Erik Sandewall from Linköping University, Sweden, has been invited for three months.

Prof. Jacques Cohen from Brandeis University, USA, Carolyn Talcott from the Stanford Research Institute, USA, Albert Goldbeter from Université Libre de Bruxelles, Belgium, Kazunori Ueda from Waseda University, Tokyo, and Dr. Vladislav Vyshemirsky from the University of Glasgow, Scotland, 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.

Several Ph.D. students and members of Contraintes teach in the first cycles of Universities or Engineering schools. Our involvement in Masters courses is the following:

- Course on **Constraint Programming**, Master Parisien de Recherche en Informatique (MPRI) by Sylvain Soliman (12h) and François Fages (6h).
- Course on **Formal Bio-informatics** by François Fages (12h) and Laurence Calzone (3h).
- 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 new “Association Française pour la Programmation par Contraintes” **AFPC**, formerly AFPLC. He is in charge of international relationships of INRIA with Brazil and Portugal and co-organized with the Brazilian Ministry of Science the “Information Science and Technology in Brazil” days (October 11-12) with a technical exhibition.
- 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 of the Editorial Board of **RAIRO Operations Research**, member of the Scientific Councils of the Epigenomics project of the Genopole of Evry, and of the French-Russian **Liapunov Institute**.
- Sylvain Soliman is the Secretary of the ERCIM Working Group on **Constraints**.

10. Bibliography

Books and Monographs

- [1] F. FAGES, S. SOLIMAN (editors). *Proceedings of PPSWR'05, Third International Workshop on Principles and Practice of Sematic Web Reasoning*, Lecture Notes in Computer Science, vol. 3703, Springer-Verlag, Dagstuhl Castle, Germany, 2005.
- [2] B. FALTINGS, A. PETCU, F. FAGES, F. ROSSI. *New Trends in Constraints, selected papers from the Joint ERCIM/Colognet Workshop CSCLP'04*, Lecture Notes in Artificial Intelligence, Springer-Verlag, 2005.

Articles in refereed journals and book chapters

- [3] L. CALZONE, N. CHABRIER-RIVIER, F. FAGES, S. SOLIMAN. *Langages formels dans la machine abstraite biochimique BIOCHAM*, in "Techniques et Sciences Informatiques", submitted 2005, <http://contraintes.inria.fr/~fages/Papers/CCFS05tsi.pdf>.
- [4] L. CALZONE, N. CHABRIER-RIVIER, F. FAGES, S. SOLIMAN. *Machine learning biochemical networks from temporal logic properties*, in "Transactions on Computational Systems Biology", submitted 2005, <http://contraintes.inria.fr/~fages/Papers/CCFS05tcsb.pdf>.

Publications in Conferences and Workshops

- [5] S. BERGER, E. COQUERY, W. DRABENT, A. WILK. *Descriptive Typing Rules for Xcerpt*, in "Proceedings of PPSWR'05, Third International Workshop on Principles and Practice of Sematic Web Reasoning, Dagstuhl Castle, Germany", F. FAGES, S. SOLIMAN (editors). , Lecture Notes in Computer Science, vol. 3703, Springer-Verlag, 2005.
- [6] L. CALZONE, N. CHABRIER-RIVIER, F. FAGES, L. GENTILS, S. SOLIMAN. *Machine learning bio-molecular interactions from temporal logic properties*, in "CMSB'05: Proceedings of the third Workshop on Computational Methods in Systems Biology", G. PLOTKIN (editor). , 2005.
- [7] L. CALZONE, N. CHABRIER-RIVIER, F. FAGES, S. SOLIMAN. *A Machine Learning approach to Biochemical Reaction Rules Discovery*, in "Proceedings of Foundations of Systems Biology and Engineering FOSBE'05, Santa Barbara", F. J. DOYLE III (editor). , 2005, p. 375–379.

-
- [8] L. CALZONE, N. CHABRIER-RIVIER, F. FAGES, S. SOLIMAN. *Apprentissage de règles de réactions biochimiques à partir de propriétés en logique temporelle*, in "Actes de JOBIM'05, Lyon", G. PERRIÈRE, A. GUÉNOCHE, C. GEOURJON (editors). , July 2005, p. 183–192.
- [9] E. COQUERY, F. FAGES. *A prescriptive type system for CHR (short paper)*, in "Proceedings of International Conference on Logic Programming ICLP 2005", Lecture Notes in Computer Science, Springer-Verlag, 2005.
- [10] E. COQUERY, F. FAGES. *A type system for CHR*, in "Proceedings of Joint ERCIM/Colognet workshop CSCLP'05", University of Uppsala, 2005.
- [11] E. COQUERY, F. FAGES. *A type system for CHR*, in "Proceedings of the second Constraint Handling Rules Workshop CHR'05", T. FRÜHWIRTH, T. SCHRIJVERS (editors). , 2005.
- [12] E. COQUERY, F. FAGES. *Un système de types pour CHR*, in "Actes de Journées Francophones de Programmation par Contraintes JFPC'05", 2005.
- [13] F. FAGES. *Biologie formelle dans la machine abstraite biochimique BIOCHAM (exposé invité)*, in "Proc. Journées Francophones des Langages Applicatifs, JFLA'04, Obernai", March 2005.
- [14] F. FAGES. *Temporal logic constraints in the biochemical abstract machine BIOCHAM*, in "Franco-Japanese Workshop on Constraint Programming, Le Croisic", October 2005.
- [15] F. FAGES. *Temporal logic constraints in the biochemical abstract machine BIOCHAM (invited talk)*, in "Proceedings of LOPSTR'05, London, joint with SAS'05", Imperial College, September 2005.
- [16] F. FAGES. *Temporal logic constraints in the biochemical abstract machine BIOCHAM (survey talk)*, in "Proceedings of Joint ERCIM/Colognet workshop CSCLP'05", University of Uppsala, June 2005.
- [17] R. HAEMMERLÉ, F. FAGES. *Closures are needed for closed module systems*, in "Proceedings of joint ERCIM/Colognet workshop on constraint solving and constraint logic programming CSCLP'05", Uppsala Univ., 2005.
- [18] R. HAEMMERLÉ, F. FAGES. *Closures for closed module systems*, in "Proc. of CICLOPS'05 workshop associated to International Conference on Logic Programming ICLP'05", University of Barcelona, September 2005.
- [19] R. HAEMMERLÉ, F. FAGES. *Un système de modules fermé pour la PLC*, in "Actes de Journées Francophones de Programmation par Contraintes JFPC'05", 2005.
- [20] R. HAEMMERLÉ. *SiLCC is linear concurrent constraint programming (doctoral consortium)*, in "Proceedings of International Conference on Logic Programming ICLP 2005", Lecture Notes in Computer Science, Springer-Verlag, 2005.
- [21] L. LANGEVINE, ET AL.. *Poster: Gentra4CP: a Generic Trace Format for Constraint Programming*, in "Proceedings of the 21st International Logic Programming Conference ICLP'05", Lecture Notes in Computer Science, Poster. Autors are The OQDymPPaC RNTL Project Team, n° 3668, Springer-Verlag, October 2005,

p. 433–434.

Internal Reports

- [22] S. AYATA. *Modeling space in BIOCHAM*, Rapport de stage du MMFAI de l'ENS Ulm Paris, INRIA, September 2005, <http://contraintes.inria.fr/~ayata>.
- [23] R. BACKOFEN, L. BADEA, P. BARAHONA, M. BERNDTSSON, A. BURGER, G. DAWELBAIT, A. DOMS, F. FAGES, A. HOTARAN, V. JAKONIENE, L. KRIPPAHL, P. LAMBRIX, K. MCLEOD, W. NUTT, B. OLSSON, M. SCHROEDER, A. SCHROIFF, L. ROYER, S. SOLIMAN, H. TAN, D. TILIVEA, S. WILL. *Requirements and specification of bioinformatics use cases*, deliverable, n° A2-D3, Dresden, August 2005, <http://idefix.pms.ifi.lmu.de:8080/reverse/index.html#REVERSE-DEL-2005-A2-D3>.
- [24] R. BACKOFEN, L. BADEA, P. BARAHONA, A. BURGER, G. DAWELBAIT, A. DOMS, F. FAGES, A. HOTARAN, V. JAKONIENE, L. KRIPPAHL, P. LAMBRIX, K. MCLEOD, S. MÖLLER, W. NUTT, B. OLSSON, M. SCHROEDER, S. SOLIMAN, H. TAN, D. TILIVEA, S. WILL. *Usage of Bioinformatics Tools and Identification of Information Sources*, deliverable, n° A2-D2, Lisbon, March 2005, <http://idefix.pms.ifi.lmu.de:8080/reverse/index.html#REVERSE-DEL-2005-A2-D2>.
- [25] N. CHABRIER-RIVIER, F. FAGES, S. SOLIMAN, L. CALZONE. *Learning transition rules from temporal logic properties*, Research Report, n° 5543, INRIA, April 2005, <http://www.inria.fr/rrrt/rr-5543.html>.
- [26] N. CHABRIER-RIVIER, F. FAGES, S. SOLIMAN. *BIOCHAM's user manual*, INRIA, 2003–2005, <http://contraintes.inria.fr/BIOCHAM/>.
- [27] E. COQUERY, F. FAGES. *A type system for CHR*, Technical report, n° RR-5565, INRIA, mai 2005, <http://www.inria.fr/rrrt/rr-5565.html>.
- [28] D. DIAZ, R. HAEMMERLÉ. *GNU Prolog RH user's manual*, INRIA, 1999–2005, <http://contraintes.inria.fr/~haemmerl/gprolog-rh>.
- [29] L. FOSSE. *Model-checking probabiliste pour les réseaux biochimiques*, Rapport de stage de l'EPITA, INRIA, December 2005.
- [30] R. HAEMMERLÉ, F. FAGES. *Closures are needed for closed module systems*, Technical report, n° RR-5575, INRIA, 2005, <http://www.inria.fr/rrrt/rr-5575.html>.
- [31] S. SÉCHERRE, F. FAGES. *Allocation distribuée de ressources dans le système CONSENSUS*, Technical report, n° RT 0308, INRIA, August 2005, <http://www.inria.fr/rrrt/rt-0308.html>.