Activity Report 2011

Section Software

Edition: 2012-03-23
1. BYMOORE Exploratory Action ................................................................. 4

**ALGORITHMS, PROGRAMMING, SOFTWARE AND ARCHITECTURE**

2. COMETE Project-Team ................................................................. 5
3. GEOMETRICA Project-Team .......................................................... 7
4. MEXICO Project-Team ................................................................. 9
5. PARSIFAL Project-Team ............................................................... 10
6. PROVAL Project-Team ................................................................. 11
7. SECSI Project-Team ................................................................. 15
8. TANC Project-Team ................................................................. 17
9. TYPICAL Project-Team ................................................................. 19

**APPLIED MATHEMATICS, COMPUTATION AND SIMULATION**

10. COMMANDS Project-Team ......................................................... 20
11. DEFI Project-Team ................................................................. 21
12. DISCO Team ................................................................. 24
13. GECO Team (section vide) ......................................................... 27
14. MAXPLUS Project-Team ........................................................... 28
15. REGULARITY Team ................................................................. 30
16. SELECT Project-Team ............................................................... 31
17. TAO Project-Team ................................................................. 32

**COMPUTATIONAL SCIENCES FOR BIOLOGY, MEDICINE AND THE ENVIRONMENT**

18. AMIB Project-Team ................................................................. 34
19. DIGIPLANTE Team ................................................................. 36
20. GALEN Team ................................................................. 37
21. PARIETAL Project-Team ............................................................ 38

**NETWORKS, SYSTEMS AND SERVICES, DISTRIBUTED Computing**

22. GRAND-LARGE Project-Team .................................................... 40
23. HIPERCOM Project-Team .......................................................... 45

**PERCEPTION, COGNITION, INTERACTION**

24. AVIZ Project-Team ................................................................. 46
25. DAHU Project-Team (section vide) ............................................ 50
26. IN-SITU Project-Team ............................................................... 51
27. LEO Team ................................................................. 60
BYMOORE Exploratory Action

3. Software

3.1. Software

- **IODC**: Framework for implementing transparent iterative optimization in data centers, see Result 4.1.
- **P & R for neuromorphic accelerator**: A place and route software which maps a neural network graph on an analog neural network hardware, see Result 4.3.
- **HPT**: A performance comparison tool based on non-parametric tests, see Result 4.2.
- **Visual cortex model**: This model is initially based on Poggio’s HMAX model, and has been reimplemented with the prospect of progressively moving it into hardware as efficiently as possible. This work is loosely connected to Result 4.4, and it is still work in progress.
5. Software

5.1. A model checker for the probabilistic asynchronous π-calculus

**Participants:** Miguel Andrés [correspondent], Catuscia Palamidessi.

In collaborations with Dave Parker and Marta Kwiatkowska, we are developing a model checker for the probabilistic asynchronous \( \pi \)-calculus. Case studies with Fair Exchange and MUTE, an anonymous peer-to-peer file sharing system, are in progress.

Technically we use MMC as a compiler to encode the probabilistic \( \pi \)-calculus into certain PRISM representation, which will then be verified against PCTL using PRISM. The transitional semantics defined in MMC can be reused to derive the symbolic transition graphs of a probabilistic process. The code for derivation will work as an add-on to MMC under XSB and invoke a graph traversal to enumerate all reachable nodes and transitions of the probabilistic process.

In the meanwhile we are also attempting a direct and more flexible approach to the development of a model checker for the probabilistic \( \pi \)-calculus, using OCaml. This should allow to extend the language more easily, to include cryptographic primitives and other features useful for the specification of security protocols. As the result of our preliminary steps in this direction we have developed a rudimentary model checker, available at the following URL: [http://vamp.gforge.inria.fr/](http://vamp.gforge.inria.fr/).

5.2. PRISM model generator

**Participants:** Konstantinos Chatzikokolakis [correspondent], Catuscia Palamidessi.

This software generates PRISM models for the Dining Cryptographers and Crowds protocols. It can also use PRISM to calculate the capacity of the corresponding channels. More information can be found in [ 33 ] and in the file README file width instructions at the URL [http://www.lix.polytechnique.fr/comete/software/README-anonmodels.html](http://www.lix.polytechnique.fr/comete/software/README-anonmodels.html).

The software can be downloaded from [http://www.lix.polytechnique.fr/comete/software/anonmodels.tar.gz](http://www.lix.polytechnique.fr/comete/software/anonmodels.tar.gz). These scripts require Perl to run and have been tested in Linux. The GUI of the corners tool also requires the Perl/Tk library. Finally some parts of the model generator tool require PRISM and gnuplot to be installed.

5.3. Calculating the set of corner points of a channel

**Participants:** Konstantinos Chatzikokolakis [correspondent], Catuscia Palamidessi.

The corner points can be used to compute the maximum probability of error and to improve the Hellman-Raviv and Santhi-Vardy bounds. More information can be found in [ 34 ] and in the file README file width instructions at the URL [http://www.lix.polytechnique.fr/comete/software/README-corners.html](http://www.lix.polytechnique.fr/comete/software/README-corners.html).

The software can be downloaded from [http://www.lix.polytechnique.fr/comete/software/corners.tar.gz](http://www.lix.polytechnique.fr/comete/software/corners.tar.gz). These scripts require Perl to run and have been tested in Linux. The GUI of the corners tool also requires the Perl/Tk library. Finally some parts of the model generator tool require PRISM and gnuplot to be installed.

5.4. MMCsp, a compiler for the \( \pi \)-calculus

**Participants:** Peng Wu [correspondent], Catuscia Palamidessi.

MMCsp is a compiler from a simple probabilistic \( \pi \)-calculus to PRISM ([http://www.prismmodelchecker.org/manual/Main/Introduction](http://www.prismmodelchecker.org/manual/Main/Introduction)), models. It is built on XSB ([http://xsbs.sourceforge.net/](http://xsbs.sourceforge.net/)), a tabled logic programming system, and generates the symbolic semantic representation of a probabilistic pi-calculus term in text. A separate Java program then translates this semantic representation into a probabilistic model for PRISM.
The tool was developed by Peng Wu during his postdoc period in Comète in the context of the collaboration between the teams Comète and PRISM under the INRIA/ARC Project ProNoBib (http://www.lsv.ens-cachan.fr/~goubault/ProNobis/index.html). It is based on the papers [41] and [38].

The source code is free and can be download from http://www.cs.ucl.ac.uk/staff/p.wu/mmc_sp_manual.html.
5. Software

5.1. CGAL, the Computational Geometry Algorithms Library

Participants: Pierre Alliez, Jean-Daniel Boissonnat, Olivier Devillers, Monique Teillaud, Mariette Yvinec.

With the collaboration of Hervé Brönnimann, Manuel Caroli, Pedro Machado Manhães de Castro, Frédéric Cazals, Frank Da, Christophe Delage, Andreas Fabri, Julia Flötotto, Philippe Guigue, Michael Hemmer, Samuel Hornus, Menelaos Karavelas, Sébastien Loriot, Abdelkrim Mebarki, Naceur Meskini, Andreas Meyer, Sylvain Pion, Marc Pouget, François Rebuffat, Laurent Rineau, Laurent Saboret, Stéphane Tayeb, Radu Ursu, and Camille Wormser. http://www.cgal.org

CGAL is a C++ library of geometric algorithms and data structures. Its development has been initially funded and further supported by several European projects (CGAL, GALIA, ECG, ACS, AIM@SHAPE) since 1996. The long term partners of the project are research teams from the following institutes: INRIA Sophia Antipolis - Méditerranée, Max-Planck Institut Saarbrücken, ETH Zürich, Tel Aviv University, together with several others. In 2003, CGAL became an Open Source project (under the LGPL and QPL licenses), and it also became commercialized by GEOMETRY FACTORY, a company Born of INRIA founded by Andreas Fabri.

The aim of the CGAL project is to create a platform for geometric computing supporting usage in both industry and academia. The main design goals are genericity, numerical robustness, efficiency and ease of use. These goals are enforced by a review of all submissions managed by an editorial board. As the focus is on fundamental geometric algorithms and data structures, the target application domains are numerous: from geological modeling to medical images, from antenna placement to geographic information systems, etc.

The CGAL library consists of a kernel, a list of algorithmic packages, and a support library. The kernel is made of classes that represent elementary geometric objects (points, vectors, lines, segments, planes, simplices, isothetic boxes, circles, spheres, circular arcs...), as well as affine transformations and a number of predicates and geometric constructions over these objects. These classes exist in dimensions 2 and 3 (static dimension) and \(d\) (dynamic dimension). Using the template mechanism, each class can be instantiated following several representation modes: one can choose between Cartesian or homogeneous coordinates, use different types to store the coordinates, and use reference counting or not. The kernel also provides some robustness features using some specifically-devised arithmetic (interval arithmetic, multi-precision arithmetic, static filters...).

A number of packages provide geometric data structures as well as algorithms. The data structures are polygons, polyhedra, triangulations, planar maps, arrangements and various search structures (segment trees, \(d\)-dimensional trees...). Algorithms are provided to compute convex hulls, Voronoi diagrams, Boolean operations on polygons, solve certain optimization problems (linear, quadratic, generalized of linear type). Through class and function templates, these algorithms can be used either with the kernel objects or with user-defined geometric classes provided they match a documented interface.

Finally, the support library provides random generators, and interfacing code with other libraries, tools, or file formats (ASCII files, QT or LEDA Windows, OpenGL, Open Inventor, Postscript, Geomview...). Partial interfaces with Python, SCILAB and the lpe drawing editor are now also available.

GEOMETRICA is particularly involved in general maintenance, in the arithmetic issues that arise in the treatment of robustness issues, in the kernel, in triangulation packages and their close applications such as alpha shapes, in meshes... Three researchers of GEOMETRICA are members of the CGAL Editorial Board, whose main responsibilities are the control of the quality of CGAL, making decisions about technical matters, coordinating communication and promotion of CGAL.

CGAL is about 700,000 lines of code and supports various platforms: GCC (Linux, Mac OS X, Cygwin...), Visual C++ (Windows), Intel C++... A new version of CGAL is released twice a year, and it is downloaded about 10000 times a year. Moreover, CGAL is directly available as packages for the Debian, Ubuntu and Fedora Linux distributions.
More numbers about CGAL: there are now 13 editors in the editorial board, with approximately 20 additional developers. The user discussion mailing-list has more than 1000 subscribers with a relatively high traffic of 5-10 mails a day. The announcement mailing-list has more than 3000 subscribers.
5. Software

5.1. Software

5.1.1. libalf: the Automata Learning Framework

Participant: Benedikt Bollig [correspondent].

libalf is a comprehensive, open-source library for learning finite-state automata covering various well-known learning techniques (such as, Angluin’s $L^*$, Biermann, and RPNI, as well as a novel learning algorithm for NFA. libalf is highly flexible and allows for facilely interchanging learning algorithms and combining domain-specific features in a plug-and-play fashion. Its modular design and virtual plug-and-play implementation in C++ make it a flexible platform for adding and engineering further, efficient learning algorithms for new target models (e.g., Büchi automata).

Details on libalf can be found at http://libalf.informatik.rwth-aachen.de/

5.1.2. Mole/Cunf: unfolders for Petri Nets

Participants: Stefan Schwoon [correspondent], César Rodríguez.

Mole computes, given a safe Petri net, a finite prefix of its unfolding. It is designed to be compatible with other tools, such as PEP and the Model-Checking Kit, which are using the resulting unfolding for reachability checking and other analyses. The tool Mole arose out of earlier work on Petri nets. Details on Mole can be found at http://www.lsv.ens-cachan.fr/~schwoon/tools/mole/

In the context of MExICo, we have created a new tool called Cunf, which is able to handle contextual nets (Petri nets with read arcs). Recent work carried out within MExICo [53] has transformed a preliminary implementation into an efficient tool. While in principle every contextual net can be transformed into an “equivalent” Petri net and then unfolded using Mole, Cunf can take advantage of their special features to do the job faster. More details can be found at http://www.lsv.ens-cachan.fr/~rodrigue/tools/cunf/

5.1.3. COSMOS: a Statistical Model Checker for the Hybrid Automata Stochastic Logic

Participants: Hilal Djafri [correspondent], Benoît Barbot.

COSMOS is a statistical model checker for the Hybrid Automata Stochastic Logic (HASL). HASL employs Linear Hybrid Automata (LHA), a generalization of Deterministic Timed Automata (DTA), to describe accepting execution paths of a Discrete Event Stochastic Process (DESP), a class of stochastic models which includes, but is not limited to, Markov chains. As a result HASL verification turns out to be a unifying framework where sophisticated temporal reasoning is naturally blended with elaborate reward-based analysis. COSMOS takes as input a DESP (described in terms of a Generalized Stochastic Petri Net), an LHA and an expression $Z$ representing the quantity to be estimated. It returns a confidence interval estimation of $Z$. COSMOS is written in C++ and is freely available to the research community.

Details on COSMOS can be found at http://www.lsv.ens-cachan.fr/~barbot/cosmos/
5. Software

5.1. Profound

Participants: Kaustuv Chaudhuri [correspondant], Nicolas Guenot, Lutz Straßburger.

Profound is an interactive proof-development tool based on the focused calculus of structures [19]. It allows the user to build proofs using direct manipulation of the current proof state using the cursor keys and the mouse, instead of learning a formal textual proof interaction language. The tool checks all user actions dynamically with the aid of a theorem prover.

We plan to investigate adaptations of a tool such as Profound for proof development in other interactive proof development systems such as Abella or Coq. We also plan to use the high degree of proof compression that is enabled by the calculus of structures to create efficient proof certificates for exchange between different proof development systems.

The first public release of Profound is expected in December 2011. The development can be followed on INRIA GForge.

5.2. Abella

Participants: Andrew Gacek, Dale Miller.

The earliest versions of the Abella theorem prover was written while Gacek was a PhD student at the University of Minnesota. Two years ago, Gacek was a post doc in the Parsifal team and more features were added to this prover. During 2011, Chaudhuri and three interns (Andrew Cave from McGill and Salil Joshi and Chris Martens from CMU) developed some news designs and new prototypes of needed features for Abella. These features will provide this prover with better ways to manipulate specification-logic contexts in the reasoning-logic and a means for outputting proper proof object (different from the scripts that are used to find a proof).

For more information, see the Abella home page: http://abella.cs.umn.edu/.

5.3. Bedwyr

Participants: Kaustuv Chaudhuri, Quentin Heath, Dale Miller [correspondant].

During 2011, our close colleagues Alwen Tui (Australian National University) and David Baelde (INRIA team Proval) have made some improvements to the Bedwyr system. In the case of Tui, he made these changes in order to build SPEC, a model checker for the spi-calculus, on top of Bedwyr.

Starting in September, Quentin Heath has joined the team as a technical staff member. He is currently working on the Bedwyr code so that it can share files with the Abella system. These two provers work within a logic that is essentially the same: Heath is working to ensure that the concrete syntax and static semantics for the logical expressions on which they work is also the same. Thus, we expect to have our model checker (Bedwyr) and interactive theorem prover (Abella) share theories and proofs.

The work of Heath is being done in the context of the BATT ADJ project funded by INRIA. The boarder goals of the BATT project is to get four software systems (Bedwyr, Abella, Tac, and Teyjus) to inter-operate.

See also the web page http://slimmer.gforge.inria.fr/bedwyr/.
5. Software

5.1. The CiME rewrite toolbox

Participants: Évelyne Contejean [contact], Claude Marché, Andrei Paskevich, Xavier Urbain.

CiME is a rewriting toolbox. Distributed since 1996 as open source, at URL http://cime.lri.fr. Beyond a few dozens of users, CiME is used as back-end for other tools such as the TALP tool developed by Enno Ohlebusch at Bielefeld university for termination of logic programs; the MU-TERM tool (http://www.dsic.upv.es/~slucas/csr/termination/muterm/) for termination of context-sensitive rewriting; the CARIBOO tool (developed at INRIA Nancy Grand-Est) for termination of rewriting under strategies; and the MTT tool (http://www.lcc.uma.es/~duran/MTT/) for termination of Maude programs. CiME2 is no longer maintained, and the currently developed version is CiME3, available at http://a3pat.ensiie.fr/pub. The main new feature of CiME3 is the production of traces for Coq. CiME3 is also developed by the participants of the A3PAT project at the CNAM, and is distributed under the Cecill-C license.

5.2. The Why platform

Participants: Claude Marché [contact], Romain Bardou, François Bobot, Jean-Christophe Filliâtre, Guillaume Melquiond, Andrei Paskevich.


The Why platform is a set of tools for deductive verification of Java and C source code. In both cases, the requirements are specified as annotations in the source, in a special style of comments. For Java (and Java Card), these specifications are given in JML and are interpreted by the Krakatoa tool. Analysis of C code must be done using the external Frama-C environment, and its Jessie plugin which is distributed in Why.

The platform is distributed as open source, under GPL license, at http://why.lri.fr/. The internal VC generator and the translators to external provers are no longer under active development, as superseded by the Why3 system described below.

The Krakatoa and Jessie front-ends are still maintained, although using now by default the Why3 VC generator. These front-ends are described in a specific web page http://krakatoa.lri.fr/. They are used for teaching (University of Evry, Ecole Polytechnique, etc.), used by several research groups in the world, e.g at Fraunhofer Institute in Berlin [71], and at Universidade do Minho in Portugal [50].

5.3. The Why3 system

Participants: Jean-Christophe Filliâtre [contact], François Bobot, Claude Marché, Guillaume Melquiond, Andrei Paskevich.


Why3 is the next generation of Why. Why3 clearly separates the purely logical specification part from generation of verification conditions for programs. It features a rich library of proof task transformations that can be chained to produce a suitable input for a large set of theorem provers, including SMT solvers, TPTP provers, as well as interactive proof assistants.

It is distributed as open source, under GPL license, at http://why3.lri.fr/.

---

1 self-evaluation following the guidelines (http://www.inria.fr/content/download/11783/409665/version/4/file/SoftwareCriteria-V2-CE.pdf) of the Software Working Group of INRIA Evaluation Committee (http://www.inria.fr/institut/organisation/instances/commission-d-evaluation)
Why3 is used as back-end of our own tools Krakatoa and Jessie, but also as back-end of the GNATprove tool (Adacore company), and in a near future of the WP plugin of Frama-C. Why3 has been used to develop and prove a significant part of the programs of our team gallery http://proval.lri.fr/gallery/index.en.html, and will be used soon for teaching (Master Parisien de Recherche en Informatique).

5.4. The Alt-Ergo theorem prover

Participants: Sylvain Conchon [contact], Évelyne Contejean, Stéphane Lescuyer, Alain Mebsout, Mohamed Iguernelala.

Criteria for Software Self-Assessment: A-3-up, SO-4, SM-4-up, EM-4, SDL-5, OC-4.

Alt-Ergo is an automatic, little engine of proof dedicated to program verification, whose development started in 2006. It is fully integrated in the program verification tool chain developed in our team. It solves goals that are directly written in the Why’s annotation language; this means that Alt-Ergo fully supports first order polymorphic logic with quantifiers. Alt-Ergo also supports the standard [90] defined by the SMT-lib initiative. It is currently used in our team to prove correctness of C and Java programs as part of the Why platform and the new Why3 system. Alt-Ergo is also called as an external prover by the Pangolin tool developed by Y. Regis Gianas, INRIA project-team Gallium http://code.google.com/p/pangolin-programming-language/. Alt-Ergo is usable as a back-end prover in the SPARK verifier for ADA programs, since Oct 2010. It is planed to be integrated in next generation of Airbus development process.

Alt-Ergo is distributed as open source, under the CeCILL-C license, at URL http://alt-ergo.lri.fr/.

5.5. Bibtex2html

Participants: Jean-Christophe Filliâtre [contact], Claude Marché.


Bibtex2html is a generator of HTML pages of bibliographic references. Distributed as open source since 1997, under the GPL license, at http://www.lri.fr/~filliatr/bibtex2html/. We estimate that between 10000 and 100000 web pages have been generated using Bibtex2html. Bibtex2html is also distributed as a package in most Linux distributions. Package popularity contests show that it is among the 20% most often installed packages.

5.6. OCamlgraph

Participants: Jean-Christophe Filliâtre [contact], Sylvain Conchon.

OCamlgraph is a graph library for Objective Caml. It features many graph data structures, together with many graph algorithms. Data structures and algorithms are provided independently of each other, thanks to OCaml module system. OCamlgraph is distributed as open source, under the LGPL license, at http://ocamlgraph.lri.fr/. It is also distributed as a package in several Linux distributions. OCamlgraph is now widely spread among the community of OCaml developers.

5.7. Mlpost

Participants: Jean-Christophe Filliâtre [contact], Stéphane Lescuyer, Romain Bardou, François Bobot.

Mlpost is a tool to draw scientific figures to be integrated in LaTeX documents. Contrary to other tools such as TikZ or MetaPost, it does not introduce a new programming language; it is instead designed as a library of an existing programming language, namely Objective Caml. Yet it is based on MetaPost internally and thus provides high-quality PostScript figures and powerful features such as intersection points or clipping. Mlpost is distributed as open source, under the LGPL license, at http://mlpost.lri.fr/. Mlpost was presented at JFLA’09 [51].
5.8. Functory

**Participants:** Jean-Christophe Filliâtre [contact], Kalyan Krishnamani.

Functory is a distributed computing library for Objective Caml. The main features of this library include (1) a polymorphic API, (2) several implementations to adapt to different deployment scenarios such as sequential, multi-core or network, and (3) a reliable fault-tolerance mechanism. Functory was presented at JFLA 2011 [31] and at TFP 2011 [27].

5.9. The Flocq library

**Participants:** Sylvie Boldo [contact], Guillaume Melquiond.


The Flocq library for the *Coq* proof assistant is a comprehensive formalization of floating-point arithmetic: core definitions, axiomatic and computational rounding operations, high-level properties [23]. It provides a framework for developers to formally certify numerical applications.

It is distributed as open source, under a LGPL license, at http://flocq.gforge.inria.fr/. It was first released in 2010.

5.10. The Gappa tool

**Participant:** Guillaume Melquiond [contact].


Given a logical property involving interval enclosures of mathematical expressions, Gappa tries to verify this property and generates a formal proof of its validity. This formal proof can be machine-checked by an independent tool like the *Coq* proof-checker, so as to reach a high level of confidence in the certification [66] [19].

Since these mathematical expressions can contain rounding operators in addition to usual arithmetic operators, Gappa is especially well suited to prove properties that arise when certifying a numerical application, be it floating-point or fixed-point. Gappa makes it easy to compute ranges of variables and bounds on absolute or relative roundoff errors.

Gappa is being used to certify parts of the mathematical libraries of several projects, including CRlibm, FLIP, and CGAL. It is distributed as open source, under a Cecill-B / GPL dual-license, at http://gappa.gforge.inria.fr/. Part of the work on this tool was done while in the Arénaire team (INRIA Rhône-Alpes), until 2008.

5.11. The Interval package for Coq

**Participant:** Guillaume Melquiond [contact].


The Interval package provides several tactics for helping a *Coq* user to prove theorems on enclosures of real-valued expressions. The proofs are performed by an interval kernel which relies on a computable formalization of floating-point arithmetic in *Coq*.

It is distributed as open source, under a LGPL license, at http://www.lri.fr/~melquion/soft/coq-interval/. Part of the work on this library was done while in the Mathematical Components team (Microsoft Research–INRIA Joint Research Center).

In 2010, the Flocq library was used to straighten and fill the floating-point proofs of the Interval package.

5.12. The Alea library for randomized algorithms

**Participants:** Christine Paulin-Mohring [contact], David Baelde.

The ALEA library is a Coq development for modeling randomized functional programs as distributions using a monadic transformation. It contains an axiomatisation of the real interval $[0,1]$ and its extension to positive real numbers. It introduces definition of distributions and general rules for approximating the probability that a program satisfies a given property.

It is distributed as open source, at http://www.lri.fr/~paulin/ALEA. It is currently used as a basis of the Certicrypt environment (MSR-INRIA joint research center, Imdea Madrid, INRIA Sophia-Antipolis) for formal proofs for computational cryptography [53]. It is also experimented in LABRI as a basis to study formal proofs of probabilistic distributed algorithms.

5.13. The Coccinelle library for term rewriting

**Participant:** Évelyne Contejean [contact].

Coccinelle is a Coq library for term rewriting. Besides the usual definitions and theorems of term algebras, term rewriting and term ordering, it also models some of the algorithms implemented in the CiME toolbox, such as matching, matching modulo associativity-commutativity, computation of the one-step reducts of a term, RPO comparison between two terms, etc. The RPO algorithm can effectively be run inside Coq, and is used in the Color development (http://color.inria.fr/) as well as for certifying Spike implicit induction theorems in Coq (Sorin Stratulat).

Coccinelle is developed by Évelyne Contejean, available at (http://www.lri.fr/~contejea/Coccinelle), and is distributed under the Cecill-C license.
5. Software

5.1. Tookan

Participants: Graham Steel [correspondant], Romain Bardou.

See also the web page http://secgroup.ext.dsi.unive.it/projects/security-apis/pkcs11-security/tookan/.

Tookan is a security analysis tool for cryptographic devices such as smartcards, security tokens and Hardware Security Modules that support the most widely-used industry standard interface, RSA PKCS#11. Each device implements PKCS#11 in a slightly different way since the standard is quite open, but finding a subset of the standard that results in a secure device, i.e. one where cryptographic keys cannot be revealed in clear, is actually rather tricky. Tookan analyses a device by first reverse engineering the exact implementation of PKCS#11 in use, then building a logical model of this implementation for a model checker, calling a model checker to search for attacks, and in the case where an attack is found, executing it directly on the device. Tookan has been used to find at least a dozen previously unknown flaws in commercially available devices.

The first results using Tookan were published in 2010 [56] and a six-month licence was granted to Boeing to use the tool. In 2011, this transfer activity has continued, principally in combination with a major UK bank. In June, Tookan was used by Steel and Focardi two days of testing on devices belonging to the bank. Following these results, in September, a more significant contract was signed granting the bank 18 months of use of Tookan to test all their in-house equipment. Initial feedback has been very positive.

Tookan is the subject of a CSATT transfer action resulting in the hiring of an engineer, Romain Bardou, who started on September 1st. Early progress in re-implementing key parts of Tookan to improve modularity and overall code quality has been excellent. The next steps for Tookan are still being investigated: the Tookan project is the subject of a ‘qualification’ procedure by IT2 who will evaluate its suitability as the basis for a start-up company. At the same time other options are being considered, such as partnership with an existing SME. A decision is expected in mid-2012.

5.2. Orchids

Participants: Jean Goubault-Larrecq [correspondant], Hedi Benzina, Baptiste Gourdin, Nasr-Eddine Yousfi.

The ORCHIDS real-time intrusion detection system was created in 2003-04 at SECSI. After a few years where research and development around ORCHIDS was relatively quiet, several new things happened, starting from the end of 2010.

First, several companies and institutions expressed interest in ORCHIDS, among which, notably, EADS Cassidian, Thalès, Galois Inc. (USA), the French Direction Générale de l’Armement (DGA).

Second, Baptiste Gourdin was hired as a development engineer (Dec. 2010-Nov. 2011) on an Action de Développement Technologique (ADT). He improved Orchids in several ways. Its user interface benefitted from a complete revamping. New features were implemented, such as conformance with the IODEF and IDMEF standards, connection with vulnerability and network topology databases, the possibility to do forensics that synchronize past events to the state that the above databases were in at the time of the events, among others.

Nasr-Eddine Yousfi has followed up on Baptiste Gourdin, starting from December 2011, on an ITI engineer position allotted by INRIA’s CSATT.

Hedi Benzina implemented a tool on top of ORCHIDS, RuleGen, which allows one to write simple security policies that compile to ORCHIDS rules.

The efforts done in 2011 around ORCHIDS should be seen as the first steps in the creation of an open source consortium, which will be consolidated in the next years.
5.3. AKISS and SubVariant

Participant: Ștefan Ciobăcă.

AKISS (http://www.lsv.ens-cachan.fr/~ciobaca/akiss/) is a tool implementing a procedure for verifying trace equivalence (or equivalently may-testing equivalence) for bounded security processes with no else branches employing cryptographic primitives modeled by an optimally reducing rewrite system.

Trace equivalence can be used to model strong secrecy, vote-privacy and other security properties.

AKISS uses a fully-abstract encoding of symbolic traces into Horn clauses, thereby extending the KISS tool (http://www.lsv.ens-cachan.fr/~ciobaca/kiss/), which can only check static equivalence.

In order to get rid of the equational theory modeling the cryptographic primitives, AKISS employs algorithms for computing strongly complete sets of variants and complete set of unifiers of the SubVariant tool. AKISS is described in an article submitted to ESOP, in Chapter 5 of Ștefan Ciobăcă’s PhD thesis [12].

SubVariant (http://www.lsv.ens-cachan.fr/~ciobaca/subvariant/) is a tool for computing finite strongly complete set of variants modulo a convergent optimally reducing term rewriting system. SubVariant can also compute complete sets of equational unifiers for equational theories implemented by a convergent optimally reducing term rewriting system.

Complete sets of variants and the finite variant property were introduced in [59]. In [33], Ștefan Ciobăcă defines strongly complete sets of variants, which are more natural and more useful. Chapter 3 in Ștefan Ciobăcă’s PhD thesis describes extensively the algorithms behind SubVariant.
5. Software

5.1. ECPP

F. Morain has been continuously improving his primality proving algorithm called ECPP, originally developed in the early 1990s. Binaries for version 6.4.5 have been available since 2001 on his web page. Proving the primality of a 512 bit number requires less than a second on an average PC. His personal record is around 25,000 decimal digits, with the fast version he started developing in 2003. All of the code is written in C, and based on publicly available packages (GMP, mpfr, mpc, mpfrx).

5.2. SEA

Together with E. Schost and L. DeFeo, F. Morain has developed a new implementation of the SEA algorithm that computes the cardinality of elliptic curves over finite fields (large prime case, case \( p = 2 \)). It uses NTL and includes the more recent algorithms for solving all subtasks. The large prime case is relevant to cryptographical needs. The \( p = 2 \) case, though not directly useful, is a good testbed for the FAAST program of LDeFeo (see 5.4). This program forms a gforge project.

5.3. TIFA

The TIFA library (short for Tools for Integer FActorization) was initially developed in 2006 and has been continuously improved during the last few years. TIFA is made up of a base library written in C99 using the GMP library, together with stand-alone factorization programs and a basic benchmarking framework to assess the performance of each algorithm.

As of November 2011, the library includes the following algorithms:

- CFRAC (Continued Fraction factorization \([64]\))
- ECM (Elliptic Curve Method)
- Fermat (McKee’s “fast” variant of Fermat’s algorithm \([62]\))
- SIQS (Self-Initializing Quadratic Sieve \([35]\))
- SQUFOF (SQUARE FORM Factorization \([54]\))

The complete TIFA package has been registered at the French Agency for Software Protection (APP – \text{http://app.legalis.net/}) on June, 1st 2011 with the Inter Deposit Digital Number:

\text{IDDN.FR.001.220019.000.S.A.2011.000.31235}.

It is now available online at \text{http://www.lix.polytechnique.fr/Labo/Jerome.Milan/tifa/tifa.xhtml} and distributed under the Lesser General Public License, version 2.1 or later.

5.4. FAAST

The FAAST library is developed in C++ by L. De Feo and makes use of the NTL library. It implements the algorithms presented in \([4]\), plus other algorithms needed by the author for his research on explicit isogenies. Version 0.2.0, released on July 11 2009, is available at \text{http://www.lix.polytechnique.fr/Labo/Luca.De-Feo/FAAST/}. The source code is distributed under the General Public License version 2 or higher.

FAAST is a very efficient library for lattices of extensions of finite fields. Our aim is to add support for arbitrary finite fields, making it an essential building block for efficient computer algebra systems.
5.5. Quintix

The Quintix library is a Mathemagix package available at http://www.mathemagix.org/www/main/index.en.html. It is developed in C++ within the Mathemagix computer algebra system. It implements basic arithmetic for Galois rings and their unramified extensions, basic functions for the manipulation of Reed-Solomon codes and the complete Sudan list-decoding algorithm. It also implements the root-finding algorithms presented in [30]. The source code is distributed under the General Public License version 2 or higher.

Quintix is a very efficient library for Galois rings, extensions of Galois rings and root-finding in Galois rings.

5.6. APIP

As part of his activity in the PACE ANR, J. Milan completed, under the supervision of A. Enge, the development of APIP (Another Pairings Implementation in PARI), a PARI/GP module to compute state-of-the-art cryptographic pairings over elliptic curves. This module was intended to be an experimental framework for comparing the performances of the main cryptographic pairings with an emphasis on the standard 128, 192 and 256 bit high security levels.

APIP implements the Tate, Weil, ate and twisted ate pairings together with some optimal variants of the ate and twisted ate pairings for some elliptic curve families. Due to its very flexible architecture, it makes it easy to select several algorithm variants for each step of a pairing computation for a finer analysis.

Due to its emphasis on pairings for cryptographic purposes only, it is doubtful that the APIP module will be integrated in the upstream PARI/GP code base. We hope to be able to distribute APIP as an independent module in the near future, ideally under an open-source licence.
TYPICAL Project-Team

5. Software

5.1. Coq

Participants: Bruno Barras [Contact], Jean-Marc Notin, Arnaud Spiwack, Enrico Tassi.

Coq is a major proof system an the primary object and / or tool of our research. Its development is now mainly coordinated by the πr² INRIA Paris-Rocquencourt project-team, and some members of the TypiCal team are active developers of the system.

5.2. Coqfinitegroup

Participants: Cyril Cohen, Assia Mahboubi [Contact].

Coqfinitegroup is the development corresponding to the ongoing effort to formalize the proof of the Feit-Thompson theorem. It is probably the most advanced formal development of group theory today. Its current size is about 80.000 lines of (compact) Coq code. Assia Mahboubi and Cyril Cohen are actively participating to this long term formalization project.

5.3. Dedukti

Participants: Mathieu Boespflug [Contact], Gilles Dowek.

Dedukti is a universal proof checker, based on the λπ-calculus modulo formalism. Mainly developed by Mathieu Boespflug, it is distributed under the GNU licence. The main system includes about 2000 lines of Haskell.

5.4. Ssreflect

Participants: Assia Mahboubi [Contact], Enrico Tassi.

SSReflect is a proof language extension of Coq developed under Georges Gonthier (Microsoft Research). It was originally designed to make the formalization of the Four Color Theorem possible and has been evolving since. It is important to note that it is shipped with redesigned basic proof libraries. Members of the Typical are in charge of the documentation and distribution of this extension.
5. Software

5.1. Bocop

Participants: Pierre Martinon [correspondant], Vincent Grélard, Frédéric Bonnans.

The Bocop project aims to develop an open-source toolbox for solving optimal control problems, with collaborations with industrial and academic partners. Optimal control (optimization of dynamical systems governed by differential equations) has numerous applications in transportation, energy, process optimization, and biology. This project is supported by INRIA in the framework of an ADT, Action de Développement Technologique, 2010-2012.

The software uses some packages from the COIN-OR library, in particular the well-known interior-point nonlinear programming solver Ipopt. It also features a user-friendly interface in Scilab. See the web page http://www.bocop.org.

5.2. BiNoPe-HJ

Participants: Hasnaa Zidani [correspondant], Olivier Bokanowski, Nicolas Forcadel, Jun-Yi Zhao.

This project aims at developing sequential and parallel MPI/openMP C++ solvers for the approximation of Hamilton-Jacobi-Bellman (HJB) equations in a d-dimensional space. The main goal is to provide an HJB solvers that can work in dimension d (limited by the machine’s capacity). The solver outputs can be visualized with Matlab or Paraview (via VTK files).

The HJB Solver has been actively developed under a partnership between COMMANDS and the SME HPC-project in the period between December 2009 to November 2011. See also http://www.ensta-paristech.fr/~zidani/BiNoPe-HJ.

We release two versions:

- HJB-SEQUENTIAL-REF: sequential version that can run on any machine
- HJB-PARALLEL-REF: parallel version that can run only on multi-core architectures.

5.3. Shoot

Participant: Pierre Martinon [correspondant].

Shoot was designed for the resolution of optimal control problems via indirect methods (necessary conditions, Pontriagyn’s Maximum Principle). Such methods transform the original problem into finding a zero of a certain shooting function. The package offers several choices of integrators and solvers, and can handle control discontinuities. Features also include the use of variational equations to compute the Jacobian of the shooting function, as well as homotopy and grid shooting techniques for easier initialization.

See also the web page http://www.cmap.polytechnique.fr/~martinon/codes.html.
5. Software

5.1. RODIN

Participant: Grégoire Allaire [correspondant].

One of the aims of the RODIN project is to develop a new shape optimization software for solid structures in the framework of the SYSTUS code developed by ESI-group. The work has just started.

5.2. FreeFem++ Toolboxes

5.2.1. Shape optimization toolbox in FreeFem++

Participants: Grégoire Allaire, Olivier Pantz.

We propose several FreeFem++ routines which allow the users to optimize the thickness, the geometry or the topology of elastic structures. All examples are programmed in two space dimensions. These routines have been written by G. Allaire, B. Boutin, C. Dousset, O. Pantz. A web page of this toolbox is available at http://www.cmap.polytechnique.fr/~allaire/freefem_en.html.

We also have written a C++ code to solve the Hamilton Jacoby equation used in the Level-set shape optimization method. This code has been linked with FreeFem++ routines.

5.2.2. Inverse Problems for Stokes Flows

Participants: Armin Lechleiter [correspondant], Tobias Rienmüller.

This software solves shape reconstruction inverse problems for the Stokes-Brinkmann equations modelling porous penetrable inclusions inside a free flow. This problem is motivated by non-destructive testing in pipes and reservoirs. The factorization method is used to solve the inverse problem.

5.2.3. Inverse shape and medium problem for thin coatings

Participant: Nicolas Chaulet.

We developed a FreeFem++ toolbox which retrieve an obstacle and two coefficients that define a generalized impedance boundary condition form a few far field data in dimension 2. The reconstruction algorithm relies on regularized non linear optimization technique. The toolbox also contains a forward solver for the scattering of acoustic waves by obstacle on which a generalized impedance boundary condition is applied using an approximate Dirichlet-to-Neuman map to bound the computational domain.

5.2.4. Inverse shape problems for axisymmetric eddy current problems

Participants: Armin Lechleiter, Zixian Jiang [correspondant].

This FreeFem++ toolbox solves inverse problems for an axisymmetric eddy current model using shape optimization techniques. The underlying problem is to find inclusions in a tubular and unbounded domain. The direct scattering problems are solved using an adaptive finite element method, and Dirichlet-to-Neumann operators are used to implement the transparent boundary conditions. Based on the shape derivative of an inclusion with respect to the domain, the toolbox offers regularized iterative algorithms to solve the inverse problem.

5.2.5. Contact managements

Participant: Olivier Pantz.
We have developed a toolbox running under Freefem++ in order to take into account the non-intersection constraints between several deformable bodies. This code has been used to treat contacts between red blood cells in our simulations, but also between genuine non linear elastic structure. It can handle both contacts and self-contacts.

Moreover, a toolbox based on the Penalization method has also been developed.

5.2.6. De-Homogenization

Participant: Olivier Pantz.

We have developed a code under Freefem++ that implements our De-Homogenization method. It has been used to solve the compliance minimization problem of the compliance of an elastic shape. In particular, it enables us to recover well known optimal Michell’s trusses for shapes of low density.

5.3. Scilab and Matlab Toolboxes

5.3.1. Shape optimization toolbox in Scilab

Participant: Grégoire Allaire [correspondant].

Together with Georgios Michailidis, we improved a Scilab toolbox for 2-d shape and topology optimization by the level set method which was originally produced by Anton Karrman and myself. The routines, a short user’s manual and several examples are available on the web page.

5.3.2. Conformal mapping method

Participant: Houssem Haddar [correspondant].

This Scilab toolbox is dedicated to the resolution of inverse 2-D electrostatic problems using the conformal mapping method introduced by Akdumann, Kress and Haddar. The toolbox treats the cases of a simply connected obstacle with Dirichlet, Neumann or impedance boundary conditions or a simply connected inclusion with a constant conductivity.

5.3.3. Direct and inverse problems in waveguides

Participants: Armin Lechleiter [correspondant], Dinh Liem Nguyen.

This Matlab toolbox includes fast solvers for direct and inverse scattering problems in planar 3D waveguides for inhomogeneous media. The direct scattering problems are solved using an spectral integral equation approach relying on the Lippmann-Schwinger integral equation, discretized as a Galerkin method via the fast Fourier transform. The toolbox includes preconditioning by a two-grid scheme and multipole expansions coupled to the spectral solver to allow for multiple scattering objects. The inverse problem to find the shape of the scattering object from near-field measurements is solved using a Factorization method.

5.4. Sampling methods for inverse problems

5.4.1. Samplings-2d

Participant: Houssem Haddar [correspondant].

This software is written in Fortran 90 and is related to forward and inverse problems for the Helmholtz equation in 2-D. It includes three independent components. The first one solves to scattering problem using integral equation approach and supports piecewise-constant dielectrics and obstacles with impedance boundary conditions. The second one contains various samplings methods to solve the inverse scattering problem (LSM, RGLSM(s), Factorization, MuSiC) for near-field or far-field setting. The third component is a set of post processing functionalities to visualize the results.
See also the web page http://sourceforge.net/projects/samplings-2d/.

- License: GPL
- Type of human computer interaction: sourceforge
- OS/Middleware: Linux
- Programming language: Fortran
- Documentation: fichier

5.4.2. **Samplings-3d**  
**Participant:** Houssem Haddar [correspondant].

This software is written in Fortran 90 and is related to forward and inverse problems for the Helmholtz equation in 3-D. It contains equivalent functionalities to samplings-2d in a 3-D setting.

5.4.3. **Time domain samplings-2d**  
**Participants:** Houssem Haddar [correspondant], Armin Lechleiter.

This software is written in Fortran 90 and is related to forward and inverse problems for the time dependent wave equation in 2-D. The forward solver is based on a FDTD method with PMLs. The inverse part is an implementation of the linear sampling method in a near field setting and the factorization method in a far field setting.

5.4.4. **Factorization Method for EIT**  
**Participant:** Giovanni Migliorati.

We developed a numerical code that implements the Factorization Method applied to the Continuous Model, in the framework of Electrical Impedance Tomography featuring an inhomogeneous background. The numerical scheme relies on the approximation by the finite element method of the solution to the dipole-like Neumann boundary-value problem. Two regularization techniques are implemented, i.e. the Tikhonov regularization embedding Morozov principle, and the classical Picard Criterion. The numerical analysis of the method and the results obtained are presented in the INRIA RR-7801, November 2011.

5.5. **FVforBlochTorrey**  
**Participant:** Jing-Rebecca Li [correspondant].

Finite volume code in Fortran 90 to solve the multiple compartment Bloch Torrey equation in 2D and 3D to simulate the bulk magnetization of a sample under the influence of a diffusion gradient. We couple a mass-conserving finite volume discretization in space with a stable time discretization using an explicit Runge-Kutta-Chebyshev method and we are able to solve the Bloch-Torrey PDE in multiple compartments for an arbitrary diffusion sequence with reasonable accuracy for moderately complicated geometries in computational time that is on the order of tens of minutes per bvalue on a laptop computer. See also the web page http://www.cmap.polytechnique.fr/~jingrebeccali/.
DISCO Team

5. Software

5.1. YALTA

Participants: David Avanessoff [correspondent], Catherine Bonnet, André Fioravanti [UNICAMP, correspondent].

The YALTA package is dedicated to the study of classical and fractional systems with delay in the frequency-domain. Its objective is to provide basic but important information such as, for instance, the position of the neutral chains of poles and unstable poles, as well as the root locus with respect to the delay of the system. The corresponding algorithms are based on recent theoretical results (see, for instance, [14] and [112]) and on classical continuation methods exploiting the particularities of the problem [113], [18]. Some refinements have been included this year in order to deal with systems with tricky numerical behaviour. The YALTA package will be available in the first semester of 2012.

5.2. OreModules

Participants: Alban Quadrat [correspondent], Daniel Robertz [Univ. Aachen], Frédéric Chyzak [INRIA Rocquencourt, Algorithms Project].

The OreModules package [106], based on the commercial Maple package Ore_algebra [107], is dedicated to the study of linear multidimensional systems defined over certain Ore algebras of functional operators (e.g., ordinary or partial differential systems, time-delay systems, discrete systems) and their applications in mathematical systems theory, control theory and mathematical physics. OreModules is original because it combines the recent developments of the Gröbner bases over some noncommutative polynomial rings [115], [117] and new algorithms of algebraic analysis in order to effectively check classical properties of module theory (e.g., existence of a non-trivial torsion submodule, torsion-freeness, reflexiveness, projectiveness, stably freeness, freeness), it gives their system-theoretical interpretations (existence of autonomous elements or successive parametrizations, existence of minimal/injective parametrizations or Bézout equations) [121], [120], [105] and it computes important tools of homological algebra (e.g., (minimal) free resolutions, split exact sequences, extension functors, projective or Krull dimensions, Hilbert power series). The abstract language of homological algebra used in the algebraic analysis approach carries over to the implementations in OreModules: up to the choice of the domain of functional operators which occurs in a given system, all algorithms are stated and implemented in sufficient generality such that linear systems defined over the Ore algebras developed in the Ore_algebra package are covered at the same time. Applications of the OreModules package to mathematical systems theory, control theory and mathematical physics are illustrated in a large library of examples. The binary of the package is freely available at http://wwwb.math.rwth-aachen.de/OreModules/.

5.3. Stafford

Participants: Alban Quadrat [correspondent], Daniel Robertz [Univ. Aachen].

The Stafford package of OreModules [106] contains an implementation of two constructive versions of Stafford’s famous but difficult theorem [132] stating that every ideal over the Weyl algebra $A_n(k)$ (resp., $B_n(k)$) of partial differential operators with polynomial (resp., rational) coefficients over a field $k$ of characteristic 0 (e.g., $k = \mathbb{Q}, \mathbb{R}$) can be generated by two generators. Based on this implementation and algorithmic results developed in [128] by the authors of the package, two algorithms which compute bases of free modules over the Weyl algebras $A_n(\mathbb{Q})$ and $B_n(\mathbb{Q})$ have been implemented. The development of the Stafford package was motivated by the problem of computing injective parametrizations of underdetermined linear systems of partial differential equations with polynomial or rational coefficients (the so-called Monge problem), differential flatness, the reduction and decomposition problems and Serre’s reduction problem. To our knowledge, the Stafford package is the only implementation of Stafford’s theorems nowadays available. The binary of the package is freely available at http://wwwb.math.rwth-aachen.de/OreModules/.
5.4. QuillenSuslin

Participants: Alban Quadrat [correspondent], Anna Fabiańska [Univ. Aachen].

The Quillen-Suslin package [110] contains an implementation of the famous Quillen-Suslin theorem [130], [133]. In particular, this implementation allows us to compute bases of free modules over a commutative polynomial ring with coefficients in a field (mainly $\mathbb{Q}$) and in a principal ideal domain (mainly $\mathbb{Z}$). The development of the Quillen-Suslin package was motivated by different constructive applications of the Quillen-Suslin theorem in multidimensional systems theory [110] (e.g., the Lin-Bose conjectures, the computation of (weakly) left/right/doubly coprime factorizations of rational transfer matrices, the computation of injective parametrizations of flat linear multidimensional systems with constant coefficients, the reduction and decomposition problems, Serre’s reduction problem). To our knowledge, the Quillen-Suslin package is the only implementation of the Quillen-Suslin theorem nowadays available. The binary of the package is freely available at http://wwwb.math.rwth-aachen.de/QuillenSuslin .

5.5. OreMorphisms

Participants: Alban Quadrat [correspondent], Thomas Cluzeau [ENSIL, Univ. Limoges].

The OreMorphisms package [109] of OreModules [105] is dedicated to the implementation of homological algebraic tools such as the computations of homomorphisms between two finitely presented modules over certain noncommutative polynomial algebras (Ore algebras), of kernel, coimage, image and cokernel of homomorphisms, Galois transformations of linear multidimensional systems and idempotents of endomorphism rings. Using the packages Stafford and Quillen-Suslin, the factorization, reduction and decomposition problems can be constructively studied for different classes of linear multidimensional systems. Many linear systems studied in engineering sciences, mathematical physics and control theory have been factorized, reduced and decomposed by means of the OreMorphisms package. The binary of the package is freely available at http://www-sop.inria.fr/members/Alban.Quadrat/OreMorphisms/index.html .

5.6. JanetMorphisms

Participants: Alban Quadrat [correspondent], Thomas Cluzeau [ENSIL, Univ. Limoges], Daniel Robertz [Univ. Aachen].

The JanetMorphisms package is dedicated to a new mathematic approach to quasilinear systems of partial differential equations (e.g., Burger’s equation, shallow water equations, Euler equations of a compressible fluid) based on algebraic analysis and differential algebra techniques [44]. See Section 6.2. This package computes symmetries, first integrals of motion, conservation laws, study Riemann invariants... The JanetMorphisms package is based on the Janet package (http://wwwb.math.rwth-aachen.de/Janet/).

5.7. PurityFiltration

Participant: Alban Quadrat [correspondent].

The PurityFiltration package, built upon the OreModules package, is an implementation of a new effective algorithm obtained in [97], [77], [76] which computes the purity/grade filtration [102], [103] of linear functional systems (e.g., partial differential systems, differential time-delay systems, difference systems) and equivalent block-triangular matrices. See Section 6.1. This package is used to compute closed form solutions of over/underdetermined linear partial differential systems which cannot be integrated by the standard computer algebra systems such as Maple and Mathematica. This package will soon be available.

5.8. AbelianSystems

Participants: Alban Quadrat [correspondent], Mohamed Barakat [Univ. Kaiserslautern].
The **ABELIANSYSTEMS** package is an implementation of an algorithm developed in [97], [77], [76] for the computation of the purity/grade filtration [102], [103] in the powerful homalg package of GAP 4 dedicated to constructive homological algebra methods, and developed by Barakat (University of Kaiserslautern) and his collaborators (http://homalg.math.rwth-aachen.de/). This package both supersedes the existing **PURITYFILTRATION** package which uses the non-efficient Maple Gröbner basis computation (see Section 5.7), and the original homalg procedure which computes purity filtration by means of time-consuming spectral sequences. Using the homalg package philosophy, the **ABELIANSYSTEMS** package can be used for the computation of the purity filtration of objects in different constructive abelian categories such as sheaves over projective varieties as demonstrated in the homalg package called **Sheaves** (see http://homalg.math.rwth-aachen.de/).

### 5.9. SystemTheory

**Participants:** Alban Quadrat [correspondent], Thomas Cluzeau [ENSIL, Univ. Limoges], Markus Lange-Hegermann [Univ. Aachen], Mohamed Barakat [Univ. Kaiserslautern].

The **SYSTEMTHEORY** package is a homalg based package dedicated to mathematical systems. This package, still in development, will include the algorithms developed in the **OREMODULES** and **OREMORPHISMS** packages. It currently contains an implementation of the **OREMORPHISMS** procedures which handle the decomposition problem aiming at decomposing a module/system into direct sums of submodules/subsystems.
GECO Team (section vide)
5. Software

5.1. Boîte à outil Maxplus de SCILAB/Maxplus toolbox of Scilab

Trois chercheurs du groupe (S. Gaubert, J.-P. Quadrat, et G. Cohen) ont développé (à partir d’une première version réalisée par M. Mc Gettrick) la boîte à outils Maxplus de Scilab, qui est téléchargeable librement parmi les contributions du site Scilab, et qui est maintenant intégrée par défaut dans Scicoslab. Cette boîte à outils implémente l’ensemble du calcul numérique linéaire max-plus, elle comprend en particulier le stockage creux des matrices, et des algorithmes efficaces pour le calcul de la valeur propre basées sur les itérations sur les politiques. Elle a été utilisées par plusieurs chercheurs, voir notamment [84], [146]. Il faut aussi noter que le groupe de L. Hardouin, du LISA/Istia, a complété la boîte à outils Maxplus en interfaçant leur propre librairie C++, qui permet le calcul des séries de transfert de graphes d’événements temporisés.

English version

Three researchers of the team (S. Gaubert, J.-P. Quadrat, and G. Cohen, building on a preliminary version of M. McGettrick) have developed and released the Maxplus toolbox of Scilab, which is freely available among the contributions on the Scilab web site, and which is now included by default in Scicoslab. It implements all basic linear algebra functionalities, with a special attention to large sparse matrices, including efficient algorithms for eigenvalue computation based on policy iteration. The software has been used by several researchers in their work, including [84], [146]. It should be noted that the team of L. Hardouin, from LISA/Istia, has completed the toolbox by interfacing their own C++ library computing the transfer series of a timed event graph.

5.2. Itérations sur les politiques pour les jeux stochastiques à somme nulle/Policy iterations for zero sum stochastic games

L’algorithme d’itérations sur les politiques pour les jeux stochastiques à somme nulle pour le cas de paiements ergodiques (gain moyen par unité de temps), et dégénérés de type “multi-chaîne” a été introduit dans [103]. Plusieurs stages ont permis l’implémentation partielle en Scilab, C ou C++, et le test de ce type d’algorithmes (voir le travail de Vishesh Dhingra [115]), ou de son couplage avec la résolution de systèmes linéaires par des méthodes multigrilles algébriques (stage de Shantanu Gangal en 2007). Le travail de thèse de Sylvie Detournay, qui porte sur le couplage entre itérations sur les politiques et méthodes multigrilles algébriques, voir le § 6.4.1 ci-dessous, a permis le développement d’un programme complet. Le code écrit par Sylvie Detournay (en C) a été déposé sur INRIAGForge. Pour le moment il n’est accessible qu’aux membres de l’équipe.

English version

The policy iteration algorithm for zero sum repeated games with ergodic payoff (i.e. mean payoff per time unit), and in degenerate “multichain” cases, has been introduced in [103]. Several internships allowed us to implement in Scilab, C or C++, and to test such algorithms (see the work of Vishesh Dhingra [115]), or its combination with the resolution of linear systems by algebraic multigrid methods (internship of Shantanu Gangal in 2007). The PhD thesis work of Sylvie Detournay, who concerns the combination of policy iterations with algebraic multigrid methods, see § 6.4.1 below, allowed us to develop a complete program. The program written by Sylvie Detournay (in C language) has been posted on INRIAGForge. For the moment it can only be seen by members of the team.
5.3. TPLib: bibliothèque pour la manipulation de polyèdres tropicaux/TPLib: tropical polyhedra library

TPLib est une bibliothèque écrite en OCaml qui permet de manipuler des polyèdres tropicaux. Elle est distribuée sous license LGPL https://gforge.inria.fr/projects/tplib .

Cette bibliothèque implémente notamment des algorithmes permettant de passer d’une représentation externe d’un polyèdre à une représentation interne, ou inversement (voir § 6.2.1 pour plus de détails). Elle fournit également toutes les primitives permettant d’utiliser les polyèdres tropicaux en tant que domaine abstrait numérique, afin de déterminer des invariants de programmes faisant intervenir les opérations min et max (voir [ 82 ]).

TPLib est aujourd’hui utilisé dans le logiciel Polymake [ 128 ], développé à la Technische Universität Darmstadt (Allemagne). Ce dernier logiciel constitue une boîte à outils permettant de manipuler des nombreux objets mathématiques (polytopes convexes, complexes polyédraux, graphes, matroïdes, polytopes tropicaux). Une interface à la bibliothèque de domaines abstraits numériques APRON [ 139 ] est également en cours de développement.

English version

TPLib is a library written in OCaml, which allows to manipulate tropical polyhedra. It is distributed under LGPL https://gforge.inria.fr/projects/tplib .

This library implements algorithms allowing to pass from an external representation of a polyhedron to an internal description, or inversely (see § 6.2.1 for more details). It also provides all the primitives allowing to use tropical polyhedra as an numerical abstract domain, in order to determine program invariants involving the operators min and max (see [ 82 ]).

TPLib is now used in the software Polymake [ 128 ], developed in Technische Universität Darmstadt (Germany). Polymake is a toolbox allowing to manipulate mathematic objects such as convex polytopes, polyhedral complexes, graphs, matroids, and tropical polytopes. An interface to the numerical abstract domain APRON [ 139 ] is also under development.
5. Software

5.1. FracLab

**Participants:** Paul Balança, Jacques Lévy Véhel [correspondant].

FracLab was developed for two main purposes:

1. propose a general platform allowing research teams to avoid the need to re-code basic and advanced techniques in the processing of signals based on (local) regularity.
2. provide state of the art algorithms allowing both to disseminate new methods in this area and to compare results on a common basis.

FracLab is a general purpose signal and image processing toolbox based on fractal, multifractal and local regularity methods. FracLab can be approached from two different perspectives:

- (multi-) fractal and local regularity analysis: A large number of procedures allow to compute various quantities associated with 1D or 2D signals, such as dimensions, Hölder and 2-microlocal exponents or multifractal spectra.
- Signal/Image processing: Alternatively, one can use FracLab directly to perform many basic tasks in signal processing, including estimation, detection, denoising, modeling, segmentation, classification, and synthesis.

A graphical interface makes FracLab easy to use and intuitive. In addition, various wavelet-related tools are available in FracLab.

FracLab is a free software. It mainly consists of routines developed in MatLab or C-code interfaced with MatLab. It runs under Linux, MacOS and Windows environments. In addition, a “stand-alone” version (i.e. which does not require MatLab to run) is available.

FracLab has been downloaded several thousands of times in the last years by users all around the world. A few dozens laboratories seem to use it regularly, with more than two hundreds registered users. Our ambition is to make it the standard in fractal softwares for signal and image processing applications. We have signs that this is starting to become the case. To date, its use has been acknowledged in more than two hundreds research papers in various areas such as astrophysics, chemical engineering, financial modeling, fluid dynamics, internet and road traffic analysis, image and signal processing, geophysics, biomedical applications, computer science, as well as in mathematical studies in analysis and statistics (see [http://fraclab.saclay.inria.fr/](http://fraclab.saclay.inria.fr/) for a partial list with papers). In addition, we have opened the development of FracLab so that other teams worldwide may contribute. Additions have been made by groups in Australia, England, the USA, and Serbia.
SELECT Project-Team

5. Software

5.1. MIXMOD software

Participants: Gilles Celeux [Correspondant], Erwan Le Pennec.

MIXMOD is being developed in collaboration with Christophe Biernacki, Florent Langrognet (Université de Franche-Comté) and Gérard Govaert (Université de Technologie de Compiègne). MIXMOD (MIXture MODelling) software fits mixture models to a given data set with either a clustering or a discriminant analysis purpose. MIXMOD uses a large variety of algorithms to estimate mixture parameters, e.g., EM, Classification EM, and Stochastic EM. They can be combined to create different strategies that lead to a sensible maximum of the likelihood (or completed likelihood) function. Moreover, different information criteria for choosing a parsimonious model, e.g. the number of mixture component, some of them favoring either a cluster analysis or a discriminant analysis view point, are included. Many Gaussian models for continuous variables and multinomial models for discrete variable are available. Written in C++, MIXMOD is interfaced with SCILAB and MATLAB. The software, the statistical documentation and also the user guide are available on the Internet at the following address: http://www.mixmod.org.

Since this year, MIXMOD has a proper graphical user interface (Version 1) which has been presented at the MIXMOD day in Lyon in December 2010. A version of MIXMOD in R is forthcoming.

Erwan Le Pennec with the help of Serge Cohen has proposed a spatial extension in which the mixture weights can vary spatially.
5. Software

5.1. MoGo

Participants: Olivier Teytaud [correspondent], Hassen Doghmen, Jean-Baptiste Hoock.

MoGo and its Franco-Taiwanese counterpart MoGoTW is a Monte-Carlo Tree Search program for the game of Go, which made several milestones of computer-Go in the past (first wins against professional players in 19x19; first win with disadvantageous side in 9x9 Go); MoGo has had new developments as follows:

- A Meta-MCTS module (inspired by the collaboration with Tristan Cazenave in the ANR EXPLO-RA project), which provided both a huge opening book in 9x9 and an approximate solving of 7x7 Go[24].
- Following the “poolRave” modification, introduction of machine learning and statistics into MCTS, such as:
  - Bernstein Races [41] (for offline educating Monte-Carlo simulations).
These developments have been summarized and compared in [93].
- Variants of Go: we tested variants of Go, in particular blind variants; this suggests that in such frameworks playing theoretically suboptimal moves helps a lot, because such unnatural moves are harder to memorize. A preliminary related publication is [25]; some additional results are to be published. Another interesting variant is random-Go, starting from a randomly generated board; such situations are much harder for humans, and, interestingly, our program was competitive in front of a 6D player (ranked 4th in a world amateur championship and former French champion) on a 19x19 board[40].

MoGo’s development team was awarded the 2010 ChessBase award for the best contribution to Computer-Games.

5.2. Covariance Matrix Adaptation Evolution Strategy

Participant: Nikolaus Hansen [correspondent].

The Covariance Matrix Adaptation Evolution Strategy (CMA-ES) is one of the most powerful continuous domain evolutionary algorithms. The CMA-ES is considered state-of-the-art in continuous domain evolutionary computation\(^3\) and has been shown to be highly competitive on different problem classes. The algorithm is widely used in research and industry as witnessed by hundreds of published applications. We provide source code for the CMA-ES in C, Java, Matlab, Octave, Python, and Scilab including the latest variants of the algorithm.

Links: [http://www.lri.fr/~hansen/cmaes_inmatlab.html](http://www.lri.fr/~hansen/cmaes_inmatlab.html)

5.3. Comparing Continuous Optimizers

Participants: Nikolaus Hansen [correspondent], Raymond Ros, Anne Auger, Marc Schoenauer.

COCO (Comparing Continuous Optimizers) is a platform for systematic and sound comparisons of real-parameter global optimizers. COCO provides benchmark function testbeds (noiseless and noisy) and tools for processing and visualizing data generated by one or several optimizers. The code for processing experiments is provided in Matlab and C. The post-processing code is provided in Python. The code has been improved and used for the GECCO 2009 and 2010 workshops on “Black Box Optimization Benchmarking” (BBOB) (see Section 6.4), and will serve as a basis for the test platform in the CSDL project.

---

5.4. GridObservatory

Participants: Michèle Sebag, Cécile Germain-Renaud [correspondent], Julien Nauroy, Yusik Kim.

The Grid Observatory software suite collects and publishes traces of the EGI (European Grid Initiative) grid usage. With the release and extensions of its portal, the Grid Observatory has made a database of grid usage traces available to the wider computer science community. These data are stored on the grid, and made accessible through a web portal without the need of grid credentials. More than 100 users are currently registered. The GO is supported by an INRIA ADT (Action de Développement Technologique).

In 2011, the suite has been extended to energy consumption. The first barrier to improved energy efficiency of IT systems is the lack of large-scale collections of experimental data. The Green Computing Observatory (GCO), part of the GO initiative monitors a large computing center (Laboratoire de l’Accélérateur Linéaire - LAL) within the EGI grid, and publishes the data through the Grid Observatory. The GCO is supported by the CNRS PEPS program, and by University Paris-Sud through the MRM (Moyens de Recherche Mutualisés) program.

Portal site: http://www.grid-observatory.org
4. Software

4.1. Varna

Participants: Yann Ponty [correspondant], Alain Denise.

Varna is a tool for the automated drawing, visualization and annotation of the secondary structure of RNA, designed as a companion software for web servers and databases. Varna implements four drawing algorithms, supports input/output using the classic formats dbn, ct, bpseq and RNAML and exports the drawing, either as a bitmap (JPEG, PNG) or as a vector picture (SVG, EPS and XFIG). It also allows manual modification and structural annotation of the resulting drawings using either an interactive point and click approach, within a web server or through command-line arguments. Varna is a free software distributed under the terms of the GPLv3.0 license and available at http://varna.lri.fr.

Varna is currently used by RNA scientists (Cited by 44 research articles since its presentation in Fall of 2009), web servers such as the BoulderAle webserver (http://www.microbio.me/boulderale/), the TFold webserver (http://tfold.ibisc.univ-evry.fr/TFold/), the CyloFold webserver (http://cylofold.abcc.ncifcrf.gov/), and by databases such as the IRESITE database (http://iresite.org/), sRNATarBase (http://ccb.bmi.ac.cn/srnatarbase/) and the RFAM database (http://rfam.sanger.ac.uk/), the main source of sequence/structure data for RNA scientist, to display secondary structures. It is also used as an integrated component within Jalview, arguably one of the leading sequence alignment editor (http://www.jalview.org/).

4.2. GeneValorization

Participant: Sarah Cohen-Boulakia [correspondant].

High-throughput technologies provide fundamental information concerning thousands of genes. Many of the current research laboratories daily use one or more of these technologies and end-up with lists of genes. Assessing the originality of the results obtained includes being aware of the number of publications available concerning individual or multiple genes and accessing information about these publications. Faced with the exponential growth of publications available and number of genes involved in a study, this task is becoming particularly difficult to achieve. We introduce GeneValorization, a web-based tool which gives a clear and handful overview of the bibliography available corresponding to the user input formed by (i) a gene list (expressed by gene names or ids from ENTREZGENE) and (ii) a context of study (expressed by keywords). From this input, GeneValorization provides a matrix containing the number of publications with co-occurrences of gene names and keywords. Graphics are automatically generated to assess the relative importance of genes within various contexts. Links to publications and other databases offering information on genes and keywords are also available. To illustrate how helpful GeneValorization is, we have considered the gene list of the OncotypeDX prognostic marker test. it is available at http://bioguide-project.net/gv.

4.3. HSIM

Participant: Patrick Amar [correspondant].

HSIM is a simulation tool for studying the dynamics of biochemical processes in a virtual bacteria. The model is given using a language based on probabilistic rewriting rules that mimics the reactions between biochemical species. HSIM is a stochastic automaton which implements an entity-centered model of objects. This kind of modelling approach is an attractive alternative to differential equations for studying the diffusion and interaction of the many different enzymes and metabolites in cells which may be present in either small or large numbers. This software is freely available at http://www.lri.fr/~pa/Hsim; A compiled version is available for the Windows, Linux and MacOSX operating systems.
4.4. Cartaj

Participants: Alain Denise [correspondant], Alexis Lamiable.

Cartaj is a software that automatically predicts the topological family of three-way junctions in RNA molecules, from their secondary structure only. The Cartaj software http://cartaj.lri.fr that implements our method can be used online. It is also meant for being part of RNA modelling softwares and platforms. The methodology and the results of CARTAJ are presented in [16].
4. Software

4.1. PYGMALION

PyGMAlion (Plant Growth Model Analysis, Identification and Optimization) has become the leading development project in the group. The objective is on one hand to provide modelers with mathematical and statistical tools for model analysis, and on the other hand to capitalize in the same software the different methods developed in the group. The basic idea is that provided the modeler writes his dynamic system of plant growth in a simple frame (defining model state variables, state function, parameters, external inputs, and model observations) then some parameter estimation methods are available, as well as sensitivity analysis, evaluation of criteria for model selection and data assimilation.
5. Software

5.1. Deformable Registration Software

Participants: Nikos Paragios [Correspondant], Ben Glocker, Aristeidis Sotiras, Nikos Komodakis.

DROP is a deformable registration platform in C++ for the medical imaging community (publicly available at http://www.mrf-registration.net) developed mainly at Ecole Centrale, Technical University of Munich and University of Crete. This is the first publicly available platform which contains most of the existing metrics to perform registration under the same concept. The platform is used for clinical research from approximately 3,000 users worldwide.

5.2. Fast Primal Dual Strategies for Optimization of Markov Random Fields

Participants: Nikos Komodakis [Correspondant], Nikos Paragios, George Tziritas.

FASTPD is an optimization platform in C++ for the computer vision and medical imaging community (publicly available at http://www.csd.uoc.gr/~komod/FastPD/) developed mainly at Ecole Centrale and University of Crete. This is the most efficient publicly available platform in terms of a compromise of computational efficiency and ability to converge to a good minimum for the optimization of generic MRFs. The platform is used from approximately 1,500 users worldwide.

5.3. imaGe-based Procedural Modeling Using Shape Grammars

Participants: Olivier Teboul [Correspondant], Iasonas Kokkinos, Panagiotis Koutsourakis, Loïc Simon, Nikos Paragios.

GRAPEs is a generic image parsing library based on reinforcement learning. It can handle grammars (binary-split, four-color, Hausmannian) and image-based rewards (Gaussian mixtures, Randomized Forests) of varying complexity while being modular and computationally efficient both in terms of grammar and image rewards. The platform is used from approximately 500 users worldwide.

5.4. Texture Analysis Using Modulation Features and Generative Models

Participants: Iasonas Kokkinos [Correspondant], Georgios Evangelopoulos.

TeXMeG is a front-end for texture analysis and edge detection platform in Matlab that relies on Gabor filtering and image demodulation (publicly available at http://cvsp.cs.ntua.gr/software/texture/). Includes frequency- and time-based definition of Gabor- and other Quadrature-pair filterbanks, demodulation with the Regularized Energy Separation Algorithm and Texture/Edge/Smooth classification based on MDL criterion. The platform is used from approximately 250 users worldwide.
5. Software

5.1. Mayavi

Participant: Gaël Varoquaux [Correspondant].

Mayavi is the most used scientific 3D visualization Python software (http://mayavi.sourceforge.net/). It has been developed by Prabhu Ramachandran (IIT Bombay) and Gaël Varoquaux (PARIETAL, INRIA Saclay). Mayavi can be used as a visualization tool, through interactive command line or as a library. It is distributed under Linux through Ubuntu, Debian, Fedora and Mandriva, as well as in PythonXY and EPD Python scientific distributions. Mayavi is used by several software platforms, such as PDE solvers (fipy, sfepy), molecule visualization tools (http://pyrx.scripps.edu) and brain connectivity analysis tools (connectomeViewer).

See also the web page http://mayavi.sourceforge.net/ and the following paper http://hal.inria.fr/inria-00528985/en.

- Version: 3.4.0

5.2. Nipy

Participants: Bertrand Thirion [correspondant], Virgile Fritsch, Gaël Varoquaux.

Nipy is an open-source Python library for neuroimaging data analysis, developed mainly at Berkeley, Stanford, MIT and Neurospin. It is open to any contributors and aims at developing code and tools sharing. Some parts of the library are completely developed by Parietal and LNAO (CEA, DSV, Neurospin). It is devoted to algorithmic solutions for various issues in neuroimaging data analysis. All the nipy project is freely available, under BSD licence. It is available in NeuroDebian.

See also the web page http://nipy.org.

- Version: 0.2

5.3. MedINRIA

Participants: Pierre Fillard [correspondant], Sergio Medina, Viviana Siless.

MedINRIA is a free collection of softwares developed within the ASCLEPIOS, ATHENA and VISAGES research projects. It aims at providing to clinicians state-of-the-art algorithms dedicated to medical image processing and visualization. Efforts have been made to simplify the user interface, while keeping high-level algorithms. MedINRIA is available for Microsoft windows XP/Vista, Linux Fedora Core, MacOSX, and is fully multithreaded.

See also the web page http://med.inria.fr/.

- Version: 2.0

5.4. Scikit learn

Participants: Bertrand Thirion [correspondant], Gaël Varoquaux, Alexandre Gramfort, Fabian Pedregosa, Virgile Fritsch.
Scikit-learn is open-source a machine learning toolkit written in Python/C that provides generic tools to learn information for the classification of various kinds of data, such as images or texts. It is tightly associated to the scientific Python software suite (numpy/scipy) for which it aims at providing a complementary toolkit for machine learning (classification, clustering, dimension reduction, regression). There is an important focus on code quality (API consistency, code readability, tests, documentation and examples), and on efficiency, as the scikit-learn compares favorably to state-of-the-art modules developed in R in terms of computation time or memory requirements. Scikit-learn is currently developed by about 30 contributors, but the core developer team has been with the Parietal INRIA team at Saclay-Île-de- France since January 2010. The scikit-learn has recently become the reference machine learning library in Python.

- Version: 0.9
- Programming language: Python, C/Cython
5. Software

5.1. APMC-CA

Participants: Sylvain Peyronnet [correspondant], Joel Falcou, Pierre Esterie, Khaled Hamidouche, Alexandre Borghi.

The APMC model checker implements the state-of-the-art approximate probabilistic model checking methods. Last year we develop a version of the tool dedicated to the CELL architecture. Clearly, it was very pedagogic, but the conclusion is that the CELL is not adapted to sampling based verification methods.

This year we develop, thanks to the BSP++ framework, a version compatible with SPM/multicores machines, clusters and hybrid architectures. This version outperforms all previous ones, thus showing the interest of both these new architectures and of the BSP++ framework.

5.2. YML

Participants: Serge Petiton [correspondant], Nahid Emad, Maxime Hugues.

Scientific end-users face difficulties to program P2P large scale applications using low level languages and middleware. We provide a high level language and a set of tools designed to develop and execute large coarse grain applications on peer-to-peer systems. Thus, we introduced, developed and experimented the YML for parallel programming on P2P architectures. This work was done in collaboration with the PRiSM laboratory (team of Nahid Emad).

The main contribution of YML is its high level language for scientific end-users to develop parallel programs for P2P platforms. This language integrates two different aspects. The first aspect is a component description language. The second aspect allows to link components together. A coordination language called YvetteML can express graphs of components which represent applications for peer-to-peer systems.

Moreover, we designed a framework to take advantage of the YML language. It is based on two component catalogues and an YML engine. The first one concerns end-user’s components and the second one is related to middleware criteria. This separation enhances portability of applications and permits real time optimizations. Currently we provide support for the XtremWeb Peer-to-Peer middleware and the OmniRPC grid system. The support for Condor is currently under development and a beta-release will be delivered soon (in this release, we plan to propagate semantic data from the end-users to the middleware). The next development of YML concerns the implementation of a multi-backend scheduler. Therefore, YML will be able to schedule at runtime computing tasks to any global computing platform using any of the targeted middleware.

We experimented YML with basic linear algebra methods on a XtremWeb P2P platform deployed between France and Japan. Recently, we have implemented complex iterative restarted Krylov methods, such as Lanczos-Bisection, GMRES and MERAM methods, using YML with the OmniRPC back-end. The experiments are performed either on the Grid5000 testbed of on a Network of Workstations deployed between Lille, Versailles and Tsukuba in Japan. Demos was proposed on these testbeds from conferences in USA. We recently finished evaluations of the overhead generated using YML, without smart schedulers and with extrapolations due to the lack of smart scheduling strategies inside targeted middleware.

In the context of the FP3C project funded by ANR-JST, we have recently extended YML to support a directive distributed parallel language, XcalableMP http://www.xcalablemp.org/. This extension is based on the support of the XcalableMP language inside YML components. This allows to develop parallel programs with two programming paradigm and thus two parallelism levels. This work is a part of the project that targets post-Petascale supercomputer that would be composed of heterogeneous and massively parallel hardware.

The software is available at http://yml.prism.uvsq.fr/
5.3. The Scientific Programming InterNet (SPIN)

Participant: Serge Petiton [correspondant].

SPIN (Scientific Programming on the InterNet), is a scalable, integrated and interactive set of tools for scientific computations on distributed and heterogeneous environments. These tools create a collaborative environment allowing the access to remote resources.

The goal of SPIN is to provide the following advantages: Platform independence, Flexible parameterization, Incremental capacity growth, Portability and interoperability, and Web integration. The need to develop a tool such as SPIN was recognized by the GRID community of the researchers in scientific domains, such as linear algebra. Since the P2P arrives as a new programming paradigm, the end-users need to have such tools. It becomes a real need for the scientific community to make possible the development of scientific applications assembling basic components hiding the architecture and the middleware. Another use of SPIN consists in allowing to build an application from predefined components ("building blocks") existing in the system or developed by the developer. The SPIN users community can collaborate in order to make more and more predefined components available to be shared via the Internet in order to develop new more specialized components or new applications combining existing and new components thanks to the SPIN user interface.

SPIN was launched at ASCI CNRS lab in 1998 and is now developed in collaboration with the University of Versailles, PRiSM lab. SPIN is currently under adaptation to incorporate YML, cf. above. Nevertheless, we study another solution based on the Linear Algebra KErnel (LAKE), developed by the Nahid Emad team at the University of Versailles, which would be an alternative to SPIN as a component oriented integration with YML.

5.4. V-DS

Participant: Franck Cappello [correspondant].

This project started officially in September 2004, under the name V-Grid. V-DS stands for Virtualization environment for large-scale Distributed Systems. It is a virtualization software for large scale distributed system emulation. This software allows folding a distributed systems 100 or 1000 times larger than the experimental testbed. V-DS virtualizes distributed systems nodes on PC clusters, providing every virtual node its proper and confined operating system and execution environment. Thus compared to large scale distributed system simulators or emulators (like MicroGrid), V-DS virtualizes and schedules a full software environment for every distributed system node. V-DS research concerns emulation realism and performance.

A first work concerns the definition and implementation of metrics and methodologies to compare the merits of distributed system virtualization tools. Since there is no previous work in this domain, it is important to define what and how to measure in order to qualify a virtualization system relatively to realism and performance. We defined a set of metrics and methodologies in order to evaluate and compared virtualization tools for sequential system. For example a key parameter for the realism is the event timing: in the emulated environment, events should occur with a time consistent with a real environment. An example of key parameter for the performance is the linearity. The performance degradation for every virtual machine should evolve linearly with the increase of the number of virtual machines. We conducted a large set of experiments, comparing several virtualization tools including Vserver, VMware, User Mode Linux, Xen, etc. The result demonstrates that none of them provides both enough isolation and performance. As a consequence, we are currently studying approaches to cope with these limits.

We have made a virtual platform on the GDX cluster with the Vserver virtualization tool. On this platform, we have launched more than 20K virtual machines (VM) with a folding of 100 (100 VM on each physical machine). However, some recent experiments have shown that a too high folding factor may cause a too long execution time because of some problems like swapping. Currently, we are conducting experiments on another platform based on the virtualization tool named Xen which has been strongly improved since 2 years. We expect to get better result with Xen than with Vserver. Recently, we have been using the V-DS version based on Xen to evaluate at large scales three P2P middleware [89].
5.5. PVC: Private Virtual Cluster

Participant: Franck Cappello [correspondant].

Current complexity of Grid technologies, the lack of security of Peer-to-Peer systems and the rigidity of VPN technologies make sharing resources belonging to different institutions still technically difficult.

We propose a new approach called "Instant Grid" (IG), which combines various Grid, P2P and VPN approaches, allowing simple deployment of applications over different administration domains. Three main requirements should be fulfilled to make Instant Grids realistic: simple networking configuration (Firewall and NAT), no degradation of resource security, no need to re-implement existing distributed applications.

Private Virtual Cluster, is a low-level middle-ware that meets Instant Grid requirements. PVC turns dynamically a set of resources belonging to different administration domains into a virtual cluster where existing cluster runtime environments and applications can be run. The major objective of PVC is to establish direct connections between distributed peers. To connect firewall protected nodes in the current implementation, we have integrated three techniques: UPnP, TCP/UDP Hole Punching and a novel technique Traversing-TCP.

One of the major application of PVC is the third generation desktop Grid middleware. Unlike BOINC and XtremWeb (which belong to the second generation of desktop Grid middleware), PVC allows the users to build their Desktop Grid environment and run their favorite batch scheduler, distributed file system, resource monitoring and parallel programming library and runtime software. PVC ensures the connectivity layer and provide a virtual IP network where the user can install and run existing cluster software.

By offering only the connectivity layer, PVC allows to deploy P2P systems with specific applications, like file sharing, distributed computing, distributed storage and archive, video broadcasting, etc.

5.6. OpenWP

Participant: Franck Cappello [correspondant].

Distributed applications can be programmed on the Grid using workflow languages, object oriented approaches (Proactive, IBIS, etc), RPC programming environments (Grid-RPC, DIET), component based environments (generally based on Corba) and parallel programming libraries like MPI.

For high performance computing applications, most of the existing codes are programmed in C, Fortran and Java. These codes have 100,000 to millions of lines. Programmers are not inclined to rewrite them in a "non standard" programming language, like UPC, CoArray Fortran or Global Array. Thus environments like MPI and OpenMPI remain popular even if they require hybrid approaches for programming hierarchical computing infrastructures like cluster of multi-processors equipped with multi-core processors.

Programming applications on the Grid add a novel level in the hierarchy by clustering the cluster of multi-processors. The programmer will face strong difficulties in adapting or programming a new application for these runtime infrastructures featuring a deep hierarchy. Directive based parallel and distributed computing is appealing to reduce the programming difficulty by allowing incremental parallelization and distribution. The programmer add directives on a sequential or parallel code and may check for every inserted directive its correction and performance improvement.

We believe that directive based parallel and distributed computing may play a significant role in the next years for programming High performance parallel computers and Grids. We have started the development of OpenWP. OpenWP is a directive based programming environment and runtime allowing expressing workflows to be executed on Grids. OpenWP is compliant with OpenMP and can be used in conjunction with OpenMP or hybrid parallel programs using MPI + OpenMP.
The OpenWP environment consists of a source to source compiler and a runtime. The OpenWP parser, interprets the user directives and extracts functional blocks from the code. These blocks are inserted in a library distributed on all computing nodes. In the original program, the functional blocks are replaced by RPC calls and calls to synchronization. During the execution, the main program launches non-blocking RPC calls to functions on remote nodes and synchronize the execution of remote functions based on the synchronization directives inserted by the programmer in the main code. Compared to OpenMP, OpenWP does not consider a shared memory programming approach. Instead, the source to source compiler inserts data movements calls in the main code. Since the dataset can be large in Grid application, the OpenWP runtime organizes the storage of data sets in a distributed way. Moreover, the parameters and results of RPC calls are passed by reference, using a DHT. Thus, during the execution, parameter and result references are stored in the DHT along with the current position of the datasets. When a remote function terminates its execution, it stores the result datasets and stores a reference to the data set in the DHT.

We are evaluating OpenWP from an industrial application (Amibe), used by the European aerospace company EADS. Amibe is the mesher module of jCAE. Amibe generates a mesh from a CAD geometry in three steps. It first creates edges between every patch of the CAD (mesh in one dimension), then generates a surface mesh for every unfolded patch (mesh in two dimensions) and finally adds the third dimension to the mesh by projecting the 2D mesh into the original CAD surfaces. The first and third operation cannot be distributed. However the second step can easily be distributed following a master/worker approach, transferring the mesh1d results to every computing node and launching the distributed execution of the patches.

5.7. Parallel solvers for solving linear systems of equations

Participant: Laura Grigori.

In the last several years, there has been significant research effort in the development of fully parallel direct solvers for computing the solution of large unsymmetric sparse linear systems of equations. In this context, we have designed and implemented a parallel symbolic factorization algorithm, which is suitable for general sparse unsymmetric matrices. The symbolic factorization is one of the steps that is sequential and represents a memory bottleneck. The code is intended to be used with very large matrices when because of the memory usage, the sequential algorithm is not suitable. This code is available in the SuperLU_DIST, a widely used software, developed at UC Berkeley and LBNL by Professor James W. Demmel and Dr. Xiaoye S. Li. The algorithm is presented in [77]. The SuperLU_DIST is available at http://crd.lbl.gov/~xiaoye/SuperLU/.

5.8. OpenScop

Participant: Cédric Bastoul.

OpenScop is an open specification which defines a file format and a set of data structures to represent a static control part (SCoP for short), i.e., a program part that can be represented in the polyhedral model, an algebraic representation of programs used for automatic parallelization and optimization (used, e.g., in GNU GCC, LLVM, IBM XL or Reservoir Labs R-Stream compilers). The goal of OpenScop is to provide a common interface to various polyhedral compilation tools in order to simplify their interaction.

OpenScop provides a single format for tools that may have different purposes (e.g., as different as code generation and data dependence analysis). We could observe that most available polyhedral compilation tools during the last decade were manipulating the same kind of data (polyhedra, affine functions...) and were actually sharing a part of their input (e.g., iteration domains and context concepts are nearly everywhere). We could also observe that those tools may rely on different internal representations, mostly based on one of the major polyhedral libraries (e.g., Polylib, PPL or isl), and this representation may change over time (e.g., when switching to a more convenient polyhedral library). OpenScop aims at providing a stable, unified format that offers a durable guarantee that a tool can use an output or provide an input to another tool without breaking a compilation chain because of some internal changes in one element of this chain. The other promise of

---

1 project page: http://jcae.sourceforge.net
OpenScop is the ability to assemble or replace the basic blocks of a polyhedral compilation framework at no, or at least low engineering cost. The OpenScop Library (licensed under the 3-clause BSD license) has been developed as an example, yet powerful, implementation of the OpenScop specification.

5.9. CALU for multicore architectures

Participant: Laura GRIGORI [correspondant].

The communication avoiding algorithms are implemented in the form of a portable library. In its current form, this library is designed for multicore architectures and uses a hybrid scheduling technique that exploits well the data locality and can adapt to dynamic changes in the machine. The library will be publicly available since February 2012.

See also the web page http://www-rocq.inria.fr/who/Laura.Grigori/COALA2010/coala.html.

• Version: 1.0

5.10. Fast linear system solvers in public domain libraries

Participant: Marc Baboulin [correspondant].

Hybrid multicore+GPU architectures are becoming commonly used systems in high performance computing simulations. In this research, we develop linear algebra solvers where we split the computation over multicore and graphics processors, and use particular techniques to reduce the amount of pivoting and communication between the hybrid components. This results in efficient algorithms that take advantage of each computational unit [12]. Our research in randomized algorithms yields to several contributions to propose public domain libraries PLASMA and MAGMA in the area of fast linear system solvers for general and symmetric indefinite systems. These solvers minimize communication by removing the overhead due to pivoting in LU and LDLT factorization.

See also the web page http://icl.cs.utk.edu/magma/.

5.11. cTuning: Repository and Tools for Collective Characterization and Optimization of Computing Systems

Participant: Grigori Fursin [correspondant].

Designing, porting and optimizing applications for rapidly evolving computing systems is often complex, ad-hoc, repetitive, costly and error prone process due to an enormous number of available design and optimization choices combined with the complex interactions between all components. We attempt to solve this fundamental problem based on collective participation of users combined with empirical tuning and machine learning.

We developed cTuning framework that allows to continuously collect various knowledge about application characterization and optimization in the public repository at cTuning.org. With continuously increasing and systematized knowledge about behavior of computer systems, users should be able to obtain scientifically motivated advices about anomalies in the behavior of their applications and possible solutions to effectively balance performance and power consumption or other important characteristics.

Currently, we use cTuning repository to analyze and learn profitable optimizations for various programs, datasets and architectures using machine learning enabled compiler (MILEPOST GCC). Using collected knowledge, we can quickly suggest better optimizations for a previously unseen programs based on their semantic or dynamic features [7].

We believe that such approach will be vital for developing efficient Exascale computing systems. We are currently developing the new extensible cTuning2 framework for automatic performance and power tuning of HPC applications.

For more information, see the web page http://cTuning.org.
5. Software

5.1. RPL P2P

Participants: Emmanuel Baccelli [correspondant], Matthias Philipp.

P2P-RPL is an implementation of draft-ietf-roll-p2p-rpl, providing reactive discovery of point-to-point routes in low power and lossy networks such as wireless sensor networks. The implementation is based on the Contiki operating system. See also the web page http://contiki-p2p-rpl.gforge.inria.fr/.

- Version: 0.4

5.2. OPERA infrastructure

Participants: Cédric Adjih [correspondant], Ichrak Amdouni, Pascale Minet, Ridha Soua.

OPERA-infrastructure is the system support code of OPERA, the Optimized Protocol for Energy efficient Routing with node Activity scheduling.

5.3. OPERA perf simul

Participants: Cédric Adjih [correspondant], Ichrak Amdouni.

OPERA-perf-simul is a set of tools for simulation and performance evaluation as well as large scale tests of OPERA, the Optimized Protocol for Energy efficient Routing with node Activity scheduling.

5.4. OPERA protocol

Participants: Cédric Adjih [correspondant], Ichrak Amdouni, Pascale Minet, Saoucene Mahfoudh.

OPERA-protocol is the heart of OPERA, the Optimized Protocol for Energy efficient Routing with node Activity scheduling. It includes EOND a neighborhood discovery protocol, EOSTC a protocol byuiding and maintaining a n energy efficient routing tree and SERENA a node coloring algorithm.

5.5. OPERA validation and tools

Participant: Cédric Adjih [correspondant].

OPERA-validation and tools is a set of tools for validation, debugging, analysis and visualization of OPERA protocol, the Optimized Protocol for Energy efficient Routing with node Activity scheduling. It operates either in a real embedded system or in simulation.
5. Software

5.1. The Obvious Toolkit

Participants: Pierre-Luc Hémery, Jean-Daniel Fekete [correspondant].

The Obvious Toolkit is a new Interactive Graphics Toolkit written in Java to facilitate the interoperability between Information Visualization toolkits and components (Fig. 1).

The Obvious Toolkit is an abstraction layer above visualization toolkits. Currently, it connects the most popular toolkits in Java: Prefuse, the InfoVis Toolkit, Improvise, as well as other libraries such as the Java Database Communication Toolkit (JDBC) and some others.

It is meant to provide an abstraction layer for information visualization application builders so that they can postpone their choice of a concrete toolkit to use. When faced with the final choice, application builders can use one of the toolkits or connect all of them dynamically to Obvious. Obvious is available at http://code.google.com/p/obvious. A paper on Obvious was presented at the IEEE Visual Analytics Science and Technology conference (VAST 2011).

5.2. GeneaQuilts

Participants: Jean-Daniel Fekete [correspondant], Pierre Dragicevic, Anastasia Bezerianos, Julie Bae, Ben Watson, Maike Gilliot [correspondant].

GeneaQuilts [2] is a new genealogy exploration software that allows genealogists and historians to visualize and navigate in large genealogies of up to several thousand individuals (Fig. 2). The visualization takes the form of a diagonally-filled matrix, where rows are individuals and columns are nuclear families. The GeneaQuilts system includes an overview, a timeline, search and filtering components, and a new interaction technique called Bring & Slide that allows fluid navigation in very large genealogies. The tool has been featured in several InfoVis and genealogy Websites and the website has been visited over 9000 times.
Figure 2. The genealogy of the Simpsons family (left) and of the Greek Pantheon (right), produced by the GeneaQuilts software.

See also the web page http://www.aviz.fr/geneaquilts/.

- Version: 1.0.4

5.3. Diffamation

Participants: Fanny Chevalier, Pierre Dragicevic [correspondant], Anastasia Bezerianos, Jean-Daniel Fekete.

Figure 3. Screenshot the Diffamation system during a transition: (a) the document view, (b) the overview scrollbar and (c) the timeline.

The Diffamation system [3] allows rapid exploration of revision histories such as Wikipedia or subversion repositories by combining text animated transitions with simple navigation and visualization tools. Diffamation can be used for example to get a quick overview of the entire history of a Wikipedia article or to see what has happened to one’s contributions. Diffamation complements classical diff visualizations: once moments of interest have been identified, classical diff visualizations can come in useful to compare two given revisions in detail.
The Diffamation revision exploration system is available at http://www.aviz.fr/diffamation/. It has been presented at the plenary session of the Ubuntu Developer Summit.

5.4. The InfoVis Toolkit

**Participant:** Jean-Daniel Fekete [correspondant].

The InfoVis Toolkit [5] is an Interactive Graphics Toolkit written in Java to facilitate the development of Information Visualization applications and components.

The InfoVis Toolkit implements several visualization techniques, as well as interaction techniques related. It has been used for teaching the Information Visualization course (Masters level, Univ. of Paris-Sud) and is the basis for all AVIZ contracts. It is our main development platform for information visualization; most of our Information Visualization prototypes rely on it. It is available at http://ivtk.sourceforge.net.

In the forthcoming years, it will be superseded by extensions of the Obvious Toolkit (see 5.1).

- Version: version0.9 beta 2

5.5. GraphDice

**Participants:** Jean-Daniel FEKETE [correspondant], Pierre Dragicevic, Niklas Elmqvist, Anastasia Bezerianos.

![Figure 4. Screenshot the GraphDice system.](image)

GraphDice [1] is a visualization system for exploring multivariate networks (Fig. 4). GraphDice builds upon our previous system ScatterDice (best paper award at the IEEE InfoVis 2008 conference) [4]: it shows a scatter plot of 2 dimensions among the multiple ones available and provides a very simple paradigm of 3D rotation to change the visualized dimensions. The navigation is controlled by a scatter plot matrix that is used as a high-level overview of the dataset as well as a control panel to switch the dimensions.

While ScatterDice works on any tabular dataset (e.g., CSV file), the GraphDice system show networks using a node-link diagram representation as a scatter plot with links drawn between connected nodes. See the web page http://graphdice.gforge.inria.fr for more information.

- Version: version 1.0

5.6. Gliimpse

**Participants:** Pierre Dragicevic [correspondant], Stéphane Huot, Fanny Chevalier.
Gliimpse is a quick preview technique that smoothly transitions between document markup code (HTML, LaTeX,...) and its visual rendering. This technique allows users to regularly check the code they are editing in-place, without leaving the text editor. This method can complement classical preview windows by offering rapid overviews of code-to-document mappings and leaving more screen real-estate. A proof-of-concept editor can be downloaded for free at http://www.aviz.fr/gliimpse/.
DAHU Project-Team (section vide)
5. Software

5.1. jBricks

Participants: Stéphane Huot, Emmanuel Pietriga [correspondant], Mathieu Nancel, Romain Primet.

jBricks (Figure 1) is a Java toolkit that integrates a high-quality 2D graphics rendering engine based on ZVTM (section 5.2) and a versatile input configuration module (based on ICon [40] and FlowStates 5.4) into a coherent framework, enabling the exploratory prototyping of interaction techniques and rapid development of post-WIMP applications running on cluster-driven interactive visualization platforms such as wall-sized displays. The goal of this framework is to ease the development, testing and debugging of interactive visualization applications. It also offers an environment for the rapid prototyping of novel interaction techniques and their evaluation through controlled experiments.

Figure 1. jBricks applications running on the WILD platform (32 tiles for a total resolution of 20 480 × 6 400 pixels). (a) Zoomed-in visualization of the North-American part of the world-wide air traffic network (1 200 airports, 5 700 connections) overlaid on NASA’s Blue Marble Next Generation images (86 400 × 43 200 pixels) augmented with country borders ESRI shapefiles. (b) Panning and zooming in Spitzer’s Infrared Milky Way (396 032 × 12 000 pixels). (c) Controlled laboratory experiment for the evaluation of mid-air multi-scale navigation techniques.

- ACM: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)
- OS/Middleware: Java (Linux, Mac OS X, Windows)
- Required library or software: several, managed through Maven
- Programming language: Java

5.2. The Zoomable Visual Transformation Machine

Participants: Caroline Appert, Rodrigo de Almeida, Olivier Chapuis, Arjit Gupta, Julien Husson, Emmanuel Pietriga [correspondant], Mathieu Nancel, Romain Primet.
ZVTM provides application programmers with building blocks for implementing complex multi-scale interface components that cannot be handled by traditional WIMP widgets. Featuring off-the-shelf visualisation and navigation components that are easy to combine, ZVTM provides a simple yet powerful API and handles low-level operations such as multi-threading, clipping, repaint requests and animation management. The toolkit is based on the metaphor of universes that can be observed through smart movable/zoomable cameras. The graphical object model permits management of a large number of complex geometrical shapes. It emphasizes perceptual continuity via an advanced animation module that can animate virtually any on-screen modification. This ranges from camera movements and activation of distortion lenses to modification of the visual variables of graphical objects. Various temporal pacing functions are available to control the execution of these animations. ZVTM is now one of the core components of our jBricks toolkit for wall-sized displays (Section 5.1), and current development activities around the toolkit focus on making applications run transparently on cluster-driven ultra-high-resolution wall-sized displays such as that of the WILD visualization platform. The toolkit is also used to develop advanced visualization components for the ALMA observatory’s operations monitoring and control software [29].

Initially developed by Xerox Research Centre Europe and the World Wide Web Consortium (W3C) team at MIT, ZVTM has been available as open-source software under the GNU Lesser General Public License (LGPL) since early 2002. It is used in both academic and industrial projects such as IsaViz (http://www.w3.org/2001/11/IsaViz/), W3C’s visual browser/editor for RDF, Blast2GO (Figure 2 - left) (http://www.blast2go.org/), or ZGRViewer (http://zvtm.sourceforge.net/zgrviewer.html) for viewing large graphs generated by AT&T GraphViz 4 (Figure 2 - right). The development of the toolkit is now supported by INRIA. More information can be found at http://zvtm.sourceforge.net and [43] and [24].

5.3. The SwingStates Toolkit
Participants: Caroline Appert [correspondant], Michel Beaudouin-Lafon.

4 http://www.graphviz.org
SwingStates [1] is a library that adds state machines and a graphical canvas to the Java Swing user interface toolkit. It was motivated by the lack of widely disseminated toolkits that support advanced interaction techniques and the observation that HCI research toolkits are little used outside the lab. By extending the popular Java Swing toolkit rather than starting from scratch, the goal is to facilitate the dissemination and adoption of SwingStates by practitioners.

SwingStates uses state machines to specify interaction. It provides programmers with a natural syntax to specify state machines and reduces the potential for an explosion of the number of states by allowing multiple state machines to work together or separately. SwingStates can be used to add new interaction techniques to existing Swing widgets, e.g. to select buttons and checkboxes by crossing rather than clicking. It can also be used with the SwingStates canvas (see below) and to control high-level dialogues.

SwingStates also provides a powerful canvas widget. The canvas can contain any Java2D shape, including geometric shapes, images, text strings and even Swing widgets. Shapes can be manipulated individually or collectively, through tags. An intensive use of polymorphism allows to apply almost any command to a tag: the command is then applied to all objects with this tag. Tags are also used in conjunction with state machines, to specify transitions that occur only on objects with a given tag. For example, pie menus can be implemented by creating a canvas in the overlay layer of any Swing application (Figure 3).

Figure 3. A numeric text field whose value can be set by a joystick-like interaction (left) and a semi-transparent menu to change the background color of Swing widgets (right)

SwingStates tightly integrates state machines, the Java language and the Swing toolkit to provide programmers with a natural and powerful extension to their natural programming environment. SwingStates is available at http://swingstates.sf.net under the GNU Lesser General Public License (LGPL).

- ACM: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)
- OS/Middleware: Mac OS X, Linux, Windows
- Required library or software: Java virtual machine
- Programming language: Java

5.4. The FlowStates Toolkit

Participants: Caroline Appert [correspondant], Michel Beaudouin-Lafon, Stéphane Huot.

FlowStates [33], is a new toolkit to program advanced interaction techniques which require non standard input (e.g., two different mice that act independently, a joystick, a tablet, etc.). It is built on top of two existing toolkits: SwingStates [1] and ICon [40].
With FlowStates the developer can program interaction logic using state machines like SwingStates does but does not restrict the set of possible input channels to Java AWT standard input (a single couple <mouse, keyboard>). The state machines just have to define the virtual input events that are required to trigger their transitions so that FlowStates turns these machines into ICon devices which can be plugged to any physical input channels (Figure 4). An ICon device is a data flow building block that has input and output slots in order to be connected to other devices in the simple graphical environment provided by ICon. State machines can also send out events which appear as output slots in the data flow model.

With FlowStates we showed how two models for programming interaction (state machines and data flow) can be fully integrated to offer a huge power of expression. The explicit decision to not set strict limits between the roles of each model makes this hybrid approach highly flexible, the developer setting himself the limit between the two according to his needs and habits.


- ACM: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)
- OS/Middleware: Mac OS X, Linux, Windows
- Required library or software: ICon, Java virtual machine
- Programming language: Java

5.5. TouchStone

Participants: Caroline Appert [correspondant], Michel Beaudouin-Lafon, Wendy Mackay.

TouchStone [5] is a platform for designing, running and analyzing the results of controlled experiments (Figure 5). While it focuses on experiments comparing interaction techniques, it can be used in a wide variety of contexts.

With the Touchstone design platform, a user specifies the factors and the measures of the experiment, the blocking and counterbalancing of trials, and assess the time it will take to run the experiment. Multiple designs can be explored in parallel to assess the various trade-offs. The output of the design platform is an XML file that can be used as input for the run platform.
The **Touchstone run platform** provides a framework to implement and run an experiment and to collect experimental data. It uses a flexible plug-in architecture to manage a variety of input devices and interaction techniques. The runs of the experiment are controlled by an XML script that can be produced by the design platform.

The analysis platform currently consists of data analysis tools such as JMP, R or Excel. Log data produced by the run platform can be directly loaded into any of these tools. In a future version, analysis sketches will be derived from the experimental design to assist with the analysis.

Touchstone has been used heavily at INSITU over the past three years for the many experiments that we design and run. It has also been used for teaching for the first time in 2011. Students used it to design various experiments during tutorial classes in Master 2 Interaction (“Introduction to HCI” module).


- ACM: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)
- OS/Middleware: Mac OS X, Linux, Windows
- Required library or software: Java virtual machine
- Programming language: Java

**5.6. Metisse**

**Participant:** Olivier Chapuis [correspondant].

Metisse [38] is a window system that facilitates the design, implementation and evaluation of innovative window management techniques. The system is based on a compositing approach, making a clear distinction between the rendering and the interactive compositing processes. The Metisse server is a modified X server that supports both input and output redirection. The default compositor is a combination of a slightly modified version of FVWM, a standard window manager, with an interactive viewer application called *FvwmCompositor*.
FvwmCompositor uses OpenGL to display windows, which offers a rich graphics model well adapted to the exploration of new window management techniques. Texture mapping, for example, makes it possible to transform the window shapes in real-time (Figure 6, left). Alpha blending makes it easy to create translucent objects and shadows. Scaling, rotation and translation can also be used to position windows in 2D or 3D (Figure 6, middle and right). Input redirection makes it still possible to interact with applications no matter the visual transformations applied to the windows. It also makes it possible to adapt, reconfigure or re-combine existing graphical interfaces [45]. This year we used again Metisse to implement novel desktop interaction techniques [3].

![Figure 6. Sample window management techniques implemented with Metisse: extended paper metaphor (left), interactive table configuration that allows to duplicate and rotate windows (middle) and zoomable 3D desktop (right).](image)

- Web: [http://insitu.lri.fr/metisse/](http://insitu.lri.fr/metisse/)
- ACM: H.5.2 [User Interfaces]: Windowing systems
- Software benefit: see [38], [45], [39], [41] and [3].
- License: GPL
- Type of human computer interaction: Graphique
- OS/Middleware: X Window et Mac OS X
- Required library or software: OpenGL via nucleo and some usual C/C++ libraries
- Programming language: * C/C++

**5.7. Wmtrace**

**Participant:** Olivier Chapuis [correspondant].

Wmtrace [37] includes two tools that help us study an individual user’s window management activity. The first tool runs in the background of an X Window session and continuously logs information about windows and how they are being manipulated. The second uses a VCR-like interface (Figure 7) to replay the resulting logs and analyze the entire session. This tool provides several ways to filter the logs and extract high-level information, including interactive move events and mouse speed. Both tools allow HCI researchers to perform qualitative and quantitative statistical analyses of window management activity.

- ACM: H.5.2 [User Interfaces]: Windowing systems
- Software benefit: see [37], [41], [36].

---

5.8. The Substance Middleware

Participants: Michel Beaudouin-Lafon [correspondent], Clemens Klokmose, Tony Gjerlufsen, James Eagan, Clement Pillias.

Substance is a middleware based on a novel programming paradigm called data-oriented programming and was designed to facilitate the development of multi-surface interactive applications [20]. Such applications are distributed by nature as they involve a varying number of display and interaction surfaces that are controlled by different computers. For example, our WILD room includes a 32-monitor display wall driven by 16 computers plus a front-end, a multi-touch table, various mobile devices such as iPodTouch and iPads, and the laptops that the users of the room may bring with them. We want to support seamless interaction techniques across these surfaces, such as the pick-and-drop technique pioneered by Rekimoto [44].

Data-oriented programming consists of attaching functionality to a tree data structure through facets attached to the individual nodes of the tree. Facets can be added and removed dynamically, and notified of changes in the tree. Substance supports two powerful ways to share nodes and facets: mounting, where access to the shared tree is managed through remotely, and replication, where the shared tree is replicated at each site and synchronized.

Substance has been used to create two full-scale applications (Figure 8): a generalized Canvas that can display and manage graphics, PDF files, image files and other content (through an extensible content manager) across surfaces spanning multiple displays and computers; SubstanceGrise, which uses multiple instances of the Anatomist/BrainVISA application to display coordinated 3D imagery of many brains in parallel on the WILD wall and control from a physical model of the brain.
Figure 8. The Canvas (left) and SubstanceGrise (right) applications developed with Substance. (©CNRS-Phototheque - Cyril FRESILLON for SubstanceGrise).

Substance is available at http://substance-env.sourceforge.net/ under a GNU GPL 3.0 licence.

- ACM: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)
- OS/Middleware: Mac OS X, Linux
- Required library or software: several, managed by Python install
- Programming language: Python

5.9. Scotty

Participants: Michel Beaudouin-Lafon [correspondant], James Eagan.

The goal of Scotty is to support malleable interfaces, i.e. interfaces that can be modified at run-time in ways not anticipated by the designers [18]. Scotty is a toolkit that allows a programmer to extend an existing Mac OS X application without access to its source code. Scotty provides the following abstractions: hooks to alter the appearance of windows and widgets, event funnels to alter their behavior, glass sheets to overlay graphics and add new interaction methods, dynamic code loading and object proxies to redefine and extend existing objects. Scotty also provides a higher-level interface based on instrumental interaction [34]. Scotty currently runs on Mac OS X for applications written with the Cocoa user interface framework.

Scotty has been used to create a number of extensions (Figure 9). Scribbler is a generic extension that uses glass sheets to allow handwritten annotations of any Cocoa window. Teleportation is another generic extension that can teleport and resize the content of any Cocoa window onto another computer, including an iPhone or iPad. The user can interact with the teleported content as if it was on the original computer. It was used to create a content provider for the Substance Canvas (see above), making it possible to display any application running on a laptop onto the WILD wall display and/or table. When vector-based content is available, e.g., for text, Scotty provides smooth rescaling without the typical pixelation apparent when enlarging bitmap images. Finally Stylesheet is an extension to the Pages word processor that provides a semi-transparent toolglass for specifying the styles of paragraphs.
Scotty is available at [http://insitu.lri.fr/Projects/Scotty](http://insitu.lri.fr/Projects/Scotty) under a GNU GPL 3.0 licence.

- ACM: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)
- OS/Middleware: Mac OS X
- Required library or software: none
- Programming language: Objective-C, Python
5. Software

5.1. AlignViz

Name: AlignViz  
Contact: Fayçal Hamdi (hamdi@lri.fr)  
Other contacts: Brigitte Safar (safar@lri.fr) and Chantal Reynaud (chantal.reynaud@lri.fr)  
Presentation: a visualization tool for alignments between ontologies

5.2. AnnoViP

Name: AnnoViP  
Contact: Konstantinos Karanasos (konstantinos.karanasos@inria.fr)  
Other contacts: Ioana Manolescu (ioana.manolescu@inria.fr) and Jesús Camacho_Rodriguez (jesus.camacho-rodriguez@inria.fr)  
Presentation: a tool for editing and exploiting XML documents with annotations in a distributed P2P setting

5.3. EAP Framework

Name: EAP Framework  
Contact: Nadjet Zémirline (nadjet.zemirline@supelec.fr)  
Other contacts: Chantal Reynaud (chantal.reynaud@lri.fr)  
Presentation: a prototype helping to express adaptation strategies based on the use and a semi-automatic combination of patterns

5.4. EdiFlow

Name: EdiFlow (http://scidam.gforge.inria.fr)  
Contact: Wael Khemiri (wael.khemiri@inria.fr)  
Other contacts: Ioana Manolescu (ioana.manolescu@inria.fr), Jean-Daniel Fekete (jean-daniel.fekete@inria.fr), Pierre-Luc Hémery (pierre-luc.hemery@inria.fr), Véronique Benzaken (veronique.benzaken@lri.fr)  
Presentation: A platform for data-intensive visual analytics

5.5. KD2R

Name: KD2R  
Contact: Danai Symeonidou (danai.symeonidou@lri.fr)  
Other contacts: Nathalie Pernelle (nathalie.pernelle@lri.fr), Fatiha Saïs (fatiha.sais@lri.fr)  
Presentation: a tool for OWL2 key discovery on RDF datasets

5.6. Glucose2

Name: Glucose 2.0
5.7. GlucosER

Name: GlucosER
Contact: Laurent Simon (simon@lri.fr)
Other contacts: George Katsirelos (gkatsi@gmail.com) and Gilles Audemard (audemard@cril.univ-artois.fr)
Presentation: a SAT Solver based on Glucose 1.0 with Extended Resolution.

5.8. LiquidXML

Name: Liquid XML (http://vip2p.saclay.inria.fr/?page=liquidxml)
Contact: Asterios Katsifodimos (asterios.katsifodimos@inria.fr)
Other contacts: Ioana Manolescu (ioana.manolescu@inria.fr)
Presentation: a prototype for automatically recommending XML materialized views in order to improve the performance of a query workload

5.9. LN2R

Name: LN2R
Contact: Fatiha Saïs (sais@lri.fr)
Other contacts: Nathalie Pernelle (pernelle@lri.fr)
Presentation: a logical and numerical tool for reference reconciliation.

5.10. MESAM

Name: MESAM
Contact: Nadjet Zémirline (nadjet.zemirline@supelec.fr)
Other contacts: Chantal Reynaud (chantal.reynaud@lri.fr)
Presentation: a plug-in for Protege 2000 to merge generic and specific models

5.11. RDFViewS

Name: RDFViewS (http://tripleo.saclay.inria.fr/rdfs/)
Contact: Konstantinos Karanasos (konstantinos.karanasos@inria.fr)
Other contacts: François Goasdoué (fg@lri.fr), Julien Leblay (julien.leblay@inria.fr), and Ioana Manolescu (ioana.manolescu@inria.fr)
Presentation: a storage tuning wizard for RDF applications

5.12. SomeWhere

Name: SomeWhere
Contact: François Goasdoué (fg@lri.fr)
5.13. SpyWhere

Name: SpyWhere
Contact: François-Elie Calvier (fcalvier@gmail.com)
Other contacts: Chantal Reynaud (chantal.reynaud@lri.fr)
Presentation: a generator of mapping candidates for enriching peer ontologies

5.14. TaxoMap

Name: TaxoMap (http://taxomap.lri.fr)
Contact: Fayçal Hamdi (hamdi@lri.fr)
Other contacts: Brigitte Safar (safar@lri.fr) and Chantal Reynaud (chantal.reynaud@lri.fr)
Presentation: a prototype to automate semantic mappings between taxonomies

5.15. TaxoMap Framework

Name: TaxoMap Framework
Contact: Fayçal Hamdi (hamdi@lri.fr)
Other contacts: Brigitte Safar (safar@lri.fr) and Chantal Reynaud (chantal.reynaud@lri.fr)
Presentation: an environment to specify treatments to refine mappings and to enrich ontologies

5.16. ViP2P

Contact: Ioana Manolescu (ioana.manolescu@inria.fr)
Other contacts: Jesús Camacho_Rodriguez (jesus.camacho-rodriguez@inria.fr), Asterios Katsifodimos (asterios.katsifodimos@inria.fr), Konstantinos Karanasos (konstantinos.karanasos@inria.fr)
Presentation: a P2P platform for disseminating and querying XML and RDF data in large-scale distributed networks.

5.17. XUpOp

Name: XUpOp (XML Update Optimization)
Contact: Dario Colazzo (colazzo@lri.fr)
Other contacts: Nicole Bidoit (bidoit@lri.fr), Marina Sahakian (Marina.Sahakyan@lri.fr), and Mohamed Amine Baazizi (baazizi@lri.fr)
Presentation: a general purpose type based optimizer for XML updates

5.18. XUpIn

Name: XUpIn (XML Update Independence)
Contact: Federico Ulliana (Federico.Ulliana@lri.fr)
Other contacts: Dario Colazzo (colazzo@lri.fr), Nicole Bidoit (bidoit@lri.fr)
Presentation: an XML query-update independence tester