



RESEARCH CENTER
Saclay - Île-de-France

FIELD

Activity Report 2012

Section Software

Edition: 2013-04-24

1. AMIB Project-Team	4
2. AVIZ Project-Team	6
3. BYMOORE Exploratory Action	11
4. COMETE Project-Team	12
5. COMMANDS Project-Team	14
6. DAHU Project-Team (section vide)	16
7. DEFI Project-Team	17
8. DISCO Project-Team	20
9. GALEN Team	23
10. GECO Team	25
11. GEOMETRICA Project-Team	26
12. GRACE Team	28
13. GRAND-LARGE Project-Team	29
14. HIPERCOM Project-Team	35
15. IN-SITU Project-Team	36
16. MACS Project-Team	45
17. MAXPLUS Project-Team	47
18. MEXICO Project-Team	49
19. OAK Team	51
20. PARIETAL Project-Team	53
21. PARSIFAL Project-Team	54
22. POEMS Project-Team	56
23. POPIX Exploratory Action	57
24. REGULARITY Project-Team	58
25. SECSI Project-Team	59
26. SELECT Project-Team	60
27. TAO Project-Team	61
28. TOCCATA Team	63
29. TYPICAL Project-Team	68

AMIB Project-Team

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 92 research articles since its presentation in Fall of 2009, according to Google scholar), web servers such as the BOULDEALE (<http://www.microbio.me/boulderale/>), TFOLD (<http://tfold.ibisc.univ-evry.fr/TFold/>), CYLOFOLD (<http://cylofold.abcc.ncifcrf.gov/>) webservers, and by databases such as IRESITE (<http://iresite.org/>), SRNATARBASE (<http://ccb.bmi.ac.cn/srnatarbase/>) and RFAM (<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/>), and Y. Ponty co-supervised with Jim Procter (University of Dundee, Jalview Project Leader) two internships (including a Google Summer of Code) in the summer of 2012 to further the interactions between the two software.

4.2. SPFlow

Participants: Jiuqiang Chen, Sarah Cohen-Boulakia [correspondant], Christine Froidevaux [correspondant].

SPFLOW is a scientific workflow rewriting tool. SPFlow aims at transforming complex workflow structures (non series-parallel structures) into provenance-equivalent simple workflow structures (series-parallel structures). SPFlow takes as an input a file representing one scientific workflow from Taverna and produces another file in which the structure of the original workflow is made series-parallel while ensuring that both workflows have the same provenance (more information available at [32], [39]). The tool is freely available at <http://www.lri.fr/~chenj/SPFlow>.

4.3. GeneValorization

Participants: Bryan Brancotte, 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>.

4.4. 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.

The new version of HSIM includes a Stochastic Simulation Algorithm *a la* Gillespie that can be used with the same model in a standalone way or in a mixed way with the entity-centered algorithm. This new version offers also the possibility to export the model in SciLab for a ODE integration. Last, HSIM can export the differential equations system, equivalent to the model, to LaTeX for pretty-printing.

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.

4.5. Cartaj

Participant: Alain Denise [correspondant].

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 [14].

AVIZ Project-Team

5. Software

5.1. Graph Cuisine

Participants: Évelyne Lutton [correspondant], Benjamin Bach, André Spritzer, Jean-Daniel Fekete.

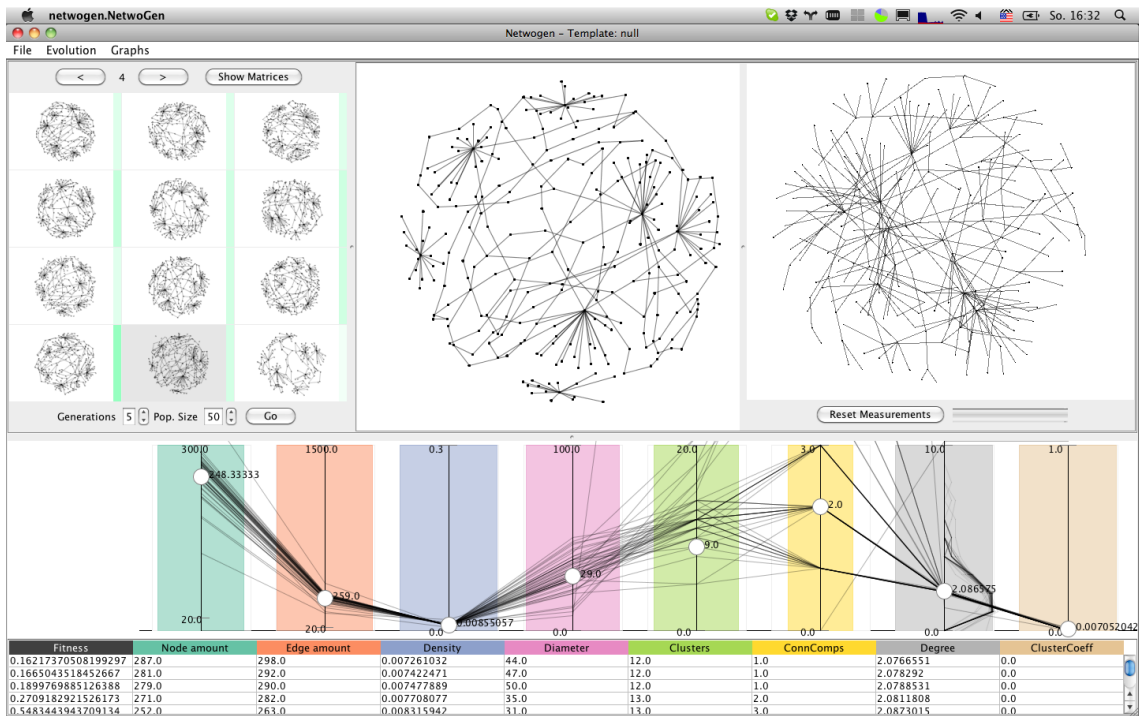


Figure 1. GraphCuisine interface showing one real graph (right), the measures extracted from it (bottom circles), several graphs with similar measures (left) and one of them selected (middle).

GraphCuisine lets users steer an Evolutionary Algorithm (EA) to create random graphs that match user-specified measures. Generating random graphs with particular characteristics is crucial for evaluating graph algorithms, layouts and visualization techniques. Current random graph generators provide limited control of the final characteristics of the graphs they generate. The situation is even harder when one wants to generate random graphs similar to a given one, all-in-all leading to a long iterative process that involves several steps of random graph generation, parameter changes, and visual inspection. Our system follows an approach based on interactive evolutionary computation. Fitting generator parameters to create graphs with pre-defined measures is an optimization problem, while assessing the quality of the resulting graphs often involves human subjective judgment. GraphCuisine has been proved to be able to generate graphs that mimic a given real-world network. <http://www.aviz.fr/Research/Graphcuisine>

5.2. Histomages

Participants: Fanny Chevalier, Pierre Dragicevic [correspondant], Christophe Hurter.

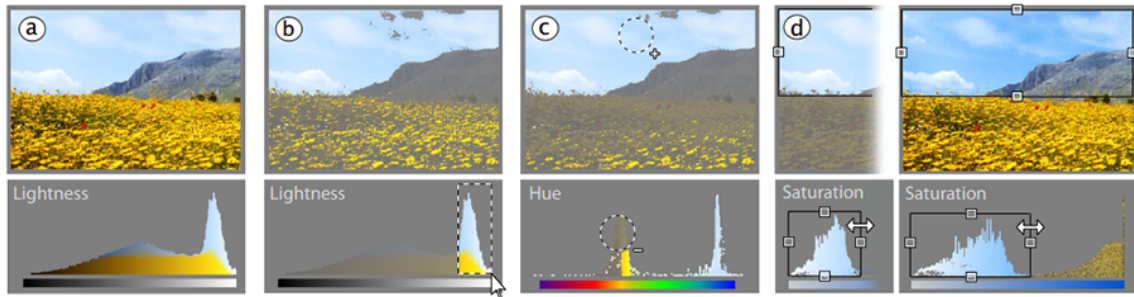


Figure 2. Example of sky enhancement with Histomages: (a) the image is duplicated and its pixels rearranged into a lightness histogram ; (b) bright pixels are selected with the rubber-band selection tool; (c) all pixels are rearranged into a hue histogram and yellow pixels are filtered out with the subtract selection brush (bottom). Missing pixels are added with the add selection brush on the image (top); (d) the sky is enhanced by resizing the selection on the saturation histogram.

Histomages is an image editor based on a new interaction model that considers histogram views as spatial rearrangements of image pixels. Users can select pixels on image histograms as they would select image regions and directly manipulate them to adjust their colors. Histomages are affected by other image tools such as paintbrushes. We explored some possibilities offered by this interaction model, and discussed the four key principles behind it as well as their implications for the design of feature-rich software in general [29]. <http://www.aviz.fr/histomages/>.

5.3. Glimpse

Participants: Pierre Dragicevic [correspondant], Stéphane Huot, Fanny Chevalier.



Figure 3. Glimpse: A detail of the animation between an article and its LaTeX source code.

Glimpse 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/glimpse/>.

5.4. The Obvious Toolkit

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

Information Visualization, Java, Toolkit

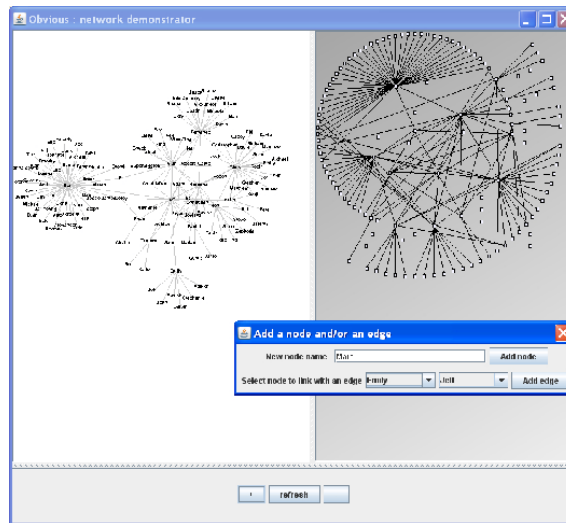


Figure 4. The Obvious toolkit showing the same graph with a Prefuse and an IVTK rendering.

The Obvious Toolkit is a new Interactive Graphics Toolkit written in Java to facilitate the interoperability between Information Visualization toolkits and components (Fig. 4).

The Obvious Toolkit is an abstraction layer above visualization toolkits. Currently, it connects the most popular toolkits in Java: Prefuse, the InfoVis Toolkit, Improvise, JUNG, as well as other libraries such as the Java Database Communication Toolkit (JDBC) and two Machine-Learning toolkits: Weka and RapidMiner.

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. A paper on Obvious was presented at the IEEE Visual Analytics Science and Technology conference (VAST 2011) [53]. Obvious is available at <http://code.google.com/p/obvious>.

5.5. GeneaQuilts

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

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. 5). 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. It has been integrated in commercial and open-source implementations (4 to date). See also the web page <http://www.aviz.fr/geneaquilts/>.

5.6. Diffamation

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

Animation, Edit histories, Wikipedia, Revision Control

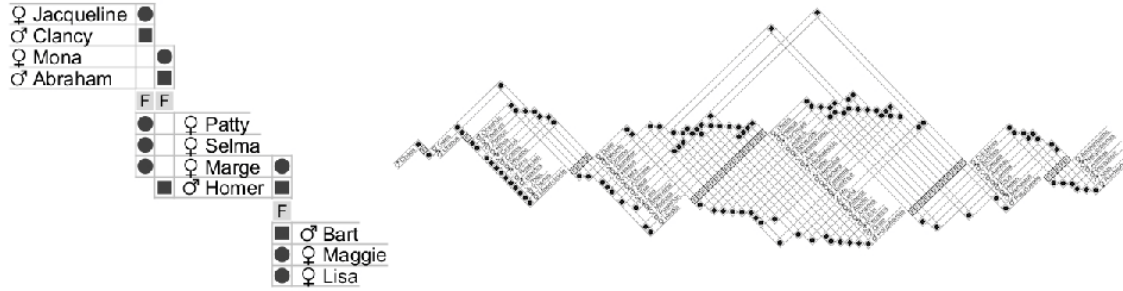


Figure 5. The genealogy of the Simpsons family (left) and of the Greek Pantheon (right), produced by the GeneaQuilts software.

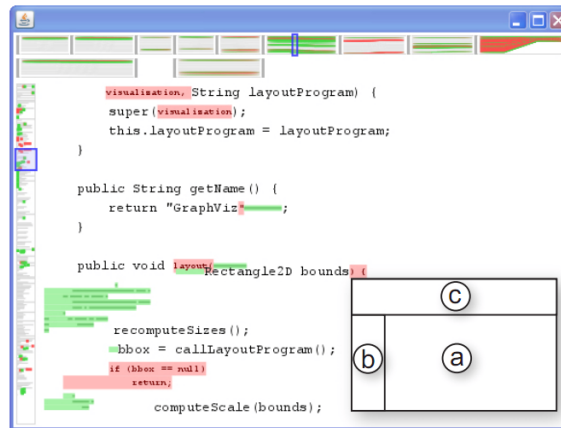


Figure 6. Screenshot the Diffamation system during a transition: (a) the document view, (b) the overview scrollbar and (c) the timeline.

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 has been presented at the plenary session of the Ubuntu Developer Summit. It is available at <http://www.aviz.fr/diffamation/>.

5.7. The InfoVis Toolkit

Participant: Jean-Daniel Fekete [correspondant].

Information Visualization, Java, Toolkit

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 section 5.4).

5.8. GraphDice

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

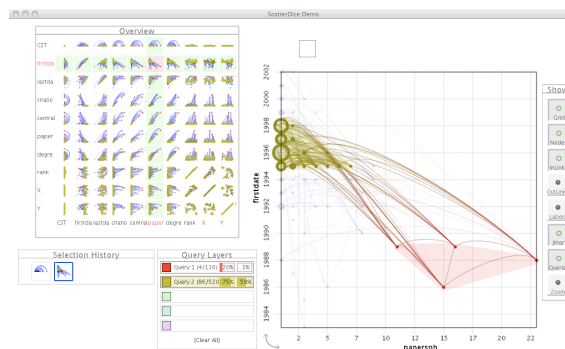


Figure 7. Screenshot the GraphDice system.

GraphDice [1] is a visualization system for exploring multivariate networks (Fig. 7). 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. For more information, see the web page at <http://graphdice.gforge.inria.fr>.

BYMOORE Exploratory Action

3. Software

3.1. Software

- **IODC:** Framework for implementing transparent iterative optimization in data centers, see Result [4.5](#).
- **P & R for neuromorphic accelerator:** A place and route software which maps a neural network graph on an analog neural network hardware.
- **Spiking neural network model:** A model for investigating and comparing various unsupervised neural network models.
- **Component programming:** A framework for programming heterogeneous multi-cores. The framework is compatible with a broad range of accelerators, including accelerators with distinct memory address spaces.

We first list the results corresponding to hardware accelerators (especially neural network accelerators), and then results corresponding to finishing, ALCHEMY (former Inria project)-related, research.

COMETE Project-Team

5. Software

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

Participants: Miguel Andrés [correspondant], Catuscia Palamidessi.

In collaborations with Dave Parker and Marta Kwiatkowska, we are developing a model checker for the probabilistic asynchronous π -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 π -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 π -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/>.

5.2. PRISM model generator

Participants: Konstantinos Chatzikokolakis [correspondant], 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 [39] and in the file README file with instructions at the URL <http://www.lix.polytechnique.fr/comete/software/README-anonmodels.html>.

The software can be download at <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 [correspondant], 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 [40] and in the file README file with instructions at the URL <http://www.lix.polytechnique.fr/comete/software/README-corners.html>.

The software can be download at <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 π -calculus

Participants: Peng Wu [correspondant], Catuscia Palamidessi.

MMCsp is a compiler from a simple probabilistic π -calculus to PRISM models. It is built on XSB, 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 2005-2007, in the context of the collaboration between the teams Comète and PRISM under the Inria/ARC **Project ProNoBis**. It is based on the papers [44] and [42].

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

COMMANDS Project-Team

5. Software

5.1. Bocop

Participants: Pierre Martinon [corresponding author], Vincent Grélard, Daphné Giorgi, Frédéric Bonnans.

Web page: <http://bocop.org>

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. The software reuses some packages from the COIN-OR library, in particular the well-known nonlinear programming solver Ipopt, and also features a user-friendly interface.

The project is supported by Inria with the arrival of Vincent Grelard as developer in October 2010, and then Daphné Giorgi in October 2012. The first prototype was released in 2011, and version 1.4 is scheduled for the end of 2012. Bocop was first successfully tested on several academic problems, see [28] available on <http://bocop.org>. In 2012, several research collaborations were initiated in fields such as bio-reactors for energy production (ref ECC), swimming micro-robots, and quantum control for medical imaging. Bocop was also featured during our participation in the Imatch "Optimisation and Control" in October.

5.2. CollAv

Participants: Hasnaa Zidani [corresponding author], Olivier Bokanowski, Anna Desilles.

This software simulates the evolution of controlled dynamical systems (possibly under uncertainties). The numerical algorithm here is based on HJB or viability approaches, and allows the design of optimal planning strategies (according to a criterion determined by the user: time, energy, ...). It also provides conflict resolution and avoidance of collisions with fixed or moving obstacles. So far, the software is used in collaboration with DGA for avoidance collision of UaVs, and by Volkswagen in some studies related to collision avoidance of cars.

5.3. OCOPHyS

Participants: Hasnaa Zidani [corresponding author], Giovanni Granato.

This is a software for optimisation-based controller design for operating in different regimes or modes of operation. The software can be used, for example, to determine the optimal management for hybrid vehicles or hybrid engines with multiple energy sources. However, the methods used in software are still quite general and can be used in many applications.

5.4. BiNoPe-HJ

Participants: Hasnaa Zidani [corresponding author], Olivier Bokanowski, Anna Desilles, Jun-Yi Zhao.

Web page: <http://www.ensta-paristech.fr/zidani/BiNoPe-HJ>

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 development of the HJB Solver has been initiated under a partnership between COMMANDS and the SME HPC-project in the period between December 2009 to November 2011. Currently, it is still maintained and improved by COMMANDS.

In 2012, two versions were released:

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

5.5. Shoot

Participant: Pierre Martinon [corresponding author].

Web page: <http://www.cmap.polytechnique.fr/~martinon/codes.html>

Shoot was designed for the resolution of optimal control problems via indirect methods (necessary conditions, Pontryagin'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.

DAHU Project-Team (section vide)

DEFI Project-Team

5. Software

5.1. RODIN

Participant: Grégoire Allaire [correspondant].

RODIN project: developing an industrial software for geometry and topology optimization of solid structures, based on the level set method.

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 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 from 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.

5.2.3. Inverse shape problems for axisymmetric eddy current problems

Participant: Zixian Jiang.

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.

5.2.4. 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.

5.2.5. 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: http://www.cmap.polytechnique.fr/~allaire/levelset_en.html

5.3.2. Conformal mapping method

Participant: Housseem 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. The latest development includes the extension of the method to the inverse scattering problem at low frequencies as introduced by Haddar-Kress (2012).

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: Housseem 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: Housseem 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: Housseem 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 code now supports the case of piecewise-wise constant by unknown background.

5.5. BlochTorreyPDESolver

Participants: Jing-Rebecca Li [correspondant], Dang Van Nguyen.

We developed numerical codes to solve the multiple compartment Bloch Torrey partial differential equation in 2D and 3D to simulate the bulk magnetization of a sample under the influence of a diffusion-encoding gradient magnetic field. We coupled a mass-conserving spatial discretization 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.

This code has been implemented in Fortran90, C++, as well as Matlab. A Matlab Toolbox with graphical user interface for the simulation of DMRI signals in 2D and 3D cellular geometries using this numerical method is being developed.

The version of the code using Finite Volume discretization on a Cartesian grid is complete (written by Jing-Rebecca Li). The version of the code using linear Finite Elements discretization is in the final testing phase (written by Dang Van Nguyen)

See the web page <http://www.cmap.polytechnique.fr/~jingrebeccali/> for more details.

DISCO Project-Team

5. Software

5.1. OreModules

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

The OREMODULES package [88], based on the commercial Maple package Ore_algebra [89], 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 [97], [99] 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) [102], [101], [87] 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/>.

5.2. Stafford

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

The STAFFORD package of OREMODULES [88] contains an implementation of two constructive versions of Stafford's famous but difficult theorem [114] 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 [109] 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 rest of Stafford's results developed in [114] have recently been made constructive in [112] (e.g., computation of unimodular elements, decomposition of modules, Serre's splitting-off theorem, Stafford's reduction, Bass' cancellation theorem, minimal number of generators) and implemented in the STAFFORD package. The development of the STAFFORD package was motivated by applications to linear systems of partial differential equations with polynomial or rational coefficients (e.g., computation of injective parametrization, 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/>.

5.3. QuillenSuslin

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

The QUILLEN-SUSLIN package [93] contains an implementation of the famous Quillen-Suslin theorem [113], [115]. 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 [93] (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>.

5.4. OreMorphisms

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

The ORE MORPHISMS package [91] of ORE MODULES [87] 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 ORE MORPHISMS package. The binary of the package is freely available at <http://www-sop.inria.fr/members/Alban.Quadrat/OreMorphisms/index.html>.

5.5. JanetMorphisms

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

The JANET MORPHISMS 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 [86]. This package computes symmetries, first integrals of motion, conservation laws, study Riemann invariants... The JANET MORPHISMS package is based on the Janet package (<http://wwwb.math.rwth-aachen.de/Janet/>).

5.6. PurityFiltration

Participant: Alban Quadrat [correspondent].

The PURITY FILTRATION package, built upon the ORE MODULES package, is an implementation of a new effective algorithm obtained in [108] which computes the purity/grade filtration [82], [83] 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.

5.7. AbelianSystems

Participants: Alban Quadrat [correspondent], Mohamed Barakat [Univ. Kaiserslautern].

The ABELIANSYSTEMS package is an implementation of an algorithm developed in [34] for the computation of the purity/grade filtration [82], [83] in the powerful `homa1g` package of GAP 4 dedicated to constructive homological algebra methods, and developed by Barakat (University of Kaiserslautern) and his collaborators (<http://homa1g.math.rwth-aachen.de/>). This package both supersedes the existing PURITYFILTRATION package which uses the non-efficient Maple Gröbner basis computation (see Section 5.6), and the original `homa1g` procedure which computes purity filtration by means of time-consuming spectral sequences. Using the `homa1g` package philosophy, the ABELIANSYSTEMS package can be used for the computation of the purity filtration of objects in different constructive abelian categories such as coherent sheaves over projective schemes as demonstrated in the `homag` package called `Sheaves` (see <http://homa1g.math.rwth-aachen.de/>).

5.8. SystemTheory

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

The SYSTEMTHEORY package is a `homa1g` 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, and Serre's reduction problem aiming at finding an equivalent system defined by fewer unknowns and fewer equations.

5.9. YALTA

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

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, [84] and [95]) and on classical continuation methods exploiting the particularities of the problem [96], [14]. We have included this year a Pade2 approximation scheme as well as H_∞ -stability properties. The package is freely available at <http://team.inria.fr/disco/fr/software/>.

GALEN Team

5. Software

5.1. Deformable Registration Software

Participant: Nikos Paragios [Correspondant].

deformable image and volume registration, 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. Dense image and surface descriptors

Participant: Iasonas Kokkinos [Correspondant].

Scale-Invariant Descriptor, Scale-Invariant Heat Kernel Signatures DISD (publicly available at <http://vision.mas.ecp.fr/Personnel/iasonas/descriptors.html>) implements the SID, SI-HKS and ISC descriptors. SID (Scale-Invariant Descriptor) is a densely computable, scale- and rotation- invariant descriptor. We use a log-polar grid around every point to turn rotation/scalings into translation, and then use the Fourier Transform Modulus (FTM) to achieve invariance. SI-HKS (Scale-Invariant Heat Kernel Signatures) extract scale-invariant shape signatures by exploiting the fact that surface scaling amounts to multiplication and scaling of a properly sampled HKS descriptor. We apply the FTM trick on HKS to achieve invariance to scale changes. ISC (Intrinsic Shape Context) constructs a net-like grid around every surface point by shooting outwards and tracking geodesics. This allows us to build a meta-descriptor on top of HKS/SI-HKS that takes neighborhood into account, while being invariant to surface isometries.

5.3. Dissimilarity Coefficient learning

Participant: Pawan Kumar [Correspondant].

weakly supervised learning, dissimilarity coefficient, structured prediction DISC (publicly available at <http://www.centrale-ponts.fr/personnel/pawan/code/DISCAPI.zip>) software provides a convenient API for dissimilarity coefficient (DISC) based learning. DISC allows the use of weakly supervised datasets (with missing information) by jointly learning a structured prediction classifier and a conditional probability distribution of the missing information. The parameters of the classifier and the distribution are learned by minimizing a user-specified dissimilarity coefficient between them.

5.4. Efficient bounding-based object detection

Participant: Iasonas Kokkinos [Correspondant].

branch-and-bound, parts detection, segmentation, DPMS implements branch-and-bound object detection, cutting down the complexity of detection from linear in the number of pixels to logarithmic (publicly available at <http://vision.mas.ecp.fr/Personnel/iasonas/dpms.html>). The results delivered are identical to those of the standard deformable part model detector, but are available in 5 to 20 times less time. This website has been visited 1500 times in 10 months.

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

Participant: Nikos Komodakis [Correspondant].

discrete optimization, Markov random field, duality, graph cuts, 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.

5.6. imaGe-based Procedural Modeling Using Shape Grammars

Participant: Iasonas Kokkinos [Correspondant].

procedural modeling, image-based building reconstruction, shape grammars GRAPES is a generic image parsing library based on re-inforcement learning (publicly available at <http://cvc.centrale-ponts.fr/>). 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.

5.7. Learning-based symmetry detection

Participant: Stavros Tsogkas [Correspondant].

Scale-Invariant Descriptor, Scale-Invariant Heat Kernel Signatures LBSD (publicly available at <http://www.centrale-ponts.fr/personnel/tsogkas/code.html>) implements the learning-based approach to symmetry detection published in [32]. It includes the code for running a detector, alongside with the ground-truth symmetry annotations that we have introduced for the Berkeley Segmentation Dataset (BSD) benchmark.

5.8. Texture Analysis Using Modulation Features and Generative Models

Participant: Iasonas Kokkinos [Correspondant].

Texture, modulation, generative models, segmentation, 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.

5.9. Sparse Prediction

Participant: Andreas Argyriou [Correspondant].

Sparse prediction, K-support norm, SPARSE_K is a sparse prediction code (publicly available at http://www.centrale-ponts.fr/personnel/andreas/code/sparse_k/sparse_k.tar) using regularization with the k -support norm, which we have introduced [36]. The algorithm uses an accelerated first-order method similar to Nesterov's method.

GECO Team

5. Software

5.1. IRHD

We developed a first version of a software for reconstruction of corrupted and damaged images, named IRHD (for Image Reconstruction via Hypoelliptic Diffusion). One of the main features of the algorithm on which the software is based is that we don't use any information about the location and character of the corrupted places; this allows us to work with real non-academic images. Another important advantage is that this method is massively parallelizable; this allows to work with sufficiently large images. Theoretical background of the presented method is based on the model of geometry of vision due to Petitot, Citti and Sarti. The main step is numerical solution of the equation of 3D hypoelliptic diffusion. IRHD is based on Fortran. Alexey Remizov is in charge with the development of the software, in collaboration with Ugo Boscain and Jean-Paul Gauthier.

GEOMETRICA Project-Team

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 Lorient, Abdelkrim Mebarki, Naceur Meskini, Andreas Meyer, Sylvain Pion, Marc Pouget, François Rebufat, Laurent Rineau, Laurent Saboret, Stéphane Tayeb, Jane Tournois, 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 Ipe 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 14 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.

GRACE Team

4. Software

4.1. ECPP

F. Morain has been continually 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, using the fast version that he started developing in 2003. All of the code is written in C, and based on publicly available packages (GMP, mpfr, mpc, mpfrx).

4.2. 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 most recent algorithms for solving all subtasks. The large prime case is relevant to cryptographic needs. The $p = 2$ case, though not directly useful, is a good testbed for the FFAST program of Luca De Feo. This program forms a gforge project.

4.3. TIFA

The TIFA library (short for Tools for Integer FActorization), initially developed in 2006, 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.

It is now available online at <http://www.lix.polytechnique.fr/Labo/Jerome.Milan/tifa/tifa.xhtml>; it is distributed under the Lesser General Public License (version 2.1 or later).

4.4. Quintix

The Quintix library is a Mathemagix package, available at <http://www.mathemagix.org/www/main/index.en.html>. Quintix is a very efficient library for Galois rings, extensions of Galois rings and root-finding in Galois rings, 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 [23]. The source code is distributed under the General Public License (version 2 or higher).

4.5. finitefieldz

G. Quintin wrote the finitefieldz package which provides arithmetic for finite fields (of any characteristic) and towers of finite fields. He wrote this package with the help of Grégoire Lecerf during the first year of his PhD thesis. The package uses univariate polynomials and multiprecision integers, and also provides univariate polynomial root finding and factorization over finite fields.

4.6. Decoding

Decoding is a standalone C library licensed under the GPLv2. Its primary goal is to implement Guruswami–Sudan list decoding-related algorithms, as efficiently as possible. Its secondary goal is to give an efficient tool for the implementation of decoding algorithms (not necessarily list decoding algorithms) and their benchmarking.

For now (2012/12/13) you can use the library and have a working list decoding algorithm, but there is no unique decoding algorithm (though you can tell decoding to list decode up to half the minimum distance). The library is being further developed and more algorithms will be added.

The library was presented at the 2012 International Symposium on Symbolic and Algebraic Computation.

GRAND-LARGE Project-Team

4. Software

4.1. 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.

4.2. 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 or 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/>

4.3. 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.

4.4. 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 [83].

This software is available at <http://v-ds.lri.fr/>

4.5. 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.

4.6. 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 in 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 insert data movements calls in the main code. Since the data set can be large in Grid application, the OpenWP runtime organize 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 is called, the DHT is consulted to obtain the position of the parameter data sets in the system. When a remote function terminates its execution, it stores the result data sets and store 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 ¹. 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 meshId results to every computing node and launching the distributed execution of the patches.

4.7. 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 [72]. The SuperLU_DIST is available at <http://crd.lbl.gov/~xiaoye/SuperLU/>.

4.8. 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

¹project 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.

4.9. Clay

Participant: Cédric Bastoul.

Clay is a free software and library devoted to semi-automatic optimization using the polyhedral model. It can input a high-level program or its polyhedral representation and transform it according to a transformation script. Classic loop transformations primitives are provided. Clay is able to check for the legality of the complete sequence of transformation and to suggest corrections to the user if the original semantics is not preserved (experimental at this document redaction time). Main authors include Joël Poudroux and Cédric Bastoul.

4.10. 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

4.11. MIDAPACK for CMB data analysis

Participants: Laura GRIGORI [correspondant], Mikolaj SZYDLARSKI [correspondant].

Midapack is a library aiming at the crucial stages down the CMB data analysis pipeline. In its current form, the library provides tools for computing spherical harmonic transforms on heterogeneous architectures, and algorithms for finding the solution to a generalized least squares problem. The algorithms are described in [36] and [44]. See also the web page <http://pages.saclay.inria.fr/laura.grigori/soft.html>.

- Version: 1.0

4.12. 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 [14]. 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. Different approaches to reduce communication are compared in [26].

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

4.13. 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 [6].

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>.

HIPERCOM Project-Team

5. Software

5.1. RPL P2P

Participants: Emmanuel Baccelli [correspondant], Oliver Hahm, 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. MPR-OSPF

Participants: Emmanuel Baccelli [correspondant], Juan-Antonio Cordero.

MPR-OSPF is an implementation of RFC5449, providing OSPF-compatible routing in hybrid networks composed of both mobile ad hoc routers and fixed wired networks. The implementation is based on Quagga/Zebra. See also the web page <http://ospfmanet.gforge.inria.fr>.

- Version: 0.4

5.3. 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.

5.4. 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.

5.5. OPERA protocol

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

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.

5.6. 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.

IN-SITU Project-Team

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 [45] 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 $20\,480 \times 6\,400$ 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 ($86\,400 \times 43\,200$ pixels) augmented with country borders ESRI shapefiles. (b) Panning and zooming in Spitzer's Infrared Milky Way ($396\,032 \times 12\,000$ 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)
- Software benefit: See E. Pietriga, S. Huot, M. Nancel, R. Primet, Rapid Development of User Interfaces on Cluster-Driven Wall Displays with jBricks, EICS '11: Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems, pages 185-190, June 2011
- 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, 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 [26].

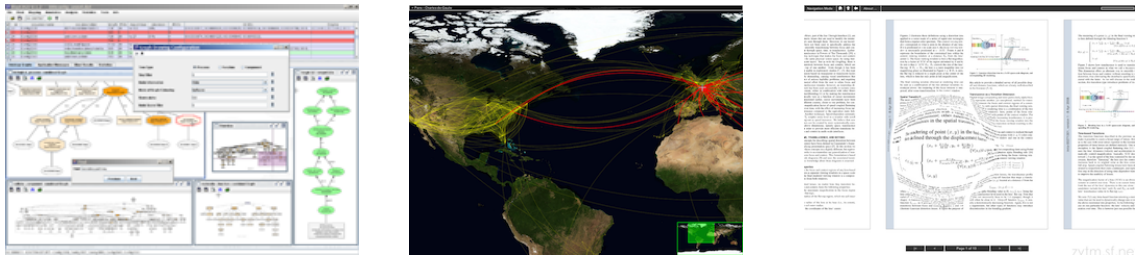


Figure 2. ZVTM used in various applications

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 GraphViz¹ (Figure 2 - right). The development of the toolkit is now supported by Inria. More information can be found at <http://zvtm.sourceforge.net> and [52] and [51].

- ACM: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)
- Software benefit: See Pietriga, A Toolkit for Addressing HCI Issues in Visual Language Environments, IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC '05), pages 145-152, September 2005
- License: LGPL
- Type of human computer interaction: Graphique
- OS/Middleware: Java (Linux, Mac OS X, Windows)
- Required library or software: several, managed through Maven
- Programming language: Java

5.3. The SwingStates Toolkit

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

¹<http://www.graphviz.org>

SwingStates [37] 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)
- Software benefit: See C. Appert and M. Beaudouin-Lafon (2008) SwingStates: Adding State Machines to Java and the Swing Toolkit. *Software: Practice and Experience*, 38(11):1149 - 1182.
- 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 [38], 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 [37] and ICon [45].

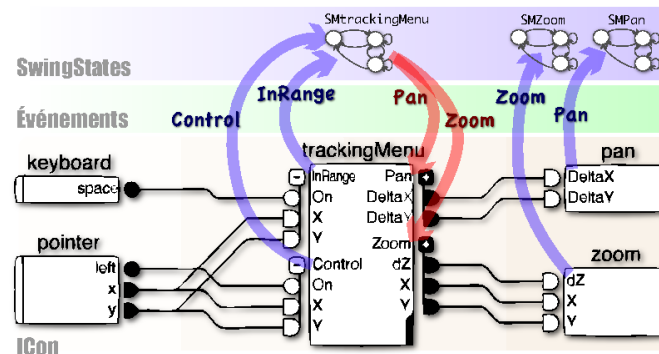


Figure 4. State machines and data flow in FlowStates

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.

FlowStates is available at <http://www.lri.fr/~appert/FlowStates/>.

- ACM: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)
- Software benefit: See C. Appert, S. Huot, P. Dragicevic and M. Beaudouin-Lafon (2009) FlowStates: Prototypage d'applications interactives avec des flots de données et des machines à états. In Proceedings of IHM 2009. ACM, pages 119-128.
- 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 [8] 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.

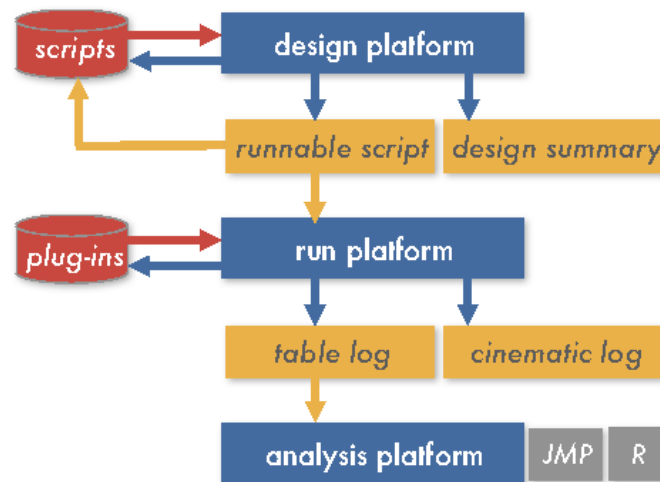


Figure 5. The architecture of the Touchstone platform

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).

Touchstone is available at <http://code.google.com/p/touchstone-platforms/> under a BSD License.

- ACM: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)
- Software benefit: See W. Mackay, C. Appert, M. Beaudouin-Lafon, O. Chapuis, Y. Du, JD. Fekete and Y. Guiard (2007) TouchStone: Exploratory Design of Experiments. In Proceedings of ACM CHI 2007 Conference on Human Factors and Computing Systems. ACM, pages 1425-1434.
- 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 [43] 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 $2D\frac{1}{2}$ 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 [54]. This year we used again Metisse to implement novel desktop interaction techniques [4].

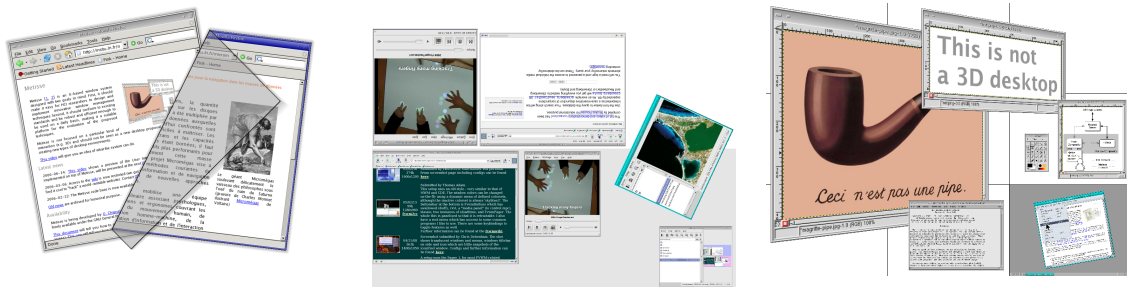


Figure 6. Sample window management techniques implemented with Metisse: extended paper metaphor (left), interactive table configuration that allows to duplicate and rotate windows (middle) and zoomable 3D desktop (right).

- Web: <http://insitu.lri.fr/metisse/>
- ACM: H.5.2 [User Interfaces]: Windowing systems
- Software benefit: see [43], [54], [44], [47] and [4].
- License: GPL
- Type of human computer interaction: Graphique
- OS/Middleware: X Window et Mac OS X
- Required library or software: OpenGL via nucleo² and some usual C/C++ libraries
- Programming language: * C/C++

5.7. Wmtrace

Participant: Olivier Chapuis [correspondant].

Wmtrace [42] 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.

- Web: <http://insitu.lri.fr/~chapuis/software/wmtrace/>.
- ACM: H.5.2 [User Interfaces]: Windowing systems
- Software benefit: see [42], [47], [41].
- License: GPL

²<http://interaction.lille.inria.fr/~rousseau/projects/nucleo/index.html>

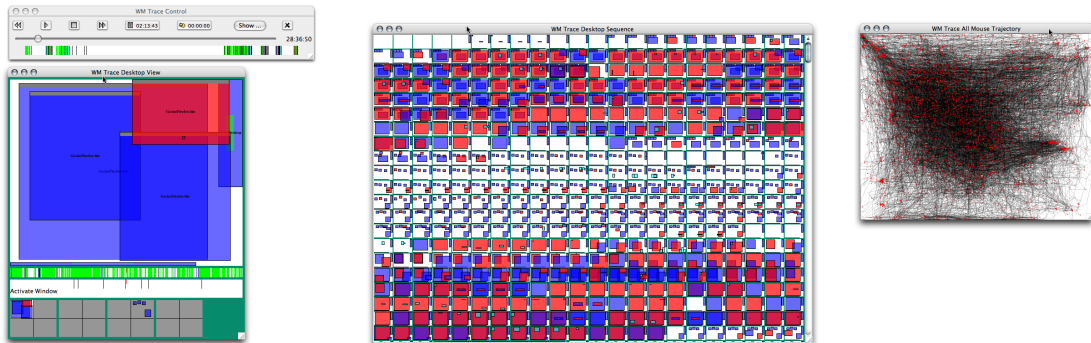


Figure 7. VCR-like interface, session overview and sample plots of mouse trajectories (black) and mouse clicks (red)

- Type of human computer interaction: Daemon and Graphique
- OS/Middleware: X Window (daemon) and Java (VCR interface)
- Required library or software: all X libraries (daemon) and Java (VCR interface)
- Programming language: * C and Java

5.8. The Substance Middleware

Participants: Michel Beaudouin-Lafon [correspondant], Clemens Klokrose, 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 [48]. 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 [53].

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.

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)

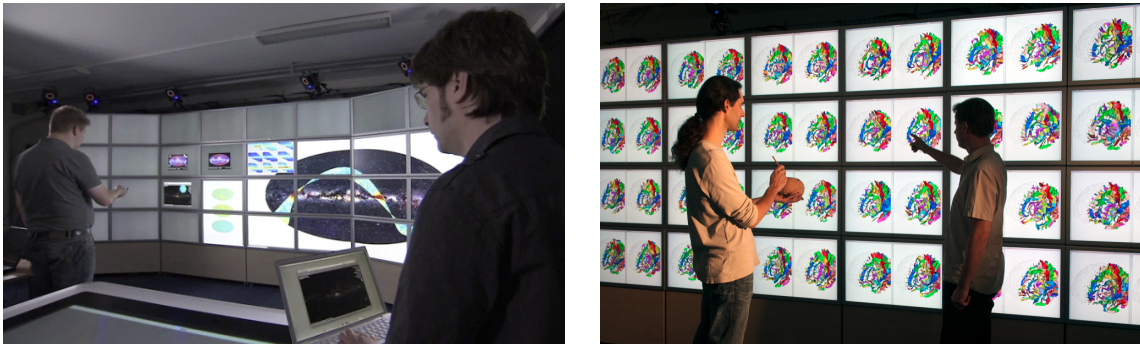


Figure 8. The Canvas (left) and SubstanceGrise (right) applications developed with Substance.
(©CNRS-Phototheque - Cyril FRESILLON for SubstanceGrise).

- Software benefit: See T. Gjerlufsen, C. Klokmoose, J. Eagan, C. Pillias and M. Beaudouin-Lafon (2011) Shared Substance: Developing Flexible Multi-Surface Applications. In CHI '11: Proceedings of the 29th international conference on Human factors in computing systems. ACM, pages 3383-3392.
- 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, Wendy Mackay.

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 [46]. 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 [39]. 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> under a GNU GPL 3.0 licence.

- ACM: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)
- Software benefit: See J. Eagan, W. Mackay and M. Beaudouin-Lafon (2011) Cracking the Cocoa Nut: User Interface Programming at Runtime. In UIST 2011: Proceedings of the 24th ACM Symposium on User Interface Software and Technology. ACM, pages 225-234.

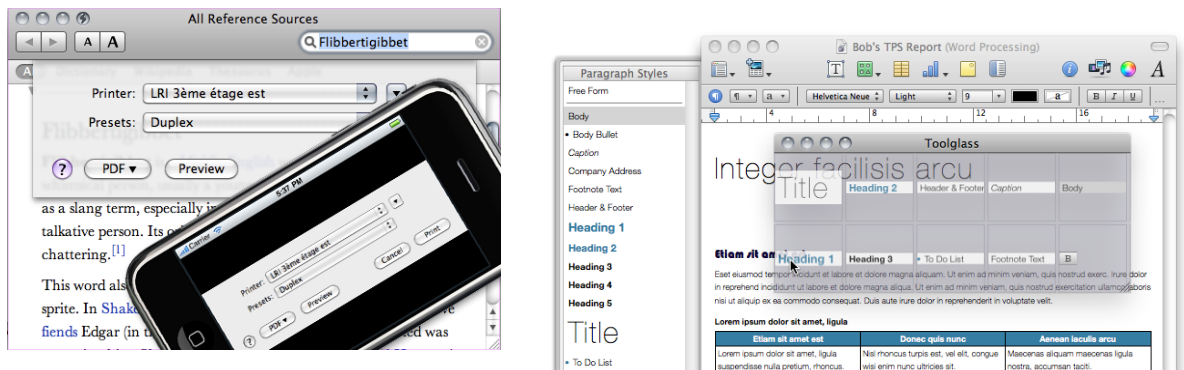


Figure 9. Using Scotty to teleport a window of a Mac OS X application onto an iPhone (left) and to create a toolglass in the Pages word processor (right).

- OS/Middleware: Mac OS X
- Required library or software: none
- Programming language: Objective-C, Python

MACS Project-Team

5. Software

5.1. FELISCE

Participants: Dominique Chapelle, Sébastien Gilles [correspondant], Philippe Moireau.

FELISCE – standing for “Finite Elements for Life SCiences and Engineering” – is a new finite element code which the MACS and REO teams have decided to jointly develop in order to build up on their respective experiences concerning finite element simulations. One specific objective of this code is to provide in a unified software environment all the state-of-the-art tools needed to perform simulations of the complex cardiovascular models considered in the two teams – namely involving fluid and solid mechanics, electrophysiology, and the various associated coupling phenomena. FELISCE is written in C++, and may be later released as an opensource library. <https://gforge.inria.fr/projects/felisce/>

5.2. HeartLab

Participants: Matthieu Caruel, Dominique Chapelle, Alexandre Imperiale, Philippe Moireau [correspondant].

The heartLab software is a library written in (64 bits compatible) Matlab and C (mex functions) designed to perform both simulation and estimation (based on various types of measurements, e.g. images) of the heart mechanical behavior. Started in 2006, it is already quite large (about 60,000 lines), and is used within the CardioSense3D community.

The code relies on OpenFEM for the finite element computations, and the implementation was performed with a particular concern for modularity, since modeling and estimation use the same finite element operators. This modularity also allows to couple the code with other FEM solvers, such as LifeV and Mistral developed in the Reo team-project. In particular, we are now able to include perfusion and electrical coupling with LifeV using PVM, and fluid-structure interaction using Mistral.

We also included geometric data and tools in the code to define heart anatomical models compatible with the simulation requirements in terms of mesh quality, fiber direction data defined within each element, and referencing necessary for handling boundary conditions and estimation, in particular. These geometries are analytical or come from computerized tomography (CT) or magnetic resonance (MR) image data of humans or animals.

We recently incorporated numerous non-linear data assimilation observation operators based on medical imaging post-processing to be able to now perform estimation with a large variety of medical imaging modalities.

5.3. MITCNL

Participants: Dominique Chapelle [correspondant], Marina Vidrascu [REO team].

The package MITCNL is a set of subroutines that implements the triangular MITC3, MITC6 and quadrilateral MITC4 and MITC9 shell elements for large displacements [2]. We use it as a basis for new developments of shell elements, in particular within Modulef (<http://www-rocq.inria.fr/modulef/>). It can be easily interfaced with most finite element codes as well. We also license this package to some of our partners for use with their own codes.

5.4. OpenFEM: a Finite Element Toolbox for Matlab and Scilab

Participants: Dominique Chapelle, Philippe Moireau [correspondant].

OpenFEM (<http://www.openfem.net>) is an *opensource* finite element toolbox for linear and nonlinear structural mechanics within the Matlab and Scilab matrix computing environments. This software is developed in a collaboration between Macs and the SDTools company ¹. Performing finite element analyses within a matrix computing environment is of considerable interest, in particular as regards the ease of new developments, integration of external software, portability, post-processing, etc.

This Library is the core of the finite element computations of HeartLab where a specific version have been developed with the help of Cesare Corrado from Reo.

5.5. SHELDDON

Participants: Dominique Chapelle [correspondant], Marina Vidrascu [REO team].

SHELDDON (SHELLs and structural Dynamics with DOrain decomposition in Nonlinear analysis) is a finite element library based on the Modulef package which contains shell elements, nonlinear procedures and PVM subroutines used in domain decomposition or coupling methods.

5.6. Verdandi

Participants: Dominique Chapelle, Marc Fragu, Vivien Mallet [Clime team], Philippe Moireau [correspondant].

Verdandi is an opensource (LGPL) software library aiming at providing assimilation data methods and related tools. Mainly targeted at large systems arising from the discretization of PDEs, it is intentionally devised as generic, which allows for applications in a wide range of problems (biology and medicine, environment, image processing...). See also the web page <http://verdandi.gforge.inria.fr/>, with a complete documentation in English. The first stable version (1.0) was released in June 2012 and contains most of the major data assimilation algorithms of both variational and sequential types. The actual version (1.4) contains additional estimation algorithm and parallel capabilities. Note that some specific developments are performed with particular regard to cardiac modeling applications, as Verdandi is partly funded by – and distributed within – the euHeart project and is now referenced in the following peer-reviewed article [15]

- ACM: Mathematical software
- AMS: System theory; control
- Software benefit: Verdandi is the only *generic* data assimilation library
- License: LGPL (2.1 or any later version)
- Type of human computer interaction: Command line and configuration files
- OS/Middleware: Linux, MacOS ou Windows
- Required library or software: Seldon (LGPL, <http://seldon.sourceforge.net/>)
- Programming language: C++, ISO/IEC 14882: I998(E) Python, version 2.6
- Documentation: Doxygen and utilisation manual in english

¹<http://www.sdtools.com>

MAXPLUS Project-Team

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 [70], [134]. 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 [70], [134]. 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 [87]. 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 [100]), 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 [87]. Several internships allowed us to implement in Scilab, C or C++, and to test such algorithms (see the work of Vishesh Dhingra [100]), or its combinaison 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 combinaison 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 permet aussi de réaliser d'autres opérations fondamentales, comme le calcul du complexe polyédral associé à un polyèdre donné (au sens de Develin et Sturmfels [98]), ou le calcul de cônes tangents tropicaux. Enfin, elle fournit toutes les primitives permettant d'utiliser les polyèdres tropicaux en tant que domaine abstrait numérique, afin de déterminer des invariants de programmes ou systèmes faisant intervenir les opérations min et max (voir [67]).

TPLib est aujourd'hui utilisé dans le logiciel Polymake [117], 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).

Le développement d'interfaces avec d'autres logiciels est désormais facilité grâce à la présence de *bindings* dans le langage C. Grâce à cela, un prototype d'interface a été réalisé entre TPLib et l'outil VerifyTAPN (<https://launchpad.net/verifytapn>), qui permet la vérification de réseaux de Pétri avec arcs temporisés (voir §6.5.7). De même, une interface à la bibliothèque de domaines abstraits numériques APRON [127] 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). Besides, the library allows to perform several fundamental operations over tropical polyhedra, such as computing the associated polyhedral complex (see Develin and Sturmfels [98]), or determining the tropical tangent cone at any point. Finally, it provides all the primitives allowing to use tropical polyhedra as an numerical abstract domain, in order to determine program/system invariants involving the operations min and max (see [67]).

TPLib is now used in the software Polymake [117], 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.

The development of further interfaces is now easier thanks to the distribution of bindings in C language. Using these bindings, a prototype of interface has been created between TPLib and the model-checker VerifyTAPN (<https://launchpad.net/verifytapn>), which allows the verification of timed-arc Petri Nets (see §6.5.7). An interface to the numerical abstract domain APRON [127] is also under development.

MEXICO Project-Team

5. Software

5.1. Software

5.1.1. *libalf: the Automata Learning Framework*

Participant: Benedikt Bollig [correspondant].

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 facily interchanging learning algorithms and combining domain-specific features in a plug-and-play fashion. Its modular design and its 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 [correspondant], 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, i.e. Petri nets with read arcs [42],[48]. 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 and produce a smaller unfolding. *Cunf* has recently been extended with a verification component that takes advantage of these features [70]. More details can be found at <http://www.lsv.ens-cachan.fr/~rodrigue/tools/cunf/>. Moreover, *Cunf* has been integrated into the *CosyVerif* environment (see section 5.1.4).

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

Participants: Hilal Djafri, Benoît Barbot [correspondant].

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 ; recently, it has been equipped with functionalities for rare event analysis. *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/>

5.1.4. *COSYVERIF*

Participants: Serge Haddad [correspondant ?], Benoît Barbot.

CosyVerif is a software environment whose goal is the formal specification and verification of dynamic systems.

It has been designed in order to:

- support different formalisms with the ability to easily create new ones, - provide a graphical interface for every formalism, - include verification tools called via the interface as a web service, - offer the possibility for a developer to integrate his/her own tool, also allowing it to interact with the other tools.

This environment consists of two software tools: Coloane, the graphical interface, and Alligator, an integration framework based web services. It is enlarged with the existing verification tools developed in our laboratories (founding members or partners). Why ?

The development of Cosyverif has been decided and it is supported by three partners of the Parisian verification group, MeFoSyLoMa. This group is composed of seven teams. and the founding members of are LIP6, LIPN and LSV. First, these members aim at sharing their tools, comparing and supporting industrial case studies and finally making them long-lasting. Second, they also want to promote the practice of formal verification in industry and thus they intend to ease the task of integration of new formalisms and tools.

It is managed by a steering committee consisting of researchers and engineers. It decides strategic orientations as well as technical choices. Current Tools

Two formalisms are supported: automata and Petri nets, both with extensions. Most of the tools are related to Petri nets. Some of them perform structural analyses like invariant computations. while other tools perform behavioural analyses: symbolic reachability graph building, unfolding, stochastic simulations, etc. Finally some of them transform high-level nets into low-level ones. All the developed software are open source and free software tools. Alligator is published under the GNU Affero General Public License (AGPL) version 3 ; Coloane is published under the Eclipse Public License (EPL) version 1 .

Three engineers have worked or are currently working on COSYVERIF:

- Francis Hulin-Hubard, part-time (CNRS) in 2012;
- Clément Desmoulins , full-time (ANR), 6 months; and
- Alban Linard, full-time Inria engineer, for 2 years.

OAK Team

5. Software

5.1. Amada

Name: Amada (<https://team.inria.fr/oak/amada/>)

Contact: Jesús Camacho-Rodríguez (jesus.camacho-rodriguez@inria.fr)

Other contacts: Zoi Kaoudi (zoi.kaoudi@inria.fr), Ioana Manolescu (ioana.manolescu@inria.fr), Dario Colazzo (dario.colazzo@lri.fr), François Goasdoué (fg@lri.fr)

Presentation: A platform for Web data management in the Amazon cloud

5.2. Nautilus Analyzer

Name: Nautilus Analyzer (<http://nautilus.saclay.inria.fr/>)

Contact: Melanie Herschel (melanie.herschel@lri.fr)

Other contacts: n.a.

Presentation: A tool for analyzing and debugging SQL queries using why-provenance and why-not provenance.

5.3. RDFViewS

Name: RDFViewS (<http://tripleo.saclay.inria.fr/rdfvs/>)

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

5.4. ViP2P

Name: ViP2P (views in peer-to-peer, <http://vip2p.saclay.inria.fr>)

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.

5.5. 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

5.6. 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

5.7. XUpTe

Name: XUpTe (XML Update for Tempora documents)

Contact: Dario Colazzo (colazzo@lri.fr)

Other contacts: Nicole Bidoit (bidoit@lri.fr), Mohamed-Amine Baazizi (amine.baazizi@gmail.com)

Presentation: a type-based optimizer for representing and updated XML temporal sata

5.8. XPUQ

Name: XPUQ (XML Partitioning for Updates and Queries)

Contact: Dario Colazzo (colazzo@lri.fr)

Other contacts: Nicole Bidoit (bidoit@lri.fr), Noor Malla (noorwm@hotmail.com)

Presentation: a static analyzer and partitioner for XML queries and updates

PARIETAL Project-Team

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, Elvis Dohmatob, 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 license. It is available in NeuroDebian.

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

- Version: 0.3

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 multi-threaded.

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

- Version: 2.0

5.4. Scikit learn

Participants: Bertrand Thirion, Gaël Varoquaux [correspondant], Jaques Grobler, Alexandre Gramfort, Fabian Pedregosa, Virgile Fritsch.

Scikit-learn is an open-source 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 more than 60 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: 0.12
- Programming language: Python, C/Cython

PARSIFAL Project-Team

5. Software

5.1. Abella

Participants: Kaustuv Chaudhuri [correspondant], Matteo Cimini, Dale Miller.

Abella is an interactive theorem prover based on the two-level logic logic approach. It consists of a sophisticated reasoning logic that supports induction, co-induction, and generic reasoning, and a specification logic that is based on logic programming. Abella was initially designed to reason about simple second-order Lambda Prolog programs, which is sufficient for the computational specifications.

During 2012, as part of the RAPT Associated Team, Chaudhuri and Yuting Wang (intern from Univ. Minnesota) have been working on extending the expressive power of both levels of the Abella system. The following modifications have been made.

- We have extended the specification logic to support the full Lambda Prolog, which can be used to provide succinct higher-order specifications that tend to be unnatural and difficult to reason about with only second-order Lambda Prolog programs.
- We have extended the type system of Abella from simple types to parametrically polymorphic types. This is a significant improvement in the user-friendliness of the system as a lot of code does not have to be manually monomorphised and duplicated any more.
- We have experimented with extending the type system of Abella even further to higher-order predicate quantification. The theoretical basis of this work is a part of ongoing research, although we already have a number of examples of practical benefits of this extension.
- Finally, several improvements have been made to Abella's proof language to make the proofs more robust and reusable. We intend to make a more drastic change to the proof language in the future that will make proofs more declarative and high level.

The core development of Abella has also been centralized, with a single canonical repository and a new webpage: <http://abella-prover.org>. These resources are managed jointly by members of Parsifal and our colleagues at the University of Minnesota.

The next version of Abella, version 2.0, is in beta testing with expected release early in 2013.

5.2. Bedwyr

Participants: Quentin Heath, Dale Miller [correspondant].

During 2012, Quentin Heath has made the following important improvements to the Bedwyr model checking system.

- The concrete syntax for Bedwyr and Abella have been unified. Now, both systems can load the definitions and theorems developed in the other system. Eventually, we expect to have our model checker (Bedwyr) and interactive theorem prover (Abella) share theories and proofs.
- The documentation, distribution, and testing of Bedwyr were all improved, greatly increasing the usability of this system.
- The underlying support for logic has also been increased. In particular, the Bedwyr system contains a *tabling* mechanism which is capable of remembering past successful proofs (it can even support a finite failure as a successful proof of a negation). The most recent version of Bedwyr allows one to actually program the table in rather sophisticated ways. For example, simple lemmas can be loaded into the table and these lemmas can be used to greatly extend the range of what is tabled (remembered). We are currently examining different trade-offs between different styles of reasoning in the table (backchaining vs forwardchaining).

The work of Heath is being done in the context of the BATT ADJ project funded by Inria.

See also the web page <http://slimmer.gforge.inria.fr/bedwyr/>.

5.3. Psyche

Participants: Mahfuza Farooque, Stéphane Graham-Lengrand [correspondant].

Psyche (*Proof-Search factorY for Collaborative HEuristics*) is a modular programme for universal proof-search in classical logic. The motivation is twofold:

On the one hand, prove some mathematics of the broadest range while making the most of problem-specific techniques; On the other hand, gain high confidence about the correctness of the proofs produced without having to rely on a proof-checker.

The architecture is that of an interaction between a trusted universal kernel and smart plugins that are meant to be efficient at solving certain kinds of problems:

The kernel contains the mechanisms for exploring the proof-search space in a sound and complete way, taking into account branching and backtracking. The output of Psyche comes from the (trusted) kernel and is therefore correct by construction. The plugins then drive the kernel by specifying how the branches of the search space should be explored, depending on the kind of problem that is being treated. The quality of the plugin is then measured by how fast it drives the kernel towards the final answer.

Version 1.0 of Psyche (released 4/9/2012) handles classical propositional logic, and its proof-search mechanism is simply the incremental construction of proof-trees in the polarised and focussed sequent calculus. The mechanism is driven by a plugin that emulates the behaviour of a SAT-solver (DPLL), with non-trivial features such as the eager application of the Unit Propagation rule, conflict analysis, backjumping and clause learning.

Psyche's input for that kind of SAT-problem is a file given in the standard DIMACS format.

See also the web page <http://www.lix.polytechnique.fr/~lengrand/PSI/index.php?page=Psyche/index>.

POEMS Project-Team

5. Software

5.1. Introduction

We are led to develop two types of software. The first one is prototype softwares : various softwares are developed in the framework of specific research contracts (and sometimes sold to the contractor) or during PhD theses. They may be also contributions to already existing softwares developed by other institutions such as CEA, ONERA or EDF. The second category is an advanced software which are intended to be developed, enriched and maintained over longer periods. Such software is devoted to help us for our research and/or promote our research. We have chosen to present here only our advanced software.

5.2. XLIFE++

XLIFE++ is a new Finite Element library in C++ based on philosophy of the previous library MELINA in Fortran but with new capabilities (boundary element, discontinuous Galerkin methods, more integrated tools (in particular mesh tools) and high performance computing skills (multithread and GPU computation). It is licensed under LGPL and it is developed in the context of the european project SIMPOSIUM (FP7/ICT, leader CEA/LIST, from september 2011 to august 2014). There are also academic partners : Irmar-univ. Rennes and Lama-univ. Paris-Est .

In 2012, all development tools have been set up : versioning using Git, repository on Inria-Gforge, compiling and installing tools using Cmake, documentation in TeX and using Doxygen), test processing. All fundamentals library have been developed and checked (String, Function, Messages, Matrix, Vector, ...) and major libraries are done : geometry (Mesh description and tools), space,form, operators to deal with variational descriptions of PDE problem, finiteElements describing all finite elements description, term wich deals with sparse matrix representation of linear or bilinear forms involved in variational problems and finally the solvers library. A first version of the library should be published soon.

POPIX Exploratory Action

5. Software

5.1. MONOLIX

Participant: Marc Lavielle.

MONOLIX is an easy, fast and powerful tool for parameter estimation in non-linear mixed effect models, model diagnosis and assessment, and advanced graphical representation. It is a platform of reference for model-based drug development. Pharmacometricians and biostatisticians can rely on MONOLIX for population analysis and to model PK/PD and other complex biochemical and physiological processes.

MONOLIX was developed by Inria until June 2011. The spin-off Lixoft now develops and supports MONOLIX.

POPIX collaborates closely with Lixoft to convert research results into new user features available in MONOLIX.

5.2. MLXTRAN

Participant: Marc Lavielle.

MONOLIX is associated with MLXTRAN, a powerful and immediately readable declarative language for describing complex pharmacometric and statistical models. MLXTRAN can be used and interfaced with various environments, eg. R, Matlab, etc.

POPIX collaborates closely with LIXOFT on the definition of the specifications and the syntax of MLXTRAN. Implementation is then ensured by LIXOFT.

5.3. Clinical trial simulator

Participants: Marc Lavielle, Elodie Maillot, Laura Brocco.

A clinical trial simulator (CTS) enables effective implementation of the learn-and-confirm paradigm in drug development. Through simulations the anticipated success rate of a future trial can be estimated. For various reasons industry has not embraced currently available software for trial simulation. A new tool is essential for Model Based Drug Development (MBDD).

POPIX is responsible for developing a new CTS within the DDMoRe project (see below). Version 2 of the CTS is available since June 2012. The capabilities of this new version comprise:

- Flexible study designs used in Phase 2 of clinical drug development: parallel group studies, crossover studies, complex treatments defined as a combination of different treatments
- Simulation of patients sampled from a joint distributions or using an external data file
- Simulation of exposure to the investigated drug and several types of drug effects related to drug exposure (continuous, categorical, count, time-to-event)
- Graphics and statistical tests
- Automatic reporting

REGULARITY Project-Team

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/> 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, France, the USA, and Serbia.

We have produced this year a major release of FracLab (version 2.1).

SECSI Project-Team

5. Software

5.1. Tookan

Participants: Graham Steel [correspondant], Romain Bardou.

See also the web page <http://tookan.gforge.inria.fr/>.

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 [48] and a six-month licence was granted to Boeing to use the tool. In 2011, a contract was signed with a major UK bank. Tookan is now the subject of a CSATT transfer action resulting in the hiring of an engineer, Romain Bardou, who started on September 1st, 2011. During 2012 Bardou and Steel implemented a new version of Tookan that is intended to form the technological basis for a spin-off company to be created in 2013. As a result of the transfer of Graham Steel and Romain Bardou to team Prosecco, this project is being continued in that team.

5.2. Orchids

Participants: Jean Goubault-Larrecq [correspondant], Hedi Benzina, 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.

Nasr-Eddine Yousfi followed up on Baptiste Gourdin, starting from December 2011, on an ITI engineer position allotted by Inria's CSATT. He mostly explored ways of writing security meta-policies for confidentiality of sensitive data.

Orchids will be the core of a contract between Inria and DGA, to be signed in December 2012, for three years.

SELECT Project-Team

5. Software

5.1. MIXMOD software

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

Mixture model, cluster analysis, discriminant analysis

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 2010, 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 now available <http://cran.r-project.org/web/packages/Rmixmod/index.html>.

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

TAO Project-Team

5. Software

5.1. Metis

Participants: Olivier Teytaud [correspondent], Adrien Couëtoux, Jérémie Decock, Jean-Joseph Christophe.

Energy, Optimization, Planning

Many works in Energy Optimization, in particular in the case of high-scale sequential decision making, are based on one software per application, because optimizing the software eventually implies losing generality. Our goal is to develop with Artelys a platform, Metis, which can be used for several applications. In 2012 we interfaced existing codes in Artelys and codes developed in the Tao team; experiments have been performed and test cases have been designed. A main further work is the introduction of generic tools for stochastic dynamic programming into the platform, for comparison and hybridization with other tools from the UCT-SIG.

Our favorite challenge is the hybridization of “classical” tools (based on constraint satisfaction problems, or mixed integer linear programming or mixed integer quadratic programming), which are fast and accurate, with non-linear solvers which can take care of a sophisticated (non-linear) model.

5.2. MoGo

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

Go, Multi-armed bandit

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). Recent results include 7 wins out of 12 against professional players (in Brisbane, 2012). However, the work in the UCT-SIG has now shifted to energy management.

5.3. CMA-ES: Covariance Matrix Adaptation Evolution Strategy

Participant: Nikolaus Hansen [correspondent].

Evolutionary Computation, stochastic optimization, real-parameter optimization

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 computation³ 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

5.4. COmparing Continuous Optimizers

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

Evolutionary Computation, stochastic optimization, real-parameter optimization, benchmarking, derivative free optimization

³H.-G. Beyer (2007). Evolution Strategies, *Scholarpedia*, page 1965.

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 is under continuous development and has been used for the GECCO 2009, 2010 and 2012 workshops on “Black Box Optimization Benchmarking” (BBOB) (see Section 3.3), and serves as a basis for the test platform in the CSDL project.

Link: <http://coco.gforge.inria.fr/doku.php>

5.5. MultiBoost

Participants: Balázs Kégl [correspondent], Djalel Benbouzid.

multi-class, multi-label classification

The MultiBoost package [4] provides a fast C++ implementation of multi-class/multi-label/multi-task boosting algorithms. It is based on ADABOOST.MH but it also implements popular cascade classifiers, ARC-GV, and FILTERBOOST. The package contains common multi-class base learners (stumps, trees, products, Haar filters). Further base learners and strong learners following the boosting paradigm can be easily implemented in a flexible framework.

Link: <http://multiboost.org>

5.6. Grid Observatory

Participants: Cécile Germain [correspondent], Julien Nauroy, Michèle Sebag.

Autonomic Computing, Green Computing 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 140 users are currently registered. The GO is supported by an Inria ADT (Action de Développement Technologique).

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. A tool has been made available to provide interactive visualization of the site activity (load, power usage and more) at the global and individual level. The GCO is supported by the CNRS PEPS program, and by University Paris-Sud through the MRM (Moyens de Recherche Mutualisés) program. The first barrier to improved energy efficiency is the the lack of overall data collection on the energy consumption of individual components of data centers. The Green Computing Observatory collects monitoring data on energy consumption of a large computing center, and publishes them through the Grid Observatory portal. These data include the detailed monitoring of the processors and motherboards, as well as global site information, such as overall consumption and external temperature, as global optimization is a promising way of research. A second barrier is making the collected data usable. The difficulty is to make the data readily consistent and complete, as well as understandable for further exploitation. For this purpose, the GCO opts for an ontological approach in order to rigorously define the semantics of the data (what is measured) and the context of their production (how are they acquired and/or calculated). The first achievements of these developments have been described in [40], and presented at the ICT-COST meeting and GreenDays@Lyon.

Link: <http://grid-observatory.org>

TOCCATA Team

5. Software

5.1. The CiME rewrite toolbox

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

Keywords: Equational reasoning, Rewriting, Termination, Confluence, Completion

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/~lucas/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], François Bobot, Jean-Christophe Filliâtre, Guillaume Melquiond, Andrei Paskevich.

Keywords: Deductive verification, Java programming language, Java modeling language, Java Card, ANSI C programming language.

Criteria for Software Self-Assessment ⁵: A-3, SO-4, SM-3, EM-2, SDL-5-down, OC-4.

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 [86], at Universidade do Minho in Portugal [49], at Moscow State University, Russia (<http://journal.ub.tu-berlin.de/eceasst/article/view/255>).

5.3. The Why3 system

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

Keywords: Deductive verification

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

⁵Self-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 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/>.

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 used for teaching (Master Parisien de Recherche en Informatique).

Why3 is used by other academic research groups, e.g. within the CertiCrypt/EasyCrypt project (<http://easycrypt.gforge.inria.fr/>) for certifying cryptographic programs.

5.4. The Alt-Ergo theorem prover

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

Keywords: Automated theorem proving, Combination of decision procedures, Satisfiability modulo theories
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 [103] 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 Ganas, 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. The Cubicle model checker modulo theories

Participants: Sylvain Conchon [contact], Alain Mebsout.

Partners: A. Goel, S. Krstić (Intel Strategic Cad Labs in Hillsboro, OR, USA), F. Zaïdi (LRI, Université Paris-sud)

Keywords: Satisfiability modulo theories, model checking, array-based systems

Cubicle is an open source model checker for verifying safety properties of array-based systems. This is a syntactically restricted class of parametrized transition systems with states represented as arrays indexed by an arbitrary number of processes. Cache coherence protocols and mutual exclusion algorithms are typical examples of such systems.

Cubicle model-checks by a symbolic backward reachability analysis on infinite sets of states represented by specific simple formulas, called cubes. Cubicle is based on ideas introduced by MCMT (<http://users.mat.unimi.it/users/ghilardi/mcmt/>) from which, in addition to revealing the implementation details, it differs in a more friendly input language and a concurrent architecture. Cubicle is written in OCaml. Its SMT solver is a tightly integrated, lightweight and enhanced version of *Alt-Ergo*; and its parallel implementation relies on the Functor library.

5.6. Bibtex2html

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

Keywords: Bibliography, Bibtex format, HTML, World Wide Web.

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

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.

5.7. OCamlgraph

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

Keywords: Graph, Library, *OCaml*.

OCamlgraph is a graph library for *OCaml*. 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.

5.8. Mlpost

Participants: Jean-Christophe Filliâtre [contact], François Bobot.

Keywords: Library, *OCaml*.

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 *OCaml*. 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].

5.9. Functory

Participant: Jean-Christophe Filliâtre [contact].

Keywords: Library, *OCaml*.

Functory is a distributed computing library for *OCaml*. 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 [84] and at TFP 2011 [83].

5.10. The Pff library

Participant: Sylvie Boldo [contact].

Keywords: Interactive theorem proving, floating-point arithmetic. Criteria for Software Self-Assessment: A-2, SO-3, SM-3, EM-3, SDL-5, OC-4.

The Pff library for the *Coq* proof assistant is a formalization of floating-point arithmetic with high-level definitions and high-level properties [58].

It is distributed as open source, under a LGPL license, at <http://lipforge.ens-lyon.fr/www/pff/>, and is packaged in Debian and Ubuntu as “coq-float”.

It was initiated by M. Dumas, L. Rideau and L. Théry in 2001, and then developed and maintained by S. Boldo since 2004. It is now only maintained by S. Boldo. The development has ended as this library is now subsumed by the Flocq library (see below).

5.11. The Flocq library

Participants: Sylvie Boldo [contact], Guillaume Melquiond.

Keywords: Interactive theorem proving, floating-point arithmetic.

Criteria for Software Self-Assessment: A-2, SO-3, SM-3, EM-3, SDL-4, OC-4.

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 [5]. 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.

5.12. The Gappa tool

Participant: Guillaume Melquiond [contact].

Keywords: Automated theorem proving, floating-point arithmetic, fixed-point arithmetic.

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

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 [79] [109].

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.

5.13. The Interval package for Coq

Participant: Guillaume Melquiond [contact].

Keywords: Interactive theorem proving, interval arithmetic, floating-point arithmetic.

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

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.

5.14. The Alea library for randomized algorithms

Participants: Christine Paulin-Mohring [contact], Pierre Courtieu.

Keywords: Interactive theorem proving, randomized algorithms, probability

Criteria for Software Self-Assessment: A-2, SO-3, SM-2, EM-3, SDL-4, OC-4.

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 [54]. It is also experimented in LABRI as a basis to study formal proofs of probabilistic distributed algorithms.

5.15. The Coccinelle library for term rewriting

Participant: Évelyne Contejean [contact].

Keywords: Interactive theorem proving, Coq, rewriting, termination certificate 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 a 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.

5.16. The Coquelicot library for real analysis

Participants: Sylvie Boldo [contact], Catherine Lelay, Guillaume Melquiond.

Keywords: Interactive theorem proving, real analysis

Criteria for Software Self-Assessment: A-2, SO-4, SM-2, EM-3, SDL-1, OC-4.

Coquelicot is a *Coq* library dedicated to real analysis: differentiation, integration, and so on. It is a conservative extension of the standard library of *Coq*, but with a strong focus on usability.

Coquelicot is available at <http://coquelicot.saclay.inria.fr/>.

5.17. CFML

Participant: Arthur Charguéraud [contact].

Keywords: Program verification, Interactive theorem proving, *OCaml*

Criteria for Software Self-Assessment: A-2, SO-4, SM-2, EM-3, SDL-1, OC-4. The *CFML* tool supports the verification of *OCaml* programs through interactive *Coq* proofs. The tool is made of two parts: on the one hand, a characteristic formula generator implemented as a *OCaml* program that parses *OCaml* code and produces Coq formulae; and, on the other hand, a Coq library that provides notation and tactics for manipulating characteristic formulae interactively in Coq.

CFML is distributed under the LGPL license, and is available at <http://arthur.chargueraud.org/softs/cfml/>. The tool has been initially developed by A. Charguéraud in 2010, and has been maintained and improved since by the author.

TYPICAL Project-Team

4. Software

4.1. Coq

Participants: Bruno Barras [Contact], Jean-Marc Notin, 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 πr^2 Inria Paris-Rocquencourt project-team, and some members of the TypiCal team are active developers of the system.

4.2. Coqfinitgroup

Participants: Cyril Cohen, Assia Mahboubi [Contact], Enrico Tassi.

Coqfinitgroup is the development corresponding to the full formalization of 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 have been actively participating to this long term formalization project.

4.3. 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. Enrico Tassi has worked on an extended language of patterns for term selection now included in the distribution of this extension. Members of the Typical are in charge of the documentation and distribution of this extension.