Activity Report 2013

Section Software

AMIB Project-Team

4. Software and Platforms

4.1. VARNA

Participants: Yann Ponty [correspondant], Alain Denise.

A lightweight Java Applet dedicated to the quick drawing of an RNA secondary structure. VARNA is open-source and distributed under the terms of the GNU GPL license. Automatically scales up and down to make the most out of a limited space. Can draw multiple structures simultaneously. Accepts a wide range of documented and illustrated options, and offers editing interactions. Exports the final diagrams in various file formats (svg,eps,jpeg,png,xfig) [55]...
VARNA currently ships in its 3.9 version, and consists in ~50 000 lines of code in ~250 classes.
Impact: Downloaded ~10 000 times and is cited by more than ~170 research manuscripts (source: Google Scholar).
Availability: Distributed under the terms of the GPL v3 licence since 2009 on simple demand to the author(s) at http://varna.lri.fr.

4.2. 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 sequence and the canonical Watson–Crick pairings. 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 [63]. More than 300 visits since its release in January 2012.

4.3. Rna3Dmotif

Participant: Alain Denise [correspondant].

Rna3Dmotif is a free bundle of three easy-to-install programs aimed to be used in combination to automatically extract recurrent RNA local tertiary motifs. The approach used is based on a graph representation of the RNA tertiary structure using LW nomenclature. It was applied to several widely studied ribosomal RNA structures and the motifs thus found were deposited in a dedicated repository.
Impact: Cited in 17 research manuscripts (source: Google Scholar).
Availability: Distributed under the terms of the licence since 24/03/2009 on simple demand to the author(s) at http://rna3dmotif.lri.fr.

4.4. GenRGenS

Participants: Yann Ponty [correspondant], Alain Denise.

A software dedicated to the random generation of sequences. Supports different lasses of models, including weighted context-free grammars, Markov models, PROSITE patterns... [72] GENRGENS currently ships in its 2.0 version, and consists in ~25 000 lines of code in ~120 Java classes.
Impact: Downloaded ~5 000 times and is cited by more than ~50 research manuscripts (source: Google Scholar).
Availability: Distributed under the terms of the GPL v3 licence since 2006 on simple demand to the author(s) at https://www.lri.fr/ genrgens/.

4.5. DiMoVo

Participant: Julie Bernauer [correspondant].

DIMOVO, *DIscriminate between Multimers and MOnomers by VOronoi tessellation* : Knowing the oligomeric state of a protein is necessary to understand its function. his tool, accessible as a webserver and still used and maintained, provides a reliable discrimination function to obtain the most favorable state of proteins.
Availability : released in 2008.

4.6. VorScore

Participant: Julie Bernauer [correspondant].

VORSCORE, *Voronoi Scoring Function Server* : Scoring is a crucial part of a protein-protein procedure and having a quantitave function to evaluate conformations is mandatory. This server provides access to a geometric knowledge-based evaluation function. It is still maintained and widely used. See Bernauer et al., Bioinformatics, 2007 23(5):555-562 for further details.

4.7. GeneValorization

Participants: Bryan Brancotte, Sarah Cohen-Boulakia [correspondant].

High-throughput technologies provide fundamental informations concerning thousands of genes. Most of the current biological research laboratories daily use one or more of these technologies and identify lists of genes. Understanding the results obtained includes accessing to the latest publications concerning individual or multiple genes. Faced to the exponential growth of publications avaliable, this task is becoming particularly difficult to achieve.
Here, we introduce a web-based Java application tool named GeneValorization which aims at making the most of the text-mining effort done downstream to all high throughput technology assays. Regular users come from the Curie Institute, but also the EBI.
Impact : 925 distinct international users have used GeneValorization and about a hundred use it on a regular basis. The tool is on average used once to twice every day.
Availability : it is available at http://bioguide-project.net/gv with Inter Deposit Digital Number (*depot APP*, June 2013).

4.8. SPFlow

Participant: Sarah Cohen-Boulakia [correspondant].

Scientific workflow systems are numerous and equipped of provenance modules able to collect data produced and consumed during workflow runs to enhance reproducibility. An increasing number of approaches have been developed to help managing provenance information. Some of them are able to process data in a polynomial time but they require workflows to have series-parallel (SP) structures. Rewriting any workflow into an SP workflow is thus particularly important.
SPFLOW answers this need and takes in a workflow (from the Taverna system) and provide a runnable and provenance equivalent (Taverna) workflow."
Impact: The tool is currently used by Taverna's users from the University of Manchester and more generally by myExperiment users.
Availability: Distributed under the terms of the licence since 04/02/2013 on simple demand to the author(s) at http://www.lri.fr/ chenj/SPFlow/.

4.9. SPChecker

Participant: Sarah Cohen-Boulakia [correspondant].

Scientific workflow systems are numerous and equipped of provenance modules able to collect data produced and consumed during workflow runs to enhance reproducibility. An increasing number of approaches have been developed to help managing provenance information. Some of them are able to process data in a polynomial time but they require workflows to have series-parallel (SP) structures.

SPChecker is able to detect whether or not any Taverna workflow has a series-parallel structure.

Impact: The tool is currently used by Taverna's users from the University of Manchester and more generally by myExperiment users (a collaboration with Manchester has started and should significantly augment the number of potential users).

Availability: Distributed under the terms of the licence since 01/02/2013 on simple demand to the author(s) at http://www.lri.fr/ chenj/SPChecker/.

4.10. BioGuide

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

BioGuide/BioGuideSRS : this software helps the scientists choose suitable sources and tools, find complementary information in sources, and deal with divergent data.

Reference : Sarah Cohen-Boulakia, Olivier Biton, Susan Davidson, Christine Froidevaux, BioGuideSRS: Querying Multiple Sources with a user-centric perspective, Bioinformatics, March, 23(10), 1301-1303, 2007.

Impact: The paper related to the tool has been cited by ∼26 research manuscripts (source: Google Scholar) so far. Since 2007 and up to now, BioGuide has 8,030 distinct users including regular users from the EBI (European Bioinformatics Institute), the Institut Curie and the Children's Hospital of Philadelphia.

Availability: Distributed under the terms of the licence since 01/09/2006 on simple demand to the author(s) at http://bioguide-project.net/.

4.11. HSIM

Participant: Patrick Amar [correspondant].

HSIM (Hyperstructure Simulator) 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 that 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.

AVIZ Project-Team

5. Software and Platforms

5.1. Graph Cuisine

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

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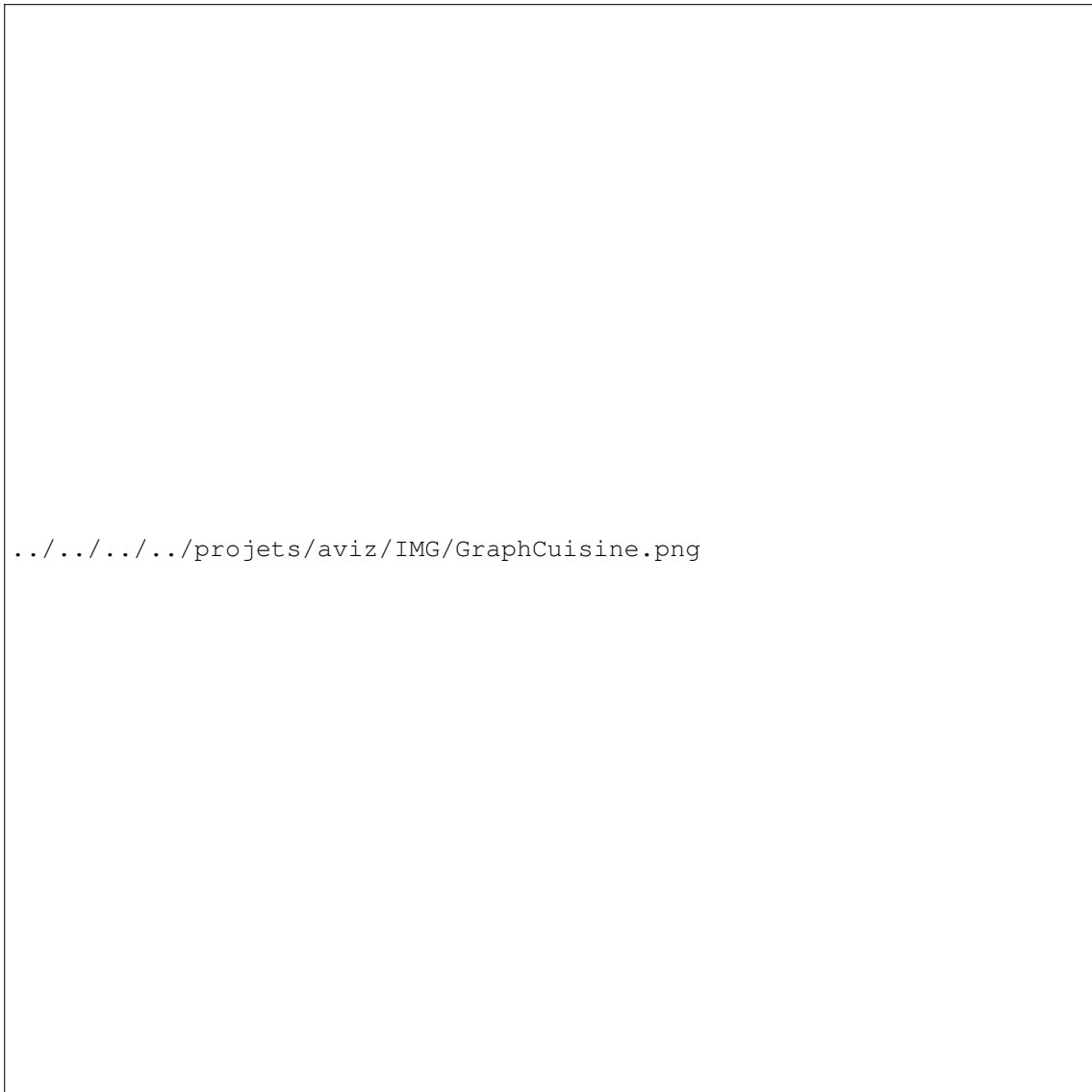
*Figure 1. GraphCuisine interface showing one real graph (right), the measures extracted from it (bottom circles),
several graphs with similar measures (left) and one of them selected (middle).*

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.

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. http://www.aviz.fr/histomages/.

5.3. Gliimpse

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

Gliimpse is a quick preview technique that smoothly transitions between document markup code (HTML, LaTeX,...) and its visual rendering. This technique allows users to regularly check the code they are editing in-place, without leaving the text editor. This method can complement classical preview windows by offering rapid overviews of code-to-document mappings and leaving more screen real-estate. A proof-of-concept editor can be downloaded for free at http://www.aviz.fr/gliimpse/.

5.4. The Obvious Toolkit

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

Information Visualization, Java, Toolkit

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) [61]. 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.

../../../../projets/aviz/IMG/hiistograms-teaser.png
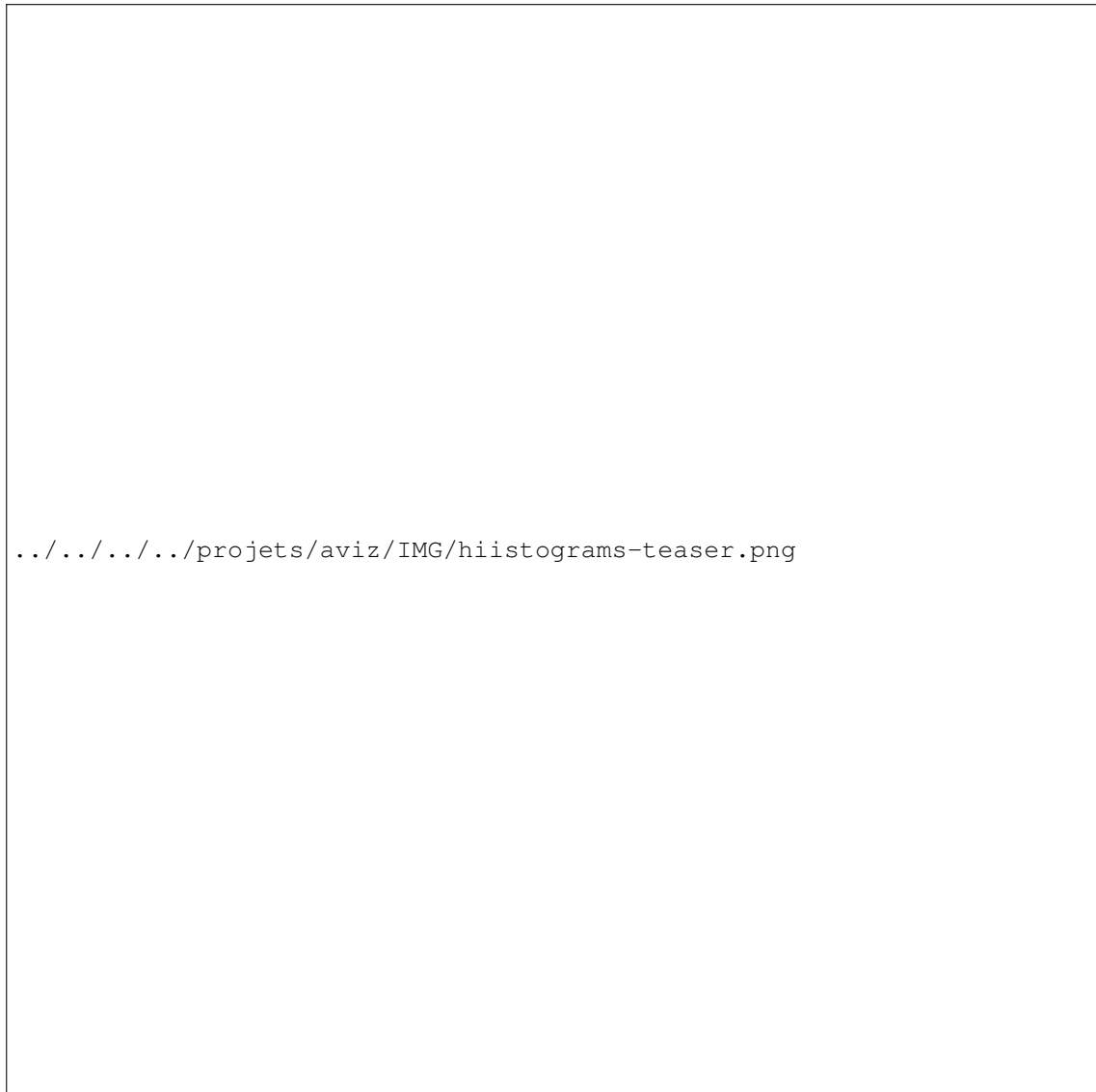
Figure 2. Example of sky enhancement with Histomages: (a) the image is duplicated and its pixels rearranged into a lightness histogram ; (b) bright pixels are selected with the rubber-band selection tool; (c) all pixels are rearranged into a hue histogram and yellow pixels are filtered out with the subtract 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.

```
../../../../projets/aviz/IMG/gliimpse.png
```

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

../../../../projets/aviz/IMG/obvious.png

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

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

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

```
../../../../projets/aviz/IMG/geneaquilts.png
```

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

../../../../projets/aviz/IMG/diffamation.png

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

The InfoVis Toolkit [60] 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 Elmqvist, Anastasia Bezerianos.

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) [59]: 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.

../../../../projets/aviz/IMG/graphdice.png

Figure 7. Screenshot the GraphDice system.

<p style="text-align:center;color:red;font-weight:bold;">COMETE Project-Team</p>

5. Software and Platforms

5.1. Location Guard

Participants: Konstantinos Chatzikokolakis [correspondant], Marco Stronati.

The purpose of *Location Guard* is to implement obfuscation techniques for achieving location privacy, in a an easy and intuitive way that makes them available to the general public. Various modern applications, running either on smartphones or on the web, allow third parties to obtain the user's location. A smartphone application can obtain this information from the operating system using a system call, while web application obtain it from the browser using a javascript call.

Although both mobile operating systems and browsers require the user's permission to disclose location information, the user faces an "all-or-nothing" choice: either disclose his exact location and give up his privacy, or stop using the application. This forces many users to disclose their location, although ideally they would like to enjoy some privacy.

The API level of a browser or an operating system would be an ideal place for integrating a location obfuscation technique, in a way that is easy to understand for the average user, and readily available to all applications. When an application asks for the user's location, the browser or operating system can ask the user's permission, but including the option to provide an obfuscated location instead of the real one! Different levels of obfuscation can be also offered, so that the user can chose to provide more accurate location to applications that really need it, and more noisy location to those that don't.

A prototype of Location Guard has been already implemented for Google Chrome. In the future we plan to extend it to other desktop and mobile browsers (Firefox, Internet Explorer, etc), as well as to implement it in modern mobile operating systems, primarily on Android.

https://github.com/chatziko/location-guard

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 [29] and in the file README file width 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 [30] and in the file README file width 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

Participant: Catuscia Palamidessi [correspondant].

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 [32] and [31].

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

<div align="center">

COMMANDS Project-Team

</div>

4. Software and Platforms

4.1. Bocop

Participants: Pierre Martinon [corresponding author], Daphné Giorgi, Joseph 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, features a user-friendly interface and can be deployed on Windows / Mac / Linux.

The project is supported by Inria with the recruitment of Vincent Grelard as developer in 2010-2012, and then Daphné Giorgi since October 2012. The first prototype was released at the end of 2011, Bocop is currently at version 1.1.4 and has been downloaded more than 700 times. The software was first succesfully tested on several academic problems, see [55] available on http://bocop.org. Starting in 2012, several research collaborations were initiated in fields such as bio-reactors for energy production ([30], [27]), swimming micro-robots ([39]), and quantum control for medical imaging ([25]). Bocop was also featured during our participation in the Imatch "Optimisation and Control" in october, which resulted in a contract with the startup Safety Line (aeronautics).

Bocop auto-assessment according to Inria notice: A3up4, SO3, SM3, EM3up4, SDL4up5

Figure 1. BOCOP

4.2. CollAv

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

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.

4.3. OCOPHyS

Participant: Hasnaa Zidani [corresponding author].

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.

4.4. BiNoPe-HJ

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

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.

4.5. Shoot

Participant: Pierre Martinon [corresponding author].

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. Shoot is an academic software, and was used during several research contracts with the CNES (french space agency).

DAHU Project-Team (section vide)

<p align="center" style="color:red">DEFI Project-Team</p>

5. Software and Platforms

5.1. RODIN

Participant: Grégoire Allaire [correspondant].

In the framework of the RODIN project we continue to develop with our software partner ESI the codes Topolev and Geolev for topology and geometry shape optimization of mechanical structures using 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. *Eddy current problems*

Participants: Zixian Jiang, Kamel Riahi.

We developed a FreeFem++ toolbox that solves direct and inverse problems for an axisymmetric and 3D eddy current problems related to non destructive testing of deposits on the shell side of PWR fuel tubes. For the 3-D version, one can refer to http://www.cmap.polytechnique.fr/~riahi

5.2.3. *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.

5.2.4. *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: Houssem Haddar [correspondant].

This Scilab toolbox is dedicated to the resolution of inverse 2-D electrostatic problems using the conformal mapping method introduced by Akdumann, Kress and Haddar. The toolbox treats the cases of a simply connected obstacle with Dirichlet, Neumann or impedance boundary conditions or a simply connected inclusion with a constant conductivity. The latest development includes the extension of the method to the inverse scattering problem at low frequencies as introduced by Haddar-Kress (2012).

5.4. Sampling methods for inverse problems

5.4.1. *Samplings-2d*

Participant: Houssem Haddar [correspondant].

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

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

5.4.2. *Samplings-3d*

Participant: Houssem Haddar [correspondant].

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

5.4.3. *Time domain samplings-2d*

Participant: Houssem Haddar [correspondant].

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 two numerical codes to solve the multiple-compartments Bloch-Torrey partial differential equation in 2D and 3D to simulate the water proton magnetization of a sample under the influence of diffusion-encoding magnetic field gradient pulses.

We coupled the spatial discretization with an efficient time discretization adapted to diffusive problems called the (explicit) Runge-Kutta-Chebyshev method.

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 complete (written by Dang Van Nguyen and Jing-Rebecca Li).

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

DISCO Project-Team

5. Software and Platforms

5.1. OreModules

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

The OREMODULES package [92], based on the commercial Maple package Ore_algebra [93], 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 [100], [102] 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) [109], [108], [91] 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/.

A Mathematica version of the OREMODULES package is in development. It is being developed by Maris Tõnso (Institute of Cybernetics, University of Tallinn), Thomas Cluzeau (ENSIL, University of Limoges) and A. Quadrat within the PHC Parrot project CASCAC. The Mathematica version of the OREMODULES package is based on the implementation of Gröbner bases over Ore algebras available in the Mathematica HolonomicFunctions package developed by Christoph Koutschan.

5.2. Stafford

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

The STAFFORD package of OREMODULES [92] contains an implementation of two constructive versions of Stafford's famous but difficult theorem [124] 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 [119] 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 [124] have recently been made constructive in [121] (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 [96] contains an implementation of the famous Quillen-Suslin theorem [123], [125]. 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 [96] (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 OREMORPHISMS package [95] of OREMODULES [91] is dedicated to the implementation of homological algebraic tools such as the computations of homomorphisms between two finitely presented modules over certain noncommutative polynomial algebras (Ore algebras), of kernel, coimage, image and cokernel of homomorphisms, Galois transformations of linear multidimensional systems and idempotents of endomorphism rings. Using the packages STAFFORD and QUILLEN-SUSLIN, the factorization, reduction and decomposition problems can be constructively studied for different classes of linear multidimensional systems. Many linear systems studied in engineering sciences, mathematical physics and control theory have been factorized, reduced and decomposed by means of the OREMORPHISMS package. The binary of the package is freely available at http://www-sop.inria.fr/members/Alban.Quadrat/OreMorphisms/index.html.

A Mathematica version of the OREMORPHIMS package is in development. It is being developed by Maris Tõnso (Institute of Cybernetics, University of Tallinn), Thomas Cluzeau (ENSIL, University of Limoges) and Alban Quadrat within the PHC Parrot project CASCAC. The Mathematica version of the OREMORPHIMS package is based on the implementation of Gröbner bases over Ore algebras available in the Mathematica HolonomicFunctions package developed by Christoph Koutschan.

5.5. JanetMorphisms

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

The JANETMORPHISMS package is dedicated to a new mathematic approach to quasilinear systems of partial differential equations (e.g., Burger's equation, shalow water equations, Euler equations of a compressible fluid) based on algebraic analysis and differential algebra techniques [90]. This package computes symmetries, first integrals of motion, conservation laws, study Riemann invariants... The JANETMORPHISMS package is based on the Janet package (http://wwwb.math.rwth-aachen.de/Janet/).

5.6. PurityFiltration

Participant: Alban Quadrat [correspondent].

The PURITYFILTRATION package, built upon the OREMODULES package, is an implementation of a new effective algorithm obtained in [30] which computes the purity/grade filtration [86], [87] 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 [30] for the computation of the purity/grade filtration [86], [87] in the powerful `homalg` package of GAP 4 dedicated to constructive homological algebra methods, and developed by Barakat (University of Kaiserslautern) and his collaborators (http://homalg.math.rwth-aachen.de/). This package both supersedes the existing PURITYFILTRATION package which uses the non-efficient Maple Gröbner basis computation (see Section 5.6), and the original `homalg` procedure which computes purity filtration by means of time-consuming spectral sequences. Using the `homalg` package philosophy, the ABELIANSYSTEMS package can be used for the computation of the purity filtration of objects in different constructive abelian categories such as coherent sheaves over projective schemes as demonstrated in the `homag` package called Sheaves (see http://homalg.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 `homalg` based package dedicated to mathematical systems. This package, still in development, will include the algorithms developed in the OREMODULES and OREMORPHISMS packages. It currently contains an implementation of the OREMORPHISMS procedures which handle the decomposition problem aiming at decomposing a module/system into direct sums of submodules/subsystems, and Serre's reduction problem aiming at finding an equivalent system defined by fewer unknowns and fewer equations.

5.9. YALTA

Participants: David Avanessoff, Catherine Bonnet [correspondent], Hugo Cavalera, André R. Fioravanti [UNICAMP], Jim Pioche.

The YALTA toolbox 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, [88] and [97]) and on classical continuation methods exploiting the particularities of the problem [98], [99]. For classical delay systems, a Pade2 approximation scheme is available as well as a finite-dimensional approximation of the system.

Binaries are freely available at http://yalta-toolbox.gforge.inria.fr/.

GALEN Project-Team

5. Software and Platforms

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:// cvn.ecp.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://vision.mas.ecp.fr/Personnel/teboul/grapesPage/index.php). 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://cvn.ecp.fr/personnel/tsogkas/code.html implements the learning-based approach to symmetry detection. 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://cvn.ecp.fr/personnel/andreas/code/sparse_k/sparse_k.tar) using regularization with the k-support norm, which we have introduced [39]. The algorithm uses an accelerated first-order method similar to Nesterov's method.

GECO Project-Team

5. Software and Platforms

5.1. IRHD

We develop 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 it does not require any information about the location and character of the corrupted places. 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.

GEOMETRICA Project-Team

5. Software and Platforms

5.1. CGAL, the Computational Geometry Algorithms Library

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

With the collaboration of Pierre Alliez, 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, Clément Jamin, Menelaos Karavelas, Sébastien Loriot, 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).

The transfer and diffusion of CGAL in industry is achieved through the company GEOMETRY FACTORY (http://www.geometryfactory.com). GEOMETRY FACTORY is a *Born of Inria* company, founded by Andreas Fabri in January 2003. The goal of this company is to pursue the development of the library and to offer services in connection with CGAL (maintenance, support, teaching, advice). GEOMETRY FACTORY is a link between the researchers from the computational geometry community and the industrial users.

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 number 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 mesh generation and related packages. Two 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 12 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 Project-Team

5. Software and Platforms

5.1. ECPP

François Morain has been continually improving his primality proving algorithm, ECPP, originally developed in the early 1990s. Proving the primality of a 512-bit number requires less than a second on an average PC. Morain's personal record is around 25000 decimal digits, using the FASTECPP variant that he started developing in 2003. The code is written in C, and based on publicly available packages (GMP, MPFR, MPC, MPFRCX).

5.2. SEA

Together with E. Schost and L. De Feo, François Morain has developed a new implementation of the SEA algorithm for computing the cardinality of elliptic curves over large prime and binary finite fields. This program is a `gforge` project, based on the NTL library. The large prime case is relevant to cryptographic needs; the binary case, while not useful in contemporary cryptography, is a good testbed for De Feo's FAAST package.

5.3. TIFA

TIFA (Tools for Integer FActorization), initially developed in 2006, has been continuously improved during the last few years. TIFA includes a base library written in C99 using the GMP library, stand-alone factorization programs, and a basic benchmarking framework. Available online at http://www.lix.polytechnique.fr/Labo/Jerome.Milan/tifa/tifa.xhtml, TIFA is distributed under the Lesser General Public License (version 2.1 or later).

5.4. Quintix

Guillaume Quintin's Quintix library implements efficient arithmetic in Galois rings and their unramified extensions, the root-finding algorithms presented in [7], basic functions for the manipulation of Reed–Solomon codes, and the complete Sudan list-decoding algorithm. Part of the Mathemagix computer algebra system (http://www.mathemagix.org/), the source code is distributed under the General Public License (version 2 or higher).

5.5. finitefieldz

Within the Mathemagix CAS (http://www.mathemagix.org/), Guillaume Quintin wrote the finitefieldz package, which provides arithmetic for finite fields and towers of finite fields, as well as univariate polynomial root finding and factorization over finite fields.

5.6. 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 and benchmarking of general decoding algorithms. As of 2012/12/13, DECODING provides a working list decoding algorithm, but there is no unique decoding algorithm (though this can be emulated by list-decoding up to half the minimum distance). The library is being still under development, and more algorithms will be added. DECODING was presented at the 2012 International Symposium on Symbolic and Algebraic Computation.

GRAND-LARGE Project-Team

4. Software and Platforms

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 of on a Network of Workstations deployed between Lille, Versailles and Tsukuba in Japan. Demos was proposed on these testbeds from conferences in USA. We recently finished evaluations of the overhead generated using YML, without smart schedulers and with extrapolations due to the lack of smart scheduling strategies inside targeted middleware.

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

The software is available at http://yml.prism.uvsq.fr/

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 [74].

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 then 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 mesh1d results to every computing node and launching the distributed execution of the patches.

4.7. 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 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.

4.8. 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.

---

[1] project page: http://jcae.sourceforge.net

4.9. 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 [16]. 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 [2].

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

4.10. 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 [8].

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.

IN-SITU Project-Team

5. Software and Platforms

5.1. WILD Platform

Participants: Michel Beaudouin-Lafon [correspondant], Olivier Chapuis, Stéphane Huot, Romain Primet, Amani Kooli, Monireh Sanaei, Gabriel Tezier, Jonathan Thorpe.

WILD (Wall-Size Interaction with Large Datasets) is InSitu's experimental ultra-high-resolution interactive platform for studying collaborative interaction and the visualization of very large datasets [2] (Figure 1 . It features a wall-sized display with thirty-two 30" LCD screens, i.e. a 5m50 x 1m80 (18' x 6') wall displaying 20 480 x 6 400 = 131 million pixels, powered by a 16-computer cluster and two front-end computer. The platform also features a camera-based motion tracking system supporting interaction with the wall as well as within the surrounding space, a multitouch table and various mobile devices. WILD provides a unique experimental environment for interactive visualization and is part of the DIGISCOPE Equipment of Excellence. In addition to using WILD for our research, we have also developed software architectures and toolkits that enable developers to run applications on such multi-device, cluster-based systems.

Figure 1. The WILD platform (clockwise): wall display, motion tracking system, display cluster, interactive table.

5.2. jBricks

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

jBricks (Figure 2) is a Java toolkit that integrates a high-quality 2D graphics rendering engine based on ZVTM [46] and a versatile input configuration module (based on ICon [42] 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.

- 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

../../../../projets/in-situ/IMG/jbricks.png

Figure 2. jBricks applications running on the WILD platform (32 tiles for a total resolution of 20 480 × 6 400 pixels). (a) Zoomed-in visualization of the North-American part of the world-wide air traffic network (1 200 airports, 5 700 connections) overlaid on NASA's Blue Marble Next Generation images (86 400 × 43 200 pixels) augmented with country borders ESRI shapefiles. (b) Panning and zooming in Spitzer's Infrared Milky Way (396 032 × 12 000 pixels). (c) Controlled laboratory experiment for the evaluation of mid-air multi-scale navigation techniques.

5.3. The SwingStates Toolkit

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

SwingStates [36] 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).

../../../../projets/in-situ/IMG/PieMenu.png

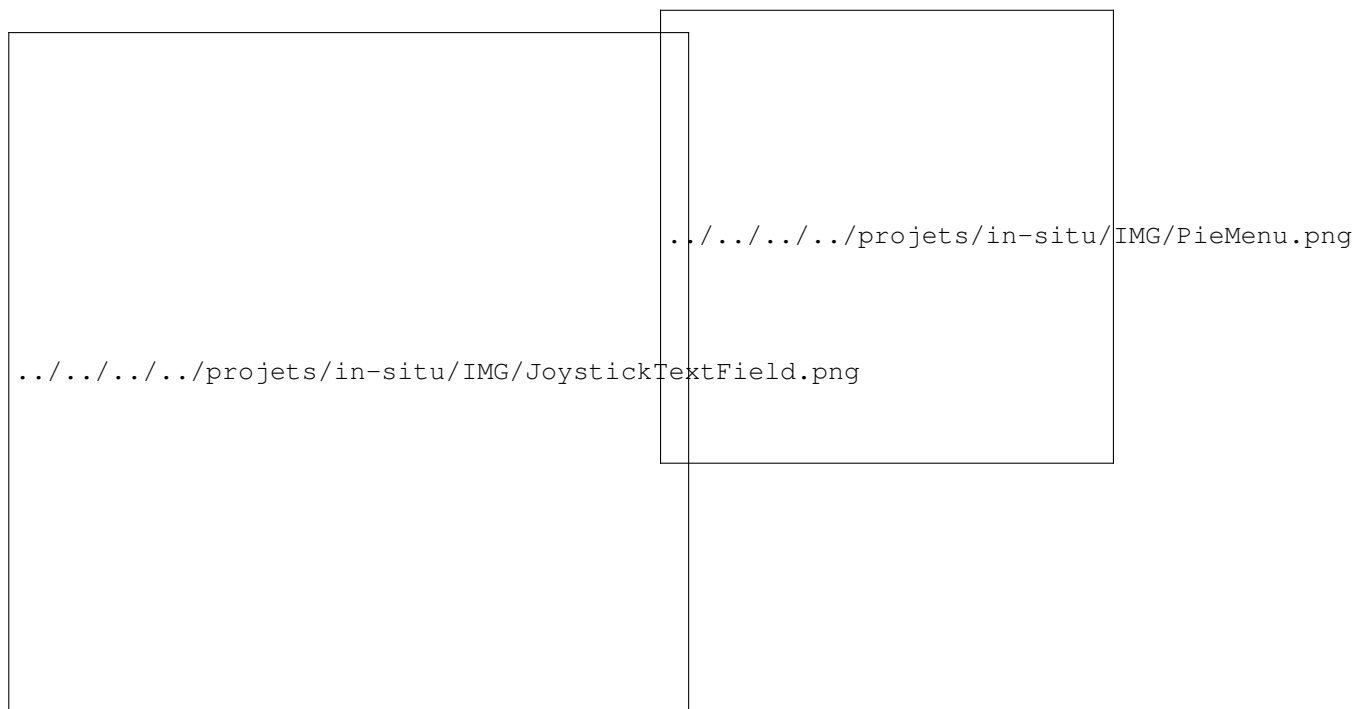../../../../projets/in-situ/IMG/JoystickTextField.png

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 [37], 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 [36] and ICon [42].

../../../../projets/in-situ/IMG/flowstates.png

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 [co-correspondant], Michel Beaudouin-Lafon, Wendy Mackay [co-correspondant].

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.

Figure 5. The architecture of the Touchstone platform

The *Touchstone Design platform* allows an experimenter to specify the factors, levels and measures in a controlled experiment, supports blocking and counterbalancing of trials and calculates how long it will take to run the experiment. Experimenters can compare the trade-offs between different experiment designs. The platform produces an XML file that serves as a protocol for the experiment and can be used as input to the Run platform.

The *Touchstone Run platform* provides a framework for implementing and running an experiment and collecting performance data. The flexible plug-in architecture supports various input devices and interaction techniques. The XML script from the Design platform can be run directly or edited to accommodate needs from specific experiments.

Log data from the Run platform can be analyzed by standard statistics tools such as JMP, R or Excel. In future, we hope to create a more elaborate *Touchstone Analysis platform* that will generate analysis scripts based on the output of the Design platform.

Members of InSitu use Touchstone for a variety of experiments and Students in the Research Masters (M2R Interaction) have been using it to design and implement experiments since 2011. 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 [40] 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 [48]. This year we used again Metisse to implement novel desktop interaction techniques [4].

- Web: http://insitu.lri.fr/metisse/
- ACM: H.5.2 [User Interfaces]: Windowing systems
- Software benefit: see [40], [48], [41], [44] and [4].
- License: GPL
- Type of human computer interaction: Graphique
- OS/Middleware: X Window et Mac OS X
- Required library or software: OpenGL via nucleo [1] and some usual C/C++ libraries
- Programming language: * C/C++

---

[1] http://interaction.lille.inria.fr/~roussel/projects/nucleo/index.html

../../../../projets/in-situ/IMG/metisse-folding.png ../../../../projets/in-situ/IMG/metisse-table-crop.png ../../../../projets/in-situ/IMG/met.

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).

5.7. The Substance Middleware

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

Substance is a middleware based on a novel programming paradigm called *data-oriented programming* and was designed to facilitate the development of multi-surface interactive applications [45]. 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 [47].

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 7): 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)
- Software benefit: See T. Gjerlufsen, C. Klokmose, 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
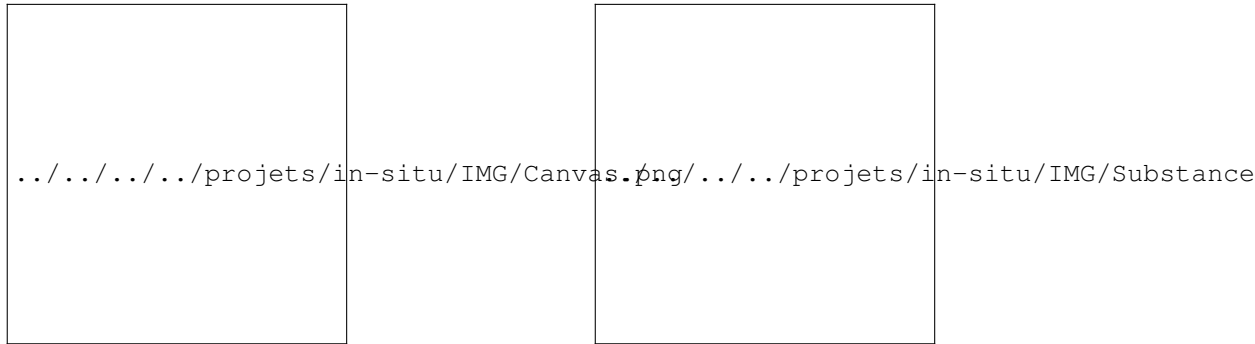- Programming language: Python

../../../../projets/in-situ/IMG/Canvas.png ../../../projets/in-situ/IMG/Substance

Figure 7. The Canvas (left) and SubstanceGrise (right) applications developed with Substance.
(©CNRS-Phototheque - Cyril FRESILLON for SubstanceGrise).

5.8. 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 [43]. 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 [38]. Scotty currently runs on Mac OS X for applications written with the Cocoa user interface framework.

../../../../projets/in-situ/IMG/iphone.png ../../../projets/in-situ/IMG/toolglass

Figure 8. 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).

Scotty has been used to create a number of extensions (Figure 8). *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.

- OS/Middleware: Mac OS X

- Required library or software: none

- Programming language: Objective-C, Python

M3DISIM Team

5. Software and Platforms

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. See 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 various collaborations.

The code relies on OpenFEM – to which the team has previously contributed, see http://www.openfem.net – 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 cardiac anatomical models compatible with the simulation requirements in terms of mesh quality, fiber direction data defined within each element, and the 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. 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 current version (1.5) 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 VPH-Share project and is now referenced in the peer-reviewed article [16].

- 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/Middelware: 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

<p align="center" style="color:red">Maxplus Project-Team</p>

5. Software and Platforms

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 [65], [135]. 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.

<p align="center">English version</p>

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 [65], [135]. 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 [84]. 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 [98]), 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 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.

<p align="center">English version</p>

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 [84]. Several internships allowed us to implement in Scilab, C or C++, and to test such algorithms (see the work of Vishesh Dhingra [98]), 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 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 [96]), 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 [63]).

TPLib est aujourd'hui utilisé dans le logiciel Polymake [116], développé à la Technische Universität Darmstadt (Allemagne). Ce dernier logiciel constitue une boite 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 Verify-TAPN (https://launchpad.net/verifytapn), qui permet la vérification de réseaux de Pétri avec arcs temporisés (voir §6.5.4). De même, une interface à la bibliothèque de domaines abstraits numériques APRON [128] 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 [96]), 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 [63]).

TPLib is now used in the software Polymake [116], 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 §6.5.4). An interface to the numerical abstract domain APRON [128] is also under development.

<div align="center">

MEXICO Project-Team

</div>

5. Software and Platforms

5.1. Software and Platform

5.1.1. Software

5.1.1.1. 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 facilely 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/

5.1.1.2. 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/. Mole served as an experimentation platform for several of our papers this year, notably [38] and [46].

In the context of MExICo, we have created a new tool called Cunf [47], which is able to handle contextual nets, i.e. Petri nets with read arcs [12].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; 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 5.1.2.1). Cunf has also participated in the Model Checking Contest held at the Petri Nets conference in 2013.

5.1.1.3. COSMOS : a Statistical Model Checker for the Hybrid Automata Stochastic Logic
Participant: 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/

5.1.2. Platforms

5.1.2.1. CosyVerif
Participants: Serge Haddad, Alban Linard [correspondant], Benoît Barbot.

CosyVerif (http://www.cosyverif.org/) is a platform dedicated to the formal specification and verification of dynamic systems. It allows to specify systems in a graphical editor, using several formalisms (such as automata and Petri nets) and to run verification tools on these models in a dedicated execution server. These tools are mainly developed by researchers of the MeFoSyLoMa group (a Parisian verification group, http://www.mefosyloma.fr/).

The platform is available as installable bundles, that contain both the client, the server, and the tools. It is also usable through two public servers: one with the latest release, one with the development version.

CosyVerif does not only handle several formalisms, but also allows to easily define new ones and integrate them within the platform. To the best of our knowledge, no other verification framework presents such a feature.

It has different kinds of users:

- Tool developers, that are usually researchers, can use the platform to distribute their tools, and have a demonstration version easily available.

- Students can use this platform in modeling and verification courses.

- Industrial case studies are also a target of the CosyVerif platform, in order to promote the practice of formal verification in industry.

The platform is managed by a steering committee consisting of researchers and engineers. This committee decides strategic orientations as well as technical choices.

This year, we have improved the platform in several ways.

- Tools: the platform handles two families of formalisms: automata and Petri nets, both with extensions. It currently integrates 10 tools with 4 new ones this year. Some of them perform structural analyses like invariant computations, while other tools perform behavioural analyses: symbolic reachability graph building, unfolding, stochastic simulations, etc.

- Server: the execution server has been enhanced with asynchronous executions, that allow to disconnect and reconnect the client in long executions. It has also been improved by the ability to communicate between servers to share their available tools.

- Client: a new command line client has been developed for scripting the executions.

- Usability: the client and server are new distributed as one bundle that can be installed easily on all platforms. The server and its tools are embedded within a virtual machine to achieve this portability.

All the developed software are open source and free software tools.

Two engineers have worked this year on CosyVerif:

- Francis Hulin-Hubard, part-time (CNRS engineer);
- Alban Linard, full-time (Inria engineer).

CosyVerif has been the subject of two international communications [28], [29]. It has been presented at the french-speaking PhD school ETR'2013 in Toulouse. and used for teaching in the master SAR of University Pierre et Marie Curie.

<p style="text-align:center;color:red;font-weight:bold;">OAK Project-Team</p>

5. Software and Platforms

5.1. Amada

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

Contact: Jesús Camacho-Rodríguez (jesus.camacho-rodriguez[at]inria.fr))

Other contacts: Ioana Manolescu (ioana.manolescu[at]inria.fr), Dario Colazzo (dario.colazzo[at]dauphine.fr), François Goasdoué (fg[at]irisa.fr)

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

5.2. FactMinder

Name: FactMinder (http://tripleo.saclay.inria.fr/xr/demo/)

Contact: Julien Leblay (julien.leblay[at]inria.fr)

Other contacts: Stamatis Zampetakis (stamatis.zampetakis[at]inria.fr), François Goasdoué (fg[at]irisa.fr), Ioana Manolescu (ioana.manolescu[at]inria.fr)

Presentation: A system for archiving, annotating, and querying semantic-rich Web content.

5.3. Nautilus Analyzer

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

Contact: Melanie Herschel (melanie.herschel[at]lri.fr)

Other contacts: n.a.

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

5.4. RDFViewS

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

Contact: Konstantinos Karanasos (kkaranasos[at]gmail.com)

Other contacts: François Goasdoué (fg[at]irisa.fr), Julien Leblay (julien.leblay[at]gmail.com), and Ioana Manolescu (ioana.manolescu[at]inria.fr)

Presentation: A storage tuning wizard for RDF applications.

5.5. ViP2P

Name: ViP2P (Views in Peer-to-Peer, http://vip2p.saclay.inria.fr)

Contact: Ioana Manolescu (ioana.manolescu[at]inria.fr)

Other contacts: Jesús Camacho-Rodríguez (jesus.camacho-rodriguez[at]inria.fr)

Presentation: A P2P platform for disseminating and querying XML and RDF data in large-scale distributed networks.

5.6. WARG

Name: WARG (https://team.inria.fr/oak/warg/)

Contact: Alexandra Roatiş (alexandra.roatis[at]lri.fr))

Other contacts: Ioana Manolescu (ioana.manolescu[at]inria.fr), Dario Colazzo (dario.colazzo[at]dauphine.fr), François Goasdoué (fg[at]irisa.fr)

Presentation: A platform for specifying and exploiting warehouses of RDF data.

5.7. XUpOp

Name: XUpOp (XML Update Optimization)

Contact: Dario Colazzo (dario.colazzo[at]dauphine.fr)

Other contacts: Nicole Bidoit (bidoit[at]lri.fr), Mohamed Amine Baazizi (baazizi[at]lri.fr)

Presentation: A general purpose type-based optimizer for XML updates.

5.8. XUpIn

Name: XUpIn (XML Update Independence)

Contact: Federico Ulliana (Federico.Ulliana[at]lri.fr)

Other contacts: Dario Colazzo (colazzo[at]lri.fr), Nicole Bidoit (bidoit[at]lri.fr)

Presentation: An XML query-update independence tester.

5.9. XUpTe

Name: XUpTe (XML Update for Temporal Documents)

Contact: Dario Colazzo (dario.colazzo[at]dauphine.fr)

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

Presentation: A type-based optimizer for representing and updating XML temporal data.

5.10. XPUQ

Name: XPUQ (XML Partitioning for Updates and Queries)

Contact: Dario Colazzo (dario.colazzo[at]dauphine.fr)

Other contacts: Nicole Bidoit (bidoit[at]lri.fr)

Presentation: A static analyzer and partitioner for XML queries and updates.

PARIETAL Project-Team

5. Software and Platforms

5.1. Scikit learn

Participants: Bertrand Thirion, Gaël Varoquaux, Olivier Grisel [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: 0.14
- Programming language: Python, C/Cython

5.2. Nilearn

Participants: Bertrand Thirion, Gaël Varoquaux [correspondant], Philippe Gervais, Jaques Grobler, Alexandre Gramfort, Fabian Pedregosa, Alexandre Abraham, Michael Eickenberg.

NiLearn is the neuroimaging library that adapts the concepts and tools of the scikit learn to neuroimaging problems. As a pure Python library, it depends on scikit learn and nibabel, the main Python library for neuroimaging I/O. It is an open-source project, available under BSD license. The two key components of NiLearn are *i)* the analysis of functional connectivity (spatial decompositions and covariance learning) and *ii)* the most common tools for multivariate pattern analysis. A great deal of efforts has been put on the efficiency of the procedures both in terms of memory cost and computation time. NiLearn is maintained both through the help of Inria: (a developer funded by Saclay CRI in 2012-2013, a 2013-2014 ADT, and through the NiConnect project (P. Gervais).

- Version: 0.1
- Programming language: Python

5.3. 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

5.4. 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

5.5. 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

5.6. PyHRF

Participants: Philippe Ciuciu [correspondant], Solveig Badillo, Aina Frau Pascual.

PyHRF is a set of tools for within-subject fMRI data analysis, focused on the characterization of the hemodynamics. Within the chain of fMRI data processing, these tools provide alternatives to the classical within-subject GLM estimation step. The inputs are preprocessed within-subject data and the outputs are statistical maps and/or fitted HRFs. The package is mainly written in Python and provides the implementation of the two following methods:

- The joint-detection estimation (JDE) approach, that divides the brain into functionally homogeneous regions and provides one HRF estimate per region as well as response levels specific to each voxel and each experimental condition. This method embeds a temporal regularization on the estimated HRFs and an adaptive spatial regularization on the response levels.

- The Regularized Finite Impulse Response (RFIR) approach, that provides HRF estimates for each voxel and experimental conditions. This method embeds a temporal regularization on the HRF shapes, but proceeds independently across voxels (no spatial model).

The development of PyHRF is now funded by an Inria ADT, in collaboration with MISTIS.

- Version: 0.1

- Keywords: Hemodynamic response function; estimation; detection; fMRI

- License: BSD 4

- Multiplatform: Windows - Linux - MacOSX

- Programming language: Python

<div align="center">

PARSIFAL Project-Team

</div>

5. Software and Platforms

5.1. Abella

Participants: Kaustuv Chaudhuri [correspondant], Matteo Cimini, Dale Miller, Olivier Savary-Bélanger, Yuting Wang.

Main web-site: http://abella-prover.org.

Abella is an interactive theorem prover based on the two-level 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 2013, as part of the RAPT Associated Team, Chaudhuri and Yuting Wang (former intern from Univ. Minnesota) released version 2.0 of Abella, a culmination of nearly two years of work and a significant improvement in its expressivity. Specifically,

> The Abella specification logic now supports the full higher-order hereditary Harrop logic of λProlog. This logic allows for very natural specifications of higher-order relations, and leads to cleaner and simpler proofs.

> The Abella reasoning logic was extended with support for arbitrary dynamic contexts and incremental backchaining. The design is based on fundamental insights from *focusing*, a core strength of the team.

> A number of illustrative examples of the use of higher-order reasoning were added to the Abella examples library, including a novel new characterization of marked β-reduction in the λ-calculus in terms of a simple higher-order inductive definition of λ-paths.

> These results were published in PPDP 2013 [26].

Abella continues to evolve as part of RAPT. In 2013, we hosted an intern from McGill University, Olivier Savary-Bélanger (supervised by Chaudhuri), who investigated extensions of Abella with *regular context schemas*. Among his contributions:

> Abella's reasoning level has been augmented with a *plugin* system that both extends the syntax of Abella theories and adds new tactics.

> The main plugin for context schemas allows definitions of regular contexts and context relations, with entirely automatic proofs of the main administrative lemmas.

> Experimentally, this extension can be used to eliminate up to 40% of the proof text, including nearly 100% of the administrative lemmas on contexts, from typical examples from the meta-theory of the λ-calculus.

We expect this extension to become part of the 2.1 release of Abella, scheduled for later in 2014.

One important application of Abella emerged in 2013: the formalization of bisimulation-up-to techniques for process calculi such as CCS and the π-calculus. Chaudhuri, Cimini, and Miller have formulated the correctness proof of a number of prominent up-to-techniques using the co-inductive and higher-order facilities of Abella. This work indicates an important emerging direction for Abella: modular reasoning.

In terms of development, we have welcomed Savary-Bélanger into the development team, and added a number of collaborators into the management team for the Abella web-site.

5.2. Bedwyr

Participants: Quentin Heath, Dale Miller [correspondant].

Main web-site: http://slimmer.gforge.inria.fr/bedwyr/.

During the first half of 2013, Quentin Heath was working as an engineer on the team, supported by the BATT ADJ project funded by Inria. During that time, we worked exclusively on making improvements to Bedwyr. In particular, he made extensive and important changes to the tabling mechanism of Bedwyr, a feature of model checking systems that is capable of remembering past successful proofs (it can even support a finite failure as a successful proof of a negation). These extension allow lemmas to be used to greatly extend the scope of what can be inferred from a table. For example, if we are attempting to show that there is a winning strategy for a given board position, we would certainly like to make use of a lemma that allows one to infer that winning strategies are preserved under symmetries of the board. There are a number of design issues that go along with the design of such a tabling mechanism: for example, should one use such lemmas in a forward-chaining or backward-chaining fashion. Quentin has tested both of these options in order to collect information as to what the trade-offs would be.

We should note that Quentin Heath is now a PhD student on the team and is addressing a number of theoretical questions related to his research on Bedwyr.

5.3. Profound

Participant: Kaustuv Chaudhuri [correspondant].

Profound is a new interactive theorem proving and proof-exploration tool based on the idea of building formal proofs *without* the use of formal proof languages. The core concepts are a generalization of *deep inference* for the underlying logical formalism, and *proof-by-pointing* for the user-interaction metaphors.

A user proves a theorem in *Profound* by using the keyboard and mouse to select subformulas of the theorem and dragging them to their suitable "destination". For instance, the formula $(A \to C) \to (A \land B \to C)$ is proved by dragging the two As and the two Cs to each other. This kind of *direct manipulation* is nevertheless constrained by the system to be both correct—meaning that no manipulation is logically unsound—and complete—meaning that every provable theorem can be proved using these metaphors.

The system is still in its early stages, but it currently supports first-order classical linear logic. It has been documented in a paper at ITP 2013 [18].

We are in the process of extending the system to intuitionistic logics, and adding a back-end exporter for more traditional proof systems with formal proof languages such as Coq and Isabelle.

5.4. Psyche

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

Psyche (*Proof-Search factorY for Collaborative HEuristics*) is a modular proof-search engine whose first version, 1.0, was released in 2012:
http://www.lix.polytechnique.fr/~lengrand/Psyche/

Its 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.

Psyche's proof-search mechanism is simply the incremental construction of proof-trees in the polarized and focused sequent calculus. Its architecture organizes an interaction between a trusted universal kernel and smart plugins that are meant 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.

In 2013, major developments were achieved in Psyche, which now handles classical propositional logic *modulo a theory* such as linear arithmetic, equality with uninterpreted symbols, arrays, etc. It therefore works in the same logic as Sat-Modulo-Theories (SMT) solvers and the architecture to handle such theories is the main contribution of 2013, in particular with the integration of the *simplex* algorithm.

Thanks to a plugin that simulates the behavior of a SAT-solver (DPLL) [21], the new version of Psyche can now simulate the behavior of SMT-solvers.

A lot of features inspired by SAT-solvers have now been lifted to proof-search in general, such as a *memoization table* to record and re-use known proofs, the technique of *2-watched literals* to efficiently propagate direct consequences of new hypotheses, machine learning techniques for *restart policies*, etc.

Psyche has been the topic of the 2013 publication [23].

<p style="text-align:center;color:red;font-weight:bold;">POEMS Project-Team</p>

5. Software and Platforms

5.1. Software

5.1.1. Introduction

We are led to develop two types of softwares. 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 own research and/or promote our research. We have chosen to present here our advanced software.

5.1.2. XLiFE++

Participants: Eric Lunéville, Nicolas Kielbasiewicz, Colin Chambeyron, Manh Ha Nguyen.

XLiFE++ is a new Finite Element library in C++ based on philosophy of the previous library MELINA in Fortran but with new capabilities (boundary elements and 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 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, University of Rennes and LAMA, University of Marne-la-Vallée.

In 2012, as a reminder, all development tools were set up and all fundamental and major libraries were done. In 2013, developments have sped up. The Finite Elements, the Spectral Elements and the Boundary Elements computation cores have been implemented and are currently under testing. In addition to the implementation of direct and iterative solvers, an internal eigen solver is operational and coupled to external solver libraries (Arpack++, Umfpack, ...).

As far as inputs/outputs are concerned, XLiFE++ allows to export a solution to the visualization tool PARAVIEW and to read mesh files from GMSH, MELINA and PARAVIEW (vtk). Furthermore, mesh tools have been enriched and a C++ interface to the mesh tool GMSH is under development. XLiFE++ can now solve the Helmholtz equation with Neumann boundary conditions in any mesh. A first version of the library should be published soon.

5. Software and Platforms

5.1. Monolix

Participants: Marc Lavielle, Hector Mesa, Célia Barthélémy.

MONOLIX is an easy, fast and powerful tool for parameter estimation in nonlinear 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 start-up Lixoft now develops and supports MONO-LIX. 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, e.g., 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, Fazia Bellal, Célia Barthélémy.

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 3 of the CTS is available since June 2013. 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 distribution 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

5.4. MLXplore

Participants: Marc Lavielle, Laura Brocco.

MLXplore is a graphical and interactive software for the exploration and visualization of complex pharmacometric models. MLXplore also includes the ability to study the statistical variability of the models, and to model and study complex administration designs.

MLXplore does not require MONOLIX, although they make for a powerful combination, enabling to use the same, human-readable model description, to finely explore the properties of the model on the one hand, and on the other hand use the same model for advanced parameter estimation in the context of population analysis and mixed effect statistics.

MLXplore is an ideal tool to learn about pharmacometric models and population analysis, and is used extensively in the online wiki WikiPopix created by POPIX, found at: https://wiki.inria.fr/popix.

MLXplore is developed by Lixoft. POPIX collaborates closely with Lixoft on on the definition of the specifications of MLXplore.

REGULARITY Project-Team

5. Software and Platforms

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 four 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 roughly three 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.

Last year, we produced a major release of FracLab (version 2.1). This year, we corrected a number of bugs.

SECSI Project-Team

5. Software and Platforms

5.1. Orchids

Participants: Jean Goubault-Larrecq [correspondant], Pierre-Arnaud Sentucq.

The ORCHIDS real-time intrusion detection system was created in 2003-04 at SECSI. Orchids is at the core of a contract between Inria and DGA, started in April 2013, for three years.

Progress in 2013 included:

- Creation of a collection of VirtualBox virtual machines with a pre-installed instance of Orchids, for easy testing and/or installation.

- A collection of scripts, allowing one to rebuild the above cited virtual machines automatically from the sources, as a nightly build (in progress).

- A new algorithm for evaluating the worst-case thread complexity of detection by Orchids, whose first principles were laid out by Jean Goubault-Larrecq, and with two prototype implementations done by Jean-Philippe Lachance, a young L2 intern from Université Laval, Québec. The purpose is to warn users of the complexity of the tasks they delegate to Orchids, and to avert denial of service attacks on Orchids itself.

Objectives for 2014 include:

- Simplifying the Orchids installation process, which has gotten complicated over the years.

- Implementing a frontend tool incorporating the full-fledged version of the worst-case thread complexity algorithm mentioned above, plus some other checks.

SELECT Project-Team

5. Software and Platforms

5.1. MIXMOD software

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

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 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.

Benjamin Auder contributes to the informatics improvement of MIXMOD. He implemented an interface to test any mathematical library (Armadillo, Eigen, ...) to replace NEWMAT. He contributed to the continuous integration setup using Jenkins tool and prepared an automated testing framework for unit and non-regression tests.

5.2. BLOCKCLUSTER software

Participants: Vincent Brault, Gilles Celeux, Christine Keribin.

Mixture model, Block cluster analysis,

Blockcluster is a software devoted on model-based block clustering. It is developed by MODAL team (Inria Lille). With Parmeet Bathia (Inria Lille), Vincent Brault has added a Bayesian point of view for the binary, categorial and continuous datas with the variational Bayes algorithm or Gibbs sampler. Criteria ICL and BIC are used for selecting a relevant block clustering.

Specfun Team

5. Software and Platforms

5.1. Mgfun

(1994–): Maple package for symbolic summation, integration, and other closure properties of multivariate special functions.

Now distributed as part of Algolib, a collection of packages for combinatorics and manipulations of special functions, available at http://algo.inria.fr/libraries/.

5.2. DDMF

(2007–): Web site consisting of interactive tables of mathematical formulas on elementary and special functions. The formulas are automatically generated by OCaml and computer-algebra routines. Users can ask for more terms of the expansions, more digits of the numerical values, proofs of some of the formulas, etc. See http://ddmf.msr-inria.inria.fr/.

5.3. DynaMoW

(2007–): Programming tool for controlling the generation of mathematical websites that embed dynamical mathematical contents generated by computer-algebra calculations. Written in OCaml. See http://ddmf.msr-inria.inria.fr/DynaMoW/.

5.4. Ring

(2004–): Coq normalization tool and decision procedure for expressions in commutative ring theories. Written in Coq and OCaml. Integrated in the standard distribution of the Coq proof assistant since 2005.

5.5. SSReflect

(2006–): Extension of the language of the Coq system. Originally written by G. Gonthier for his formal proof of the Four-Color Theorem. A. Mahboubi and E. Tassi participate to its development, maintenance, distribution, user support and have written its user manual. See http://www.msr-inria.fr/projects/mathematical-components/.

5.6. Coqfinitgroup

(2006–): Coq libraries that cover the mechanization of the proof of the Odd Order Theorem. Stable libraries are distributed with the SSReflect extension. A. Mahboubi is one of the main contributors to the code and its documentation. E. Tassi contributed to the design of core data structures and to parts of the formalization. A formal proof was completed in September 2012, and the content of the libraries, under continued improvements in view of potential reuse, is available online at http://www.msr-inria.fr/projects/mathematical-components/.

<p style="text-align:center; color:red;">**TAO Project-Team**</p>

5. Software and Platforms

5.1. METIS

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

Keywords: 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) models.

5.2. MoGo

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

Keywords:

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) and outpeforming professional players in 7x7. 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].

Keywords: Evolutionary Computation, Stochastic Optimization, Real-parameter Optimization.

The Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [72] is considered to be state-of-the-art in continuous domain evolutionary computation [69], and in stochastic optimization at large. It has been shown to be highly competitive on different problem classes even with deterministic continuous algorithms using numerically computed gradients (see the results published on COCO platform). 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.

Link: http://www.lri.fr/~hansen/cmaes_inmatlab.html

5.4. COmparing Continuous Optimizers

Participants: Nikolaus Hansen [correspondent], Anne Auger, Marc Schoenauer, Ouassim Ait Elhara, Asma Atamna.

Keywords: Evolutionary Computation, Stochastic Optimization, Real-parameter Optimization, Benchmarking, Derivative Free Optimization.

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, C, Java, and Python. The post-processing code is provided in Python. The code is under continuous developement and has been used for the GECCO 2009, 2010, 2012, and 2013 workshops on "Black Box Optimization Benchmarking" (BBOB) (see Section 6.1). It is now undergoing major changes thanks to the ANR project NumBBO that will add constraint handling and multi-objective benchmarks to the existing platform.

Link: http://coco.gforge.inria.fr/ and http://numbbo.gforge.inria.fr/

5.5. MultiBoost

Participants: Balázs Kégl [correspondent], Djalel Benbouzid.

Keywords: Multi-class, Multi-label Classification.

The MultiBoost package [68] 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-Renaud [correspondent], Julien Nauroy, Michèle Sebag.

Keywords: Autonomic Computing, Green Computing.

The Grid Observatory (GO) 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 since 2008. These data are stored on the grid, and made accessible through a web portal without the need of grid credentials. The GO is fully integrated with the evolution of EGI monitoring. More than 250 users are currently registered. The acquisition has been extended to the University cloud StratusLab hosted by the VirtualData center.

The Green Computing Observatory (GCO) monitors the VirtualData center; it collects data on energy consumption and publishes the data through the Grid Observatory. These data include the detailed monitoring of the processors and motherboards, as well as global site information. The first results on energy saving opportunities have been presented at the Green Days@Luxembourg meeting.

In order to make the GO data readily consistent and complete, as well as understandable for further exploitation, an original approach has been designed, based on a flexible data schema built in collaboration with the users. Its implementation is developed within the FUI project TIMCO. The GO has been supported by an Inria ADT (Action de Développement Technologique) up to September 2013, and by University Paris Sud through the MRM (Moyens de Recherche Mutualisés) program. Stabilization through the VirtualData initiative is currently explored.

Link: http://grid-observatory.org

TOCCATA Team

5. Software and Platforms

5.1. The CiME rewrite toolbox

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

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

5.2. The Why platform

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

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, École Polytechnique, etc.), used by several research groups in the world, e.g at Fraunhofer Institute in Berlin [93], at Universidade do Minho in Portugal [54], 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], Claude Marché, Guillaume Melquiond, Andrei Paskevich.

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

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/.

---

[5]self-evaluation following the guidelines (http://www.inria.fr/content/download/11783/409665/version/4/file/SoftwareCriteria-V2-CE.pdf) of the Software Working Group of Inria Evaluation Committee(http://www.inria.fr/institut/organisation/instances/commission-d-evaluation)

Why3 is used as back-end of our own tools *Krakatoa* and *Jessie*, but also as back-end of the GNATprove tool (Adacore company), and in a near future of the WP plugin of Frama-C. *Why3* has been used to develop and prove a significant part of the programs of our team gallery http://proval.lri.fr/gallery/index.en.html, and 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.

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 [113] defined by the SMT-lib initiative.

It is currently used in our team to prove correctness of C and Java programs as part of the *Why* platform and the new *Why3* system. *Alt-Ergo* is also called as an external prover by the Pangolin tool developed by Y. Regis Gianas, Inria project-team Gallium http://code.google.com/p/pangolin-programming-language/. Alt-Ergo is usable as a back-end prover in the SPARK verifier for ADA programs, since Oct 2010. It is planed to be integrated in next generation of Airbus development process.

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

5.5. 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)

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 Functory library.

5.6. Bibtex2html

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

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.

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

Participant: Jean-Christophe Filliâtre [contact].

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 [56].

5.9. Functory

Participant: Jean-Christophe Filliâtre [contact].

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 [92] and at TFP 2011 [91].

5.10. The Pff library

Participant: Sylvie Boldo [contact].

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 [64].

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. Daumas, 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.

Criteria for Software Self-Assessment: A-2, SO-3, SM-3, EM-3, SDL-5, 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 [6]. 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].

Criteria for Software Self-Assessment: A-3, SO-4, SM-4, EM-3, SDL-5, 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 [83] [119].

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].

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).

5.14. The Alea library for randomized algorithms

Participant: Christine Paulin-Mohring [contact].

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 used as a basis of the Certicrypt environment (MSR-Inria joint research center, Imdea Madrid, Inria Sophia-Antipolis) for formal proofs for computational cryptography [59]. It is also experimented in LABRI as a basis to study formal proofs of probabilistic distributed algorithms. ALEA version 8 distributed in May 2013 includes a module to reason with random variables with values in positive real numbers.

5.15. The Coccinelle library for term rewriting

Participant: Évelyne Contejean [contact].

Coccinelle is a *Coq* library for term rewriting. Besides the usual definitions and theorems of term algebras, term rewriting and term ordering, it also models some of the algorithms implemented in the CiME toolbox, such 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 developement (http://color.inria.fr/) as well as for certifying Spike implicit induction theorems in *Coq* (Sorin Stratulat).

Coccinelle is 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.

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

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 an *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.