Activity Report 2014

Section Software

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

5. New Software and Platforms

5.1. Location Guard

Participants: Konstantinos Chatzikokolakis [correspondant], Marco Stronati.

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

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 is 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.

Location Guard was created as a prototype for Google Chrome at the end for 2013. In 2014, Location Guard matured from a prototype to a high quality software, supporting both desktop and mobile browsers:

- Google Chrome / Chromium
- Mozilla Firefox and Firefox for Android
- Opera

After only a short period online, the extension has more than 8500 daily users, and it was presented in an article by the popular technology news site Ghacks. Our experience so far shows that end users do care about location privacy, and geo-indistinguishability is a practical approach for providing it.

In the future we plan to make Location Guard more widely available on smartphones, supporting more mobile browsers as well as providing direct integration into the operating system, primarily on Android.

5.2. libqif - A Quantitative Information Flow C++ Toolkit Library

Participants: Konstantinos Chatzikokolakis [correspondant], Marco Stronati.

https://github.com/chatziko/libqif

The goal of libqif is to provide an efficient C++ toolkit implementing a variety of techniques and algorithms from the area of quantitative information flow and differential privacy. We plan to implement all techniques produced by Comète in recent years, as well as several ones produced outside the group, giving the ability to privacy researchers to reproduce our results and compare different techniques in a uniform and efficient framework.

Some of these techniques were previously implemented in an ad-hoc fashion, in small, incompatible with each-other, non-maintained and usually inefficient tools, used only for the purposes of a single paper and then abandoned. We aim at reimplementing those – as well as adding several new ones not previously implemented – in a structured, efficient and maintainable manner, providing a tool of great value for future research. Of particular interest is the ability to easily re-run evaluations, experiments and case-studies from all our papers, which will be of great value for comparing new research results in the future.

The library is still in under heavy development but substantial progress has been made in 2014. Some of the techniques already implemented are:

- Standard leakage measures: Shannon, min-entropy, guessing entropy
- Measures from the g-leakage framework [32]
- Channel factorization
- Standard differential privacy mechanisms from the literature
- The planar Laplace mechanism of [33]
- The standard Kantorovich metric as well as the multiplicative variant from [19]
- All operations are supported for both doubles (for precision) and floats (for memory efficiency)
- All operations involving only rational quantities are supported using arbitrary precision rational arithmetic, allowing to obtain exact results
- Native linear programing for rationals

Many more are scheduled to be added in the near future.

5.3. LeakWatch: Estimating Information Leakage from Java Programs

Participant: Yusuke Kawamoto.

http://www.cs.bham.ac.uk/research/projects/infotools/leakwatch/

Comète contributed to the development of LeakWatch, a quantitative information leakage analysis tool for the Java programming language, created by several people at the University of Birmingham.

LeakWatch is based on a flexible "point-to-point" information leakage model, where secret and publicly-observable data may occur at any time during a program's execution. LeakWatch repeatedly executes a Java program containing both secret and publicly-observable data and uses robust statistical techniques to provide estimates, with confidence intervals, for min-entropy leakage (using a new theoretical result from [23]) and mutual information.

GEOMETRICA Project-Team

5. New Software and Platforms

5.1. CGAL, the Computational Geometry Algorithms Library

Participants: Jean-Daniel Boissonnat, Olivier Devillers, Marc Glisse, Aymeric Pellé, 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.

5.1.1. High-dimensional kernel Epick_d
Participant: Marc Glisse.

We implemented a new high-dimensional kernel taking advantage of the progress that was made in dimensions 2 and 3. It is meant to be used with a reimplementation of high-dimensional triangulations (in progress).

5.1.2. Number type Mpzf
Participant: Marc Glisse.

We added a new exact ring number type that can represent all finite double floating-point numbers. It makes building a Delaunay triangulation 8 times faster than with earlier CGAL releases in some degenerate cases.

5.1.3. CGALmesh: a Generic Framework for Delaunay Mesh Generation
Participants: Jean-Daniel Boissonnat, Mariette Yvinec.

In collaboration with Pierre Alliez (EPI Titane), ClémentJamin (EPI Titane)

CGALmesh is the mesh generation software package of the Computational Geometry Algorithm Library (CGAL). It generates isotropic simplicial meshes – surface triangular meshes or volume tetrahedral meshes – from input surfaces, 3D domains as well as 3D multi-domains, with or without sharp features. The underlying meshing algorithm relies on restricted Delaunay triangulations to approximate domains and surfaces, and on Delaunay refinement to ensure both approximation accuracy and mesh quality. CGALmesh provides guarantees on approximation quality as well as on the size and shape of the mesh elements. It provides four optional mesh optimization algorithms to further improve the mesh quality. A distinctive property of CGALmesh is its high flexibility with respect to the input domain representation. Such a flexibility is achieved through a careful software design, gathering into a single abstract concept, denoted by the oracle, all required interface features between the meshing engine and the input domain. We already provide oracles for domains defined by polyhedral and implicit surfaces. [27] [53]

5.1.4. Periodic Meshes
Participants: Aymeric Pellé, Monique Teillaud.

There is a growing need for a 3D periodic mesh generator for various fields, such as material engineering or modeling of nano-structures. We are writing a software package answering this need, and which will be made publicly available in the open source library CGAL. The software is based on the CGAL 3D volume mesh generator package and the CGAL 3D periodic triangulations package. [42] [63]

5.2. Gudhi library
Participants: Jean-Daniel Boissonnat, Marc Glisse, Clément Maria, Mariette Yvinec.

With the collaboration of David Salinas

https://project.inria.fr/gudhi/software/

The GUDHI open source library will provide the central data structures and algorithms that underly applications in geometry understanding in higher dimensions. It is intended to both help the development of new algorithmic solutions inside and outside the project, and to facilitate the transfert of results in applied fields. The first release of the GUDHI library includes: – Data structures to represent, construct and manipulate simplicial complexes; – Algorithms to compute persistent homology and multi-field persistent homology.

GRACE Project-Team

5. New Software and Platforms

5.1. CADO-NFS-DLOG

F. Morain is one of the developers of CADO-NFS (available at http://cado-nfs.gforge.inria.fr/), which now includes new algorithms for discrete logarithm computations over finite fields.

5.2. Fast Compact Diffie–Hellman software

Working with C. Costello (Microsoft Research) and H. Hisil (Yasar), B. Smith contributed to the development of a competitive, high-speed, open implementation of the Diffie–Hellman protocol (described in [21]), targeting the 128-bit security level on Intel platforms. The source code is freely available at http://research.microsoft.com/en-us/downloads/ef32422a-af38-4c83-a033-a7aafbc1db55/ and http://hhisil.yasar.edu.tr/files/hisil20140318compact.tar.gz.

5.3. Platforms

5.3.1. ACTIS: Contribution to Sage

In the beginning of 2014,D. Augot and C. Pernet submitted an IJD proposal (ingénieur jeune diplomé) to Inria, called Projet Actis (Algorithmic Coding Theory In Sage). The aim of this project is to vastly improve the state of the error correcting library in Sage. The existing library does not present a good and usable API, and the provided algorithms are very basic, irrelevant, and outdated. We thus have two directions for improvement: renewing the APIs to make them actually usable by researchers, and incorporating efficient programs for decoding, like J. Nielsen's CodingLib, which contains many new algorithms.

We hired D. Lucas on October 1st; he has started implementing various basic things, in a standalone manner. We plan to publish these snippets of code to the Sage community in January 2015. Our plan is to interact a lot with the Sage community, to ensure that our new APIs will cover most of the needs of various communities.

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

5. New Software and Platforms

5.1. Software

5.1.1. Software

5.1.1.1. Mole/Cunf: unfolders for Petri Nets
Participant: Stefan Schwoon [correspondant].

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 in recent years, most recently [33].

In the context of MExICo, we have created a new tool called Cunf [82], which is able to handle contextual nets, i.e. Petri nets with read arcs [80].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 5.2.1). Cunf has also participated in the Model Checking Contest held at the Petri Nets conference in 2013 and 2014.

5.1.1.2. TOURS: Testing On Unfolded Reactive Systems
Participant: Hernán Ponce de León [correspondant].

The MOLE - based testing tool TOURS [42] has been developed in 2014 with the help of intern Konstantinos Athanasiou, jointly supervised by Hernán Ponce de León and Stefan Schwoon of the MExICo team at LSV); it has served successfully to experiment the partial-order based testing methodology on a scalable benchmark example (elevator control).

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/

5.2. Platforms

5.2.1. Platform CosyVerif

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

The platform is client/server based. The modeler creates models on the client side, either programmatically, or in a dedicated graphical editor. Tools are then executed on the server side.

CosyVerif is available as installable bundles, that embed the client, the server, and also the tools. It is also usable through a public server hosted within the laboratory.

The platform offers a common language for the description of the models, in order to create interoperability between clients and tools. It also provides a way to define easily new formalisms within the platform, and to manipulate models that are instances of these formalisms. To the best of our knowledge, no other verification framework presents such a feature.

CosyVerif targets three different kinds of users:

- Students use this platform in two M2 courses in modeling and verification courses. *Citer les deux cours*

- Tool developers, that are usually researchers, use the platform to distribute their tools, and have a demonstration version easily available. They also use CosyVerif for tutorials in conferences or workshops *Citer Petri nets 2014*.

- Industrial case studies have used the platform since its creation to prove properties on systems in various fields, such as: transportation systems, scheduling, hardware, robotics, databases, banking systems, home automation...

The platform is managed by a steering committee consisting of researchers and engineers of three laboratories: LIP6, LIPN, LSV. This committee decides strategic orientations as well as technical choices.

This year, we have fully redesigned the platform, with two goals in mind: first, to use technologies that target better our users; and second, to provide more functionalities.

- We switched to lightweight web technologies, in order to ease the deployment and use of CosyVerif. For the users, it means that they can access a graphical editor within their web browser. They can also access the platform through an API, usable with any HTTP client.

- We improved the language for formalisms and models in order to allow the modular definition of new formalisms. We switched from a class/instance paradigm to a prototype one, that allows to represent complex models in a both efficient and usable way.

- We extended the server to handle not only executions. It is now primarily a repository of formalismes, models, services and executions, that belong to users or project. It also handles the tools executions, and the collaborative edition of models.

- We started working on a system to help building packages for the various components of the platform (client, server, tools, ...), to ease its installation. It is used to create the bundles of CosyVerif, that are available to download. Another team (Secsi) of the LSV laboratory is interested in this system, and will support its development in 2015.

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 used for teaching in two master programs (Universities Paris 6 and Paris 13/Villetaneuse) It has been used in a tutorial in the Petri Nets 2014 conference.

We are currently in the process of giving a better visibility to the project, by transforming it into a consortium. Our goal is to identify industrial fields where the tools of the platform can be applied successfully, by proposing services to the industry. The strength of the platform relies on the variety of techniques offered by the tools, that adapt to a wide range of problems. In order to increase the number of techniques, we have been joined by another partner from Geneva.

<p style="text-align:center">**PARSIFAL Project-Team**</p>

5. New Software and Platforms

5.1. Abella

Participants: Kaustuv Chaudhuri [correspondant], Matteo Cimini [Indiana University], Dale Miller, Olivier Savary-Bélanger [Princeton University], Mary Southern [University of Minnesota], Yuting Wang [University of Minnesota].

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

Abella is an interactive theorem prover for reasoning about data structures with binding constructs using the λ-tree approach to syntax. It consists of a sophisticated reasoning logic that supports induction, co-induction, and generic reasoning. Abella also supports the *two-level logic approach* by means of a specification logic based on the logic programming language λProlog.

In 2014, the following additions were made to the system.

- A new translation layer was added to Abella's specification layer, which was used to build an interface to the LF dependent type theory [61]. This extension was documented in the following paper: [27]. A number of examples of the use of this new specification language are available at the following URL: http://abella-prover.org/lf
- Two minor releases were made, versions 2.0.2 and 2.0.3, that fixed a number of bugs and added several convenience features. Consult the change log for more details.

Accompanying these additions were the following publications.

- A new comprehensive tutorial for the Abella system has been accept to appear in the *Journal of Formalized Reasoning* [31].
- The new tactical plugin architecture and the dynamic contexts plugin of Abella in the following paper: [26].
- The use of co-induction and higher-order relations to formalize the meta-theory of various bisimulation-up-to techniques for common process calculi: [19].

5.2. Bedwyr

Participants: Quentin Heath, Dale Miller [correspondant].

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

Quentin Heath has continued to maintain and enhance this model checking system. In particular, the tabling mechanism has been extended and formalized to a greater extent. The tabling mechanism is now able to use Horn clause lemmas in order to increase the power of the table. For example, given this enhancement it is possible to tell Bedwyr that if a given board position (in some game) has a winning strategy then symmetric versions of that board also have winning strategies. Thus, when a given board position is recognized as winning, then table will understand that all symmetric versions of that board are winning.

Significant energies have also gone into trying to understand how cyclic proofs (recognized using the tabling mechanism) can be turned into certifiable proof evidence. Good results are currently developed for treating bisimulation and non-reachability: in these cases, cyclic proofs are used to supply invariants for induction and co-induction.

5.3. Psyche

Participants: Stéphane Graham-Lengrand [correspondant], Assia Mahboubi, Jean-Marc Notin.

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/

The engine implements the ideas developed in the section "Trustworthy implementations of theorem proving techniques" above, and was the object of the system description [56].

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 how fast it drives the kernel towards the final answer.

In 2014, major developments were achieved in Psyche, whose version 2.0 was released on 20th September 2014. It is now equipped with the machinery to handle quantifiers and quantifier-handling techniques. Concretely, it uses meta-variables to delay the instantiation of existential variables, and constraints on meta-variables are propagated through the various branches of the search-space, in a way that allows local backtracking. The kernel, of about 800 l.o.c., is purely functional.

POSTALE Team

4. New Software and Platforms

4.1. New Software

4.1.1. MyNRC: image-oriented library for allocation and manipulation of SIMD 1D, 2D and 3D structures

Participant: Lionel Lacassagne.

MyNRC is multi-plateform library that can handle SSE, AVX, Neon and ST VECx registers.

4.1.2. CovTrack: agile realtime multi-target tracking algorithm

Participants: Michèle Gouiffès, Lionel Lacassagne, Florence Laguzet, Andrés Romero.

4.1.3. tmLQCD for Blue Gene/Q

Participant: Michael Kruse [correspondant].

tmLQCD is a program suite for lattice quantum chromodynamics (Lattice QCD) using the chirally twisted Wilson quarks to reduce discretization artifacts. This software is in productive use by the European Twisted Mass Collaboration (ETMC).

As to not waste precious computation time on the supercomputers it is running on, it is important to optimize the code in order to run as fast as possible. tmLQCD has already been customized for Intel Xeon processors, the Blue Gene/L and Blue Gene/P from IBM. For the latter's successor, the Blue Gene/Q, more profound changes to the program are necessary. With these changes, tmLQCD reaches a peak performance of up to 55% of the machines theoretical floating point performance.

The Blue Gene/Q optimized tmLQCD is available at: http://github.com/Meinersbur/tmLQCD

4.1.4. Molly

Participant: Michael Kruse [correspondant].

Using Polly extension, the LLVM compiler framework is able to automatically parallelize general programs for shared memory threading for by exploiting the powerful analysis and transformations of the polyhedral model.

Molly adds the ability to manage distributed memory using the polyhedral model and is therefore able to automatically parallelize even for the largest of today's supercomputer. Once the distribution of data between the computer's nodes is known, Molly determines the values that are required to be transferred between the nodes and chunks them into as few messages as possible. It also keeps tracks of the buffers required by the MPI interface. Transfers are asynchronous such that further computations take place while the data is being transferred.

Molly has not yet been officially released.

4.1.5. Dohko (http://dohko.io/)

Participant: Alessandro Ferreira Leite [correspondant].

Automating multi-cloud configuration is a difficult task. The difficulties are mostly due to clouds' heterogeneity and the lack of tools to coordinate cloud computing configurations automatically. As a result, virtual machine image (VMI) is the common approach to configure cloud environment. Although VMI can handle functional properties like minimum disk size, operating system, and software packages, it leads to a high number of configuration options, increasing the difficulty to select one that matches users' requirements. Moreover, the usage of VMI usually results in vendor lock-in. Furthermore, VMI leaves for the users the work of selecting a resource to deploy the image and for orchestrating them accordingly, i.e., the work of selecting and instantiating the VMI in each cloud. In addition, VMI migration across multiple clouds normally has a high cost due to network traffic. Finally, in case of cloud's failure, it may be difficult for users to re-create the failed environment in another cloud, since the image will be inaccessible.

Therefore, to overcome these issues, we developed a configuration management tool for cloud computing. Our tool, called Dohko, allows the users to configure a multi-cloud computing environment, following a declarative strategy. In this case, the users describe their applications and requirements and use our tool to select the resources and to set up the whole computing environment automatically, taking into account temporal and functional dependencies between the resources. Moreover, following a software product line (SPL) engineering method, Dohko captures the knowledge of configuring cloud environments in form of reusable assets. In this case, a product is a cloud computing environment that meets the user requirements, where the requirements can be either based on high or low-level descriptions. Examples of low-level descriptions include: virtualization type, disk technology, sustainable performance, among others, whereas high-level descriptions include the number of CPU cores, the RAM memory size, and the maximum monetary cost per hour. In this context, a cloud computing environment also matches cloud's configuration constraints. Besides that, thanks to the usage of an extended feature model (EFM) with attributes, our approach enables the description of the whole computing environment (i.e., hardware and software) independent of cloud provider and without requiring the usage of virtual machine image. In this case, it relies on an off-the-shelf constraint satisfaction problem (CSP) solver to implement the feature model and to select the resources.

By employing a declarative strategy, Dohko could execute a biological sequence comparison application in two distinct cloud providers (i.e., Amazon EC2 and Google Compute Engine) considering a single and a multi-cloud scenario, demanding minimal users' intervention to instantiate the whole cloud environment, as well as to execute the application. In particular, our solution tackles the lack of middleware prototypes that can support different scenarios when using services from many clouds. Moreover, it meets the functional requirements identified for multiple cloud-unaware systems [136] such as: (a) it provides a way to describe functional and non-functional requirements through the usage of an SPL engineering method; (b) it can aggregate services from distinct clouds; (c) it provides a homogeneous interface to access services of multiple clouds; (d) it allows the service selection of the clouds; (e) it can deploy its components across many clouds; (f) it provides automatic procedures for deployments; (g) it utilizes an overlay network to connect and to organize the resources; (h) it does not impose any constraint for the connected clouds.

4.2. Platforms

4.2.1. *Fast linear system solvers in public domain libraries (http://icl.cs.utk.edu/magma/)*

Participant: Marc Baboulin [correspondant].

Hybrid multicore+GPU architectures are becoming commonly used systems in high performance computing simulations. In this research, we develop linear algebra solvers where we split the computation over multicore and graphics processors, and use particular techniques to reduce the amount of pivoting and communication between the hybrid components. This results in efficient algorithms that take advantage of each computational unit [12]. Our research in randomized algorithms yields to several contributions to propose public domain libraries PLASMA and MAGMA in the area of fast linear system solvers for general and symmetric indefinite systems. These solvers minimize communication by removing the overhead due to pivoting in LU and $LDLT$ factorization. Different approaches to reduce communication are compared in [2].

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

4.2.2. cTuning Framework (http://cTuning.org): 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 [10].

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.

4.2.3. NT2 (http://www.github.com/MetaScale/nt2)

Participants: Pierre Esterie, Joël Falcou, Mathias Gaunard, Ian Masliah, Antoine Tran Tan.

NT2 is a C++ high level framework for scientific computing.[18]

4.2.4. Boost.SIMD (http://www.github.com/MetaScale/nt2)

Participants: Pierre Esterie, Joël Falcou, Mathias Gaunard.

Boost.SIMD provides a portable way to vectorize computation on Altivec, SSE or AVX while providing a generic way to extend the set of supported functions and hardwares.

SPECFUN Project-Team

5. New Software and Platforms

5.1. SSReflect

SSReflect is a language extension of the Coq system and was originally written by G. Gonthier for his formal proof of the Four-Color Theorem [0]. In the team, A. Mahboubi and E. Tassi participate to its development, maintenance, distribution, documentation, and user support. A new version (1.5) was released in March 2014. The proof language now offers fine-grained control on type-classes inference and offers new proof commands to ease forward reasoning. In particular the 'have' tactic now supports new modifiers to ease stating generalized formulas as well as hoisting out deeply nested forward steps.

5.2. The Mathematical Components library

The Mathematical Components library is a set of Coq libraries that cover the mechanization of the proof of the Odd Order Theorem, with large contributions by A. Mahboubi and E. Tassi. After the formal proof was completed in September 2012, stable libraries had been distributed [0] with the SSReflect extension, while remaining parts of the libraries had remained under continued improvements in view of potential reuse. In March 2014, version 1.5 of library was released. With it, the library includes 16 more theory files, covering in particular field and Galois theory, advanced character theory, and a construction of algebraic numbers.

5.3. Coq

The way Coq processes theory files has been improved. When used as a batch compiler, Coq is now able to decouple the checking of statements and definitions from the checking of proofs. All proofs can be checked independently taking advantage of modern parallel hardware. When used interactively in conjunction with PIDE-based interfaces, Coq is now able to process the document asynchronously by delegating most of the task to external workers.

The Coq build process was also improved to better support the Windows platform and to enable third parties to provide pre-compiled plugins for such platform.

5.4. Coq/jEdit

Building on top of the asynchronous processing of Coq proofs, we have implemented a plugin that connects the jEdit generic text editor to Coq. This plugin is an adaptation of a similar plugin, written by M. Wenzel, for the Isabelle proof assistant. The interaction using this plugin is a significant change from existing user interfaces, making full use of Coq's asynchronous processing capabilities to provide richer feedback about the proof a user is editing.

The plugin was released as a beta in November 2014 and is available at http://pages.saclay.inria.fr/carst. tankink/jedit.html.

5.5. Other maintained software

We still actively maintain the following other software, which have not had a new release this year.

---

[0] http://www.msr-inria.fr/projects/mathematical-components/
[0] http://www.msr-inria.fr/projects/mathematical-components/

5.5.1. 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/1.9.1/ddmf. We count hundreds of user sessions per month. Source code distributed under CeCILL-B. A next release is under preparation: it will base on a different, more user-friendly rendering tool (MathJax) and will display more contents.

5.5.2. DynaMoW

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

5.5.3. Ring

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

5.5.4. 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/. This software has been used this year for our formal proof of irrationality of $\zeta(3)$.

TOCCATA Project-Team

5. New Software and Platforms

5.1. 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.[0]

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/. It is also distributed as part of major Linux distributions and in the OPAM packaging system http://opam.ocaml.org/packages/why3/why3.0.85/.

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 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 (e.g., at the 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. The *Why3* web site http://why3.lri.fr lists a few other works done by external researchers and relying on the use of *Why3*.

Two versions were released in 2014: 0.83 released in March and 0.84 in September, plus a few days later a bugfix version 0.85.

5.2. 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 automated proof engine, 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 ina *Why*'s annotation language; this means that *Alt-Ergo* fully supports first order polymorphic logic with quantifiers. *Alt-Ergo* also supports the standard [116] 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. It is used as back-end prover in the environments Frama-C and CAVEAT for static analysis of C developed at CEA. In this context, Alt-Ergo has been qualified by Airbus and is integrated in the next generation of Airbus development process. Alt-Ergo is usable as a back-end prover in the SPARK verifier for ADA programs, since Oct 2010, and is also the main back-end prover of the new SPARK2014.

Alt-Ergo is integrated in several other tools and platforms: the Bware platform for discharging VCs generated by Atelier B, the EasyCrypt environment for verifying cryptographic protocols, the Pangolin programming language http://code.google.com/p/pangolin-programming-language/, etc.

Last but not least, Alt-Ergo is the solver used by the Cubicle model checker described below.

---

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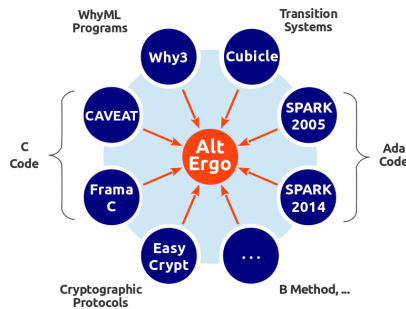the Software Working Group of Inria Evaluation Committee (http://www.inria.fr/institut/organisation/instances/commission-d-evaluation)

Figure 2.

Alt-Ergo is distributed as open source, under the CeCILL-C license, at URL http://alt-ergo.lri.fr/, and in the OPAM packaging system http://opam.ocaml.org/packages/alt-ergo/alt-ergo.0.95.2/. Latest public version is 0.99.1, released in Dec. 2014. Maintenance is done by the OcamlPro company http://alt-ergo.ocamlpro.com/.

5.3. 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, which corresponds to 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.

Cubicle is distributed as open source, under the Apache license, at URL http://cubicle.lri.fr/, and in the OPAM packaging system http://opam.ocaml.org/packages/cubicle/cubicle.1.0.1/. Latest version is 1.0.1, released in Nov. 2014.

5.4. 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.

Flocq is currently used by the CompCert certified compiler for its support of floating-point computations.

It is distributed as open source, under a LGPL license, at http://flocq.gforge.inria.fr/. It was first released in 2010. Current version is 2.4.0 released in Sep. 2014.

5.5. 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 [82], [123].

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/. Latest version is 1.1.2 released in October 2014.

Part of the work on this tool was done while in the Arénaire team (Inria Rhône-Alpes), until 2008.

5.6. The Interval package for Coq

Participants: Guillaume Melquiond [contact], Érik Martin-Dorel.

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*.

Versions 1.0 and 2.0 were released in 2014. Version 2.0 integrates the CoqApprox library for computing Taylor models, so as to greatly improve performances when bounding univariate expressions [43].

It is distributed as open source, under a CeCILL-C license, at http://coq-interval.gforge.inria.fr/. Latest version is 2.0 released in November 2014.

Part of the work on this library was done while in the Mathematical Components team (Microsoft Research–Inria Joint Research Center).

5.7. 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.

The Coquelicot library is designed with three principles in mind. The first is the user-friendliness, achieved by implementing methods of automation, but also by avoiding dependent types in order to ease the stating and readability of theorems. This latter part was achieved by defining total function for basic operators, such as limits or integrals. The second principle is the comprehensiveness of the library. By experimenting on several applications, we ensured that the available theorems are enough to cover most cases. We also wanted to be able to extend our library towards more generic settings, such as complex analysis or Euclidean spaces. The third principle is for the Coquelicot library to be a conservative extension of the Coq standard library, so that it can be easily combined with existing developments based on the standard library. Moreover, we achieved this compatibility without adding any additional axiom.

The result is the Coquelicot library available at http://coquelicot.saclay.inria.fr. Latest version is 2.0.1 released in March 2014. It contains about 1,700 theorems and 37,000 lines of Coq.

5.8. The CFML tool for verifying OCaml code

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. *CFML* proofs establish the full functional correctness of the code with respect to a specification. They may also be used to formally establish bounds on the asymptotic complexity of the code. 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/. It has been continuously extended since its first release in 2010. In particular, in 2014 support for the verification of asymptotic complexity bounds has been added.

5.9. Other Maintained Tools

5.9.1. *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.

ALEA 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 [55]. It is also experimented in LABRI as a basis to study formal proofs of probabilistic distributed algorithms.

ALEA is distributed as open source, at http://www.lri.fr/~paulin/ALEA. Latest version is 8 released in May 2013. In particular, it includes a module to reason about random variables with values in positive real numbers.

5.9.2. *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/. Latest version is 1.98 released in July 2014. Bibtex2html is also distributed as a package in most Linux distributions, and in the OPAM packaging system http://opam.ocaml.org/packages/bibtex2html/bibtex2html.1.98/.

We estimate that between 10000 and 100000 web pages have been generated using Bibtex2html.

5.9.3. *The Coccinelle library for term rewriting*
Participant: Évelyne Contejean [contact].

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

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 a number of algorithms implemented in the CiME toolbox, such as matching, matching modulo associativity-commutativity, computation of the one-step reducts of a term, recursive path ordering (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.

5.9.4. *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/. Latest version is 1.8.5, released in March 2014. It is also distributed as a package in several Linux distributions. OCamlgraph is now widely spread among the community of *OCaml* developers, and available as an OPAM package http://opam.ocaml.org/packages/ocamlgraph/ocamlgraph.1.8.5/.

5.9.5. *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 [52].

Mlpost is available as an OPAM package http://opam.ocaml.org/packages/mlpost/mlpost.0.8.1/.

5.9.6. *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 [91] and at TFP 2011 [90].

Functory is distributed as open source, under the LGPL license, at http://functory.lri.fr/, and in the OPAM packaging system http://opam.ocaml.org/packages/functory/functory.0.5/. Latest version is 0.5, release in March 2013.

5.9.7. *The Why Environment*

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

Criteria for Software Self-Assessment: 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/.

It also distributed as part of major Linux distributions and in the OPAM packaging system http://opam.ocaml.org/packages/why/why.2.34/. Version 2.34 was released in August 2014, to provide a version compatible with both *Frama-C*Neon and *Why3* 0.83.

The internal VC generator and the translators to external provers are no longer under active development, as superseded by the *Why3*system described above. 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 [92], at Universidade do Minho in Portugal [50], at Moscow State University, Russia (http://journal.ub.tu-berlin.de/eceasst/article/view/255).

<p style="text-align:center">**COMMANDS Project-Team**</p>

5. New Software and Platforms

5.1. Bocop

Participants: Pierre Martinon [corresponding author], Frédéric Bonnans, Daphné Giorgi, Olivier Tissot.

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, Daphné Giorgi (Oct. 2012-Sept. 2014), and Olivier Tissot since October 2014. The first prototype was released at the end of 2011, Bocop is currently at version 2.0.1 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 ([13], [26]), swimming micro-robots ([71]), and quantum control for medical imaging ([35]). Bocop was also featured during our participation in the Imatch "Optimisation and Control" in october 2013, leading to an ongoing contract with the startup Safety Line, on fuel optimization for civil aircrafts.

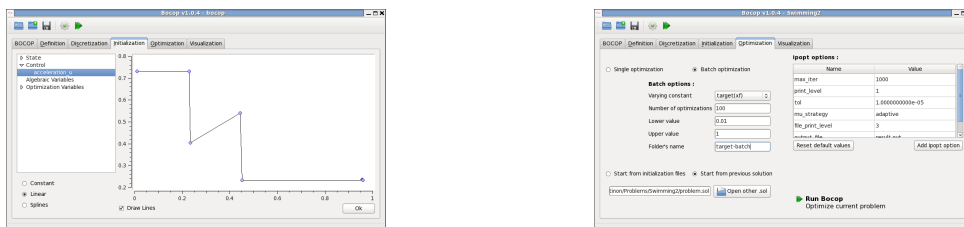*Bocop auto-assessment according to Inria notice: A3up4, SO3, SM3, EM3up4, SDL4up5*

Figure 1. BOCOP

DEFI Project-Team

5. New 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, Mohamed 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 and also to [15].

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).

5.3.3. *SAXS Utilities*

Participants: Federico Benvenuto [correspondant], Houssem Haddar.

We developed a scilab and matlab toolboxes that post treat SAXS type measurements to identify size distributions of diluted particles. We treat both axisymmetric measuremenst and anisotropic ones. The toolbox also simulates SAXS measurements associated with some canonical geometries.

5.3.4. *Direct Solver for periodic media*

Participants: Thi Phong Nguyen [correspondant], Houssem Haddar.

This Matlab toolbox solves the scattering from locally perturbed periodic layer using Floquet-Bloch transform and spectral discretization of associated volume integral equation.

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.

5.5. 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.

<p style="text-align:center">**DISCO Project-Team**</p>

5. New Software and Platforms

5.1. OreModules

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

The OREMODULES package [73], based on the commercial Maple package Ore_algebra [74], 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 [82], [84] 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) [90], [89], [72] and it computes important tools of homological algebra (e.g., (minimal) free resolutions, split exact sequences, extension functors, projective or Krull dimensions, Hilbert power series). The abstract language of homological algebra used in the algebraic analysis approach carries over to the implementations in OREMODULES: up to the choice of the domain of functional operators which occurs in a given system, all algorithms are stated and implemented in sufficient generality such that linear systems defined over the Ore algebras developed in the Ore_algebra package are covered at the same time. Applications of the OREMODULES package to mathematical systems theory, control theory and mathematical physics are illustrated in a large library of examples. The binary of the package is freely available at http://wwwb.math.rwth-aachen.de/OreModules/.

5.2. Stafford

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

The STAFFORD package of OREMODULES [73] contains an implementation of two constructive versions of Stafford's famous but difficult theorem [96] 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 [92] 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 [96] have recently been made constructive in [93] (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: Anna Fabiańska [Univ. Aachen], Alban Quadrat [correspondent].

The QUILLEN-SUSLIN package [78] contains an implementation of the famous Quillen-Suslin theorem [94], [97]. 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 [78] (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: Thomas Cluzeau [ENSIL, Univ. Limoges], Alban Quadrat [correspondent].

The OREMORPHISMS package [76] of OREMODULES [72] 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://pages.saclay.inria.fr/alban.quadrat/OreMorphisms/index.html.

5.5. PurityFiltration

Participant: Alban Quadrat [correspondent].

The PURITYFILTRATION package, built upon the OREMODULES package, is an implementation of a new effective algorithm obtained in [24] which computes the purity/grade filtration [67], [68] of linear functional systems (e.g., partial differential systems, differential time-delay systems, difference systems) and equivalent block-triangular matrices. 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. For more information, see http://pages.saclay.inria.fr/alban.quadrat/OreAlgebraicAnalysis/index.html.

5.6. OreAlgebraicAnalysis

Participants: Thomas Cluzeau [ENSIL, Univ. Limoges], Alban Quadrat [correspondent], Maris Tõnso [Institute of Cybernetics, Univ. Tallinn].

OREALGEBRAICANALYSIS is a Mathematica implementation of algorithms available in the OREMODULES and the OREMORPHISMS packages (developed in Maple). OREALGEBRAICANALYSIS is based on the implementation of Gröbner bases over Ore algebras available in the Mathematica *HolonomicFunctions* package developed by Christoph Koutschan (RICAM). OREALGEBRAICANALYSIS can handle larger classes of Ore algebras than the ones accessible in Maple, and thus we can study larger classes of linear functional systems. Finally, Mathematica internal design allows us to consider classes of systems which could not easily be considered in Maple such as generic linearizations of nonlinear functional systems defined by explicit nonlinear equations and systems containing transcendental functions (e.g., trigonometric functions, special functions). This package has been developed within the PHC Parrot project CASCAC.

5.7. YALTA

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

The YALTA toolbox is a Matlab toolbox 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, [70] and [79]) and on classical continuation methods exploiting the particularities of the problem [80], [81].

For classical delay systems, a Pade2 approximation scheme is available as well as a finite-dimensional approximation of the system.

Recently, some optimizations and features have been added. For instance, the main software function has been split into several procedures, allowing some calculus such as finding the position of the neutral chains to be processed independently of more computationnally expensive ones (for instance determining the root locus with respect to the delay of the system). In parallel, software documentation has been rewritten.

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

The YALTA GUI (graphical user interface) is a graphical application developed in Python that interacts with the Matlab toolbox 5.7 . User actions are performed through intuitive graphic elements (dialog boxes, icons, menus, scroll bars) in order to capitalize on the functionalities of YALTA. This software, still in development, is based on PyQt, a Python binding of the cross-platform GUI toolkit Qt (C++).

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

5. New 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.

5. New 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 [76], [147]. Il faut aussi noter que le groupe de L. Hardouin, du LISA/Istia, a complété la boîte à outils Maxplus en interfaçant leur propre librairie C++, qui permet le calcul des séries de transfert de graphes d'événements temporisés.

English version

Three researchers of the team (S. Gaubert, J.-P. Quadrat, and G. Cohen, building on a preliminary version of M. McGettrick) have developed and released the *Maxplus toolbox* of Scilab, which is freely available among the contributions on the Scilab web site, and which is now included by default in Scicoslab. It implements all basic linear algebra functionalities, with a special attention to large sparse matrices, including efficient algorithms for eigenvalue computation based on policy iteration. The software has been used by several researchers in their work, including [76], [147]. 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 [95]. 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 [112]), 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.

English version

The policy iteration algorithm for zero sum repeated games with ergodic payoff (i.e. mean payoff per time unit), and in degenerate "multichain" cases, has been introduced in [95]. Several internships allowed us to implement in Scilab, C or C++, and to test such algorithms (see the work of Vishesh Dhingra [112]), 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 §6.3.1 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 [108]), 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 [70]).

TPLib est utilisé dans le logiciel Polymake [128], développé à la Technische Universität Berlin (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 §6.6.4). De même, une interface à la bibliothèque de domaines abstraits numériques APRON [140] 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 §6.3.1 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 [108]), 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 [70]).

TPLib is used in the software Polymake [128], developed in Technische Universität Berlin (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 §6.6.4). An interface to the numerical abstract domain APRON [140] is also under development.

POEMS Project-Team

5. New Software and Platforms

5.1. Software

5.1.1. *XLiFE++*

Participants: Eric Lunéville, Nicolas Kielbasiewicz, Colin Chambeyron.

XLIFE++ is a Finite Element library in C++ based on philosophy of the previous library MELINA in Fortran but with new capabilities (boundary elements, 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.

After 3 years of work, the development of the finite element library XLIFE++ reached a milestone in 2014 with the first downloadable public release, after an important effort to improve the user interface and to complete the last major developments necessary for this output: essential boundary conditions, mesh construction, Dirichlet-to-Neumann maps ... among others. In June 2014, a day was organized to present to a wider audience the features of this library. We now provide support to the users (patches, new developments...).

5.1.2. *COFFEE*

Participant: Stéphanie Chaillat.

COFFEE is a 3D BEM-accelerated FMM solver for linear elastodynamics (full implementation, 30 000 lines of Fortran 90). The 3-D elastodynamic equations are solved with the boundary element method accelerated by the multi-level fast multipole method. The fundamental solutions for the infinite space are used in this implementation. A boundary element-boundary element coupling strategy is also implemented so multi-region problems (strata inside a valley for example) can be solved.

REGULARITY Project-Team

5. New Software and Platforms

5.1. FracLab

Participant: 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 and fifty 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.

SELECT Project-Team

5. New 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 2010, MIXMOD has a proper graphical user interface. 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.

This year, it has been decide to create MIXMODSTORE which proposes companion programs of MIXMOD. As a matter of fact, the program MixmodCombi of Jean-Patrick Baudry (Université Paris 6) and Gilles Celeux which allows a hierarchical clustering derived from a mixture has been associated to Rmixmod.

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. It ha been enriched by a full Bayesian version using a Gibbs sampler. This Gibbs sampler coupled with the variaotional Bayes algorithm provides solutions more stable and less dependent of the starting values of the algorithm. An exact expression of criterion ICL has been provided. This criterion or BIC are used for selecting a relevant block clustering.

TAO Project-Team

5. New Software and Platforms

5.1. METIS

Participants: Olivier Teytaud [correspondent], Jérémie Decock, Jean-Joseph Christophe, Vincent Berthier, Marie-Liesse Cauwet, Jialin Liu, Sandra Cecilia Astete Morales.

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.

Metis is the Artelys/Tao contribution to Crystal, which is at the heart of the Post project, which is selected by the European Commission for a 4-years project for energy modeling http://www.artelys.com/news/120/90/Energy-The-European-Commission-Chooses-Artelys-Crystal.

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) in 7x7, and recently an optimization of the random seed which brings a significant improvement in Go and (unpublished) on the difficult case of phantom-Go. However, the work in the UCT-SIG has now shifted to energy management.

5.3. CMA-ES: Covariance Matrix Adaptation Evolution Strategy

Participants: Emmanuel Benazera, Nikolaus Hansen [correspondent].

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

The Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [65] is considered to be state-of-the-art in continuous domain evolutionary computation [64], 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, C++11, 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 6.3). 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

Participant: Balázs Kégl [correspondent].

Keywords: Multi-class, Multi-label Classification.

The MultiBoost package [63] 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 [27]. Its implementation is developed within the FUI project TIMCO.

Link: http://grid-observatory.org

5.7. Platforms

5.7.1. *io.datascience*

This Data as a Service (DaaS) platform is developed in the context of the Center for Data Science and the TIMCO project. Its overall goals is to exploit the advances in semantic web techniques for efficient sharing and usage of scientific data. A related specific software is the Tester for Triplestore (TFT) software suite, which benchmarks the compliance of sparql databases wrt the RDF standard and publishes the results through the SparqlScore service. TFT has been selected for the Semantic Web Challenge [42].

Links: https://io.datascience-paris-saclay.fr, https://github.com/BorderCloud/TFT, http://sparqlscore.com

AMIB Project-Team

4. New Software and Platforms

4.1. 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 [59]. More than 300 visits since its release in January 2012.

4.2. 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.

4.3. 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.

4.4. ConQuR-Bio

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

ConQuR-Bio assists scientists when they query public biological databases. Various reformulations of the user query are generated using medical terminologies (MeSH, OMIM, ...). Such alternative reformulations are then used to rank the query results using a new consensus ranking strategy. The originality of our approach thus lies in using consensus ranking techniques within the context of query reformulation. The ConQuR-Bio system is able to query the Entrez-Gene NCBI database. The benefit of using ConQuR-Bio compared to what is currently provided to users has been demonstrated on a set of biomedical queries.
Availability : http://conqur-bio.lri.fr/

4.5. VARNA (Visualization Application for RNA)

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) [52]...
VARNA currently ships in its 3.9 version, and consists in ∼50 000 lines of code in ∼250 classes.
Availability : Distributed at http://varna.lri.fr since 2009 under the GPL v3 license.

Impact: Downloaded ∼15k times and cited by ∼250 research manuscripts (source: Google Scholar).

4.6. GenRGenS (GENeration of Random GENomic Sequences)

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... [69] GENRGENS currently ships in its 2.0 version, and consists in ∼25 000 lines of code in ∼120 Java classes.
Availability : Distributed at http://www.lri.fr/~genrgens/ since 2006 under the terms of the GPL v3 license.
Impact: Downloaded ∼5k times and cited by ∼60 times (source: Google Scholar).

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. 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.

4.9. Pint

Participant: Loïc Paulevé [correspondant].

PINT provides several command-line tools to model, simulate, and analyse the dynamics of automata networks. Its main application domain is systems biology for modelling and analysis of very large interaction networks. Besides a textual language for specifying networks and standard stochastic simulation algorithms, PINT implements static analysis for analysing and controlling the transient reachability. In particular, PINT provides the computation of cut sets for transient reachability, that gives sets of key automata states, whose mutation would prevent the concerned reachability to occur.

PINT has been applied to extremely large biological networks, from 100 to 10,000 interacting components, demonstrating its scalability and potential to handle full databases of interactions.

PINT is distributed under the CeCiLL licence, and is available at http://loicpauleve.name/pint.

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

5. New 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. Ranking with High-Order Information

Participant: Puneet Dokania [Correspondant].

Average precision optimization, high-order information, ranking The software (publicly available at http://cvn.ecp.fr/projects/ranking-highorder/) provides a convenient API for learning to rank with high-order information. The samples are ranked according to a scorethat is proportional to the difference of max-marginals of the positive and the negative class. The parameters of the score function are computed by minimizing an upper bound on the average precision loss. The software also provides an instantiation of the API for ranking samples according to their relevance to an action, using the poselet features.

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.

M3DISIM Team

5. New Software and Platforms

5.1. FELISCE

Participants: Dominique Chapelle, Sébastien Gilles [correspondant], Sébastien Imperiale, 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/.

In FELISCE we have prepared a branch called HappyHeart, which aims at providing a user-friendly interface able to deal efficiently with complex cardiovascular simulations. Started in 2013, the code is already quite large (about 55 000 lines of code in almost 700 different files) and its core is about to be complete in early 2015. It includes among others full HPC functionalities, high-order finite elements, physics coupling and topology capabilities. Our purpose will then be to use the library to implement the sophisticated cardiovascular models of the team and couple them with Verdandi (data assimilation library) to provide patient-specific simulations.

- Software benefit: HappyHeart is a multiphysics HPC FEM Library with cardiac simulation concerns
- Type of human computer interaction: Command line and configuration files.
- OS/Middleware: MacOS, Linux.
- Required library or software: OpenMpi (parallelism), Petsc (linear algebra), Seldon (linear algebra), Parmetis (partitioner), Mumps (solver), Ops (input parameter file management), STL and Yuni (generic C++ libraries).
- Programming language: C++ 11/14.
- Documentation: Doxygen and user's manual in English.

5.2. HeartLab

Participants: Matthieu Caruel, Dominique Chapelle, Alessandro Felder, Philippe Moireau [correspondant].

The heartLab software is a library written in (64-bit 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 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. And recently we have worked on generalized micro-macro cardiac law using stochastic formulations.

5.3. Verdandi

Participants: Aurora Armiento [Mamba team], Dominique Chapelle, Annabelle Collin, Vivien Mallet [Clime team], Karine Mauffrey, Philippe Moireau [correspondant].

Verdandi is an opensource (LGPL) software library aiming at providing data assimilation 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 (1.6) 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 and VP2HF projects and is now referenced in the peer-reviewed article [4].

- 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

Moreover a Matlab module called VerdandinMatlab is developed in the team for pedagogical and test purposes.

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

5. New Software and Platforms

5.1. Scikit learn

Participants: Olivier Grisel [correspondant], Gaël Varoquaux, Bertrand Thirion, Michael Eickenberg, Loïc Estève, Alexandre Gramfort, Fabian Pedregosa Izquierdo.

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: 0.15.2
- Programming language: Python, C/Cython

5.2. Nilearn

Participants: Gaël Varoquaux [correspondant], Bertrand Thirion, Loïc Estève, Alexandre Abraham, Michael Eickenberg, Alexandre Gramfort, Fabian Pedregosa Izquierdo, Elvis Dohmatob, Virgile Fritsch.

NiLearn is the neuroimaging library that adapts the concepts and tools of 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.

- 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], 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. It is devoted to algorithmic solutions for various issues in neuroimaging data analysis. The Nipy project is available, under BSD license, and within NeuroDebian.

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

- Version: 0.3

5.5. PyHRF

Participants: Philippe Ciuciu [correspondant], Aina Frau Pascual, Salma Torkhani.

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

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

5. New Software and Platforms

5.1. Monolix

Participants: Marc Lavielle, 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, 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). A new version of the CTS is available as a R package since December 2014. 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)
- Inter individual and intra individual variability models
- Graphics and statistical tests

5.4. MLXplore

Participant: Marc Lavielle.

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 but POPIX collaborates closely with Lixoft on on the definition of the specifications of MLXplore.

INFINE Team

5. New Software and Platforms

5.1. Software

5.1.1. MACACOapp

Participant: Aline Carneiro Viana.

MACACOapp (https://macaco.inria.fr/macacoapp/) is developed in the context of the EU CHIST-ERA MACACO project (https://macaco.inria.fr/). It consists in a mobile phone application that periodically samples phone's information on the mobility (through, e.g., GPS sensor, accelerometer and WiFi/Bluetooth/Cellular environment, connectivity type) and on the data traffic it generates (through, e.g., Internet browser history and applications data consumption). The information collected will be time-stamped and will be periodically sent to the central servers for analysis and visualization. We expect that (1) the collected information will allow us studying the correlation between mobility and content demand patterns and that (2) the results of this analysis will allow us inferring the best times and places to transfer content from/to users' phones location and/or from/to the wireless infrastructure closest to the users' phones location. Users will be also invited to fill a non-mandatory questionnaire relevant to this study. Our questionnaire collects information about the personality traits and application preferences of people. We expect that the information collected from questionnaire will allow us to analyse the correlation between users personality traits and their application preferences and interests. User's application preferences and interests will be inferred from the Internet browsing history and running app information obtained from the MACACO App.

The data collection and on-the-phone storage of MACACOapp is designed in accordance with the state-of-the-art best practices in application development. The data collected on the phone is encrypted and inaccessible by any other application installed on the same phone or to any other third party, even in case your phone gets lost or stolen. Moreover, any user's identity information available in the collected data or in the questionnaire will be completely and irreversibly anonymised before its transfer to the central servers. The on-the-phone collected data and questionnaire data will be transferred via a secure transmission protocol to the central servers. This application is in phase of test.

5.1.2. RIOT

Participants: Emmanuel Baccelli, Oliver Hahm.

RIOT (http://www.riot-os.org) is a nano operating system for the Internet of Things. While requiring as low as 1,5kB of RAM and 5kB or ROM, RIOT offers real time and energy efficiency capabilities, as well as a single API (partially POSIX compliant) across heterogeneous 8-bit, 16-bit and 32-bit low-hardware. This API is developer-friendly in that it enables multi-threading, standard C and C++ application programming and the use of standard debugging tools (which was not possible so far for embedded programming). On top of this, RIOT includes several network stacks, such as a standard IPv6/6LoWPAN stack and a information-centric network stack (based on CCN).

RIOT is developed by an international community of open-source developers that was co-founded by Inria and Freie Universitaet Berlin. The goal of RIOT is to provide a powerful, free, open-source IoT software platform that can be used like Linux is for less constrained machines, both (i) in the context of research and/or teaching, as well as (ii) in industrial contexts.

5.1.3. DragonNet

Participants: Cédric Adjih, Ichrak Amdouni, Hana Baccouch, Antonia Masucci.

DragonNet is a generic framework for network coding in wireless networks. It is a initially result of the GETRF project of the Hipercom2 team.

It is based on intra-flow coding where the source divides the flow in a sequence of payloads of equal size (padding may be used). The design keys of DragonNet are simplicity and universality; DragonNet does not use explicit or implicit knowledge about the topology (such as the direction or distance to the source, the loss rate of the links, ...). Hence, it is perfectly suited to the most dynamic wireless networks. The protocol is distributed and requires minimal coordination. DragonNet architecture is modular, it is based on 5 building blocks (LIB, SIG, Protocol, SEW and DRAGON). Each block is almost independent. This makes DragonNet generic and hence adaptable to many application scenarios. DragonNet derives from a prior protocol called DRAGONCAST. Indeed, DragonNet shares the same principles and theoretical overview of DRAGONCAST. It enriches DRAGONCAST by the information base and signaling required to perform broadcast in wireless networks and in wireless sensor networks in particular.

5.2. Platforms

5.2.1. *FIT IoT-LAB*

Participants: Cedric Adjih, Emmanuel Baccelli, Ichrak Amdouni.

FIT IoT-LAB is a platform built to help foster the development, tuning and experimentation of protocols and applications for the Internet of Things and wireless sensor networks. IoT-LAB provides both dedicated IoT hardware deployments, a front-end webportal and backend management software. Using these elements, IoT-LAB enables users to share access to this IoT hardware, set-up and manage experiments. Remote use, and large scale experiments on concrete IoT deployments are thus made possible.

The Infine team is now managing the IoT-LAB site currently located in Rocquencourt, and which was publically opened in November 2014. It consists of the following:

- A set of GPS repeaters are relaying the GPS signal indoor (used for time synchronization)
- 200 A8 nodes, all equipped with GPS (10 deployed outside – identifiers between 166 and 175)
- 24 M3 nodes
- 120 WSN430 nodes

This platform was developed as part of the Equipex FIT (see section 8.1.1).

AVIZ Project-Team

5. New Software and Platforms

5.1. MakerVis

Participants: Sai Ganesh Swaminathan, Shi Conglei, Yvonne Jansen, Pierre Dragicevic [correspondant], Lora Oehlberg, Jean-Daniel Fekete.
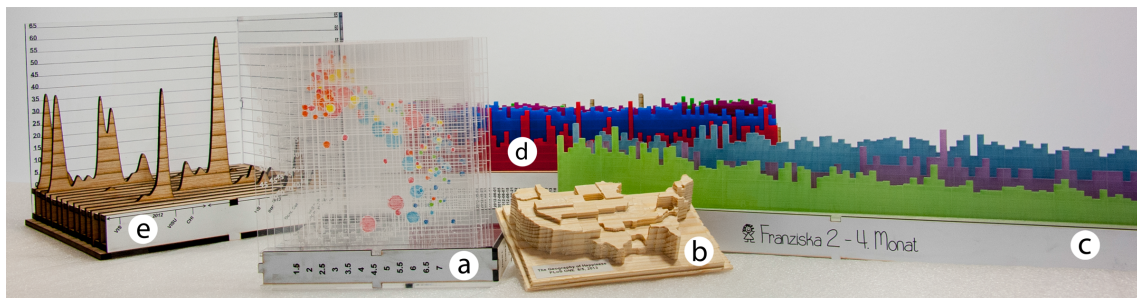

Figure 1. Physical visualizations created with our fabrication tool MakerVis: a) a scatterplot created after Hans Rosling's TED talk, b) a prism map showing happiness across the US computed from Twitter sentiments, c),d),e) visualizations created by users during design sessions.

An increasing variety of physical visualizations are being built, for purposes ranging from art and entertainment to business analytics and scientific research. However, crafting them remains a laborious process and demands expertise in both data visualization and digital fabrication. We created the MakerVis prototype [34], the first tool that integrates the whole workflow, from data filtering to physical fabrication. The design of MakerVis tries to overcome the limitations of current workflows, that we initially analyzed through three real case studies. Design sessions with three end users shows that tools such as MakerVis can dramatically lower the barriers behind producing physical visualizations. Observations and interviews also revealed important directions for future research. These include rich support for customization, and extensive software support for materials that accounts for their unique physical properties as well as their limited supply.

More details on the Web page: www.aviz.fr/makervis

5.2. Bertifier

Participants: Charles Perin, Pierre Dragicevic, Jean-Daniel Fekete.

Bertifier [20] is a web application (available at www.bertifier.com) for rapidly creating tabular visualizations from spreadsheets. Bertifier draws from Jacques Bertin's matrix analysis method, whose goal was to "simplify without destroying" by encoding cell values visually and grouping similar rows and columns. Although there were several attempts to bring this method to computers, no implementation exists today that is both exhaustive and accessible to a large audience. Bertifier remains faithful to Bertin's method while leveraging the power of today's interactive computers. Tables are formatted and manipulated through *crossets* [36], a new interaction technique for rapidly applying operations on rows and columns. Vertifier also introduces *visual reordering*, a semi-interactive reordering approach that lets users apply and tune automatic reordering algorithms in a WYSIWYG manner. We showed in an evaluation that Bertifier has the potential to bring Bertin's method to a wider audience of both technical and non-technical users, and empower them with data analysis and communication tools that were so far only accessible to a handful of specialists.
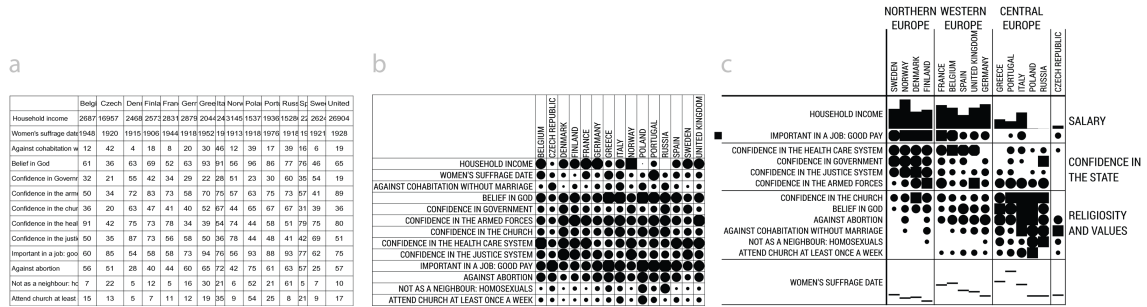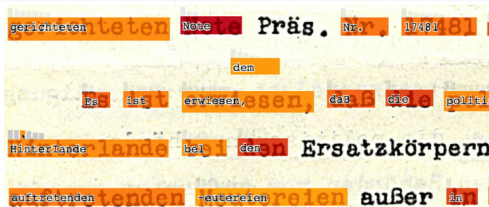
Figure 2. A spreadsheet formatted and reordered with BERTIFIER: a) the original numerical table; b) the corresponding tabular visualization; c) the final result, reordered, formatted and annotated. The final result is ready to be exported and inserted as a figure.

More details about the software are available at www.aviz.fr/bertifier

5.3. Sparklificator

Participants: Pascal Goffin, Wesley Willett, Jean-Daniel Fekete, Petra Isenberg.

Figure 3. Four examples of the integration of word-scale visualizations into HTML documents

Sparklificator [17] is a general open-source jQuery library that eases the process of integrating word-scale visualizations into HTML documents. It provides a range of options for adjusting the position (on top, to the right, as an overlay), size, and spacing of visualizations within the text. The library includes default visualizations, including small line and bar charts, and can also be used to integrate custom word-scale visualizations created using web-based visualization toolkits such as D3.

More on the project Web page: www.aviz.fr/sparklificator

5.4. GraphDiaries

Participants: Benjamin Bach [correspondant], Emmanuel Pietriga, Jean-Daniel Fekete.
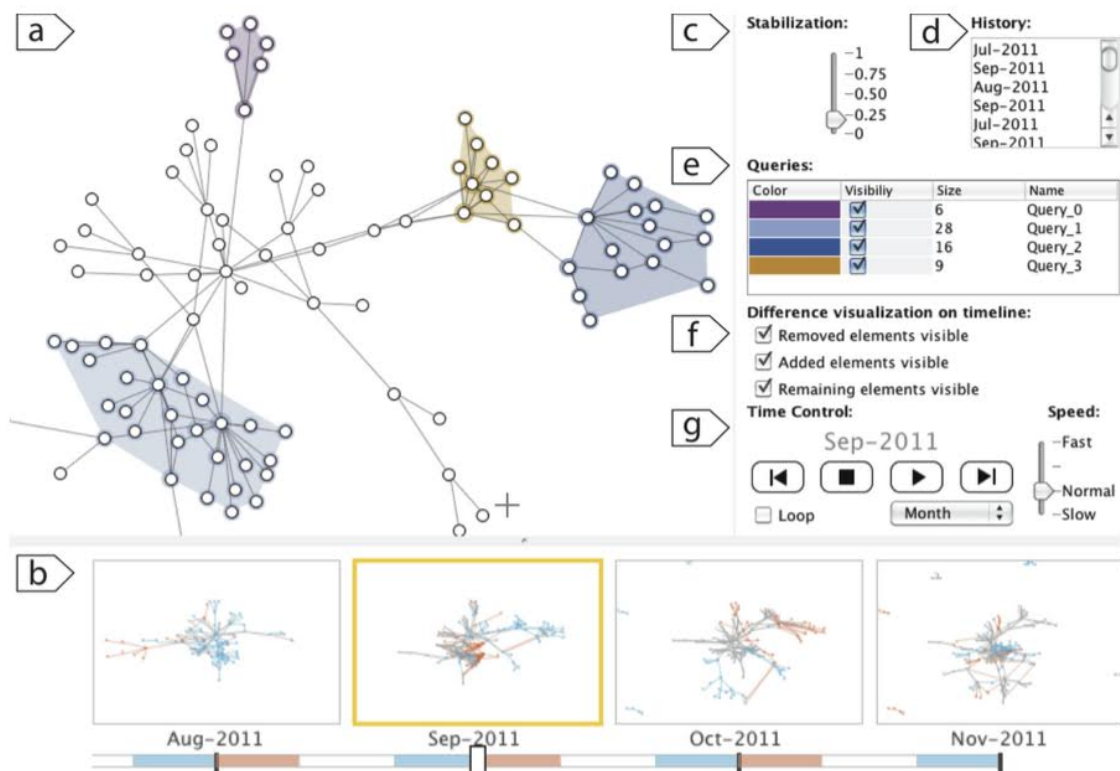
Figure 4. GraphDiaries interface: a) Network view, b) Time- line, c) Layout stabilization slider, d) Navigation history, e) Node queries, f) Panel to change visibility of red, blue or gray elements in the Timeline, g) Animation playback panel.

Identifying, tracking and understanding changes in networks that change over time, such as social networks, brain connectivy or migration flows, are complex and cognitively demanding tasks. To better understand the tasks related to the exploration of these networks, we introduced a task taxonomy which informed the design of GraphDiaries, [13], a new visual interface (Figure 4) designed to improve support for these tasks. GraphDiaries relies on animated transitions that highlight changes in the network between time steps, thus helping users identify and understand changes. GraphDiaries features interaction techniques to quickly navigate between individual time steps of the network. We conducted on a user study, based on representative tasks identified through the taxonomy, that compares GraphDiaries to existing techniques for temporal navigation in dynamic networks, showing that it outperforms them both in terms of task time and errors for several of these tasks.

5.5. Cubix

Participants: Benjamin Bach [correspondant], Emmanuel Pietriga, Jean-Daniel Fekete.
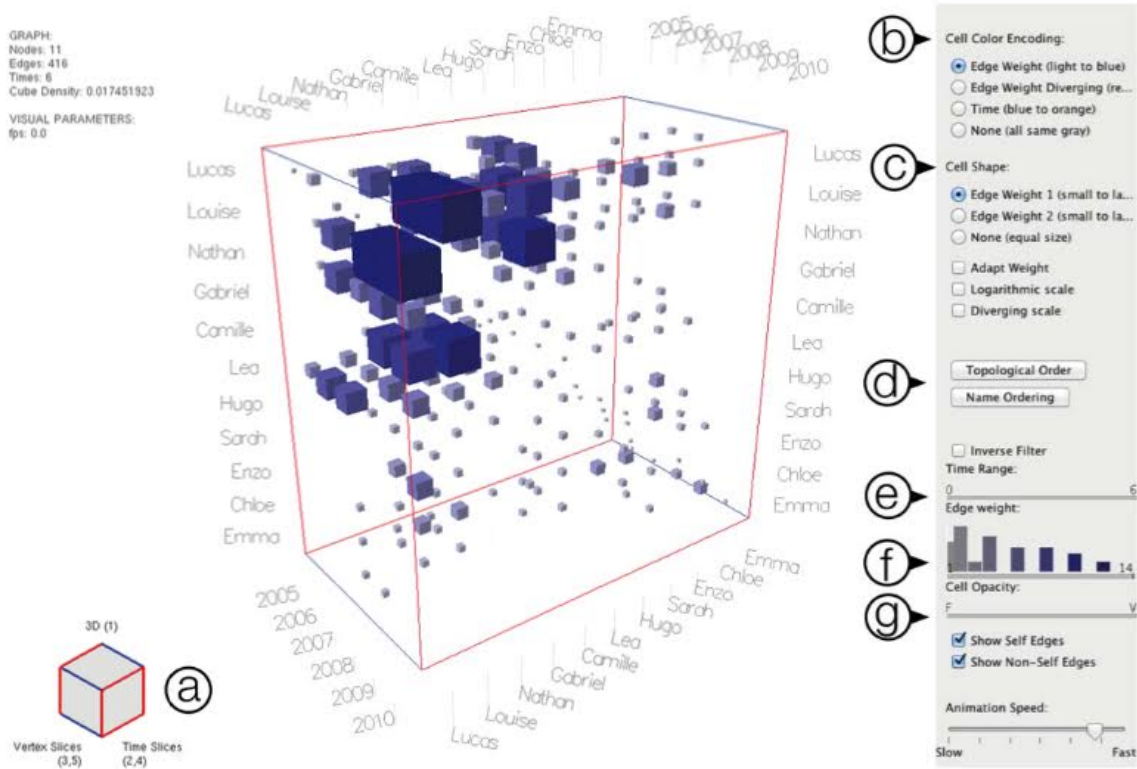

Figure 5. Cubix UI screenshot. a) Cubelet Widget, b) Cell color encod ing, c) Cell shape encoding, d) Vertex ordering, e) Time range slider, f) Cell weight filter with histogram indicating edge weight distribution, g) Cell opacity.

Designing visualizations of dynamic networks is challenging, both because the data sets tend to be complex and because the tasks associated with them are often cognitively demanding. Different tasks may require different visualizations and visual mappings, but combined in a simple interface. We developed Cubix [23] (Figure 5), a software featuring a novel visual representation and navigation model for dynamic networks,

inspired by the way people comprehend and manipulate physical cubes. Users can change their perspective on the data by rotating or decomposing the 3D cube. These manipulations can produce a range of different 2D visualizations that emphasize specific aspects of the dynamic network suited to particular analysis tasks. A range of interactions can be performed on dynamic networks using the Cubix system. We showed how two domain experts, an astronomer and a neurologist, successfully used Cubix to explore and report on their own network data.

More on the project Web page: www.aviz.fr/cubix

5.6. EditorsNotes

Participants: Jean-Daniel Fekete [correspondant], Nadia Boukhelifa, Evanthia Dimara.

Figure 6. EditorsNotes environment with its three main panes: on the left, the list of projects, in the middle the editor and related documents, on the right the visualizations showing entities appearing in the current project.

CENDARI Is a European Infrastructure project funded by the EU for 4 years: 2012-2016. Aviz is in charge of the Human-Computer Interface for the project, and develops a tool to allow historians and archivists to take notes, enter them online, manage their images in relations with the notes and documents, and visualize the entities they find in the documents and notes. This system is an extension of the original EditorsNotes project, integrating several innovative components asked by the historians: visualizations, relations with the Semantic Web, and a management of access rights respecting the researchers' desire of privacy for their notes, as well as desire of sharing entities and relations throught the notes and documents.

More on the project Web page: www.aviz.fr/Research/CENDARI

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

5. New Software and Platforms

5.1. New Software

5.1.1. The Webdamlog system

The Webdamlog system is a distributed knowledge management system. A new version of the system has been developed in collaboration with Drexel University (Prof. Julia Stoyanovich). The new version includes access control.

5. New Software and Platforms

5.1. WILDER Platform

Participants: Michel Beaudouin-Lafon [correspondant], Olivier Chapuis, Cédric Fleury, Olivier Gladin, Rémi Hellequin, Stéphane Huot, Amani Kooli, Monireh Sanaei, Gabriel Tezier, Jonathan Thorpe.

WILDER is InSitu's second experimental ultra-high-resolution interactive environment, following up on the WILD platform developed since 2009 [2] (Figure 1). It features a wall-sized display with seventy-five 20" LCD screens, i.e. a 5m50 x 1m80 (18' x 6') wall displaying 14 400 x 4 800 = 69 million pixels, powered by a 10-computer cluster and two front-end computers. The platform also features a camera-based motion tracking system supporting interaction with the wall as well as within the surrounding space, a multitouch frame making the entire wall-sized display touch sensitive and various mobile devices. WILDER is part of the DIGISCOPE Equipment of Excellence and, in combination with WILD and the other DIGISCOPE rooms, provides a unique experimental environment for collaborative interaction. In addition to using WILD and WILDER for our research, we have also developed software architectures and toolkits that enable developers to run applications on such multi-device, cluster-based systems.
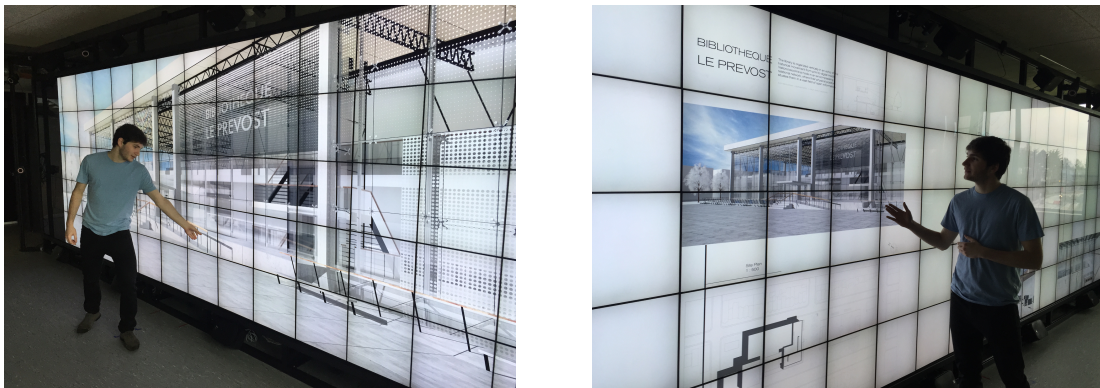

Figure 1. The WILDER platform.

5.2. Smarties

Participants: Olivier Chapuis [correspondant], Anastasia Bezerianos, Bruno Fruchard.

The Smarties system [16] provides an easy way to add mobile interactive support to collaborative applications for wall displays.

It consists of (i) a mobile interface that runs on mobile devices for input, (ii) a communication protocol between the mobiles and the wall application, and (iii) libraries that implement the protocol and handle synchronization, locking and input conflicts. The library presents the input as an event loop with callback functions and handles all communication between mobiles and wall application. Developers can customize the mobile interface from the wall application without modifying the mobile interface code.

On each mobile we find a set of cursor controllers associated with keyboards, widgets and clipboards. These controllers (pucks) can be shared by multiple collaborating users. They can control simple cursors on the wall application, or specific content (objects or groups of them). The developer can decide the types of widgets associated to pucks from the wall application side.

Smarties mobile clients currently run on Android, while server libraries have been developed in C++ and Java.

Figure 2. Left: Multiple Lenses, starting from the left a magnification lens, a DragMag and a fisheye. Right: two synchronized Smarties clients running on tablets. The four colored pucks are attached respectively to a magnification lens (left of wall), the anchor and lens of a DragMag (middle) and a fisheye (right). The active puck is the blue for the device on top, and green for the bottom. The described widgets added by the application are seen on the widget area.

Smarties is available at the http://smarties.lri.fr/ under a GNU GPL licence.

- ACM: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)
- Software benefit: See O. Chapuis, A. Bezerianos, S. Franzeskakis (2014) Smarties: An Input System for Wall Display Development. In CHI '14: Proceedings of the 32nd international conference on Human factors in computing systems. ACM, pages 225-234.
- OS/Middleware: Crossplatform
- Required library or software: none
- Programming language: C++, Java

5.3. WildOS

Participant: Michel Beaudouin-Lafon [correspondant].

WildOS is middleware to support applications running in an interactive room featuring various interaction resources, such as our WILD and WILDER rooms: a tiled wall display, a motion tracking system, tablets and smartphones, etc. The conceptual model of WildOS is a *platform*, such as the WILD or WILDER room, described as a set of devices and on which one or more applications can be run.

WildOS consists of a server running on a machine that has network access to all the machines involved in the platform, and a set of clients running on the various interaction resources, such as a display cluster or a tablet. Once *WildOS* is running, applications can be started and stopped and devices can be added to / removed from the platform.

WildOS relies on Web technologies, most notably Javascript and node.js, as well as node-webkit and HTML5. This makes it inherently portable (it is currently tested on Mac OS X and Linux). While applications can be developed only with these Web technologies, it is also possible to bridge to existing applications developed in other environments if they provide sufficient access to be remote controlled.

WildOS is used in several InSitu projects, and is also deployed on several of Google's interactive rooms in Mountain View, Dublin and Paris. It is available under on Open Source licence at https://bitbucket.org/mblinsitu/wildos.

- ACM: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)
- Software benefit: helps the development of multisurface applications.
- OS/Middleware: Crossplatform
- Required library or software: node.js, node-webkit
- Programming language: Javascript

5.4. GlideCursor

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

GlideCursor is a Mac OS X application that implements the inertial cursor described in [15]. The current version only works when moving the cursor with a trackpad. The application lets users configure gliding, and can also log cursor activity for later analyses.

GlideCursor is available under on Open Source licence at https://bitbucket.org.

- ACM: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)
- Software benefit: can improve cursor pointing on large displays.
- OS/Middleware: Mac OS X
- Required library or software: none
- Programming language: Objective-C

<h1 style="text-align:center; color:red">OAK Project-Team</h1>

5. New Software and Platforms

5.1. Amada

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

Contact: Jesús Camacho-Rodríguez (jcamachor[at]gmail.com))

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. PAXQuery

Name: PAXQuery (https://team.inria.fr/oak/projects/paxquery/)

Contact: Jesús Camacho-Rodríguez (jcamachor[at]gmail.com))

Other contacts: Ioana Manolescu (ioana.manolescu[at]inria.fr), Dario Colazzo (dario.colazzo[at]dauphine.fr), Juan Alvaro M. Naranjo (juan-alvaro.munoz-naranjo[at]inria.fr)

Presentation: A system for the massively parallel processing of XQuery queries, developed as an extension of the Apache Flink system (http://flink.apache.org/)

5.3. CliqueSquare

Name: CliqueSquare (https://team.inria.fr/oak/projects/cliquesquare/)

Contact: Stamatis Zampetakis (stamatis.zampetakis[at]inria.fr))

Other contacts: Ioana Manolescu (ioana.manolescu[at]inria.fr), François Goasdoué (fg[at]irisa.fr), Benjamin Djahandideh (benjamin.djahandideh[at]inria.fr)

Presentation: A system for the massively parallel evaluation of conjunctive SPARQL queries, built on top of Hadoop. The system has been released in open-source: https://sourceforge.net/projects/cliquesquare/.

5.4. FactMinder

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

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

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

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

5.5. 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.

5.6. PigReuse

Name: PigReuse

Contact: Jesús Camacho-Rodríguez (jcamachor[at]gmail.com)

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

Presentation: A PigLatin optimization tool based on identifying and sharing repeated subexpressions.

5.7. WARG

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

Contact: Alexandra Roatiş (alexandra.roatis[at]gmail.com)

Other contacts: Ioana Manolescu (ioana.manolescu[at]inria.fr), Sejla Cebiric (sejla.cebiric[at]inria.fr), François Goasdoué (fg[at]irisa.fr)

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