RESEARCH CENTER
Nancy - Grand Est

FIELD

Activity Report 2014

Section Software

Edition: 2015-03-24

ALGORILLE Project-Team

5. New Software and Platforms

5.1. Introduction

Software is a central part of our output. In the following we present the main tools to which we contribute. We use the Inria software self-assessment catalog for a classification.

5.2. Implementing parallel models

Several software platforms have served us to implement and promote our ideas in the domain of coarse grained computation and application structuring.

5.2.1. ORWL and P99

Participants: Jens Gustedt, Stéphane Vialle [External collaborator, SUPELEC], Mariem Saied.

ORWL is a reference implementation of the Ordered Read-Write Lock tools as described in [4]. The macro definitions and tools for programming in C99 that have been implemented for ORWL have been separated out into a toolbox called P99. ORWL is intended to become opensource, once it will be in a publishable state. P99 is available under a QPL at http://p99.gforge.inria.fr/.
Software classification: A-3-up, SO-4, SM-3, EM-3, SDL (P99: 4, ORWL: 2-up), DA-4, CD-4, MS-3, TPM-4

5.2.2. parXXL

Participants: Jens Gustedt, Stéphane Vialle [External collaborator, SUPELEC].

ParXXL is a library for large scale computation and communication that executes fine grained algorithms on coarse grained architectures (clusters, grids, mainframes). It has been one of the software bases of the InterCell project and has been proven to be a stable support, there. It is available under a GPLv2 at http://parxxl.gforge.inria.fr/. ParXXL is not under active development anymore, but still maintained in the case of bugs or portability problems.
Software classification: A-3, SO-4, SM-3, EM-2, SDL-4, DA-4, CD-4, MS-2, TPM-2

5.2.3. musl

Participant: Jens Gustedt.

musl is a re-implementation of the C library as it is described by the C and POSIX standards. It is *lightweight*, *fast*, *simple*, *free*, and strives to be correct in the sense of standards-conformance and safety. Musl is production quality code that is mainly used in the area of embedded device. It gains more market share also in other area, *e.g.* there are now Linux distributions that are based on musl instead of Gnu LibC.

In 2014, we have added an implementation of the new thread interface that had been defined in the recent C11 standard.

5.3. Parallel developments for numerical scientific application

Participant: Sylvain Contassot-Vivier.

The RAD2D/RAD3D software are co-developed with Fatmir Asllanaj, full researcher in physics at the LEMTA Laboratory, in the context of an inter-disciplinary collaboration. The object of those software is to solve and compute the radiative-transfer equation by using the finite volume method. As the amount of computations induced is very large, the resort to parallelism is mandatory [9], [15]. By its complexity and similarity with a large proportion of scientific applications, this real case application is a fully pertinent test-case for the parallel techniques and schemes we have designed in our team. Those software are not open-source and, by the way, are still in development state.

5.4. Distem

Participants: Tomasz Buchert, Emmanuel Jeanvoine, Lucas Nussbaum, Luc Sarzyniec.

Wrekavoc and Distem are distributed system emulators. They enable researchers to evaluate unmodified distributed applications on heterogeneous distributed platforms created from an homogeneous cluster: CPU performance and network characteristics are altered by the emulator.
Wrekavoc was developed until 2010, and we then focused our efforts on **Distem**, that shares the same goals with a different design. Distem is available from http://distem.gforge.inria.fr/ under GPLv3.
Software classification: A-3-up, SO-4, SM-3-up, EM-3, SDL-4, DA-4, CD-4, MS-4, TPM-4.

5.5. SimGrid

SimGrid is a toolkit for the simulation of distributed applications in heterogeneous distributed environments. The specific goal of the project is to facilitate research in the area of parallel and distributed large scale systems, such as grids, P2P systems and clouds. Its use cases encompass heuristic evaluation, application prototyping or even real application development and tuning.

5.5.1. Core distribution

Participants: Martin Quinson, Marion Guthmuller, Paul Bédaride, Gabriel Corona, Lucas Nussbaum.

SimGrid has an active user community of more than one hundred members, and is available under GPLv3 from http://simgrid.gforge.inria.fr/. One third of the source code is devoted to about 12000 unit tests and 500 full integration tests. These tests are run for each commit for 4 package configurations and on 4 operating systems thanks to the Inria continuous integration platform.
Software classification: A-5, SO-4, SM-4, EM-4, SDL-5, DA-4, CD-4, MS-4, TPM-4.

5.5.2. SimGridMC

Participants: Martin Quinson, Marion Guthmuller, Gabriel Corona.

SimGridMC is a module of SimGrid that can be used to formally assess any distributed system that can be simulated within SimGrid. It explores all possible message interleavings searching for states violating the provided properties. We recently added the ability to assess liveness properties over arbitrary C codes, thanks to a system-level introspection tool that provides a finely detailed view of the running application to the model checker. This can for example be leveraged to verify arbitrary MPI code written in C.
Software classification: A-3-up, SO-4, SM-3-up, EM-3-up, SDL-5, DA-4, CD-4, MS-4, TPM-4.

5.5.3. SCHIaaS

Participants: Julien Gossa [External collaborator, SUPELEC], Stéphane Genaud [External collaborator, SUPELEC], Rajni Aron.

The *Simulation of Clouds, Hypervisor and IaaS* (SCHIaaS) is an extension of SimGrid that can be used to comprehensively simulate clouds, from the hypervisor/system level, to the IaaS/administrator level. The hypervisor level includes models about virtualization overhead and VMs operations like boot, start, suspend, migrate, and network capping. The IaaS level includes models about instances management like image storage and deployment and VM scheduling. This extension allows to fully simulate any cloud infrastructure, whatever the hypervisor or the IaaS manager. This can be used by both cloud administrators to dimension and tune clouds, and cloud users to simulate cloud applications and assess provisioning strategies in term of performances and cost.
Software classification: A-3-up, SO-3, SM-2-up, EM-2-up, SDL-2, DA-4, CD-4, MS-4, TPM-4.

5.6. Kadeploy

Participants: Luc Sarzyniec, Stéphane Martin, Emmanuel Jeanvoine, Lucas Nussbaum [correspondant].

Kadeploy is a scalable, efficient and reliable deployment (provisioning) system for clusters and grids. It provides a set of tools for cloning, configuring (post installation) and managing cluster nodes. It can deploy a 300-nodes cluster in a few minutes, without intervention from the system administrator. It plays a key role on the Grid'5000 testbed, where it allows users to reconfigure the software environment on the nodes, and is also used on a dozen of production clusters both inside and outside INRIA. It is available from http://kadeploy3.gforge.inria.fr/ under the Cecill license.
Software classification: A-4-up, SO-3, SM-4, EM-4, SDL-4-up, DA-4, CD-4, MS-4, TPM-4.

5.7. XPFlow

Participants: Tomasz Buchert, Lucas Nussbaum [correspondant].

XPFlow is an implementation of a new, workflow-inspired approach to control experiments involving large-scale computer installations. Such systems pose many difficult problems to researchers due to their complexity, their numerous constituents and scalability problems. The main idea of the approach consists in describing the experiment as a workflow and execute it using achievements of Business Process Management (BPM), workflow management techniques and scientific workflows. The website of XPFlow is http://xpflow.gforge.inria.fr/. XPFlow was featured in a tutorial during Grid'5000 Spring School 2014.
Software classification: A-2-up, SO-3-up, SM-2-up, EM-3-up, SDL-2-up, DA-4, CD-4, MS-4, TPM-4.

5.8. Grid'5000 testbed

Participants: Luc Sarzyniec, Jérémie Gaidamour, Arthur Garnier, Clément Parisot, Emmanuel Jeanvoine, Émile Morel, Lucas Nussbaum [correspondant].

Grid'5000 (http://www.grid5000.fr) is a scientific instrument designed to support experiment-driven research in all areas of computer science related to parallel, large-scale or distributed computing and networking. It gathers 10 sites, 25 clusters, 1200 nodes, for a total of 8000 cores. It provides its users with a fully reconfigurable environment (bare metal OS deployment with Kadeploy, network isolation with KaVLAN) and a strong focus on enabling high-quality, reproducible experiments.
The AlGorille team contributes to the design of Grid'5000, to the administration of the local Grid'5000 site in Nancy, and to the design and development of Kadeploy (in close cooperation with the Grid'5000 technical team). The AlGorille engineers also administer *Inria Nancy – Grand Est*'s local production cluster, named *Talc*, leveraging the experience and tools from Grid'5000.

Software classification: A-5, SO-4, SM-4, EM-4, SDL-N/A, DA-4, CD-4, MS-4, TPM-4.

ALICE Project-Team

5. New Software and Platforms

5.1. Vorpaline

Participants: Dobrina Boltcheva, Bruno Lévy, Thierry Valentin.

Vorpaline is an automatic surfacic and volumetric mesh generation software, distributed with a commercial license. Vorpaline is based on the main scientific results stemming from projects GoodShape and VORPALINE, funded by the European Research Council, about optimal quantization, centroidal Voronoi diagrams and fast/parallel computation of Voronoi diagrams in high-dimension space. The current version provides functionalities such as isotropic/adaptive/anisotropic surface re-meshing, tolerant surface re-meshing, mesh repair and mesh decimation, constrained surface meshing, quad-dominant surface meshing and hex-dominant volume meshing. It is extensively tested on industrial data with a continuous integration platform, and extensively documented. It is now proposed (since 2014) to the sponsors of the Gocad consortium, as an extension package of the Gocad software.

5.2. IceSL

Participants: Jérémie Dumas, Jean Hergel, Sylvain Lefebvre, Frédéric Claux, Jonas Martinez-Bayona, Samuel Hornus.

In the new software IceSL, we propose to exploit recent advances in GPU and Computer Graphics to accelerate the slicing process of objects modelled via a CSG [0] language. Our target are open source low cost *fused deposition modeling* printers such as RepRaps.

Our approach first inputs a CSG description of a scene which can be composed of both meshes and analytic primitives. During display and slicing the CSG model is converted on the fly into an intermediate representation enabling fast processing on the GPU. Slices can be quickly extracted, and the tool path is prepared through image erosion. The interactive preview of the final geometry uses the exact same code path as the slicer, providing an immediate, accurate visual feedback.

IceSL is the recipient software for our ERC research project "ShapeForge", led by Sylvain Lefebvre.

Figure 1. Left. A two-colored vase is modeled in IceSL. Right. An early printed result.

[0] Constructive Solid Geometry

5.3. Graphite

Participants: Dobrina Boltcheva, Samuel Hornus, Bruno Lévy, David Lopez, Jeanne Pellerin, Nicolas Ray.

Graphite is a research platform for computer graphics, 3D modeling and numerical geometry. It comprises all the main research results of our “geometry processing” group. Data structures for cellular complexes, parameterization, multi-resolution analysis and numerical optimization are the main features of the software. Graphite is publicly available since October 2003, and is hosted by Inria GForge since September 2008. Graphite is one of the common software platforms used in the frame of the European Network of Excellence AIMShape .

Graphite and its research-plugins are actively developed and extended. The latest version was released on January 2nd, 2014 and has been downloaded 732 times as of Sept 5.

5.4. GraphiteLifeExplorer

Participant: Samuel Hornus.

GLE is a 3D modeler, developed as a plugin of Graphite, dedicated to molecular biology. It is developed in cooperation with the Fourmentin Guilbert foundation and has recently been renamed "GraphiteLifeExplorer". Biologists need simple spatial modeling tools to help in understanding the role of the relative position of objects in the functioning of the cell. In this context, we develop a tool for easy DNA modeling. The tool generates DNA along any user-given curve, open or closed, allows fine-tuning of atoms position and, most importantly, exports to PDB (the Protein Daba Bank file format).

The development of GLE is currently on hold, but it is still downloaded (freely) about twice a day (1600 downloads to date).

5.5. OpenNL - Open Numerical Library

Participants: Bruno Lévy, Nicolas Ray, Rhaleb Zayer.

OpenNL is a standalone library for numerical optimization, especially well-suited to mesh processing. The API is inspired by the graphics API OpenGL, this makes the learning curve easy for computer graphics practitioners. The included demo program implements our LSCM [24] mesh unwrapping method. It was integrated in Blender by Brecht Van Lommel and others to create automatic texture mapping methods. OpenNL is extended with two specialized modules :

- CGAL parameterization package: this software library, developed in cooperation with Pierre Alliez and Laurent Saboret, is a CGAL package for mesh parameterization.
- Concurrent Number Cruncher: this software library extends OpenNL with parallel computing on the GPU, implemented using the CUDA API.

5.6. GEOGRAM

Participant: Bruno Lévy.

GEOGRAM is a software library with geometrical algorithms. The focus is put on the ease of use, minimal memory consumption, minimal size of the code and extensively documented algorithms (whereas in existing libraries such as CGAL, the focus is put on the extensibility). GEOGRAM includes the PCK (Predicate Construction Kit), a system to automatically generate robust predicates from their equation. It provides a standalone exact number type, based on Shewchuk’s expansion arithmetics. The library is portable under Linux, Windows, MacOS, Android, and any system that has IEEE floating point arithmetics. The arithmetic kernel may be used by other programming library and proposed as extension packages (e.g. for CGAL).

5.7. LibSL

Participant: Sylvain Lefebvre.

LibSL is a Simple library for graphics. Sylvain Lefebvre continued development of the LibSL graphics library (under CeCill-C licence, filed at the APP). LibSL is a toolbox for rapid prototyping of computer graphics algorithms, under both OpenGL, DirectX 9/10, Windows and Linux. The library is actively used in both the REVES / Inria Sophia-Antipolis Méditerrannée and the ALICE / Inria Nancy Grand-Est teams.

BIGS Project-Team

4. New Software and Platforms

4.1. Online data analysis

Participants: J.-M. Monnez

An R package performing most of the methods of factorial analysis in an online way has been developed by R. Bar and J.-M. Monnez. Starting from a simulated data flow, the main goal of the program is to perform online factorial analyses (Principal Component Analyses, Canonical Correlation Analysis, Canonical Discriminant Analysis, Correspondence Analysis). Data are supposed to be independent and identically distributed observations of a random vector (whose distribution is a priori unknown). Defining stochastic approximation processes, the procedure is adaptative in the sense that the results of the analyses are updated recursively each time that a new piece of data is taken into account.

From a theoretical point of view, the i.i.d case has been recently extended to the case of an expectation and/or covariance matrix of the random vector varying with time. We plan to include these improvements into our software.

4.2. Socio-economic index

Participants: J.-M. Monnez

A R package called SesIndexCreatoR has been written by B. Lalloué and J.-M. Monnez in order to implement our socio-economic index for health inequalities. The version 1.0 of this package is currently freely available on the website of the Equit'Area project: http://www.equitarea.org/documents/packages_1.0-0/. It contains the functions needed to run the procedure (either integrally or partially) and obtain the corresponding SES index. The user may also create categories of this index with different methods (hierarchical clustering with or without k-nearest neighbors, quantiles, or intervals) and generate automatic reports of the results. Visualization and plotting functions are provided in the package.

4.3. Angio-Analytics

Participants: T. Bastogne

A software *Angio-Analytics* has been developed by J.-B. Tylcz, E. Djermoune and T. Bastogne. This tool allows the pharmacodynamic characterization of anti-vascular effects in anti-cancer treatments. It uses time series of *in vivo* images provided by intra-vital microscopy. Such *in vivo* images are obtained owing to skinfold chambers placed on mice skin, as illustrated in Fig. 1 . The automatized analysis is split up into two steps that were completely performed separately and manually before. The first steps corresponds to image processing to identify characteristics of the vascular network, as illustrated in Fig. 2 . The last step is the system identification of the pharmacodynamic response and the statistical analysis of the model parameters as shown in Fig. 3 and Fig. 4 . An article has been submitted to a journal (Biomedical Signal Processing and Control) and is currently in revision process. Moreover, the current version of the software has been registered to the *Agence de Protection des Programmes*.

4.4. In silico design of nanoparticles for the treatment of cancers by enhanced radiotherapy

Participants: T. Bastogne

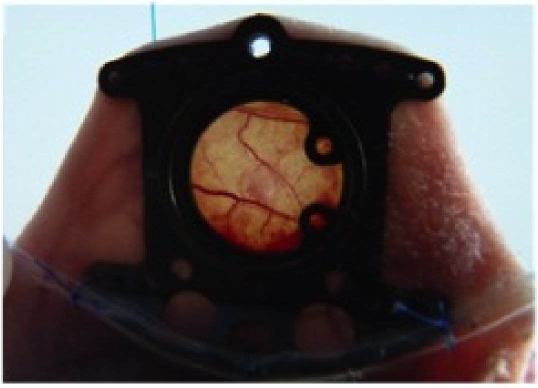


Figure 1. Example of a skinfold chamber placed on a mouse skin

More than eight million people die from cancer worldwide each year. Current treatment such as chemotherapy and radiotherapy are still limited in terms of benefit/risk ratio. Nevertheless, engineered nanoparticles have opened new interesting perspectives in cancerology, as emphasized by Brigger et al. since 2002. One of these promising solutions is based on the development of nanoparticles able to enhance the cytotoxic effect of radiotherapy. Nevertheless, the preclinical development in nano-medicine is slow, risky and expensive. Recently, Etheridge et al. (2013) highlighted the fact that many of the revolutionary nano-medicine technologies anticipated in the literature may be 20 or more years from clinical use. To speed up the preclinical development of medical engineered nanomaterials, we have designed an integrated computing platform dedicated to the virtual screening of nanostructured materials activated by X-ray making it possible to select nano-objects presenting interesting medical properties faster. That innovation gathers stochastic simulations and statistical modeling to estimate the impact of each design parameter describing the nano-object. That allows us to optimize composition factors in order to suggest one or few promising architectures regarding the medical purpose. The main advantage of this *in silico* design approach is to virtually screen a lot of possible formulations and to rapidly select the most promising ones. The platform can currently handle the accelerated design of radiation therapy enhancing nanoparticles and medical imaging nano-sized contrast agents as well as the comparison between nano-objects and the optimization of existing materials. Other applications related to nano-medicines will be subject to further developments (e.g., photodynamic therapy). That contribution has received the best innovation award from the Institut Mines-Telecom in 2014 and application results will be presented at the 36th PAMM-EORTC Winter Meeting in January 2015.

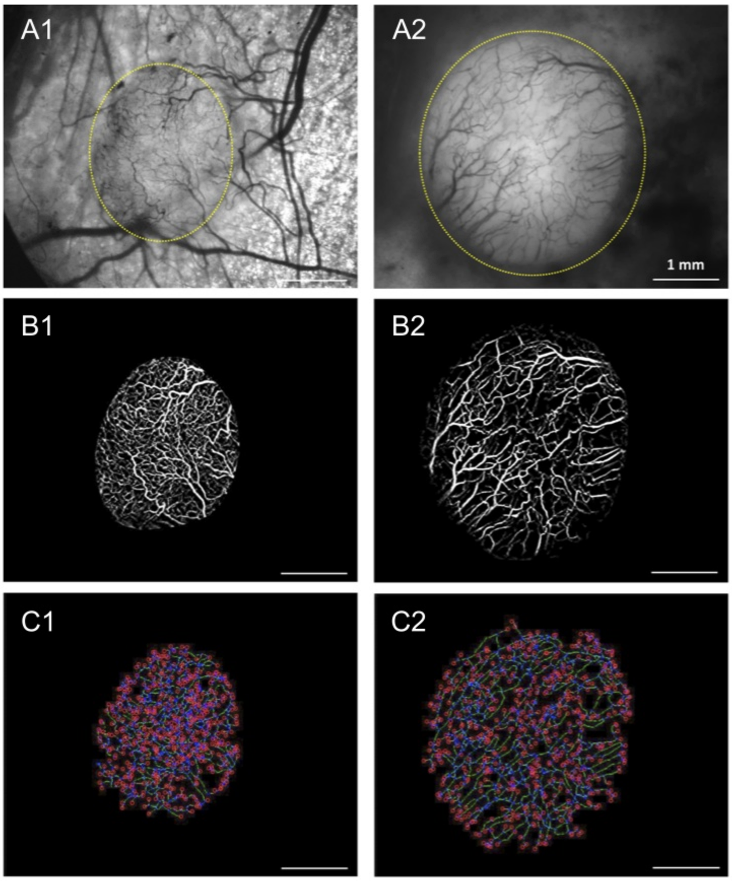


Figure 2. Example of segmentation process on a control (left) and treated tumor (right) at day -7: manual segmentation (ROI) of cancerous tissues is done in yellow on step A, vessel segmentation is performed on step B (vessels are in white), step C presents the quantification (blue and red circles) on the skeletonized vascular network (green lines)

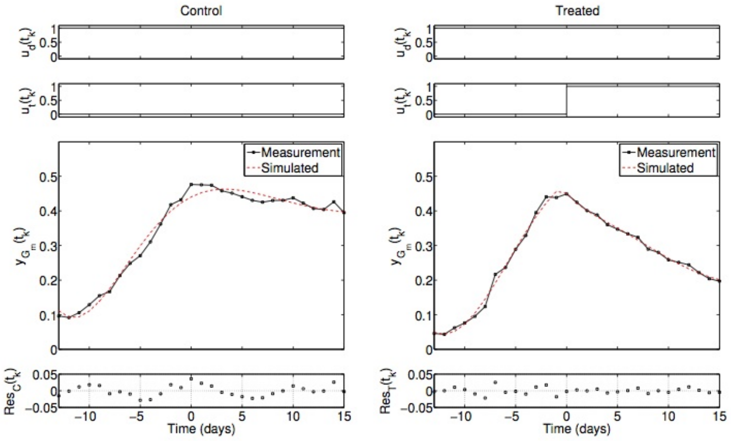
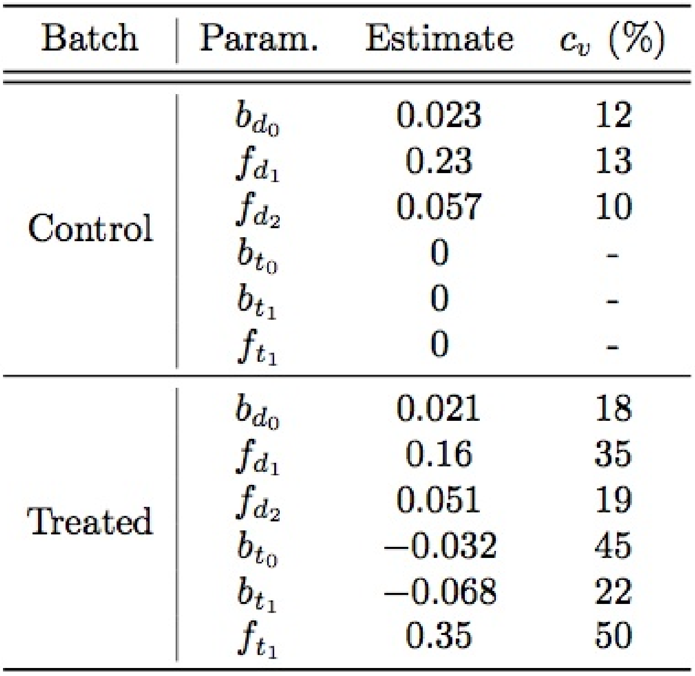


Figure 3. Measurements and estimated outputs for control and treated batches. Input signals and residuals are respectively plotted above and below



Batch	Param.	Estimate	c_v (%)
Control	b_{d_0}	0.023	12
	f_{d_1}	0.23	13
	f_{d_2}	0.057	10
	b_{t_0}	0	-
	b_{t_1}	0	-
	f_{t_1}	0	-
Treated	b_{d_0}	0.021	18
	f_{d_1}	0.16	35
	f_{d_2}	0.051	19
	b_{t_0}	-0.032	45
	b_{t_1}	-0.068	22
	f_{t_1}	0.35	50

Figure 4. Parameter estimates and coefficients of variation c_v for control and treated batches

CAMUS Team

5. New Software and Platforms

5.1. PolyLib

Participant: Vincent Loechner.

PolyLib [0] is a C library of polyhedral functions, that can manipulate unions of rational polyhedra of any dimension. It was the first to provide an implementation of the computation of parametric vertices of a parametric polyhedron, and the computation of an Ehrhart polynomial (expressing the number of integer points contained in a parametric polytope) based on an interpolation method. Vincent Loechner is the maintainer of this software. It is distributed under GNU General Public License version 3 or later.

Apart from normal maintenance, it was parallelized using OpenMP with the support of Master student Adilla Susungi, funded by the ICPS team (ICube laboratory, University of Strasbourg).

5.2. APOLLO software and LLVM

Participants: Aravind Sukumaran-Rajam, Juan Manuel Martinez Caamaño, Willy Wolff, Luis Esteban Campostrini, Matías Perez, Alexandra Jimborean, Philippe Clauss.

We are developing a framework called APOLLO (Automatic speculative POLyhedral Loop Optimizer) whose main concepts are based on our previous framework VMAD. However, several important implementation issues are now handled differently in order to improve the performance and usability of the framework, and also to open its evolution to new interesting perspectives. APOLLO is dedicated to automatic, dynamic and speculative parallelization of loop nests that cannot be handled efficiently at compile-time. It is composed of a static part consisting of specific passes in the LLVM compiler suite, plus a modified Clang frontend, and a dynamic part consisting of a runtime system. Its last extensions are presented in subsection 6.2 .

5.3. IBB source-to-source xfor compiler

Participants: Imen Fassi, Philippe Clauss, Cédric Bastoul.

Imen Fassi has developped a source-to-source xfor compiler called IBB (Iterate-But-Better) which is automatically translating any C source code containing xfor-loops into an equivalent source code where xfor-loops have been transformed into equivalent for-loops. The polyhedral code generator CLooG [27] is used to generate the corresponding code. The IBB compiler has been improved in some aspects in 2014: loop bounds can now be min and max functions, IBB uses the OpenScop format to encode statements and iteration domains.

The xfor structure is also currently incorporated in the polyhedral parser Clan [0], opening the door of xfor usage to polyhedral compilation tools. Additionally, an xfor programming wizard is currently being developed, providing automatic dependence analysis and code verification to users, thanks to the dependence analyzer Candl [0].

5.4. CLooG

Participant: Cédric Bastoul.

[0] http://icps.u-strasbg.fr/PolyLib
[0] http://icps.u-strasbg.fr/~bastoul/development/clan
[0] http://icps.u-strasbg.fr/~bastoul/development/candl

CLooG [0] is a free software and library to generate code (or an abstract syntax tree of a code) for scanning Z-polyhedra. That is, it finds a code (e.g. in C, FORTRAN...) that reaches each integral point of one or more parameterized polyhedra. CLooG has been originally written to solve the code generation problem for optimizing compilers based on the polyhedral model. Nevertheless it is used now in various area e.g. to build control automata for high-level synthesis or to find the best polynomial approximation of a function. CLooG may help in any situation where scanning polyhedra matters. While the user has full control on generated code quality, CLooG is designed to avoid control overhead and to produce a very effective code. CLooG is widely used (including by GCC and LLVM compilers), disseminated (it is installed by default by the main Linux distributions) and considered as the state of the art in polyhedral code generation.

5.5. OpenScop

Participant: Cédric Bastoul.

OpenScop [0] is an open specification that defines a file format and a set of data structures to represent a static control part (SCoP for short), i.e., a program part that can be represented in the polyhedral model. The goal of OpenScop is to provide a common interface to the different polyhedral compilation tools in order to simplify their interaction. To help the tool developers to adopt this specification, OpenScop comes with an example library (under 3-clause BSD license) that provides an implementation of the most important functionalities necessary to work with OpenScop.

5.6. Clan

Participants: Cédric Bastoul, Imen Fassi.

Clan [0] is a free software and library which translates some particular parts of high level programs written in C, C++, C# or Java into a polyhedral representation called OpenScop. This representation may be manipulated by other tools to, e.g., achieve complex analyses or program restructurations (for optimization, parallelization or any other kind of manipulation). It has been created to avoid tedious and error-prone input file writing for polyhedral tools (such as CLooG, LeTSeE, Candl etc.). Using Clan, the user has to deal with source codes based on C grammar only (as C, C++, C# or Java). Clan is notably the frontend of the two major high-level compilers Pluto and PoCC.

5.7. Clay

Participant: Cédric Bastoul.

Clay [0] is a free software and library devoted to semi-automatic optimization using the polyhedral model. It can input a high-level program or its polyhedral representation and transform it according to a transformation script. Classic loop transformations primitives are provided. Clay is able to check for the legality of the complete sequence of transformation and to suggest corrections to the user if the original semantics is not preserved. Clay is still experimental at this report redaction time but is already used during advanced compilation labs at Paris-Sud University and is one of the foundations of our ongoing work on simplifying code manipulation by programmers.

[0] http://www.cloog.org
[0] http://icps.u-strasbg.fr/~bastoul/development/openscop
[0] http://icps.u-strasbg.fr/~bastoul/development/clan
[0] http://icps.u-strasbg.fr/~bastoul/development/clay

CARAMEL Project-Team

5. New Software and Platforms

5.1. Introduction

A major part of the research done in the CARAMEL team is published within software. On the one hand, this enables everyone to check that the algorithms we develop are really efficient in practice; on the other hand, this gives other researchers — and us of course — basic software components on which they — and we — can build other applications.

5.2. GNU MPFR

Participant: Paul Zimmermann [contact].

GNU MPFR is one of the main pieces of software developed by the CARAMEL team. Since end 2006, it has become a joint project between CARAMEL and the ARÉNAIRE project-team (now ARIC, INRIA Grenoble - Rhône-Alpes). GNU MPFR is a library for computing with arbitrary precision floating-point numbers, together with well-defined semantics, and is distributed under the LGPL license. All arithmetic operations are performed according to a rounding mode provided by the user, and all results are guaranteed correct to the last bit, according to the given rounding mode.

No new release was made in 2014. However a developers meeting was organized in January 20 to 22 in Nancy, together with the developers of GNU MPC.

5.3. GNU MPC

Participant: Paul Zimmermann [contact].

GNU MPC is a floating-point library for complex numbers, which is developed on top of the GNU MPFR library, and distributed under the LGPL license. It is co-written with Andreas Enge (LFANT project-team, INRIA Bordeaux - Sud-Ouest). A complex floating-point number is represented by $x + iy$, where x and y are real floating-point numbers, represented using the GNU MPFR library. The GNU MPC library provides correct rounding on both the real part x and the imaginary part y of any result. GNU MPC is used in particular in the TRIP celestial mechanics system developed at IMCCE (*Institut de Mécanique Céleste et de Calcul des Éphémérides*), and by the Magma and Sage computational number theory systems.

Version 1.0.2 (Fagus silvatica) was released in January, with a few bug fixes, some related to the use in our own work related to the computation of Igusa class polynomials.

5.4. Finite Fields

Participants: Pierrick Gaudry, Emmanuel Thomé [contact], Luc Sanselme.

$\mathtt{mp}\mathbb{F}_q$ is (yet another) library for computing in finite fields. The purpose of $\mathtt{mp}\mathbb{F}_q$ is not to provide a software layer for accessing finite fields determined at runtime within a computer algebra system like Magma, but rather to give a very efficient, optimized code for computing in finite fields precisely known at *compile time*. $\mathtt{mp}\mathbb{F}_q$ can adapt to finite fields of any characteristic and any extension degree. However, one of the targets being the use in cryptology, $\mathtt{mp}\mathbb{F}_q$ somehow focuses on prime fields and on fields of characteristic two.

When it was first written in 2007, $\mathtt{mp}\mathbb{F}_q$ established reference marks for fast elliptic curve cryptography: the authors improved over the fastest examples of key-sharing software in genus 1 and 2, both over binary fields and prime fields. A stream of academic works followed the idea behind $\mathtt{mp}\mathbb{F}_q$ and improved over such timings, notably by Scott, Aranha, Longa, Bos, Hisil, Costello.

The library's purpose being the *generation* of code rather than its execution, the working core of mp$\mathbb{F}_q$ consists of roughly 18,000 lines of Perl code, which generate most of the C code. mp$\mathbb{F}_q$ is distributed at http://mpfq.gforge.inria.fr/.

In 2014, mp$\mathbb{F}_q$ has undergone some sanitization work, related to embedded assembly, build system, coverage test, and processor feature support. The fact that mp$\mathbb{F}_q$ is used in CADO-NFS has played an important role in fostering these changes to the mp$\mathbb{F}_q$ code. Future plans regarding the linear algebra code in CADO-NFS are expected to rely on the arithmetic part being implemented in mp$\mathbb{F}_q$. Preliminary work in this direction has been implemented by Luc Sanselme. Preliminary code by Hamza Jeljeli and Bastien Vialla from LIRMM, Montpellier, based on RNS arithmetic (Residue Number System) is also to be integrated in this context. We therefore expect more work in this area in the coming months, eventually leading to a new release.

5.5. gf2x

Participants: Pierrick Gaudry, Emmanuel Thomé [contact], Paul Zimmermann.

GF2X is a software library for polynomial multiplication over the binary field, developed together with Richard Brent (Australian National University, Canberra, Australia). It holds state-of-the-art implementation of fast algorithms for this task, employing different algorithms in order to achieve efficiency from small to large operand sizes (Karatsuba and Toom-Cook variants, and eventually Schönhage's or Cantor's FFT-like algorithms). GF2X takes advantage of specific processor instructions (SSE, PCLMULQDQ).

The current version of GF2X is 1.1, released in May 2012 under the GNU GPL. Since 2009, GF2X can be used as an auxiliary package for the widespread software library NTL, as of version 5.5. GF2X is also packaged in the Debian Linux distribution.

In 2014, the development version of GF2X has been updated to include some minor cleanups.

An LGPL-licensed portion of GF2X is also part of the CADO-NFS software package.

5.6. CADO-NFS

Participants: Cyril Bouvier, Alain Filbois, Pierrick Gaudry, Alexander Kruppa, Thomas Richard, Emmanuel Thomé [contact], Paul Zimmermann.

CADO-NFS is a program to factor integers using the Number Field Sieve algorithm (NFS), originally developed in the context of the ANR-CADO project (November 2006 to January 2010).

NFS is a complex algorithm which contains a large number of sub-algorithms. The implementation of all of them is now complete, but still leaves some places to be improved. Compared to existing implementations, the CADO-NFS implementation is already a reasonable player. Several factorizations have been completed using our implementation.

Since 2009, the source repository of CADO-NFS is publicly available for download, and is referenced from the software page at http://cado-nfs.gforge.inria.fr/. A major new release, CADO-NFS 2.1, was published in July 2014, with a bug-fix release (2.1.1) in October. Among the main improvements, the polynomial selection now runs in two stages, several unit tests have been added, various small speed-ups and bug fixes.

More and more people use CADO-NFS to perform medium to large factorizations. In February, Fabien Perigaud and Cédric Pernet from Cassidian Cybersecurity reverse-engineered a ransomware, which in the end boiled down to factoring numbers with CADO-NFS.

5.7. Belenios

Participants: Pierrick Gaudry, Stéphane Glondu [contact].

In collaboration with the CASSIS team, we develop an open-source private and verifiable electronic voting protocol, named BELENIOS. Our system is an evolution of an existing system, Helios, developed by Ben Adida, and used e.g., by UCL and the IACR association in real elections. The main differences with Helios are the following ones:

- In Helios, the ballot box publishes the encrypted ballots together with their corresponding voters. This raises a privacy issue in the sense that whether someone voted or not shall not necessarily be publicized on the web. Publishing this information is in particular forbidden by CNIL's recommendation. BELENIOS no longer publishes voters' identities, still guaranteeing correctness of the tally.
- Helios is verifiable except that one has to trust that the ballot box will not add ballots. The addition of ballots is particularly hard to detect as soon as the list of voters is not public. We have therefore introduced an additional authority that provides credentials that the ballot box can verify but not forge [18], [23].

This new version has been implemented by Stéphane Glondu [0]. The first public release has been done in January 2014. In the last public release (April 2014), BELENIOS still uses a major component of the Helios system, the booth. Since then, the booth has been reimplemented but is not yet part of a public release. This development version of BELENIOS has been used in December 2014 for selecting photos of LORIA's calendar (187 persons voted for 0 to 6 pictures, within a set of 52 choices).

5.8. CMH

Participant: Emmanuel Thomé [contact].

In collaboration with the LFANT project-team, INRIA Bordeaux – Sud-Ouest, we develop the CMH software package and library, which holds code for computing Igusa class polynomials. Those characterize principally polarized abelian varieties of dimension 2 having complex multiplication by the ring of integers of a quartic CM field.

The source repository of CMH is publicly available for download, and is referenced from the software page at http://cmh.gforge.inria.fr/.

Version 1.0 has been released in March 2014, simultaneously with the publication of a computation record.

5.9. Platforms

5.9.1. CATREL cluster

Installed in 2013, the CATREL computer cluster now plays an essential role in providing the team with the necessary resources to achieve significant computations, which illustrate well the efficiency of the algorithms developed in our research, together with their implementations.

[0] http://belenios.gforge.inria.fr/

CARTE Project-Team

5. New Software and Platforms

5.1. Morphus/MMDEX

MMDEX is a virus detector based on morphological analysis. It is composed of our own disassembler tool, on a graph transformer and a specific tree-automaton implementation. The tool is used in the EU-Fiware project and by some other partners (e.g., DAVFI project).
Written in C, 20k lines.
APP License, IDDN.FR.001.300033.000.R.P.2009.000.10000, 2009.

5.2. DynamicTracer

DynamicTracer is a new tool with a public web interface which provides run trace of executable files. The trace is obtained by recording a dynamic execution in a safe environment. The trace contain instruction addresses, instruction opcodes and other optional informations.
Written in C++, 2.5k lines.
http://www.lhs.loria.fr/wp/?page_id=96

5.3. CoDisasm

Codisasm is a new disassembly program which support self-modifying code and code overlapping. Up to our knowledge, this is the first to cope both aspects of program obfuscation. The tool is based on the notion of wave developed in the group.
Written in C, 3k lines.

CASSIS Project-Team

5. New Software and Platforms

5.1. Protocol Verification Tools

Participants: Véronique Cortier, Stéphane Glondu, Pierre-Cyrille Héam, Olga Kouchnarenko, Steve Kremer, Michaël Rusinowitch, Mathieu Turuani, Laurent Vigneron.

5.1.1. CL-AtSe

We develop *CL-AtSe*, a Constraint Logic based Attack Searcher for cryptographic protocols, initiated and continued by the European projects *AVISPA*, AVANTSSAR (for web-services) and Nessos respectively. The *CL-AtSe* approach to verification consists in a symbolic state exploration of the protocol execution for a bounded number of sessions, thus is both correct and complete. *CL-AtSe* includes a proper handling of sets, lists, choice points, specification of any attack states through a language for expressing e.g., secrecy, authentication, fairness, or non-abuse freeness, advanced protocol simplifications and optimizations to reduce the problem complexity, and protocol analysis modulo the algebraic properties of cryptographic operators such as XOR (exclusive or) and Exp (modular exponentiation).

CL-AtSe has been successfully used to analyse protocols from e.g., France Telecom R&D, Siemens AG, IETF, Gemalto, Electrum in funded projects. It is also employed by external users, e.g., from the AVISPA's community. Moreover, *CL-AtSe* achieves good analysis times, comparable and sometimes better than other state-of-the art tools.

CL-AtSe has been enhanced in various ways. It fully supports the Aslan semantics designed in the context of the AVANTSSAR project, including Horn clauses (for intruder-independent deductions, e.g., for credential management), and a large fragment of LTL-based security properties. A Bugzilla server collects bug reports, and online analysis and orchestration are available on our team server (https://cassis.loria.fr). Large models can be analysed on the TALC Cluster in Nancy with parallel processing. *CL-AtSe* also supports negative constraints on the intruder's knowledge, which reduces drastically the orchestrator's processing times and allows separation of duties and non-disclosure policies, as well as conditional security properties, like: i) an authentication to be verified iff some session key is safe; ii) relying on a leaking condition on some private data instead of an honesty predicate to trigger or block some agent's property. This was crucial for e.g., the Electrum's wallet where all clients can be dishonest but security guarantees must be preserved anyway.

5.1.2. Akiss

Akiss (Active Knowledge in Security Protocols) is a tool for verifying indistinguishability properties in cryptographic protocols, modelled as trace equivalence in a process calculus. Indistinguishability is used to model a variety of properties including anonymity properties, strong versions of confidentiality and resistance against offline guessing attacks, etc. *Akiss* implements a procedure to verify equivalence properties for a bounded number of sessions based on a fully abstract modelling of the traces of a bounded number of sessions of the protocols into first-order Horn clauses and a dedicated resolution procedure. The procedure can handle a large set of cryptographic primitives, namely those that can be modeled by an optimally reducing convergent rewrite system.

Recent developments include the possibility for checking everlasting indistinguishability properties [72]. This feature was added when analyzing everlasting privacy properties in electronic voting protocols. The tool is still under active development, including optimisations to improve efficiency, but also the addition of new features, such as the possibility to model protocols using weak secrets.

The *Akiss* tool is freely available at https://github.com/glondu/akiss.

5.1.3. Belenios

In collaboration with the Caramel project-team, we develop an open-source private and verifiable electronic voting protocol, named *Belenios*. Our system is an evolution and a new implementation of an existing system, Helios, developed by Ben Adida, and used e.g., by UCL and the IACR association in real elections. The main differences with Helios are a cryptographic protection against ballot stuffing and a practical threshold decryption system that allows to split the decryption key among several authorities, k out of n authorities being sufficient to decrypt. We will continue to add new cryptographic and protocol improvements to offer a secure, proved, and practical electronic voting system.

Belenios has been implemented (cf. http://belenios.gforge.inria.fr) by Stéphane Glondu and has been tested in December 2014 “in real conditions”, in a test election involving the members of Inria Nancy-Grand Est center and of the Loria lab (more than 500 potential voters) that had to elect the best pictures of the Loria.

5.1.4. SAPIC

SAPIC is a tool that translates protocols from a high-level protocol description language akin to the applied pi calculus into multiset rewrite rules, that can then be be analysed using the Tamarin Prover.

Its aim is the analysis of protocols that include states, for example Hardware Security Tokens communicating with a possibly malicious user, or protocols that rely on databases. It has been succesfully applied on several case studies including the Yubikey authentication protocol.

A recent extension, *SAPIC** extends SAPIC by a Kleene star operator (*) which allows to iterate a process a finite but arbitrary number of times. This construction is useful to specify for instance stream authentication protocols. We used it to analyse a simple version of the TESLA protocol.

The *SAPIC* tool is freely available at http://sapic.gforge.inria.fr/.

5.2. Testing Tools

Participants: Fabrice Bouquet, Frédéric Dadeau, Kalou Cabrera, Ivan Enderlin.

5.2.1. Hydra

Hydra is an Eclipse-like platform, based on Plug-ins architecture. Plug-ins can be of five kinds: *parser* is used to analyze source files and build an intermediate format representation of the source; *translator* is used to translate from a format to another or to a specific file; *service* denotes the application itself, i.e., the interface with the user; *library* denotes an internal service that can be used by a service, or by other libraries; *tool* encapsulates an external tool. The following services have been developed so far:

- BZPAnimator: performs the animation of a BZP model (a B-like intermediate format);
- Angluin: makes it possible to perform a machine learning algorithm (à la Angluin) in order to extract an abstraction of a system behavior;
- UML2SMT: aims at extracting first order logic formulas from the UML Diagrams and OCL code of a UML/OCL model to check them with a SMT solver.

These services involve various libraries (sometimes reusing each other), and rely on several *tool* plug-ins that are: SMTProver (encapsulating the Z3 solver), PrologTools (encapsulating the CLPS-B solver), Grappa (encapsulating a graph library). We are currently working on transferringthe existing work on test generation from B abstract machines, JML, and statecharts using constraint solving techniques.

5.2.2. jMuHLPSL

jMuHLPSL [6] is a mutant generator tool that takes as input a verified HLPSL protocol, and computes mutants of this protocol by applying systematic mutation operators on its contents. The mutated protocol then has to be analyzed by a dedicated protocol analysis tool (here, the AVISPA tool-set). Three verdicts may then arise. The protocol can still be *safe*, after the mutation, this means that the protocol is not sensitive to the realistic “fault” represented by the considered mutation. This information can be used to inform the protocol designers of the robustness of the protocol w.r.t. potential implementation choices, etc. The protocol can also become *incoherent*, meaning that the mutation introduced a functional failure that prevents the protocol from being executed entirely (one of the participants remains blocked in a given non-final state). The protocol can finally become *unsafe* when the mutation introduces a security flaw that can be exploited by an attacker. In this case, the AVISPA tool-set is able to compute an attack-trace, that represents a test case for the implementation of the protocol. If the attack can be replayed entirely, then the protocol is not safe. If the attack can not be replayed then the implementation does not contain the error introduced in the original protocol.

The tool is written in Java, and it is freely available at: http://members.femto-st.fr/sites/femto-st.fr.frederic-dadeau/files/content/pub/jMuHLPSL.jar.

5.2.3. Praspel

Praspel is both a specification language, a test data generator and test execution driver for PHP programs. These latter are annotated to describe class (resp. method) contracts using invariants (resp. pre- and postconditions). Praspel contracts allow to describe data typing informations, by means of *realistic domains*. According to the contract-driven testing principles, the tool uses the contracts to both generate test data, using dedicated test generators (random for integer variables, grammar-based for strings, constraint-based for arrays), and establish the test verdict by checking the contract assertions at run-time.

The tool is open source and freely available at: http://hoa-project.net. It has been integrated into a PHP framework named Hoa, and coupled with the atoum tool (https://github.com/atoum/atoum) that can be used to execute the tests and report on their code coverage.

5.3. Other Tools

Several software tools described in previous sections are using tools that we have developed in the past. For instance BZ-TT uses the set constraints solver CLPS. Note that the development of the SMT prover haRVey has been stopped. The successor of haRVey is called veriT and is developed by David Déharbe (UFRN Natal, Brasil) and Pascal Fontaine (Veridis team). We have also developed, as a second back-end of *AVISPA*, TA4SP (Tree Automata based on Automatic Approximations for the Analysis of Security Protocols), an automata based tool dedicated to the validation of security protocols for an unbounded number of sessions.

We have also designed tools to manage collaborative works on shared documents using flexible access control models. These tools have been developed in order to validate and evaluate our approach on combining collaborative edition with optimistic access control.

COAST Team

4. New Software and Platforms

4.1. Rivage

Participant: Claudia-Lavinia Ignat [contact].

Rivage (Real-tIme Vector grAphic Group Editor) is a real-time collaborative graphical editor. Several users can edit at the same time and in real-time a graphical document, user changes being immediately seen by the other users. The editor relies on a peer-to-peer architecture where users can join and leave the group at any time. Each user has a copy of the shared document and user changes on the document copies are merged in real-time by using a CRDT (Commutative Replicated Data Type) algorithm. The code is available at https://github.com/stephanemartin/rivage/

4.2. Replication Benchmarker

Participants: Pascal Urso [contact], Mehdi Ahmed-Nacer, Gérald Oster.

The Replication Benchmarker is a performance evaluation framework for optimistic replication mechanisms used in collaborative applications. It contains a library of implementation of several CRDT (Commutative Replicated Data Type) and OT (Operational Transformation) algorithms for different data types: text, set, trees. The framework is able to evaluate the performance of comparable algorithms on different corpus of events traces. These events traces can be produced randomly according to different parameters, can be extracted from real real-time editing session that have been recorded, or can be automatically extracted from distributed version control repositories such as the one produced with Git. Performances of the algorithms are measured in term of execution time, memory footprint and merge result quality (compared to manual merge history stored in git repositories). The source code of this evaluation framework is available at https://github.com/score-team/replication-benchmarker/.

4.3. BeGoood

Participant: Gérôme Canals.

BeGoood is a generic system for managing non-regression tests on knowledge bases. BeGoood allows to define test plans in order to monitor the evolution of knowledge-bases. Any system answering queries by providing results in the form of set of strings can be tested with BeGoood. BeGoood has been developed following a REST architecture and is independent of any application domain. BeGoood is a part of the Kolflow infrastructure and is available at https://github.com/kolflow/.

4.4. MUTE

Participants: Claudia Ignat, Luc André, François Charoy, Gérald Oster [contact].

MUTE (Multi-User Text Editor) is a web-based text editing tool that allows to edit documents collaboratively in real-time. It implements our recent work on collaborative editing algorithms and more specifically the LOGOOTSPLIT+ approach [22]. Compared to existing web-based collaborative text editing tool this editor does not require a powerful central server since the server is not performing any computation and acts as a simple broadcast server. Our editor offers support for working offline while still being able to reconnect at a later time. This prototype is distributed under the term of GNU GPLv3 licence and is freely available at https://github.com/score-team/mute-demo/. A demo server is hosted at http://mute-editorcrdt.rhcloud.com/.

CORIDA Team

5. New Software and Platforms

5.1. Simulation of viscous fluid-structure interactions

Participants: Takéo Takahashi [correspondant], Jean-François Scheid.

A number of numerical codes for the simulation for fluids and fluid-structure problems has been developed by the team. These codes are mainly written in MATLAB Software with the use of C++ functions in order to improve the sparse array process of MATLAB. We have focused our attention on 3D simulations which require large CPU time resources as well as large memory storage. In order to solve the 3D Navier-Stokes equations which model the viscous fluid, we have implemented an efficient 3D Stokes sparse solver for MATLAB and a 3D characteristics method to deal with the nonlinearity of Navier-Stokes equations. This year, we have also started to unify our 2D fluid-structure codes (fluid alone, fluid with rigid bodies and fluid with fishes).

Another code has been developed in the case of self-propelled deformable object moving into viscous fluid. Our aim is to build a deformable ball which could swim in a viscous fluid. In order to do this we have started a collaboration with a team from the CRAN (Research Centre for Automatic Control). This software solves numerically 3D Stokes equations using finite elements methods. The source code is written for use with MATLAB thanks to a C++ library developped by ALICE.

- Version: v0.5
- Programming language: MATLAB/C++

5.2. Fish locomotion in perfect fluids with potential flow

Participants: Alexandre Munnier [correspondant], Bruno Pinçon.

SOLEIL is a Matlab suite to simulate the self-propelled swimming motion of a single 3D swimmer immersed in a potential flow. The swimmer is modeled as a shape-changing body whose deformations can be either prescribed as a function of time (simulation of the direct swimming problem) or computed in such a way that the swimmer reaches a prescribed location (control problem). For given deformations, the hydrodynamical forces exerted by the fluid on the swimmer are expressed as solutions of 2D integral equations on the swimmer's surface, numerically solved by means of a collocation method.

SOLEIL is free, distributed under licence GPL v3. More details are available on the project web page http://soleil.gforge.inria.fr/.

The next step of SOLEIL (under progress) is to take into account a fluid whose flow is governed by Stokes equations.

- Version: 0.1
- Programming language:Matlab/C++

5.3. SUSHI3D : SimUlations of Structures in Hydrodynamic Interactions

Participants: Jean-François Scheid, Takéo Takahashi.

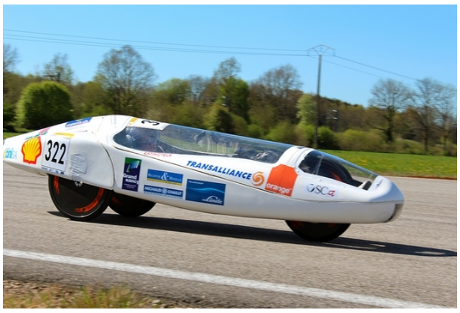
SUSHI3D is a 3D solver for numerical simulations of Fluid/Structures Interactions. The Navier-Stokes equations are coupled with the dynamics of immersed bodies which can be either rigid or deformable. The deformable body case is handled and designed for fish-swimming. The numerical method used to solve the full differential system is based on a Lagrange-Galerkin method with finite elements.

- Version: 1.0
- Programming language:Matlab/C++

5.4. The Vir'Volt prototype

Participants: Thomas Chambrion, Bruno Pinçon.

The European Shell Eco Marathon is an annual competition gathering around 200 high schools and universities. The aim of this race is to travel a given distance (changing from year to yearn, about 16 km in 2013 and 2014) within a given time (39 minutes in 2014). The winning team is the one with the lowest energy consumption (expressed in km/kWh). The EcoMotion Team (EMT) of the École Supérieure des Sciences et Technologies de l'Ingénieur de Nancy (ESSTIN) in France, has been involved for 15 years in the European Shell Eco-Marathon in the categories gasoline, hydrogen and battery electric. In 2014, the prototype *Vir'Volt 3* (see Figure 1) entered the competition in the battery electric category.



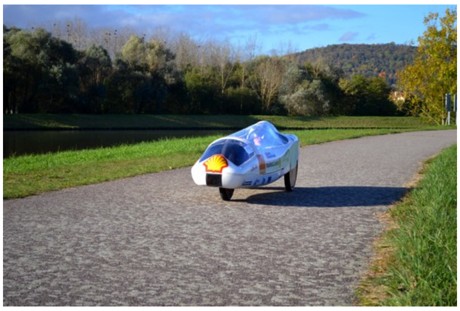


Figure 1. Vir'Volt prototype during a test run in Geoparc race track near Saint Dié in May 2014 (left) and in the neighborhood of Toul in October 2014 (right).

An automatic speed control was embedded in the vehicle. From the velocity measures and a GPS sensor, the dynamics was identified in real time. This identification was precise enough to detect changes in the slope of the track or in wind direction. This dynamics was then used to compute in real time an optimal pair of lower and upper bounds for the speed. These bounds were computed in real time with an embedded low cost micro-controller. The final performance [0] of 533 km/kWh is in line with the (human driven) performance of the team in the recent years.

[0] http://s00.static-shell.com/content/dam/shell-new/local/corporate/ecomarathon/downloads/pdf/europe/2014-results/sem-europe-2014-results-prototype-battery-electric-220514.pdf

MADYNES Project-Team

5. New Software and Platforms

5.1. Escape

Participants: Thibault Cholez [contact], Shbair Wazen.

Initially developed by Antoine Goichot during his internship [47], from reasearch results of Wazen Shbair, Thibault Cholez and Isabelle Chrisment.

Escape is a Firefox web browser addon designed and developed by the team to bypass some HTTPS filtering strategies. The extension was built in the context of evaluating HTTPS traffic filtering techniques based on the Server Name Indication (SNI) extension of TLS and which have been recently used by many firewalls for filtering websites accessed through HTTPS. Our tool mainly offers the ability to bypass such firewalls by editing on-the-fly the SNI field with alternate values and therefore access the blocked HTTPS websites. In addition, it can be used to bypass legacy filtering of DNS requests. The extension is implemented in JavaScript and is based on another security addon named Convergence. Escape is distributed under a GPL3 Open Source license and can be downloaded on the team website.

5.2. MPIGate

Participants: Mandar Harshe, Ye-Qiong Song [contact].

MPIGate stands for Multi Protocol Interface GATEway for Tele-care, Environment Monitoring and Control. It was initiated by TRIO Team of LORIA and Inria Nancy Grand-est, in October 2009 as a follow-up of wireless sensor network (WSN) projects in ambient assisted living, smart home, logistic and industry domains. Since 2012, its evolution is continuously ensured by members of MADYNES Team. It is a set of software aiming at facilitating the development of both home automation and ambient assisted living applications thanks to the abstraction of heterogeneous sensor data and the facility of access to read and write functions over the devices plugged to the networks (wired and wirelessly). The key features of MPIGate include the drivers for different networks protocols (Bluetooth, WiFi, IEEE802.15.4/Zigbee, KNX, EnOcean) and a ROS-based middleware layer offering modularity and quality of service. This year, its evolution has mainly been carried out within SATELOR project and IPL PAL project. It can be used by people working on home automation and ambient assisted living applications. Further information can be found at http://mpigate.loria.fr.

5.3. AA4MM

Participants: Laurent Ciarletta [contact], Yannick Presse, Benjamin Segault.

Benjamin Camus, Victorien Elvinger, Vincent Chevrier (contact), Julien Vaubourg, and Christine Bourjot from the MAIA team, LORIA are contributors for this software.

AA4MM (Agents and Artefacts for Multi-modeling and Multi-simulation) is a framework for coupling existing and heterogeneous models and simulators in order to model and simulate complex systems. The first implementation of the AA4MM meta-model was proposed in Julien Siebert's PhD [51] and written in Java, and a renewed Java version was submitted to the APP (Agence pour la protection des programmes).

We are using this software in a strategic action with EDF R&D in the context of the simulation of smart-grids in the frame of the MS4SG (Multi-Simulation fro Smart Grids) project. Julien Vaubourg started a PhD on this project that is co-directed by Laurent Ciarletta and Vincent Chevrier. The 2014 year was dedicated to improve existing software and to develop new components thanks to new scientific contributions.

Currently, two new pieces of software are being submitted to the APP:

1. a modelling environment software that enables the graphical definition of multi-models from preexisting elements.
2. AA4MM-Visu, a plug-in dedicated to the collection and visualization of information during simulation.

We plan to submit an enhanced version of the JAVA software and of the AA4MM-Visu. The core elements of AA4MM will be made available early in 2015 under an open licence.

5.4. Platforms

5.4.1. Android Security platform

Participants: Abdelkader Lahmadi [contact], Rémi Badonnel, Olivier Festor, Eric Finickel, Frederic Beck [SED, Inria Nancy Grand Est].

Android environments are facing several threats and attacks. Madynes team is working on the development of a monitoring platform dedicated to the security analysis and these environments. The monitoring platform relies on different components:

- a set of probes dedicated to the measurement of network activities using NetFlow protocol and logs generated by running Applications of an Android device. An OVAL agent (Ovaldroid) is also developed for vulnerability assessment.
- a set of scalable data collectors to collect and parse the data issued by our probes (NetFlow records, logs in the syslog format and vulnerability reports). The collectors are relying on Flume agents.
- a NoSQL storage (HBase) engine where all the collected data are stored for further analysis.
- A first set of analysers of the collected data, relying on a Map-Reduce engine (Spark) are also developed [41] including statistical analysis about connected services and ports but also a Self-Organising Map analyser to classify Android application patterns according to different properties including their communication patterns and also their lifecycle activities. [16].

The first version of the monitoring platform is operational and deployed within the LHS infrastructure. Further development is currently under taken to provide more analysis, data correlation and visualisation features.

5.4.2. IoT platform

Participants: Emmanuel Nataf [contact], Thibault Cholez.

This platform is a joint work between Anthony Deroche [43], Thierry Duhal [45] and Arthur Garnier [46], respectively students from TELECOM Nancy and IUT Nancy-Charlemagne. They worked under the supervision of Emmanuel Nataf and Thibault Cholez between February and August 2014.

The main goal of the IOT platform is to collect and store production and management data produced during long-run WSN experiments. The platform is open-source (https://github.com/AnthonyDeroche/iotlab/) and built with a modular architecture in order to support different types of experiment (routing algorithms, energy efficiency, security, etc.).

Based on this platform, we developed several innovative applications:

- indoor geolocalization of sensors based on RSSI strength [43]
- data collection from several concurrent points allowing better scalability with good performances on large WSN [45]
- data link to remotely control nodes from the web interface with a skeleton of API [45]

Regarding the technical aspects [44], the platform is based on a JEE architecture running on a Glassfish server, websocket full-duplex communications, secure and authenticated administrator access (HTTPS). The web interface uses the framework CSS front-end Zurb Foundation and javascript libraries to display dynamic charts and maps.

The full plateform has been instantiated with 40 TELOSB sensors deployed in TELECOM Nancy (http://iotlab.telecomnancy.eu/) during one month.

5.4.3. SCADA platform

Participants: Abdelkader Lahmadi [contact], Jérôme François, Olivier Festor.

SCADA is a term used in several industries and its stands for *Supervisory Control and Data Acquisitions*. It refers to a centralized control and monitoring system for a variety of machinery and equipment involved with many industrial activities. SCADA systems are also becoming target to different attacks exploiting traditional IT vulnerabilities, e.g. buffer overflows, script crossing, crafted network packets, or specific vulnerabilities related to control and estimation algorithms employed by control processes.

We are developing and maintaining a platform to assess and analyse the security of SCADA systems based on a testbed combining real hardware and simulation tools of physical processes. We have extended our SCADA testbed to simulate a microgrid scenario [49]. We are thus able to extract and analyse the Profinet messages at the control network level using process mining techniques. Further development will be taken to include information technology layers in the testbed (servers, firewalls, network devices, etc).

During the year 2014, we have also started the development of a scanning platform of Internet IP addresses and communication ports to identify exposed sensitive services and networks, for instance SCADA systems [42].

MAGRIT Project-Team

5. New Software and Platforms

5.1. Ralib

Our research efforts are integrated in a library called RAlib which contains our research development on image processing, registration (2D and 3D) and visualization. This library is licensed by the APP (French agency for software protection). The library was extended over the period to integrate our new research code on tongue modeling and tracking. Several applications either used internally or to demonstrate our work have been designed with this library.

5.2. PoLAR

The visualization module in RAlib has now reached a level of maturity where we believe it could be proposed to a wider audience. In the context of the ADT PoLAR (which started on October, 1st), a software engineer, Pierre-Jean Petitprez, started working on a new library called PoLAR (Portable Library for Augmented Reality). So far, the code has been cleanly made independent from our other code in RAlib, and in the process of being ported to up-to-date versions of the supporting libraries: OpenSceneGraph 3.2 and Qt5.

5.3. Ltrack

The Inria development action **LTrack** aims at developing an Android platform in order to facilitate the transfer of some of our algorithms onto mobile devices. For the moment, the tracking-by-synthesis algorithm has been implemented (up to our knowledge, for the first time on a mobile device) in order to rigidly track a real object in real time assuming that a CAD model of this object is available. The design and implementation of the platform have been guided by the need to enable easy integration of any tracking algorithm based on combining video data and other sensor information.

MAIA Project-Team

5. New Software and Platforms

5.1. AA4MM Suite

Participants: Vincent Chevrier [correspondant], Christine Bourjot, Benjamin Camus, Julien Vaubourg, Victorien Elvinger.

Laurent Ciarletta (Madynes team, LORIA) is a collaborator and correspondant for this software. Yannick Presse and Benjamin Segault (Madynes team, LORIA) are collaborator for this software.

AA4MM (Agents and Artefacts for Multi-modeling and Multi-simulation) is a framework for coupling existing and heterogeneous models and simulators in order to model and simulate complex systems. The first implementation of the AA4MM meta-model was proposed in Julien Siebert's PhD [54] and written in Java, and a renewed JAVA version was submitted to the APP (Agence pour la protection des programmes) the previous year.

The 2014 year was dedicated to improve existing software and to develop new components thanks to new scientific contributions.

Currently, two new software are submitted to the APP:

1. a modelling environment software that enables the graphical definition of multi-models from preexisting elements.
2. AA4MM-Visu, a plug in dedicated to the collection and visualization of information during simulation.

We plan to submit an enhanced version of the JAVA software and of the AA4MM-Visu.

5.2. MASDYNE

Participants: Vincent Chevrier [correspondant], Tomas Navarrete [CRP Henri Tudor].

This work was undertaken in the PhD Thesis of Julien Siebert, a joint thesis between the MAIA and MADYNES teams. It has been enhanced during the PhD of Tomas Navarrete.

MASDYNE (Multi-Agent Simulator of DYnamic Networks usErs) is a multi-agent simulator for modeling and simulating users behaviors in mobile ad hoc network. This software is part of joint work with MADYNES team, on modeling and simulation of ubiquitous networks.

5.3. FiatLux

Participant: Nazim Fatès.

FiatLux is a discrete dynamical systems simulator that allows the user to experiment with various models and to perturb them. It includes 1D and 2D cellular automata, moving agents, interacting particle systems, etc. Its main feature is to allow users to change the type of updating, for example from a deterministic parallel updating to an asynchronous random updating. FiatLux has a Graphical User Interface and can also be launched in a batch mode for the experiments that require statistics.

FiatLux is registered by the Agence pour la protection des programmes (APP). It is available under the CeCILL licence on the FiatLux website : fiatlux.loria.fr

In 2014, FiatLux was internally re-shaped in order to facilitate the reproducibility of experiments. In particular, attention was given to the generation of pseudo-random sequences for the stochastic models.

5.4. Platforms

Inria Research Center in Nancy has supported since 2010 the design and the construction of an innovative platform for favoring research in assistance for elderly people at home. This platform has been mainly funded by the CPER MISN (region of Lorraine , project Info-Situ (2010-2013). It consists of a standard apartment type F2, with a certain number of "smart and connected devices" such as sensor networks. This platform has been designed to make easy technical experimentation in an environment which is as close as possible to reality. Many technical developments have been done during the IPL PAL. In particular concerning MAIA Team, we have been working both (1) on the development of new algorithms to exploit the equipments, and (2) on the effective deployment of different kind of connected devices :

1. a network of depth cameras. These depth cameras are either fixed on the wall or are placed onboard wheeled mobile robots. One important achievement has been to connect these cameras to the ethernet network, each camera being considered as a Ros node with computation capabilities(using a NUC for each node). An other achievement has concerned the calibration of theses cameras. Today 7 cameras covers to whose HIS Platform.
2. Pressure sensing tiles which has been designed by Maia team (in cooperation which Hikob (http://www.hikob.com/applications/recherche/ and the Inria SED of Grenoble (Roger Pissard-Gibollet)) during the Pal evaluation period. Ninety tiles cover the floor of our experimental platform (HIS), which permit to sense activity through the natural interaction of people or robots with the floor when they are acting;
3. Mobile robots whose mobility allows a better coverage in term of perception of the environment.
4. recently we got a Qualisys motion capture system 5funded by Satelor Project).

These devices are all interconnected within the Robotic Operating System (ROS).

MASAIE Project-Team (section vide)

MULTISPEECH Team

5. New Software and Platforms

5.1. Introduction

This software section is organized along three main axes: tools for automatic speech processing, then visualization tools used to display different aspects of speech data and which possibly feature other functionalities; and finally tools and platforms for acquiring articulatory data.

5.2. Speech processing tools

Participants: Denis Jouvet, Dominique Fohr, Odile Mella, Irina Illina, Emmanuel Vincent, Antoine Liutkus, Vincent Colotte, Yann Salaün, Antoine Chemardin.

These automatic speech processing tools deal with audio data transcription (ANTS), audio sources separation (FASST), speech-text alignment (LASTAS) and text-to-speech synthesis (SoJA).

5.2.1. ANTS (Automatic News Transcription System)

ANTS is a multipass system for transcribing audio data, and in particular radio or TV shows. The audio stream is first split into homogeneous segments of a manageable size, and then each segment is decoded using the most adequate acoustic model with a large vocabulary continuous speech recognition engine (Julius or Sphinx). Further processing passes are run in order to apply unsupervised adaptation processes on the features (VTLN: Vocal Tract Length Normalization) and/or on the model parameters (MLLR: Maximum Likelihood Linear Regression), or to use Speaker Adaptive Training (SAT) based models. Moreover decoding results of several systems can be efficiently combined for improved decoding performance. The latest version takes advantage of the multiple CPUs available on a computer, and runs on both standalone linux machines and on clusters.

5.2.2. FASST (Flexible Audio Source Separation Toolbox)

FASST [0] is a toolbox for audio source separation distributed under the Q Public License. Version 2 in C++ has been developed in the context of the ADT FASST (conducted by MULTISPEECH in collaboration with the PANAMA and TEXMEX teams from Inria Rennes - cf. 8.1.6) and released in January 2014. Its unique feature is the possibility for users to specify easily a suitable algorithm for their use case thanks to the general modeling and estimation framework proposed in [6]. It forms the basis of most of our current research in audio source separation, some results of which will be incorporated into future versions of the software.

5.2.3. KAM (Kernel Additive Modelling)

The Kernel Additive Modelling framework for source separation [13], [42] has been proposed this year by Liutkus et al. as a new and effective approach to source separation. In 2014, two different implementations of KAM have been registered with the APP: a Matlab version matKAM and a python version pyKAM. The former is under a aGPL license, while the latter is under a proprietary license. The rationale for this choice is that the Matlab version is to be mainly disseminated for research purpose to the colleagues in the field, that mainly use Matlab, while the python version is more liable to lead to industrial transferts.

5.2.4. LASTAS (Loria Automatic Speech-Text Alignment Software)

LASTAS is a software for aligning a speech signal with its corresponding orthographic transcription. Using a phonetic lexicon and automatic grapheme-to-phoneme converters, all the potential sequences of phones corresponding to the text are generated. Then, using acoustic models, the tool finds the best phone sequence and provides together the boundaries at the phone level and at the word level.

[0] http://bass-db.gforge.inria.fr/fasst/

This year, this software has been included in a web application for speech-text automatic alignement, named ASTALI, which will soon be available [0].

5.2.5. CoALT (Comparing Automatic Labeling Tools)

CoALT is a software for comparing the results of several automatic labeling processes through user defined criteria [70].

5.2.6. SoJA (Speech synthesis platform in Java)

SOJA [0] is a software for Text-To-Speech synthesis (TTS) which relies on a non uniform unit selection algorithm. It performs all steps from text to speech signal output. Moreover, a set of associated tools is available for elaborating a corpus for a TTS system (transcription, alignment...). Currently, the corpus contains 1800 sentences (about 3 hours of speech) recorded by a female speaker. Most of the modules are in Java; some are in C. The software runs under Windows and Linux. It can be launched with a graphical user interface or directly integrated in a Java code or by following the client-server paradigm. We will consider extending and making SoJA more modular and able to handle both acoustic and visual features, in order to use it for both acoustic-only synthesis and audiovisual synthesis. In the future, the text-to-speech synthesis platform will get extended to take into account expressivity features.

5.3. Speech visualization tools

Participants: Yves Laprie, Slim Ouni, Julie Busset, Aghilas Sini, Ilef Ben Farhat.

This set of tools aims at visualizing various aspects of speech data: speech audio signal (SNOORI), Electro-Magnetographic Articulography (EMA) data (VisArtico) and speech articulators from X-ray images (Xarticulators).

5.3.1. SNOORI: speech analysis and visualization software

JSnoori is written in Java and uses signal processing algorithms developed within the WinSnoori [0] software with the double objective of being a platform independent signal visualization and manipulation tool, and also for designing exercises for learning the prosody of a foreign language. Thus JSnoori currently focuses the calculation of F0, the forced alignment of non native English uttered by French speakers and the correction of prosody parameters (F0, rhythm and energy). Several tools have been incorporated to segment and annotate speech. A complete phonetic keyboard is available, several levels of annotation can be used (phonemes, syllables and words) and forced alignment can exploit pronunciation variants. In addition, JSnoori offers real time F0 calculation which can be useful from a pedagogical point of view.

We added the possibility of developing scripts for JSnoori by using Jython which allows Java classes of JSnoori to be used from Python. This required some refactoring of JSnoori classes in order to make them more independent from the JSnoori context.

5.3.2. VisArtico: Visualization of EMA Articulatory data

VisArtico [0] is a user-friendly software which allows visualizing EMA data acquired by an articulograph (AG500, AG501 or NDI Wave). This visualization software has been designed so that it can directly use the data provided by the articulograph to display the articulatory coil trajectories, synchronized with the corresponding acoustic recordings. Moreover, VisArtico not only allows viewing the coils but also enriches the visual information by indicating clearly and graphically the data for the tongue, lips and jaw [72]. Several researchers showed interest in this application. In fact, VisArtico is very useful for the speech science community, and it makes the use of articulatory data more accessible. The software is a cross-platform application (i.e., running under Windows, Linux and Mac OS).

[0] http://astali.loria.fr

[0] http://soja-tts.loria.fr

[0] http://www.loria.fr/~laprie/WinSnoori/

[0] http://visartico.loria.fr/

Within the framework of an Inria ADT project (cf. 8.1.7), we are implementing several improvements to the software. It is possible to use VisArtico to import and export several articulatory data formats. In addition, it possible to insert images (MRI or X-Ray, for instance) to compare the EMA data with data obtained through other acquisition techniques. Finally, it is possible to generate a movie for any articulatory-acoustic sequence. These improvements (and others) extend the capabilities of VisArtico and make it more useful and widely used. The software will also provide a demonstration module that will produce articulatory synthesis from EMA data or text. It animates the vocal tract, using articulatory data and generates the corresponding acoustic signal. VisArtico is freely available for research.

5.3.3. Xarticulators: delineation of speech articulators in medical images

The Xarticulators software is intended to delineate contours of speech articulators in X-ray images, construct articulatory models and synthesize speech from X-ray films. This software provides tools to track contours automatically, semi-automatically or by hand, to make the visibility of contours easier, to add anatomical landmarks to speech articulators and to synchronize images with the sound. In addition we also added the possibility of processing digitized manual delineation results made on sheets of papers when no software is available. Xarticulators also enables the construction of adaptable linear articulatory models from the X-ray images and incorporates acoustic simulation tools to synthesize speech signals from the vocal tract shape. Recent work was on the possibility of synthesizing speech from X-ray or 2D-MRI films.

We added new articulatory model construction features intended to approximate the tongue shape more correctly when the tongue contacts the palate during the stop closure of /k/ and /t/ and we added more complete modeling of the epiglottis and the larynx region. Future developments will focus on the development of time patterns to synthesize any speech sound and on the coupling between vocal folds and vocal tract.

5.4. Data acquisition

Participants: Vincent Colotte, Slim Ouni, Yves Laprie.

The nature of our research makes us highly concerned by acquisition and processing of speech data. Besides acquisition of speech audio signals, we are concerned with the acquisition of articulatory data, mainly ElectroMagnetographic Articulography (EMA) data using an articulograph and Magnetic Resonance Imaging (MRI) data. EMA captures articulatory movements in three dimensions (3D) with a high temporal resolution by tracking tiny sensors attached to speech articulators such as the tongue, teeth, and lips. MRI is a non-invasive, hazard-free medical imaging technique allowing for high-resolution scans of the vocal tract.

5.4.1. JCorpusRecorder

JCorpusRecorder is a software for the recording of audio corpora. It provides an easy tool to record with a microphone. The audio input gain is controlled during the recording. From a list of sentences, the output is a set of wav files automatically renamed according to textual information given in input (nationality, speaker language, gender...). An easy to use tagging allows for displaying a textual/visual/audio context of the sentence to pronounce. This software is suitable for recording sentences with information to guide the speaker. The sentences can be presented randomly. The software is developed in Java. It is currently used for the recording of sentences in several projects.

5.4.2. EMA acquisition platform

Since the purchase of the articulograph AG500 in 2007, we have built a strong experience with respect to the acquisition technique and we have developed an acquisition protocol (sterilization, calibration, etc.). The platform has been improved by acquiring the latest articulograph AG501 funded by the EQUIPEX ORTOLANG project. The AG501 allows tracking the movement of 24 sensors at reasonable high frequency (250Hz) to very high frequency (1250Hz). In addition, we have developed a powerful tool, VisArtico, to visualize articulatory data acquired using an articulograph.

5.4.3. MRI acquisition platform

Magnetic Resonance Imaging (MRI) takes an increasing place in the investigation of speech production because it provides a complete geometrical information of the vocal tract. We thus initiated a cooperation with the IADI laboratory (Imagerie Adaptive Diagnostique et Interventionnelle) at Nancy Hospital, which studies in particular magnetic resonance imaging. This year, we acquired static MRI data for two speakers (approximately 90 blocked articulations corresponding to vowels and consonants followed by a vowel) and we carried out preliminary experiments intended to acquire dynamic data.

NEUROSYS Team

5. New Software and Platforms

5.1. Software

5.1.1. Visualization

- The NeuralFieldSimulator [0] computes numerically activity in two-dimensional neural fields by solving integral-differential equations involving transmission delays and visualizes the spatio-temporal activity. The tool includes a GUI that allows the user to choose field parameters. It is written in Python, open-source and is aimed to be promoted to become a major graphical visualization tool in the domain of neural field theory. We aim to establish this simulation software as the first open-source standard simulator for the neural field research community.
- *A*naesthesiaSimulator [0] simulates the activity of networks of spiking neurons subject to specific receptor dynamics. The tool is a platform to test effects of anaesthetics on neural activity and is still in its first stage of development. The neural activity is planned to be visualized in a 2D and 3D-plot evolving in time. It is written in Python, open-source and involves heavily the simulation package BRIAN [0].

5.2. Platforms

5.2.1. OpenViBE

This platform [0] is a C++ open-source software devoted to the design, test and use of Brain-Computer Interfaces. The OpenViBE platform consists of a set of software modules that can be integrated easily and efficiently to design BCI applications. Key features of the platform are its modularity, high-performance, portability, its multiple-users facilities and its connection with high-end/Virtual Reality displays. The designer tool of the platform enables to build complete scenarios based on existing software modules using a dedicated graphical language and a simple Graphical User Interface (GUI). This software is available on the Inria Forge [0] under the terms of the LGPL-V2 license. The development of OpenVibe is done in association with other Inria research teams (Hybrid, Athena, Potioc) for the national Inria project: ADT OpenViBE-NT. Neurosys is in charge of machine learning techniques and the interoperability with other tools such as Matlab, BCI2000, or TOBI.

[0] https://gforge.inria.fr/projects/nfsimulator/
[0] https://gforge.inria.fr/projects/anasim/
[0] http://briansimulator.org/
[0] http://openvibe.inria.fr/
[0] https://gforge.inria.fr/projects/openvibe/

ORPAILLEUR Project-Team

5. New Software and Platforms

5.1. Generic Symbolic KDD Systems

5.1.1. The Coron Platform

Participants: Jérémie Bourseau, Aleksey Buzmakov, Victor Codocedo, Adrien Coulet, Amedeo Napoli [contact person], Yannick Toussaint.

Keywords: data mining, frequent itemset, closed itemset, generator, association rule, rare itemset

The Coron platform [133], [120] is a KDD toolkit organized around three main components: (1) Coron-base, (2) AssRuleX, and (3) pre- and post-processing modules. The software was registered at the "Agence pour la Protection des Programmes" (APP) and is freely available (see http://coron.loria.fr).

The Coron-base component includes a complete collection of data mining algorithms for extracting itemsets such as frequent itemsets, closed itemsets, generators and rare itemsets. In this collection we can find APriori, Close, Pascal, Eclat, Charm, and, as well, original algorithms such as ZART, Snow, Touch, and Talky-G [45]. AssRuleX generates different sets of association rules (from itemsets), such as minimal non-redundant association rules, generic basis, and informative basis. In addition, the Coron system supports the whole life-cycle of a data mining task and proposes modules for cleaning the input dataset, and for reducing its size if necessary.

The Coron toolkit is developed in Java, is operational, and was already used in several research projects.

5.1.2. Orion: Skycube Computation Software

Participant: Chedy Raïssi [contact person].

Keywords: skyline, skycube

This program implements the algorithms described in a research paper published at VLDB 2010 [127]. The software provides a list of four algorithms discussed in the paper in order to compute skycubes. This is the most efficient –in term of space usage and runtime– implementation for skycube computation (see https://github.com/leander256/Orion).

5.2. Stochastic systems for knowledge discovery and simulation

5.2.1. The CarottAge System

Participants: Florence Le Ber, Jean-François Mari [contact person].

Keywords: Hidden Markov Models, stochastic process

The system CarottAge is based on Hidden Markov Models of second order and provides a non supervised temporal clustering algorithm for data mining and a synthetic representation of temporal and spatial data [92]. CarottAge is currently used by INRA researchers interested in mining the changes in territories related to the loss of biodiversity (projects ANR BiodivAgrim and ACI Ecoger) and/or water contamination. CarottAge is also used for mining hydromorphological data. Actually a comparison was performed with three other algorithms classically used for the delineation of river continuum and CarottAge proved to give very interesting results for that purpose [121].

CarottAge is freely available under GPL license (see http://www.loria.fr/~jfmari/App/). A special effort is currently aimed at designing interactive visualization tools to provide the expert a user-friendly interface.

5.2.2. The ARPEnTAge System

Participant: Jean-François Mari [contact person].

Keywords: Hidden Markov Models, stochastic process

ARPEnTAge, for "*Analyse de Régularités dans les Paysages : Environnement, Territoires, Agronomie*" (http://www.loria.fr/~jfmari/App/) is a software based on stochastic models (HMM2 and Markov Field) for analyzing spatio-temporal data-bases [124]. ARPEnTAge is built on top of the CarottAge system to fully take into account the spatial dimension of input sequences. It takes as input an array of discrete data in which the columns contain the annual land-uses and the rows are regularly spaced locations of the studied landscape. It performs a Time-Space clustering of a landscape based on its time dynamic Land Uses (LUS). Displaying tools and the generation of Time-dominant shape files have also been defined.

ARPEnTAge is freely available (GPL license) and is currently used by INRA researchers interested in mining the changes in territories related to the loss of biodiversity (projects ANR BiodivAgrim and ACI Ecoger) and/or water contamination. In these practical applications, CarottAge and ARPEnTAge aim at building a partition –called the hidden partition– in which the inherent noise of the data is withdrawn as much as possible. The estimation of the model parameters is performed by training algorithms based on the Expectation Maximization and Mean Field theories. The ARPEnTAge system takes into account: (i) the various shapes of the territories that are not represented by square matrices of pixels, (ii) the use of pixels of different size with composite attributes representing the agricultural pieces and their attributes, (iii) the irregular neighborhood relation between those pixels, (iv) the use of shape files to facilitate the interaction with GIS (geographical information system).

ARPEnTAge and CarottAge were used for mining decision rules in a territory showing environmental issues. They provide a way of visualizing the impact of farmers decision rules in the landscape and revealing new extra hidden decision rules [132].

5.3. KDD in Systems Biology

Participants: Marie-Dominique Devignes [contact person], Malika Smaïl-Tabbone.

5.3.1. IntelliGO Online

The IntelliGO measure computes semantic similarity between terms from a structured vocabulary (Gene Ontology: GO) and uses these values for computing functional similarity between genes annotated by sets of GO terms [104]. The IntelliGO measure is available on line (http://plateforme-mbi.loria.fr/intelligo/) to be used for evaluation purposes. It is possible to compute the functional similarity between two genes, the intra-set similarity value in a given set of genes, and the inter-set similarity value for two given sets of genes.

5.3.2. WAFOBI: KNIME Nodes for Relational Mining of Biological Data

KNIME (for "Konstanz Information Miner") is an open-source visual programming environment for data integration, processing, and analysis. The KNIME platform aims at facilitating the data mining experiment settings as many tests are required for tuning the mining algorithms. Various KNIME nodes were developed for supporting relational data mining using the ALEPH program (http://www.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/aleph.pl). These nodes include a data preparation node for defining a set of first-order predicates from a set of relation schemes and then a set of facts from the corresponding data tables (learning set). A specific node allows to configure and run the ALEPH program to build a set of rules. Subsequent nodes allow to test the first-order rules on a test set and to perform configurable cross validations.

5.3.3. MOdel-driven Data Integration for Mining (MODIM)

The MODIM software (MOdel-driven Data Integration for Mining) is a user-friendly data integration tool which can be summarized along three functions: (i) building a data model taking into account mining requirements and existing resources; (ii) specifying a workflow for collecting data, leading to the specification of wrappers for populating a target database; (iii) defining views on the data model for identified mining scenarios.

Although MODIM is domain independent, it was used so far for biological data integration in various internal research studies and for organizing data about non ribosomal peptide syntheses. The sources can be downloaded at https://gforge.inria.fr/projects/modim/.

5.4. Knowledge-Based Systems and Semantic Web Systems

5.4.1. The Kasimir System for Decision Knowledge Management

Participants: Nicolas Jay, Jean Lieber [contact person], Amedeo Napoli.

Keywords: classification-based reasoning, case-based reasoning, decision knowledge management, knowledge edition, knowledge base maintenance, semantic portal

The objective of the Kasimir system is decision support and knowledge management for the treatment of cancer. A number of modules have been developed within the Kasimir system for editing treatment protocols, visualization, and maintenance. Kasimir is developed within a semantic portal, based on OWL. KatexOWL (Kasimir Toolkit for Exploiting OWL Ontologies, http://katexowl.loria.fr) was developed in a generic way and is applied to Kasimir. In particular, the user interface EdHibou of KatexOWL is used for querying the protocols represented within the Kasimir system In [109], this research is presented, together with an extension of Kasimir for multi-viewpoint case-based reasoning.

CabamakA (case base mining for adaptation knowledge acquisition) is a module of the Kasimir system. This system performs case base mining for adaptation knowledge acquisition and provides information units to be used for building adaptation rules. Actually, the mining process in CabamakA is based on a frequent close itemset extraction module from the Coron platform (see §5.1.1).

The Oncologik system is a collaborative editing tool aiming at facilitating the management of medical guidelines. Based on a semantic wiki, it allows the acquisition of formalized decision knowledge also includes a graphical decision tree editor called KcatoS. A version of Oncologik was released in 2012 (http://www.oncologik.fr/).

5.4.2. Taable: a System for Retrieving and Creating New Cooking Recipes by Adaptation

Participants: Valmi Dufour-Lussier, Emmanuelle Gaillard, Florence Le Ber, Jean Lieber, Amedeo Napoli, Emmanuel Nauer [contact person].

Keywords: knowledge acquisition, ontology engineering, semantic annotation, case-based reasoning, hierarchical classification, text mining

Taable is a system whose objectives are to retrieve textual cooking recipes and to adapt these retrieved recipes whenever needed [4]. Suppose that someone is looking for a “leek pie” but has only an “onion pie” recipe: how can the onion pie recipe be adapted?

The Taable system combines principles, methods, and technologies such as case-based reasoning (CBR), ontology engineering, text mining, text annotation, knowledge representation, and hierarchical classification. Ontologies for representing knowledge about the cooking domain, and a terminological base for binding texts and ontology concepts, were built from textual web resources. These resources are used by an annotation process for building a formal representation of textual recipes. A CBR engine considers each recipe as a case, and uses domain knowledge for reasoning, especially for adapting an existing recipe w.r.t. constraints provided by the user, holding on ingredients and dish types.

The Taaable system is available on line since 2008 at http://taaable.fr, and is constantly evolving. This year, a new version of Taaable has been implemented in order to participate to the 7th Computer Cooking Contest which held during the International Case-Based Reasoning, in Cork, Ireland. The new version of Taaable is based on Tuuurbine, a generic ontology guided CBR engine over RDFS (see Section 5.4.3), and Revisor, an adaptation engine implementing various revision operators (see Section 5.4.5). In particular, Revisor is used to compute ingredient substitutions and to adjust the ingredient quantities.

5.4.3. Tuuurbine: a Generic Ontology Guided Case-Based Inference Engine

Participants: Jean Lieber, Emmanuel Nauer [contact person].

Keywords: case-based reasoning, inference engine, knowledge representation, ontology engineering, semantic web

The experience acquired since 5 years with the Taaable system conducted to the creation of a generic cased-based reasoning system, whose reasoning procedure is based on a domain ontology [63]. This new system, called Tuuurbine (http://tuuurbine.loria.fr/), takes into account the retrieval step, the case base organization, and also an adaptation procedure which is not addressed by other generic case-based reasoning tools. Moreover, Tuuurbine is built over semantic web standards that will ensure facilities for being plugged over data available on the web. The domain knowledge is represented in an RDF store, which can be interfaced with a semantic wiki, for collaborative edition and management of the knowledge involved in the reasoning system (cases, ontology, adaptation rules). The development of Tuuurbine was supported by an Inria ADT funding until October 2013. Tuuurbine is distributed under an Affero GPL License and is available from http://tuuurbine.loria.fr/.

5.4.4. BeGoood: a Generic System for Managing Non-Regression Tests on Knowledge Bases

Participant: Emmanuel Nauer [contact person].

Keywords: tests, non-regression, knowledge evolution

BeGoood is a system allowing to define test plans, independent of any application domain, and usable for testing any system answering queries by providing results in the form of sets of strings. BeGoood provides all the features usually found in test systems, such as tests, associated queries, assertions, and expected result sets, test plans (sets of tests) and test reports. The system is able to evaluate the impact of a system modification by running again test plans and by evaluating the assertions which define whether a test fails or succeeds. The main components of BeGoood are (1) the "test database" that stores every test artifacts, (2) the "remote query evaluator" which evaluates test queries, (3) the "assertion engine" which evaluates assertions over the expected and effective query result sets, (4) the "REST API" which offers the test functionalities as web services, and finally (5) the "Test controller" and (6) the "Test client".

BeGoood is available under a AGPL license on github [0]. BeGoood is used by the Taaable system (see Section 5.4.2) for managing the evolution of the knowledge base used by the CBR system.

5.4.5. Revisor: a Library of Revision Operators and Revision-Based Adaptation Operators

Participants: Valmi Dufour-Lussier, Alice Hermann, Florence Le Ber, Jean Lieber [contact person], Emmanuel Nauer, Gabin Personeni.

Keywords: belief revision, adaptation, revision-based adaptation, case-based reasoning, inference engines, knowledge representation

Revisor is a library of inference engines dedicated to belief revision and to revision-based adaptation for case-based reasoning [3]. It is open source, under a GPL license and available on the web (http://revisor.loria.fr/). It gathers several engines developed during the previous years for various knowledge representation formalisms (propositional logic—with or without the use of adaptation knowledge [93]—conjunction of linear constraints, and qualitative algebras [61], [75], [87], [14]). Some of these engines are already used in the Taaable system. Current developments on Revisor aim at defining new engines in other formalisms.

[0] https://github.com/kolflow/begoood

PAREO Project-Team

5. New Software and Platforms

5.1. ATerm

Participant: Pierre-Etienne Moreau [correspondant].

ATerm (short for Annotated Term) is an abstract data type designed for the exchange of tree-like data structures between distributed applications.

The ATerm library forms a comprehensive procedural interface which enables creation and manipulation of ATerms in C and Java. The ATerm implementation is based on maximal subterm sharing and automatic garbage collection.

We are involved (with the CWI) in the implementation of the Java version, as well as in the garbage collector of the C version. The Java version of the ATerm library is used in particular by *Tom*.

The ATerm library is documented, maintained, and available at the following address: http://www.meta-environment.org/Meta-Environment/ATerms.

5.2. Tom

Participants: Jean-Christophe Bach, Christophe Calvès, Horatiu Cirstea, Pierre-Etienne Moreau [correspondant].

Since 2002, we have developed a new system called *Tom* [27], presented in [11], [12]. This system consists of a pattern matching compiler which is particularly well-suited for programming various transformations on trees/terms and XML documents. Its design follows our experiments on the efficient compilation of rule-based systems [24]. The main originality of this system is to be language and data-structure independent. This means that the *Tom* technology can be used in a C, C++ or Java environment. The tool can be seen as a Yacc-like compiler translating patterns into executable pattern matching automata. Similarly to Yacc, when a match is found, the corresponding semantic action (a sequence of instructions written in the chosen underlying language) is triggered and executed. *Tom* supports sophisticated matching theories such as associative matching with neutral element (also known as list-matching). This kind of matching theory is particularly well-suited to perform list or XML based transformations for example.

In addition to the notion of *rule*, *Tom* offers a sophisticated way of controlling their application: a strategy language. Based on a clear semantics, this language allows to define classical traversal strategies such as *innermost*, *outermost*, *etc.* Moreover, *Tom* provides an extension of pattern matching, called *anti-pattern matching*. This corresponds to a natural way to specify *complements* (*i.e.*, what should not be there to fire a rule). *Tom* also supports the definition of cyclic graph data-structures, as well as matching algorithms and rewriting rules for term-graphs.

Tom is documented, maintained, and available at http://tom.loria.fr as well as at http://gforge.inria.fr/projects/tom.

SEMAGRAMME Project-Team

5. New Software and Platforms

5.1. ACG Development Toolkit

Participants: Sylvain Pogodalla [correspondent], Philippe de Groote, Jirí Marsík.

In order to support the theoretical work on ACG, we have been developing a support system. The objectives of such a system are twofold:

1. To make possible to implement and experiment grammars the modeling of linguistic phenomena.
2. To make possible to implement and experiment results related to the ACG formalisms. Such results can concern parsing algorithms, type extensions, language extensions, etc.

The ACG Development toolkit development effort is part of the POLYMNIE project (see Section 7.2.1.1). It will support the experimentation and evaluation parts of the project.

The current version of the ACG development toolkit prototype [0] is 1.1. It focuses on providing facilities to develop grammars. To this end, the type system currently implemented is the linear core system plus the (non-linear) intuitionistic implication, and a special attention has been paid to type error management. Since 1.0b released in Feb. 2014, ACGtk allows for transformations both from abstract terms to object terms, and from object terms to abstract terms (ACG parsing). The parsing algorithm follows [64]'s method which is being implemented for second-order ACGs. It is based on a translation of ACG grammars into Datalog programs and is well-suited to fine-grained optimization.

However, since we are interested not only by recognizability (hence whether some fact is provable) but also by the parsing structure (hence the proof), the Datalog solver has been adapted to produce not only yes/no answer to queries, but also all the proofs of the answers to the queries. The next steps concern optimization and efficiency. Note however that in the general case, the decidability of translating an object term to an abstract one is still an open problem.

We also have enriched the ACG development toolkit with graphical output. The new module includes a small functional OCaml library for manipulating images which enables users to customize the rendering of formulas as pictures.

The ACGtk has been made available as an OPAM (OCaml Package Manager) package. [0]

5.2. Grew

Participants: Bruno Guillaume [correspondent], Guy Perrier.

Grew (http://grew.loria.fr) is a Graph Rewriting tool dedicated to applications in NLP. It is freely-available and it is developed using the InriaGforge platform (http://gforge.inria.fr/projects/semagramme/).

Grew takes into account confluent and non-confluent graph rewriting and it includes several mechanisms that help to use graph rewriting in the context of NLP applications (built-in notion of feature structures, parametrization of rules with lexical information).

In 2014, an online version (http://talc2.loria.fr/grew/) of the tool based on the matching part was developped to illustrate its use (it is not possible to modify graphs). The user gives a pattern (eventually with some negative constraints) and Grew searches in a corpus the occurences on the given pattern in: the French corpus Sequoia is available (two versions are available: one containing surface annotation and one with deep annotation 6.4) and the German corpus Tiger is also avalaible for online pattern search.

[0] Available at http://acg.gforge.inria.fr with a CeCILL license.

[0] https://opam.ocaml.org/packages/acgtk/acgtk.1.1/

5.3. Leopar

Participants: Bruno Guillaume [correspondent], Guy Perrier.

Leopar is a parser for natural languages which is based on the formalism of Interaction Grammars [59]. It is open-source (under the CECILL License http://www.cecill.info) and it is developed using the InriaGforge platform (http://gforge.inria.fr/projects/semagramme/).

The main features of current version of the software are:

- automatic parsing of a sentence or a set of sentences,
- dependency and parse-tree representation of sentences,
- interactive parsing (the user chooses the couple of nodes to merge),
- visualization of grammars produced by XMG-2 or of sets of description trees associated to some word in the linguistic resources.

In 2014, a new conversion from parse-tree representation to dependency representation was implemented to take benefit of the linguistic principles that were defined and used in [36].

5.4. ZombiLingo

Participants: Bruno Guillaume [correspondent], Karën Fort.

Zombilingo (http://zombilingo.loria.fr) is a prototype of a GWAP where gamers have to give linguistic information about the syntax of French natural language sentence (see 6.6 for more details).

5.5. Other developments

Participants: Maxime Amblard [correspondent], Bruno Guillaume.

Main topics: data managment, disfluencies and dependency

- Dep2pict (http://dep2pict.loria.fr) is a program for drawing graphical representation of dependency structures of natural language sentences. An online version is available (http://wikilligramme.loria.fr/doku.php/dep2pict:demo). In 2014, the Dep2pict was modified to take into account the modified format mixing surface and deep syntactic information used in deep-sequoia 6.4 .
- A management chain of the transcriptions of interviews for the SLAM project which productes of a full anonymized randomized version of the resources.
- A program based on Distagger (disfluences) and MElt (POS and lemma) and proposes different repartition analyses.

SHACRA Project-Team

4. New Software and Platforms

4.1. SOFA

4.1.1. Description of the SOFA framework

SOFA [0] is an open-source software framework targeted at real-time multi-physics simulation, with an emphasis on medical simulation. The idea of SOFA was initiated by members of the SHACRA team, strongly supported by Inria and still actively developed within the SHACRA team. Based on C++, the SOFA engine provides many algorithms, physiological models and anatomical data, made available within a plugin architecture. With its high level of modularity, SOFA appears to be an efficient tools to benchmark and develop new medical technologies using existing algorithms.

The SOFA framework relies on a multi-model representation which allows to have several representations (e.g. mechanical, thermal and visual) of the same object. Those different representations are connected together through a mechanism called mapping. With this features, it is also possible to have models of very different nature interacting together, for instance rigid bodies, deformable objects, and fluids. CPU and GPU implementations can be transparently combined to exploit the computational power of modern hardware architectures.

SOFA is at the heart of a number of research projects, including cardiac electro-physiology modeling, interventional radiology planning and guidance, planning for cryosurgery and deep brain stimulation, robotics, percutaneous procedures, laparoscopic surgery, non-rigid registration, etc. As proof of its success, SOFA has been downloaded nearly 150,000 times, and is used today by many research groups around the world, as well as a number of companies. The mailing list used to exchange with the community includes several hundreds of researchers, from about 50 different institutions. SOFA is currently used by a number of companies (Siemens Corporate Research, Digital Trainers, Epona Medical, Moog, SenseGraphics, etc.) and also provides the key technology on which our newly created start-up (InSimo) is relying. We strongly believe that today SOFA has become a reference for academic research, and is increasingly gaining recognition for product prototyping and development. The best illustration of this worldwide positioning is the role of SOFA in the challenge set by the HelpMeSee foundation to win the contract for the development of a very ambitious and high-risk project on cataract surgery simulation.

4.1.2. Consortium

At the end of the year 2014, the creation of a consortium SOFA has been enacted. The purpose of this consortium is to define the suitable orientation in terms of development, lead to its achievement while creating a propitious ecosystem for research, industry and for the creation of numerous startups. Beside lead the development of SOFA, this consortium has to maintain the existing code, and last but not least, manage the SOFA community and help it to grow.

4.1.3. SOFA Day after ISBMS'14

On the occasion of the 6th ISBMS conference, we organized a "SOFA Day" giving us a unique opportunity to meet SOFA users from various research institutes or companies, and exchange about the future improvements and development of the engine. We use these occasions to share and discuss with SOFA users, to refine the roadmap and stay tuned with our audience.

4.1.4. A new website

Finally, a new website has been developed during the last month of the year. The final version of the website will be released in spring 2015. The website is a very important tool for the community (especially new users). The SOFA consortium will be in charge of this assignment.

[0] More information about SOFA at http://www.sofa-framework.org

TONUS Team

5. New Software and Platforms

5.1. SeLaLib

The objective of the Selalib project (SEmi-LAgrangian LIBrary) is to develop a well-designed, organized and documented library implementing several numerical methods for kinetic models of plasma physics. Its ultimate goal is to produce gyrokinetic simulations.

Another objective of the library is to provide to physicists easy-to-use gyrokinetic solvers, based on the semi-lagrangian techniques developed by Eric Sonnendrücker and his collaborators in the past CALVI project. The new models and schemes from TONUS are also intended to be incorporated into Selalib.

In addition, the CEA of Cadarache is interested by the development of this library, which picks up and extends many methods implemented in GYSELA, a code developed at CEA Cadarache for simulating turbulence in magnetic fusion plasmas, in particular, in view of the ITER project. Eric Sonnendrücker who is now in Munich continues to work on Selalib. A joint development of Selalib between Strasbourg and Munich allows both partners to benefit of each other's work.

Selalib is a library of FORTRAN modules. The CEA Cadarache has advised this language, because it is widespread in the engineering and physics communities. In this way, we hope that it will be spread among researchers interested in plasma simulations.

Selalib is under GPL license and available on the Inria Forge [0].

5.2. CLAC

CLAC is a generic Discontinuous Galerkin solver, written in C/C++, based on the OpenCL and MPI frameworks. CLAC means “Conservation Laws Approximation on many Cores”.

It is clear now that future computers will be made of a collection of thousands of interconnected multicore processors. Globally it appears as a classical distributed memory MIMD machine. But at a lower level, each of the multicore processors is itself made of a shared memory MIMD unit (a few classical CPU cores) and a SIMD unit (a GPU). When designing new algorithms, it is important to adapt them to this kind of architecture. Our philosophy will be to program our algorithms in such a way that they can be run efficiently on this kind of computers. Practically, we will use the MPI library for managing the coarse grain parallelism, while the OpenCL library will efficiently operate the fine grain parallelism.

We have invested for several years until now into scientific computing on GPUs, using the open standard OpenCL (Open Computing Language). We were recently awarded a prize in the international AMD OpenCL innovation challenge thanks to an OpenCL two-dimensional Vlasov-Maxwell solver that fully runs on a GPU. OpenCL is a very interesting tool because it is an open standard now available on almost all brands of multicore processors and GPUs. The same parallel program can run on a GPU or a multicore processor without modification.

CLAC is also a joint project with a Strasbourg small company, AxesSim, which develops software for electromagnetic simulations.

[0] http://selalib.gforge.inria.fr/

Because of the envisaged applications of CLAC, which may be either academic or commercial, it is necessary to conceive a modular framework. The heart of the library is made of generic parallel algorithms for solving conservation laws. The parallelism can be both fine-grained (oriented towards GPUs and multicore processors) and coarse-grained (oriented towards GPU clusters). The separate modules allow managing the meshes and some specific applications. In this way, it is possible to isolate parts that should be protected for trade secret reasons. The open source part of CLAC will be made freely available on the web later on. We have made an APP deposit of the first version of CLAC in October 2012. The versioning of CLAC project is also registered in the Inria Forge [0].

[0] http://clac.gforge.inria.fr

TOSCA Project-Team

5. New Software and Platforms

5.1. Triton

Participant: Antoine Lejay [correspondant].

The Triton software aims at providing a toolbox to analyze nearshore waves images recorded by a camera on the beach. More precisely, it aims at estimating the height, length and speed of waves, to find speed and direction of currents, and to reconstruct the bathymetry from these images.

This is a joint work with Rafael Almar (LEGOS, IRD, Toulouse) and with Stanislas Larnier (LAAS-CNRS, Toulouse), a former post-doctoral student in the Tosca team.

- Version: 1.0

5.2. SDM

Participants: Mireille Bossy [correspondant], Sélim Karia.

The computation of the wind at small scale and the estimation of its uncertainties is of particular importance for applications such as wind energy resource estimation. To this aim, starting in 2005, we have developed a new method based on the combination of an existing Numerical Weather Prediction model providing a coarse prediction, and a Lagrangian Stochastic Model for turbulent flows. This Stochastic Downscaling Method (SDM) requires a specific modelling of the turbulence closure, and involves various simulation techniques whose combination is totally original (such as Poisson solvers, optimal transportation mass algorithm, original Euler scheme for confined Langevin stochastic processes, and stochastic particle methods).

In 2013, the SDM code became the kernel of the wind farm modelling of the Fundacion Inria Chile. In France, its development is pursuing through the collaborative Modéol project on the evaluation of wind potential.

This is a joint work with Antoine Rousseau from the team LEMON.

- Version: 2.0

5.3. CarbonQuant

Participants: Mireille Bossy [correspondant], Sélim Karia.

CarbonQuant is a simulator project of CO2 allowances prices on a EU-ETS type market, by an indifference price approach.

It aims to demonstrate the high potentiality of stochastic control solvers, to quantify sensibilities of a carbon market with respect to its design.

See also the web page http://carbonvalue.gforge.inria.fr, from where CarbonQuant can be now downloaded for various architectures.

A new version of CarbonQuant is under devellopment that includes a N players game approache on an auction carbon market.

- Version: 2.0

VEGAS Project-Team

4. New Software and Platforms

4.1. QI: Quadrics Intersection

QI stands for “Quadrics Intersection”. QI is the first exact, robust, efficient and usable implementation of an algorithm for parameterizing the intersection of two arbitrary quadrics, given in implicit form, with integer coefficients. This implementation is based on the parameterization method described in [5] [29] and represents the first complete and robust solution to what is perhaps the most basic problem of solid modeling by implicit curved surfaces.

QI is written in C++ and builds upon the LiDIA computational number theory library [20] bundled with the GMP multi-precision integer arithmetic [19]. QI can routinely compute parameterizations of quadrics having coefficients with up to 50 digits in less than 100 milliseconds on an average PC; see [29] for detailed benchmarks.

Our implementation consists of roughly 18,000 lines of source code. QI has being registered at the Agence pour la Protection des Programmes (APP). It is distributed under a free for non-commercial use Inria license and will be distributed under the QPL license in the next release. The implementation can also be queried via a web interface [21].

Since its official first release in June 2004, QI has been downloaded six times a month on average and it has been included in the geometric library EXACUS developed at the Max-Planck-Institut für Informatik (Saarbrücken, Germany). QI is also used in a broad range of applications; for instance, it is used in photochemistry for studying the interactions between potential energy surfaces, in computer vision for computing the image of conics seen by a catadioptric camera with a paraboloidal mirror, and in mathematics for computing flows of hypersurfaces of revolution based on constant-volume average curvature.

4.2. Isotop: Topology and geometry of planar algebraic curves

ISOTOP is a Maple software for computing the topology of an algebraic plane curve, that is, for computing an arrangement of polylines isotopic to the input curve. This problem is a necessary key step for computing arrangements of algebraic curves and has also applications for curve plotting. This software has been developed since 2007 in collaboration with F. Rouillier from Inria Paris - Rocquencourt. It is based on the method described in [3] which incorporates several improvements over previous methods. In particular, our approach does not require generic position.

Isotop is registered at the APP (June 15th 2011). This version is competitive with other implementations (such as ALCIX and INSULATE developed at MPII Saarbrücken, Germany and TOP developed at Santander Univ., Spain). It performs similarly for small-degree curves and performs significantly better for higher degrees, in particular when the curves are not in generic position.

We are currently working on an improved version integrating our new bivariate polynomial solver.

4.3. CGAL: Computational Geometry Algorithms Library

Born as a European project, CGAL (http://www.cgal.org) has become the standard library for computational geometry. It offers easy access to efficient and reliable geometric algorithms in the form of a C++ library. CGAL is used in various areas needing geometric computation, such as: computer graphics, scientific visualization, computer aided design and modeling, geographic information systems, molecular biology, medical imaging, robotics and motion planning, mesh generation, numerical methods...

In computational geometry, many problems lead to standard, though difficult, algebraic questions such as computing the real roots of a system of equations, computing the sign of a polynomial at the roots of a system, or determining the dimension of a set of solutions. We want to make state-of-the-art algebraic software more accessible to the computational geometry community, in particular, through the computational geometric library CGAL. On this line, we contributed a model of the *Univariate Algebraic Kernel* concept for algebraic computations [23] (see Sections 8.2.2 and 8.4). This CGAL package improves, for instance, the efficiency of the computation of arrangements of polynomial functions in CGAL [30]. We are currently developing a model of the *Bivariate Algebraic Kernel* based on a new bivariate polynomial solver.

4.4. Fast_polynomial: fast polynomial evaluation software

The library *fast_polynomial*[0] provides fast evaluation and composition of polynomials over several types of data. It is interfaced for the computer algebra system *Sage* and its algorithms are documented [0]. This software is meant to be a first step toward a certified numerical software to compute the topology of algebraic curves and surfaces. It can also be useful as is and is submitted for integration in the computer algebra system *Sage*.

This software is focused on *fast online computation*, *multivariate evaluation*, *modularity*, and *efficiency*.

Fast online computation. The library is optimized for the evaluation of a polynomial on several point arguments given one after the other. The main motivation is numerical path tracking of algebraic curves, where a given polynomial criterion must be evaluated several thousands of times on different values arising along the path.

Multivariate evaluation. The library provides specialized fast evaluation of multivariate polynomials with several schemes, specialized for different types such as *mpz* big ints, *boost* intervals with hardware precision, *mpfi* intervals with any given precision, etc.

Modularity. The evaluation scheme can be easily changed and adapted to the user needs. Moreover, the code is designed to easily extend the library with specialization over new *C++* objects.

Efficiency. The library uses several tools and methods to provide high efficiency. First, the code uses templates, such that after the compilation of a polynomial for a specific type, the evaluation performance is equivalent to low-level evaluation. Locality is also taken into account: the memory footprint is minimized, such that an evaluation using the classical Hörner scheme will use $O(1)$ temporary objects and divide and conquer schemes will use $O(\log n)$ temporary objects, where n is the degree of the polynomial. Finally, divide and conquer schemes can be evaluated in parallel, using a number of threads provided by the user.

[0] http://trac.sagemath.org/sage_trac/ticket/13358
[0] http://arxiv.org/abs/1307.5655

VERIDIS Project-Team

5. New Software and Platforms

5.1. The veriT Solver

Participants: Haniel Barbosa, David Déharbe, Pablo Federico Dobal, Pascal Fontaine [contact].

The veriT solver is an SMT (Satisfiability Modulo Theories) solver developed in cooperation with David Déharbe from the Federal University of Rio Grande do Norte in Natal, Brazil. The solver can handle large quantifier-free formulas containing uninterpreted predicates and functions, and arithmetic over integers and reals. It features a very efficient decision procedure for uninterpreted symbols, as well as a simplex-based reasoner for linear arithmetic. It also has some support for user-defined theories, quantifiers, and lambda-expressions. This allows users to easily express properties about concepts involving sets, relations, etc. The prover can produce explicit proof traces when it is used as a decision procedure for quantifier-free formulas with uninterpreted symbols and arithmetic. To support the development of the tool, non-regression tests use Inria's grid infrastructure; it allows us to extensively test the solver on thousands of benchmarks in a few minutes. The veriT solver is available as open source under the BSD license at the veriT Web site.

Efforts in 2014 have been focused on efficiency and stability. The decision procedures for uninterpreted symbols and linear arithmetic have been further improved. There has also been some progress in the integration of the solver Redlog (section 5.4) for non-linear arithmetic in the context of the SMArT project (section 8.2).

The veriT solver participated in the SMT competition SMT-COMP 2014, part of the Vienna Summer Of Logic Olympic Games, and received the gold medal for SMT. The success of the different solvers was measured as a combination of the number of benchmark problems solved in the various categories, the number of erroneous answers, and the time taken.

We target applications where validation of formulas is crucial, such as the validation of TLA$^+$ and B specifications, and work together with the developers of the respective verification platforms to make veriT even more useful in practice. The solver is available as a plugin for the Rodin platform for discharging proof obligations generated in Event-B [50]; on a large repository of industrial and academic cases, this SMT-based plugin decreased by 75% the number of proof obligations requiring human interactions, compared to the original B prover.

5.2. The TLA+ Proof System

Participants: Stephan Merz [contact], Hernán Pablo Vanzetto.

TLAPS, the TLA$^+$ proof system developed at the Joint MSR-Inria Centre, is a platform for developing and mechanically verifying proofs about TLA$^+$ specifications. The TLA$^+$ proof language is hierarchical and explicit, allowing a user to decompose the overall proof into independent proof steps. TLAPS consists of a *proof manager* that interprets the proof language and generates a collection of proof obligations that are sent to *backend verifiers*. The current backends include the tableau-based prover Zenon for first-order logic, Isabelle/TLA$^+$, an encoding of TLA$^+$ as an object logic in the logical framework Isabelle, an SMT backend designed for use with any SMT-lib compatible solver, and an interface to a decision procedure for propositional temporal logic.

The current version 1.3.2 of TLAPS was released in May 2014, it is distributed under a BSD-like license at http://tla.msr-inria.inria.fr/tlaps/content/Home.html. The prover fully handles the non-temporal part of TLA$^+$. The SMT backend, developed in Nancy, has been further improved in 2014, in particular through the development of an appropriate type synthesis procedure, and is now the default backend. A new interface with a decision procedure for propositional temporal logic has been developed in 2014, so that simple temporal proof obligations can now be discharged. It is based on a technique for "coalescing" first-order subformulas of temporal logic, described in section 6.2 . The standard proof library has also been further developed, partly in response to the needs of the ADN4SE project on verifying a real-time micro-kernel system (section 7.2).

TLAPS was presented at tutorials at the TLA$^+$ community event organized during ABZ 2014 in Toulouse in June and at the SPES_XT summer school at the University of Twente (The Netherlands) in September.

5.3. SPASS: An Automated Theorem Prover for First-Order Logic With Equality

Participants: Martin Bromberger, Arnaud Fietzke, Thomas Sturm, Marco Voigt, Uwe Waldmann, Christoph Weidenbach [contact].

SPASS is an automated theorem prover based on superposition that handles first-order logic with equality and several extensions for particular classes of theories. It has been developed since the mid-1990s at the Max-Planck Institut für Informatik in Saarbrücken. Version 3.7 is the current stable release; it is distributed under the FreeBSD license at http://www.spass-prover.org.

The next major release of SPASS will mainly focus on improved theory support: many applications of automated deduction require reasoning in first-order logic modulo background theories, in particular some form of arithmetic. In 2014, we have continued our efforts to improve the superposition calculus as well as to develop dedicated arithmetic decision procedures for various arithmetic theories. Our results are:

- specialized reasoning support for finite subsets,
- specialized decision procedures for linear real arithmetic with one quantifier alternation,
- new efficient and complete procedures for (mixed) linear integer arithmetic,
- decidability results and respective procedures for various combinations of linear arithmetic with first-order logic.

5.4. The Redlog Computer Logic System

Participants: Thomas Sturm [contact], Marek Košta.

Redlog is an integral part of the interactive computer algebra system Reduce. It supplements Reduce's comprehensive collection of powerful methods from symbolic computation by supplying more than 100 functions on first-order formulas. Redlog has been publicly available since 1995 and is constantly being improved. The name Redlog stands for Reduce Logic System. Andreas Dolzmann from Schloss Dagstuhl Leibniz-Zentrum is a co-developer of Redlog.

Reduce and Redlog are open-source and freely available under a modified BSD license at http://reduce-algebra.sourceforge.net/. The Redlog homepage is located at http://www.redlog.eu/. Redlog generally works with interpreted first-order logic in contrast to free first-order logic. Each first-order formula in Redlog must exclusively contain atoms from one particular Redlog-supported theory, which corresponds to a choice of admissible functions and relations with fixed semantics. Redlog-supported theories include Nonlinear Real Arithmetic (Real Closed Fields), Presburger Arithmetic, Parametric QSAT, and many more.

Effective quantifier elimination procedures for the various supported theories establish an important class of methods available in Redlog. For the theories supported by Redlog, quantifier elimination procedures immediately yield decision procedures. Besides these quantifier elimination-based decision methods there are specialized, and partly incomplete, decision methods, which are tailored to input from particular fields of application.

In 2014, Redlog made two important steps into distinct but equally important future directions. On the one hand, it integrated for the first time learning strategies, as they are known from CDCL-based SMT solving, into a classical real quantifier elimination procedure, viz. virtual substitution for linear formulas [28]. On the other hand, there was important progress concerning incomplete decision procedures for the reals. A journal submission currently under review describes identification of a Hopf bifurcation for the important MAPK model within less than a minute. The corresponding polynomial relevant for root-finding has dimension 10, total degree 100, and contains more than 850,000 monomials.

Redlog is a widely accepted tool and highly visible in mathematics, informatics, engineering and the sciences. The seminal article on Redlog [4] has received more than 300 citations in the scientific literature so far.