



RESEARCH CENTER
Sophia Antipolis - Méditerranée

FIELD

Activity Report 2014

Section Software

Edition: 2015-03-24

1. ABS Project-Team	4
2. AOSTE Project-Team	6
3. APICS Project-Team	9
4. ASCLEPIOS Project-Team	14
5. ATHENA Project-Team	17
6. AYIN Team (section wide)	20
7. BIOCORE Project-Team	21
8. CASTOR Project-Team	22
9. COATI Project-Team	25
10. COFFEE Project-Team	26
11. DEMAR Project-Team	27
12. DIANA Team	30
13. ECUADOR Project-Team	34
14. FOCUS Project-Team	36
15. GALAAD2 Team	39
16. GEOMETRICA Project-Team	41
17. GRAPHIK Project-Team	44
18. HEPHAISTOS Team	45
19. INDES Project-Team	47
20. LAGADIC Project-Team	50
21. LEMON Team	57
22. MAESTRO Project-Team	60
23. MARELLE Project-Team	61
24. MCTAO Project-Team	62
25. MODEMIC Project-Team	63
26. MORPHEME Project-Team	64
27. NACHOS Project-Team	65
28. NEUROMATHCOMP Project-Team	67
29. OPALE Project-Team	69
30. REVES Project-Team	71
31. SCALE Team	73
32. STARS Project-Team	75
33. TITANE Project-Team	81
34. TOSCA Project-Team	82
35. VIRTUAL PLANTS Project-Team	83
36. WIMMICS Project-Team	85
37. ZENITH Project-Team	88

ABS Project-Team

4. New Software and Platforms

4.1. Software

Until October 2014, ABS was distributing isolated programs to solve selected tasks in computational structural biology, including:

- `vorpatch` and `compatch`: Modeling and Comparing Protein Binding Patches,
- `intervor`: Modeling Macro-molecular Interfaces,
- `vorlume`: Computing Molecular Surfaces and Volumes with Certificates,
- `ESBTL`: the Easy Structural Biology Template Library.

This software has been completely repackaged within the *Structural Bioinformatics Library*, a C++ library developed in the scope of an Inria supported *ADT*. The SBL will be released early 2015. Below, we briefly review its spirit and contents.

The Structural Bioinformatics Library (SBL): overview. The Structural Bioinformatics Library (SBL) is a generic C++/python library providing combinatorial, geometric and topological tools to solve problems in computational structural biology (CSB). Its design is meant to accommodate both the variety of models coding the physical and chemical properties of macro-molecular systems, and the variety of operations undertaken on these models. The models supported either consist of unions of balls (van der Waals models, solvent accessible models), or representations of conformations based on Cartesian or internal coordinates (distances and angles between the atoms). The operations provided revolve around the problem of understanding the relationship between the structure and the function of macro-molecules and their complexes, and deal with complementary aspects, namely geometric, topological, and combinatorial methods are used to foster our understanding of bio-physical and biological properties. Software development in this context is especially challenging due to the interactions between these complex models and operations.

To accommodate this complexity, software components of the SBL are organized into four categories:

- **SBL-APPLICATIONS:** end-user applications solving specific applied problems.
- **SBL-CORE:** low-level generic C++ classes templated by traits classes specifying C++ concepts⁰.
- **SBL-MODELS:** C++ *models* matching the C++ concepts required to instantiate classes from SBL-CORE.
- **SBL-MODULES:** C++ classes instantiating classes from the SBL-CORE with specific biophysical models from SBL-MODELS. A module may be seen as a black box transforming an input into an output. With modules, an application workflow consists of interconnected modules.

The SBL for end-users. End users will find in the SBL portable applications running on Linux, and MacOS. These applications split into the following categories:

- **Space Filling Models:** applications dealing with molecular models defined by unions of balls. Current statistics are:
 - # classes: 151
 - # lines of C++/python: 65,000
 - # pages of documentation (user + reference manuals): ~ 1000
- **Conformational Analysis:** applications dealing with molecular flexibility. Current statistics are:
 - # classes: 110

⁰The design has been guided by that used in the Computational Geometry Algorithm Library (CGAL), see <http://www.cgal.org>

- # lines of C++/python: 49,000
- # pages of documentation (user + reference manuals): ~ 800
- **Data Analysis:** applications to handle input data and results, using standard tools revolving around the XML file format (in particular the XPath query language). These tools allow automating data storage, parsing and retrieval, so that upon running calculations with applications, statistical analysis and plots are a handful of python lines away.
- **Large assemblies:** applications dealing with macro-molecular assemblies involving from tens to hundreds of macro-molecules.

The SBL for developers. Development with the SBL may occur at two levels.

Low level developments may use classes from SBL-CORE and SBL-MODELS. In fact, such developments are equivalent to those based upon C++ libraries such as CGAL (<http://www.cgal.org/>) or boost C++ libraries (<http://www.boost.org/>). It should be noticed that the SBL heavily relies on these libraries. The SBL-CORE is organized into four sub-sections:

- CADS : Combinatorial Algorithms and Data Structures.
- GT : Computational geometry and computational topology.
- CSB : Computational Structural Biology.
- IO : Input / Output.

It should also be stressed that these packages implement algorithms not available elsewhere, or available in a non-generic guise. Due to the modular structure of the library, should valuable implementations be made available outside the SBL (e.g. in CGAL or boost), a substitution may occur.

Intermediate level developments should be based upon modules, since modules allow the development of applications without the burden of instantiating low level classes. In fact, once modules are available, designing an application merely consists of connecting modules.

Interoperability. The SBL is interoperable with existing molecular modeling systems, at several levels:

- At the library level, our state-of-the-art algorithms (e.g. the computation of molecular surfaces and volumes) can be integrated within existing software (e.g. molecular dynamics software), by instantiating the required classes from SBL-CORE, or using the adequate modules.
- At the application level, our applications can easily be integrated within processing pipelines, since the format used for input and output are standard ones. (For input, the PDB format can always be used. For output, our applications generate XML files.)
- Finally, for visualization purposes, our applications generate outputs for the two reference molecular modeling environments, namely Visual Molecular Dynamics (<http://www.ks.uiuc.edu/Research/vmd/>) and Pymol (<http://www.pymol.org/>).

Releases, distribution, and licence. The SBL will be released under a proprietary open source licence. In a nutshell, academic users can use and modify the code at their discretion, for private purposes. But distributing these changes, or doing business with the SBL is forbidden. However, novel capabilities matching the design choices of the library will be welcome, and may be integrated.

The source code will be distributed from <http://structural-bioinformatics-library.org/>, as a tarball and also via a git repository. Bugzilla will be used to handle user's feedback and bug tracking.

The releases are scheduled as follows:

- February 2015: applications from the *space filling model* group, and the accompanying low level classes.
- April 2015: applications from *conformational analysis* group, and the accompanying low level classes.
- July 2015: applications from *large assemblies* group, and the accompanying low level classes.

AOSTE Project-Team

5. New Software and Platforms

5.1. TimeSquare

Participants: Nicolas Chleq, Julien Deantoni, Frédéric Mallet [correspondant].

TimeSquare is a software environment for the modeling and analysis of timing constraints in embedded systems. It relies specifically on the Time Model of the MARTE UML profile (see section 3.2), and more accurately on the associated Clock Constraint Specification Language (CCSL) for the expression of timing constraints.

TimeSquare offers five main functionalities:

1. graphical and/or textual interactive specification of logical clocks and relative constraints between them;
2. definition and handling of user-defined clock constraint libraries;
3. automated simulation of concurrent behavior traces respecting such constraints, using a Boolean solver for consistent trace extraction;
4. call-back mechanisms for the traceability of results (animation of models, display and interaction with waveform representations, generation of sequence diagrams...).
5. compilation to pure java code to enable embedding in non eclipse applications or to be integrated as a time and concurrency solver within an existing tool.

In practice TimeSquare is a set of plug-ins developed for the Eclipse modeling framework. The software is registered by the *Agence pour la Protection des Programmes*, under number IDDN.FR.001.170007.000.S.P.2009.001.10600. It can be downloaded from the site <http://timesquare.inria.fr/>. It has been integrated in the **OpenEmbeDD** ANR RNTL platform and recently in the **Gemoc Studio**.

5.2. K-Passa

Participants: Jean-Vivien Millo [correspondant], Robert de Simone.

This software is dedicated to the simulation, analysis, and static scheduling of Event/Marked Graphs, SDF and KRG extensions. A graphical interface allows to edit the Process Networks and their time annotations (*latency*, ...). Symbolic simulation and graph-theoretic analysis methods allow to compute and optimize static schedules, with best throughputs and minimal buffer sizes. In the case of KRG the (ultimately k-periodic) routing patterns can also be provided and transformed for optimal combination of switching and scheduling when channels are shared. KPASSA also allows for import/export of specific description formats such as UML-MARTE, to and from our other TimeSquare tool.

The tool was originally developed mainly as support for experimentations following our research results on the topic of Latency-Insensitive Design. This research was conducted and funded in part in the context of the CIM PACA initiative, with initial support from ST Microelectronics and Texas Instruments.

KPASSA is registered by the *Agence pour la Protection des Programmes*, under the number IDDN.FR.001.310003.000.S.P.2009.000.20700. It can be downloaded from the site <http://www-sop.inria.fr/aoste/index.php?page=software/kpassa>.

5.3. SynDEx

Participants: Yves Sorel [correspondant], Meriem Zidouni.

SynDEx is a system level CAD software implementing the AAA methodology for rapid prototyping and for optimizing distributed real-time embedded applications. Developed in OCaml it can be downloaded free of charge, under Inria copyright, from the general SynDEx site <http://www.syndex.org>.

The AAA methodology is described in section 3.3. Accordingly, SYNDEX explores the space of possible allocations (spatial distribution and temporal scheduling), from application elements to architecture resources and services, in order to match real-time requirements; it does so by using schedulability analyses and heuristic techniques. Ultimately it generates automatically distributed real-time code running on real embedded platforms. The last major release of SYNDEX (V7) allows the specification of multi-periodic applications.

Application algorithms can be edited graphically as directed acyclic task graphs (DAG) where each edge represents a data dependence between tasks, or they may be obtained by translations from several formalisms such as Scicos (<http://www.scicos.org>), Signal/Polychrony (<http://www.irisa.fr/espresso/Polychrony/download.php>), or UML2/MARTE models (http://www.omg.org/technology/documents/profile_catalog.htm).

Architectures are represented as graphical block diagrams composed of programmable (processors) and non-programmable (ASIC, FPGA) computing components, interconnected by communication media (shared memories, links and busses for message passing). In order to deal with heterogeneous architectures it may feature several components of the same kind but with different characteristics.

Two types of non-functional properties can be specified for each task of the algorithm graph. First, a period that does not depend on the hardware architecture. Second, real-time features that depend on the different types of hardware components, ranging amongst *execution and data transfer time, memory, etc.*. Requirements are generally constraints on deadline equal to period, latency between any pair of tasks in the algorithm graph, dependence between tasks, etc.

Exploration of alternative allocations of the algorithm onto the architecture may be performed manually and/or automatically. The latter is achieved by performing real-time multiprocessor schedulability analyses and optimization heuristics based on the minimization of temporal or resource criteria. For example while satisfying deadline and latency constraints they can minimize the total execution time (makespan) of the application onto the given architecture, as well as the amount of memory. The results of each exploration is visualized as timing diagrams simulating the distributed real-time implementation.

Finally, real-time distributed embedded code can be automatically generated for dedicated distributed real-time executives, possibly calling services of resident real-time operating systems such as Linux/RTAI or Osek for instance. These executives are deadlock-free, based on off-line scheduling policies. Dedicated executives induce minimal overhead, and are built from processor-dependent executive kernels. To this date, executive kernels are provided for: TMS320C40, PIC18F2680, i80386, MC68332, MPC555, i80C196 and Unix/Linux workstations. Executive kernels for other processors can be achieved at reasonable cost following these examples as patterns.

5.4. Lopht

Participants: Thomas Carle, Manel Djemal, Dumitru Potop Butucaru [correspondant].

The Lopht (Logical to Physical Time Compiler) has been designed as an implementation of the AAA methodology. Like SynDEx, Lopht relies on off-line allocation and scheduling techniques to allow real-time implementation of dataflow synchronous specifications onto multiprocessor systems. But there are several originality points: a stronger focus on efficiency, which results in the use of a compilation-like approach, a focus on novel target architectures (many-core chips and time-triggered embedded systems), and the possibility to handle multiple, complex non-functional requirements covering real-time (release dates and deadlines possibly different from period, major time frame, end-to-end flow constraints), ARINC 653 partitioning, the possibility to preempt or not each task, and finally SynDEx-like allocation.

Improved efficiency is attained through the use of classical and novel data structures and optimization algorithms pertaining to 3 fields: synchronous language compilation, classical compiler theory, and real-time scheduling. A finer representation of execution conditions allows us to make a better use of double

resource reservation and thus improve latency and throughput. The use of software pipelining allows the improvement of computation throughput. The use of post-scheduling optimizations allows a reduction in the number of preemptions. The focus on novel architectures means that architecture descriptions need to define novel communication media such as the networks-on-chips (NoCs), and that real-time characteristics must include those specific to a time-triggered execution model, such as the Major Time Frame (MTF). Attaining efficiency also requires a fine-grain description of more classical platform resources, such as the multi-bank RAMs, to allow efficient allocation during scheduling.

Significant contributions to the Lopht tool have been brought by T. Carle (the extensions concerning time-triggered platforms), M. Djemal (the extensions concerning many-core platforms), and Zhen Zhang under the supervision of D. Potop Butucaru. The tool has been used and extended during the PARSEC project. It is currently used in the direct collaboration with Airbus Defence and Space and the CNES, in the IRT SystemX/FSF project, and in the CAPACITES project. It has been developed in OCaml.

5.5. SAS

Participants: Daniel de Rauglaudre [correspondant], Yves Sorel.

The SAS (Simulation and Analysis of Scheduling) software allows the user to perform the schedulability analysis of periodic task systems in the monoprocessor case.

The main contribution of SAS, when compared to other commercial and academic softwares of the same kind, is that it takes into account the exact preemption cost between tasks during the schedulability analysis. Beside usual real-time constraints (precedence, strict periodicity, latency, etc.) and fixed-priority scheduling policies (Rate Monotonic, Deadline Monotonic, Audsley⁺⁺, User priorities), SAS additionally allows to select dynamic scheduling policy algorithms such as Earliest Deadline First (EDF). The resulting schedule is displayed as a typical Gantt chart with a transient and a permanent phase, or as a disk shape called "dameid", which clearly highlights the idle slots of the processor in the permanent phase.

For a schedulable task system under EDF, when the exact preemption cost is considered, the period of the permanent phase may be much longer than the least common multiple (LCM) of the periods of all tasks, as often found in traditional scheduling theory. Specific effort has been made to improve display in this case. The classical utilization factor, the permanent exact utilization factor, the preemption cost in the permanent phase, and the worst response time for each task are all displayed when the system is schedulable. Response times of each task relative time can also be displayed (separately).

SAS is written in OCaml, using CAMLP5 (syntactic preprocessor) and OLIBRT (a graphic toolkit under X). Both are written by Daniel de Rauglaudre. It can be downloaded from the site <http://pauillac.inria.fr/~ddr/sas-dameid/>.

APICS Project-Team

5. New Software and Platforms

5.1. RARL2

Participant: Martine Olivi [corresponding participant].

Status: Currently under development. A stable version is maintained.

This software is developed in collaboration with Jean-Paul Marmorat (Centre de mathématiques appliquées (CMA), École des Mines de Paris).

RARL2 (Réalisation interne et Approximation Rationnelle L2) is a software for rational approximation (see Section 3.3.2.2) <http://www-sop.inria.fr/apics/RARL2/rarl2.html>.

The software RARL2 computes, from a given matrix-valued function in $\overline{H}^{2m \times l}$, a local best rational approximant in the L^2 norm, which is *stable and of prescribed McMillan degree* (see Section 3.3.2.2). It was initially developed in the context of linear (discrete-time) system theory and makes an heavy use of the classical concepts in this field. The matrix-valued function to be approximated can be viewed as the transfer function of a multivariable discrete-time stable system. RARL2 takes as input either:

- its internal realization,
- its first N Fourier coefficients,
- discretized (uniformly distributed) values on the circle. In this case, a least-square criterion is used instead of the L^2 norm.

It thus performs model reduction in case 1) and 2) and frequency data identification in case 3). In the case of band-limited frequency data, it could be necessary to infer the behavior of the system outside the bandwidth before performing rational approximation (see Section 3.2.2). An appropriate Möbius transformation allows to use the software for continuous-time systems as well.

The method is a steepest-descent algorithm. A parametrization of MIMO systems is used, which ensures that the stability constraint on the approximant is met. The implementation, in Matlab, is based on state-space representations.

The number of local minima can be large so that the choice of an initial point for the optimization may play a crucial role. In this connection, two methods can be used: 1) An initialization with a best Hankel approximant. 2) An iterative research strategy on the degree of the local minima, similar in principle to that of RARL2, increases the chance of obtaining the absolute minimum by generating, in a structured manner, several initial conditions.

RARL2 performs the rational approximation step in our applications to filter identification (see Section 4.5) as well as sources or cracks recovery (see Section 4.2). It was released to the universities of Delft, Maastricht, Cork, Brussels and Macao. The parametrization embodied in RARL2 was also used for a multi-objective control synthesis problem provided by ESTEC-ESA, The Netherlands. An extension of the software to the case of triple poles approximants is now available. It is used by FindSources3D (see Section 5.6).

5.2. RGC

Participant: Fabien Seyfert [corresponding participant].

Status: A stable version is maintained.

This software is developed in collaboration with Jean-Paul Marmorat (Centre de mathématiques appliquées (CMA), École des Mines de Paris).

The identification of filters modeled by an electrical circuit that was developed by the team (see Section 4.5) led us to compute the electrical parameters of the underlying filter. This means finding a particular realization (A, B, C, D) of the model given by the rational approximation step. This 4-tuple must satisfy constraints that come from the geometry of the equivalent electrical network and translate into some of the coefficients in (A, B, C, D) being zero. Among the different geometries of coupling, there is one called “the arrow form” [57] which is of particular interest since it is unique for a given transfer function and is easily computed. The computation of this realization is the first step of RGC. Subsequently, if the target realization is not in arrow form, one can nevertheless show that it can be deduced from the arrow-form by a complex-orthogonal change of basis. In this case, RGC starts a local optimization procedure that reduces the distance between the arrow form and the target, using successive orthogonal transformations. This optimization problem on the group of orthogonal matrices is non-convex and has many local and global minima. In fact, there is not even uniqueness of the filter realization for a given geometry. Moreover, it is often relevant to know all solutions of the problem, because the designer is not even sure, in many cases, which one is being handled. The assumptions on the reciprocal influence of the resonant modes may not be equally well satisfied for all such solutions, hence some of them should be preferred for the design. Today, apart from the particular case where the arrow form is the desired form (this happens frequently up to degree 6) the RGC software is not guaranteed to provide a solution. In contrast, the software Dedale-HF (see Section 5.4), which is the successor of RGC, is guaranteed to solve this constraint realization problem.

5.3. PRESTO-HF

Participant: Fabien Seyfert [corresponding participant].

Status: Currently under development. A stable version is maintained.

PRESTO-HF: a toolbox dedicated to lowpass parameter identification for microwave filters <http://www-sop.inria.fr/apics/Presto-HF>. In order to allow the industrial transfer of our methods, a Matlab-based toolbox has been developed, dedicated to the problem of identification of low-pass microwave filter parameters. It allows one to run the following algorithmic steps, either individually or in a single shot:

- determination of delay components caused by the access devices (automatic reference plane adjustment),
- automatic determination of an analytic completion, bounded in modulus for each channel,
- rational approximation of fixed McMillan degree,
- determination of a constrained realization.

For the matrix-valued rational approximation step, Presto-HF relies on RARL2 (see Section 5.1). Constrained realizations are computed by the RGC software. As a toolbox, Presto-HF has a modular structure, which allows one for example to include some building blocks in an already existing software.

The delay compensation algorithm is based on the following assumption: far off the passband, one can reasonably expect a good approximation of the rational components of S_{11} and S_{22} by the first few terms of their Taylor expansion at infinity, a small degree polynomial in $1/s$. Using this idea, a sequence of quadratic convex optimization problems are solved, in order to obtain appropriate compensations. In order to check the previous assumption, one has to measure the filter on a larger band, typically three times the pass band.

This toolbox is currently used by Thales Alenia Space in Toulouse, Thales airborne systems and a license agreement has been recently negotiated with TAS-España. XLIM (University of Limoges) is a heavy user of Presto-HF among the academic filtering community and some free license agreements are currently being considered with the microwave department of the University of Erlangen (Germany) and the Royal Military College (Kingston, Canada). A time-limited license has been bought by Flextronics for testing purposes.

5.4. Dedale-HF

Participant: Fabien Seyfert [corresponding participant].

Status: Currently under development. A stable version is maintained.

Dedale-HF is a software dedicated to solve exhaustively the coupling matrix synthesis problem in reasonable time for the filtering community. Given a coupling topology, the coupling matrix synthesis problem (C.M. problem for short) consists in finding all possible electromagnetic coupling values between resonators that yield a realization of given filter characteristics. Solving the latter problem is crucial during the design step of a filter in order to derive its physical dimensions as well as during the tuning process where coupling values need to be extracted from frequency measurements (see Figure 3).

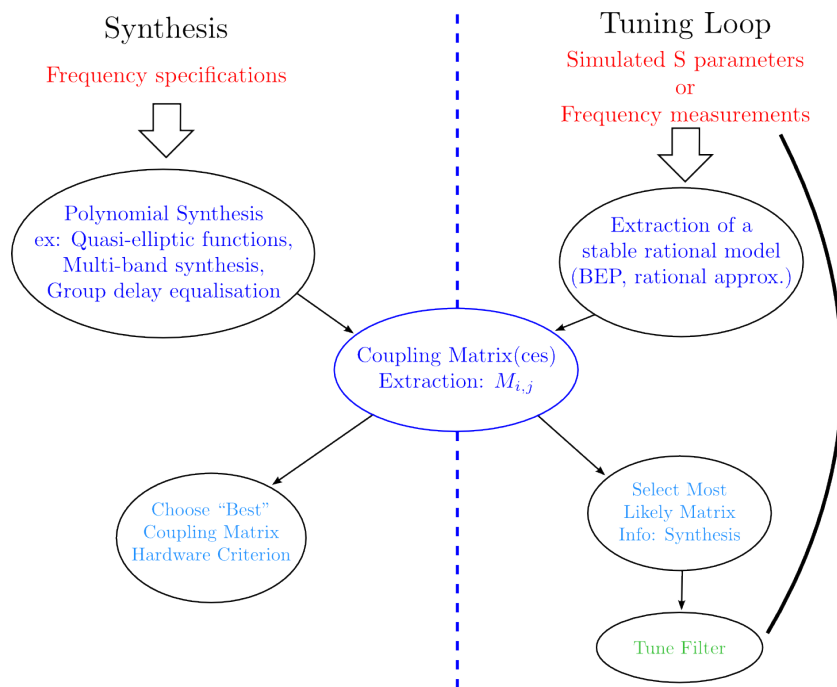


Figure 3. Overall scheme of the design and tuning process of a microwave filter.

Dedale-HF consists in two parts: a database of coupling topologies as well as a dedicated predictor-corrector code. Roughly speaking each reference file of the database contains, for a given coupling topology, the complete solution to the C.M. problem associated to particular filtering characteristics. The latter is then used as a starting point for a predictor-corrector integration method that computes the solution to the C.M. corresponding to the user-specified filter characteristics. The reference files are computed off-line using Gröbner basis techniques or numerical techniques based on the exploration of a monodromy group. The use of such continuation techniques, combined with an efficient implementation of the integrator, drastically reduces the computational time.

Access to the database and integrator code is done via the web on <http://www-sop.inria.fr/apics/Dedale/WebPages>. The software is free of charge for academic research purposes: a registration is however needed in order to access full functionality. Up to now 90 users have registered world wide (mainly: Europe, U.S.A, Canada and China) and 4000 reference files have been downloaded.

A license for this software has been sold end of 2011 to TAS-Espagna, in order to tune filters with topologies having multiple solutions. For this, Dedale-HF teams up with Presto-HF.

5.5. easyFF

Participant: Fabien Seyfert.

Status: A stable version is maintained.

This software has been developed by Vincent Lunot (Taiwan Univ.) during his PhD. He still continues to maintain it.

EasyFF is a software dedicated to the computation of complex, in particular multi-band filtering functions. The software takes as input, specifications on the modulus of the scattering matrix (transmission and rejection), the filter's order and the number of transmission zeros. The output is an "optimal" filtering characteristic in the sense that it is the solution of an associated min-max Zolotarev problem. Computations are based on a Remez-type algorithm (if transmission zeros are fixed) or on linear programming techniques if transmission zeros are part of the optimization [11].

5.6. FindSources3D

Participant: Juliette Leblond [corresponding participant].

Status: Currently under development. A stable version is maintained.

This software is developed in collaboration with Maureen Clerc and Théo Papadopoulo from the Athena Project-Team, and with Jean-Paul Marmorat (Centre de mathématiques appliquées - CMA, École des Mines de Paris).

FindSources3D⁰ is a software dedicated to source recovery for the inverse EEG problem, in 3-layer spherical settings, from point-wise data (see <http://www-sop.inria.fr/apics/FindSources3D/>). Through the algorithm described in [9] and Section 4.2, it makes use of the software RARL2 (Section 5.1) for the rational approximation step in plane sections.

A new release of FindSources3D is now available, which will be demonstrated and distributed, in particular to the medical team we maintain contact with (hosp. la Timone, Marseille). The preliminary step ("cortical mapping") is now solved using expansion in spherical harmonics, along with a constrained approximation scheme.

Another release is being prepared, due to strong interest by the German company BESA GmbH⁰, which develops EEG software for research and clinical applications. A deeper collaboration with this company started last year. Figure 4 shows good results on a two sources distribution recovered by FindSources3D from values of the potential at electrodes on a sphere (scalp) generated by BESA's simulator. There, the localization error is satisfactory, see [28]. Altogether FindSources3D provides suitable initial guess to heavier dedicated recovery tools, including an estimate of the number of sources see Section 6.1.1.

5.7. Sollya

Participant: Sylvain Chevillard [corresponding participant].

Status: Currently under development. A stable version is maintained.

This software is developed in collaboration with Christoph Lauter (LIP6) and Mioara Joldeş (LAAS).

Sollya is an interactive tool where the developers of mathematical floating-point libraries (libm) can experiment before actually developing code. The environment is safe with respect to floating-point errors, *i.e.* the user precisely knows when rounding errors or approximation errors happen, and rigorous bounds are always provided for these errors.

⁰CeCILL license, APP version 2.0 (2012): IDDN.FR.001.45009.001.S.A.2009.000.10000

⁰<http://www.besa.de/>

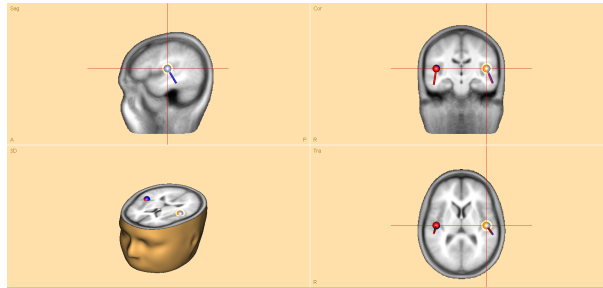


Figure 4. Recovered 2 sources by FindSources3D (courtesy of BESA).

Among other features, it offers a fast Remez algorithm for computing polynomial approximations of real functions and also an algorithm for finding good polynomial approximants with floating-point coefficients to any real function. As well, it provides algorithms for the certification of numerical codes, such as Taylor Models, interval arithmetic or certified supremum norms.

It is available as a free software under the CeCILL-C license at <http://sollya.gforge.inria.fr/>.

ASCLEPIOS Project-Team

4. New Software and Platforms

4.1. SOFA

Participants: Hervé Delingette [correspondent], Federico Spadoni, Stéphanie Marchesseau, Hugo Talbot, Sophie Giffard-Roisin, Roch-Philippe Mollero.

SOFA is an Open Source framework primarily targeted at real-time simulation, with an emphasis on medical simulation. It is mostly intended for the research community to help develop new algorithms, but can also be used as an efficient prototyping tool. Based on an advanced software architecture, it allows : the creation of complex and evolving simulations by combining new algorithms with algorithms already included in SOFA; the modification of most parameters of the simulation (deformable behavior, surface representation, solver, constraints, collision algorithm, etc.) by simply editing an XML file; the building of complex models from simpler ones using a scene-graph description; the efficient simulation of the dynamics of interacting objects using abstract equation solvers; the reuse and easy comparison of a variety of available methods. It was developed mainly by the Inria team projects Shacra, Evasion and Asclepios.

See also the web page <http://www.sofa-framework.org/>.

- ACM: J.2 Physics, J.3 LIFE AND MEDICAL SCIENCES
- Software benefit:- Simulation of the human body
- License: LGPL
- Type of human computer interaction: console, opengl, qt
- OS/Middleware: linux, windows, mac
- Required library or software: Qt - GPL - GLEW - BSD/MIT - Tinyxml - zlib
- Programming language: C/C++
- Documentation: - each function of the core API and each class in the SOFA modules - doxygen
- ACM: J.3
- Programming language: C/C++

4.2. MedInria

Participants: Maxime Sermesant [correspondent], Florian Vichot, Hakim Fadil, Loïc Cadour, Michael Buckingham.

MedInria is a medical imaging software platform developed by the Asclepios research project in collaboration with the Athena, Parietal and Visages Inria research projects. It aims at providing clinicians with state-of-the-art algorithms dedicated to medical image processing and visualization. Efforts have been made to simplify the user interface, while keeping high-level algorithms.

The core of medInria is open source with a BSD license; additional plug-ins can have any license.

The latest release of medInria, 2.2.1, was made in September 2014. See also the web page <https://med.inria.fr>.

- Version: 2.2.1
- License: BSD
- Keywords: Medical Image Processing
- Dependencies: Qt, DTK, VTK, ITK, TTK, MIPS
- Programming language: C++
- Supported OSes: Windows (XP/Vista/7/8), Linux (Fedora/Ubuntu), Mac OS X (10.6-10.9)

4.3. MUSIC

Participants: Maxime Sermesant [correspondent], Florian Vichot, Hakim Fadil, Loïc Cadour, Florent Collot, Mathilde Merle [Software Engineer IHU LIRYC].

MUSIC is a software developed by the Asclepios research project in close collaboration with the IHU LIRYC in order to propose functionalities dedicated to cardiac interventional planning and guidance. This includes specific tools (algorithms of segmentation [1](#), registration, etc.) as well as pipelines. The software is based on the MedInria platform.

For more information, see the web page <https://team.inria.fr/asclepios/software/music/>. See also: <http://videotheque.inria.fr/videotheque/media/28294> for a video on the MUSIC software application.

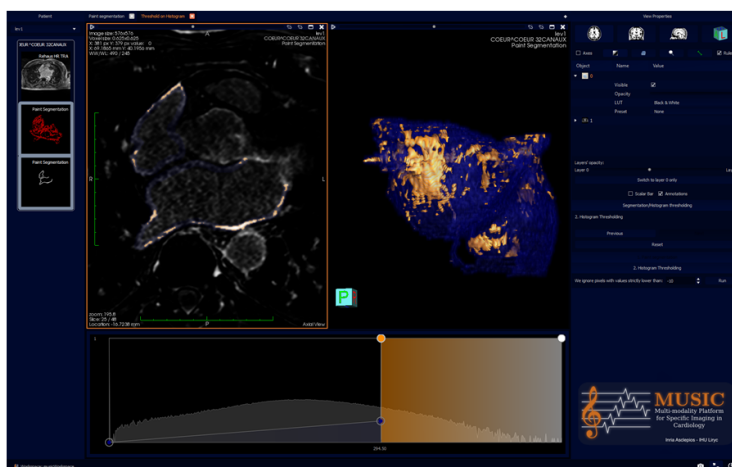


Figure 1. Segmentation of atrial fibrosis using adaptive histogram thresholding based on the MUSIC Software.

- Version: 1.0
- License: Proprietary
- Dependencies: MedInria, Qt, DTK, VTK, ITK, TTK, MIPS
- Programming language: C++
- Supported OSes: Windows (XP/Vista/7/8), Linux (Fedora/Ubuntu), Mac OS X (10.6-10.10)

4.4. VP2HF platform

Participants: Maxime Sermesant [correspondent], Hakim Fadil, Loïc Cadour.

The VP2HF software is developed by the Asclepios team and brings together all the research produced by the VP2HF's partners. It contains MedInria plugins implemented by teams such as UPF Barcelona, KCL, and specific tools provided by Philips (algorithms of segmentation [2](#), scar segmentation, ...). It aims at integrating in a single clinical workflow, tools to improve the therapy selection and treatment optimisation for patients suffering from heart failure.

- Version: 1.0
- License: Proprietary
- Keywords: Medical Image Processing
- Dependencies: MedInria, Qt, DTK, VTK, ITK, TTK, MIPS
- Programming language: C++
- Supported OSes: Windows (XP/Vista/7/8), Linux (Fedora/Ubuntu), Mac OS X (10.6-10.10)

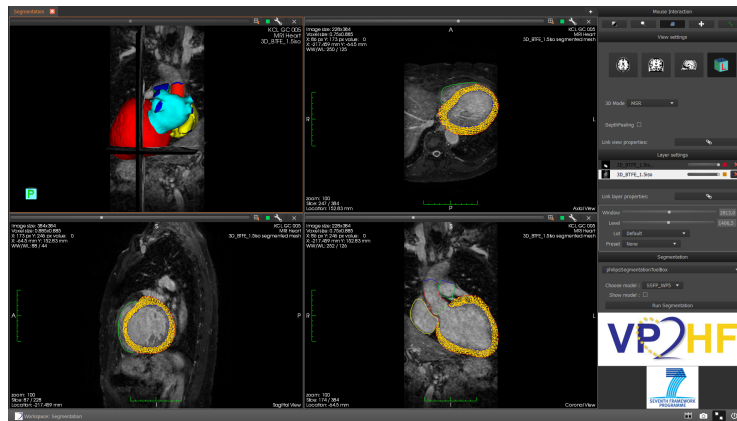


Figure 2. Philips segmentation tool within the VP2HF platform

ATHENA Project-Team

5. New Software and Platforms

5.1. OpenMEEG

Participants: Théodore Papadopoulo, Maureen Clerc, Kai Dang, Alexandre Gramfort [Telecom ParisTech].

OpenMEEG provides state-of-the art tools for low-frequency bio-electromagnetism, notably solving forward problems related to EEG and MEG [60], [61]. It implements the symmetric BEM which provides excellent accuracy and versatility. OpenMEEG is a free open software written in C++. It can be accessed either through a command line interface or through Python/Matlab interfaces. The first release has been directly downloaded about 600 times since October 2008. Our last release (in September 2011) has been downloaded more than 2000 times to this date. OpenMEEG has been integrated in the neuro-debian distribution (<http://neuro.debian.net/>) and matlab suites (such as BrainStorm, FieldTrip or SPM) which may represent many more indirect downloads. Work is under progress to integrate it into the BESA commercial software, and discussions with other software companies are also ongoing.

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

- Version: 2.2
- License: French opensource license CeCILL-B
- Multiplatform: Windows - Linux - MacOSX
- Programming language: C++
- 17 000 lines of code.
- 1800 downloads in 2012-2013.
- Web: <http://openmeeg.gforge.inria.fr>

5.2. High Performance Diffusion MRI

Participants: Aurobrata Ghosh, Théodore Papadopoulo, Rachid Deriche.

We have been closely involved in pushing the frontiers of the diffusion MRI (dMRI) in the recent years, especially in the mathematical modelling and processing of the dMRI signal and have developed state-of-the-art software implementations in the form of a C++ library that can be effectively used to infer the complex microstructure of the cerebral white matter. These algorithms and software fall into four categories : (i) local tissue modelling, which includes both popular 2nd order models and advanced higher than 2nd order models such as DTI, higher order Cartesian tensors (HOTs), ODF, FOD, EAP, maxima extraction, regularization and segmentation; (ii) generation of scalar indices (or biomarkers), which include DTI biomarkers, Diffusion Kurtosis Imaging (DKI) and invariants of 4th order tensors; (iii) global structure estimation, which includes deterministic and probabilistic tractography; and (iv) data visualisation for scalar indices, local models and global structures.

So far, ODF estimation from the ATHENA-dMRI C++ library has been successfully included in medInria 1.9, and in the process to be re-adapted for medInria 2.1. Otherwise, the ATHENA-dMRI C++ library has been mostly used internally for research purposes. However, this is now changing with a fresh restructuring of the entire library so that it can be successfully ported and used externally – primarily to be included in parts with the cutting-edge software developed by OLEA MEDICAL.

- License: French opensource license CeCILL-B - To change when it is to be sourced to OLEA MEDICAL.
- Platform: Linux and (medInria platforms)
- Programming language: C++

5.3. Contributions to the open source dMRI platform DIPY

Participants: Demian Wassermann, Rutger Fick.

DIPY (Diffusion Imaging in Python) is a fast growing open source platform for dMRI image processing. It aims to be a reference implementation platform for most dMRI processing technologies and it has several contributors around the world including Stanford University, USA; Berkeley University, USA; Sherbrooke University, Canada; and University of Cambridge, UK. This aims to provide a dMRI library easy to use in research-intensive cases where developments of new technologies are simpler than in high performance C++ libraries.

In 2014 D. Wassermann and R. Fick got involved in this open source platform. Their work spans from minor public extensions to private developments within this framework. They developed an improved implementation of the 3D-SHORE [72] basis, which is designed to reconstruct the three-dimensional diffusion propagator from three-dimensional q-space measurements. Moreover, they optimized the computation of the basis coefficients and introduced the analytical Laplacian regularization [19]. They also implemented the MAP-MRI basis [73], which is an extension of the 3D-SHORE basis to better deal with highly anisotropic data. Finally, they extended this work by again introducing the analytical Laplacian regularization. Also, we implemented a novel generalized basis that fits diffusion MRI data over both three-dimensional q-space and diffusion times (3D+t). The theoretical developments related to these two last contributions have been submitted to ISBI 2015 and IPMI 2015 respectively.

- License: Revised BSD license.
- Platform: Multiplatform
- Programming language: Python & C

5.4. medInria

Participants: Jaime Garcia Guevara, Théodore Papadopoulo.

The ATHENA team is heavily involved in the development of **medInria** 2.0 along with the ASCLEPIOS, PARIETAL and VISAGES research teams. **medInria** is a free software platform dedicated to medical data visualization and processing. **medInria** 2.0, it is a complete re-write of the first version of medInria in order to be modular and allow a distributed development. It aims at providing an integrative platform for medical image processing and to be a framework for disseminating various research tools not only to other researchers but also to clinicians. New algorithms or data formats can be added as plugins.

It aims at providing to clinicians and researchers state-of-the-art algorithms developed at Inria and elsewhere (for the future), through an intuitive user interface. **medInria** offers from standard to cutting-edge processing functionalities for medical images such as 2D/3D/4D image visualization, image registration, diffusion MR processing and tractography.

ATHENA's contributions so far consist in various improvements on the infrastructure, the core application as well as several plugins which are already available with version 2.1 (ODF visualization) or in future ones: advanced dMRI processing, M/EEG signal visualisation (by integrating code from the software AnyWave developed by Bruno Colombet and J.-M. Badier [INSERM U1106 and Aix-Marseille University](#)).

In 2013, the source code of the core of **medInria** was made public. Regular releases and bug fixes are provided on a large number of Linux, Windows and Mac versions, thanks to the Continuous Integration platform proposed at Inria.

After 4 years of important development, **medInria** is now rather mature and can be used as a basis for collaborations and projects. We now receive regular feedback through the forum and the mailing list, from both academic and clinical users.

- Version: 2.1
- Keywords: Medical Image Processing and Visualization
- License: BSD 4

- Multiplatform: Windows - Linux - MacOSX
- Programming language: C++
- 250 000 lines of code.
- 5000 downloads on 2012-2013.
- Web: <http://med.inria.fr>.

5.5. FindSources3D

Participants: Maureen Clerc, Juliette Leblond [APICS project-team], Jean-Paul Marmorat [APICS project-team], Théodore Papadopoulo.

FindSources3D is a Matlab software program dedicated to solving inverse source localization problems in electroencephalography (EEG), and in the future, magnetoencephalography (MEG). FindSources3D implements a new formalism for source localization, based on rational approximations in the complex plane. It is able to estimate, with high precision, and with no a priori on the number of sources, pointwise dipolar current sources within the brain. The head model used is a spherical model with concentric layers of homogenous conductivity.

Contributors: APICS and ATHENA Project Teams, Inria Sophia-Antipolis Méditerranée, Centre de Mathématiques Appliquées (CMA), Ecole des Mines de Paris.

- Version: 1.0
- Keywords: Medical Image Processing and Visualization
- License: CeCILL
- Multiplatform: Windows - Linux - MacOSX
- Programming language: Matlab
- Web: <http://www-sop.inria.fr/apics/FindSources3D/fr/index.html>

5.6. CoAdapt P300 Stimulator

Participants: Maureen Clerc, Théodore Papadopoulo, Loïc Mahé, Nathanaël Foy, Jérémie Mattout [Centre de Recherche en Neurosciences de Lyon, INSERM], Emmanuel Maby [Centre de Recherche en Neurosciences de Lyon, INSERM].

In the domain of Brain Computer Interfaces, extracting relevant features requires a precise timing of all events occurring in the system. In particular, when dealing with evoked responses as in the P300 speller, the timing of the visual stimulations must be well controlled. To alleviate some timing issues with the P300 speller initially provided with OpenViBE, we have implemented an external visual stimulator that allows to flash the visual targets, in a time-robust manner. This software was developed in the context of the ANR project CoAdapt. It runs with OpenViBE as an external plugin.

- Version: 1.0
- Keywords: Brain Computer Interfaces
- Multiplatform: Windows - Linux - MacOSX
- Programming language: C++
- APP IDDN FR.001.020003.000.S.P.2015.000.31235

AYIN Team (section vide)

BIOCORE Project-Team

5. New Software and Platforms

5.1. Supervision software

5.1.1. ODIN

Participants: Olivier Bernard, Francesco Novellis.

The latest developments of the bioreactor supervision platform **ODIN** were dedicated to software re-structuration (together with Méline Gauthier, from Inria Chile) in order to get more fluidity and more flexibility between modules and in order to support an on line simulator. The connection with a local data base has simplified the management of previous data acquisition and it also allows to “replay” data which were previously recorded. The coupling with the software developed by INRA (Silex) was refactored into a software named MEMO.

ODIN has been tested on four different processes especially (with Eric Latrielle) to supervise the 66m² high rate pond at the LBE, INRA Narbonne. It has also been used at Lesaffre facilities by the BioEnTech company. New algorithms have been successfully tested to control a high-rate anaerobic digestion process.

5.1.2. In@lgae

Participants: Etienne Delclaux, Francis Mairet, Quentin Béchet, Olivier Bernard.

The In@lgae platform has been optimised to make it faster. Some of the key models have been rewritten in C++ to allow a faster computation. Models have been improved to include, in the growth rate computation, the composition of the light spectrum. The graphical user interface has been enhanced and several sets of parameters describing different microalgal species have been stored. Post treatments with Matlab have been implemented to account for slope of the land, its nature, and the distance to CO₂ and nutrient sources. The platform supported a study for the French Agency for the development and master of energy (ADEME) managed by ENEA consulting. We could simulate the potential of micro -and macro-algal cultivation in France in 2030, after using the NEF cluster with 300 CPUs (it took 10 days of computation).

CASTOR Project-Team

5. New Software and Platforms

5.1. Free boundary equilibrium codes

5.1.1. CEDRES++

Participants: Jacques Blum, Cédric Boulbe, Blaise Faugeras, Holger Heumann, Sylvain Bremond [CEA], Eric Nardon [CEA].

In Tokamaks, at the slow resistive diffusion time scale, the magnetic configuration in the plasma can be described by the MHD equilibrium equations inside the plasma and the Maxwell equations outside. Moreover, the magnetic field is often supposed not to depend on the azimuthal angle.

Under this assumption of axisymmetric configuration, the equilibrium in the whole space reduces to solving a 2D problem in which the magnetic field in the plasma is described by the well known Grad Shafranov equation. The unknown of this problem is the poloidal magnetic flux. The P1 finite element code CEDRES++ solves this free boundary equilibrium problem in direct and inverse mode. The direct problem consists in the computation of the magnetic configuration and of the plasma boundary, given a plasma current density profile and the total current in each poloidal field coils (PF coils). The aim of the inverse problem is to find currents in the PF coils in order to best fit a given plasma shape. An evolutive version of the code has also been recently developed. This version takes into account the circuit equations in the PF coils. These equations give a time dependent relation between the voltages, the total current in the coils and the time derivative of the magnetic flux. Induced currents in passive structures like the vacuum vessel are also considered in this dynamic equilibrium problem. This new version of the code is an important tool for plasma scenario development and Tokamak design studies. A version of CEDRES++ is available in the environment of the european projet Eurofusion WPCD.

5.1.2. FEEQS.M

Participant: Holger Heumann.

FEEQS.M (Finite Element Equilibrium Solver in Matlab) is a MATLAB implementation of the numerical methods in [15] to solve equilibrium problems for toroidal plasmas. Direct and inverse problems for both the static and transient formulations of plasma equilibrium can be solved. FEEQS.M exploits MATLAB's evolved sparse matrix methods and uses heavily the vectorization programming paradigm, which results in running times comparable to C/C++ implementations. FEEQS.M complements the production code CEDRES++ in being considered as fast prototyping test bed for computational methods for equilibrium problems. This includes aspects of numerics such as improved robustness of the Newton iterations or optimization algorithms for inverse problems. The latest developments aim at incorporating the resistive diffusion equation.

5.2. Equinox

Participants: Jacques Blum, Cédric Boulbe, Blaise Faugeras.

EQUINOX is a code dedicated to the numerical reconstruction of the equilibrium of the plasma in a Tokamak. The problem solved consists in the identification of the plasma current density, a non-linear source in the 2D Grad-Shafranov equation which governs the axisymmetric equilibrium of a plasma in a Tokamak. The experimental measurements that enable this identification are the magnetics on the vacuum vessel, but also polarimetric and interferometric measures on several chords, as well as motional Stark effect measurements. The reconstruction can be obtained in real-time and the numerical method implemented involves a finite element method, a fixed-point algorithm and a least-square optimization procedure.

5.3. VacTH

Participants: Jacques Blum, Cédric Boulbe, Blaise Faugeras.

VacTH implements a method based on the use of toroidal harmonics and on a modelization of the poloidal field coils and divertor coils for the 2D interpolation and extrapolation of discrete magnetic measurements in a tokamak. The method is generic and can be used to provide the Cauchy boundary conditions needed as input by a fixed domain equilibrium reconstruction code like EQUINOX (see [45]). It can also be used to extrapolate the magnetic measurements in order to compute the plasma boundary itself. The proposed method and algorithm are detailed in [13] and results from numerous numerical experiments are presented. The method is foreseen to be used in the real-time plasma control loop on the WEST tokamak (see [46]).

5.4. FBGKI

Participants: Sébastien Minjeaud, Richard Pasquetti.

The Full Braginskii solver considers the equations proposed by Braginskii (1965), in order to describe the plasma turbulent transport in the edge part of tokamaks. These equations rely on a two fluid (ion - electron) description of the plasma and on the electroneutrality and electrostatic assumptions. One has then a set of 10 coupled non-linear and strongly anisotropic PDEs. FBGKI makes use in space of high order methods: Fourier in the toroidal periodic direction and spectral elements in the poloidal plane. The integration in time is based on a Strang splitting and Runge-Kutta schemes, with implicit treatment of the Lorentz terms (DIRK scheme). The spectral vanishing viscosity (SVV) technique is implemented for stabilization. Static condensation is used to reduce the computational cost. In its sequential version, a matrix free solver is used to compute the potential. The parallel version of the code is under development.

5.5. Platforms

5.5.1. FluidBox

Participants: Boniface Nkonga [contact], Hervé Guillard.

FluidBox is a software dedicated to the simulation of inert or reactive flows. It is also able to simulate multiphase, multi-material and MDH flows. There exist 2D and 3D dimensional versions. The 2D version is used to test new ideas that are later implemented in 3D. Two classes of schemes are available : a classical finite volume scheme and the more recent residual distribution schemes. Several low Mach number preconditioning are also implemented. The code has been parallelized with and without domain overlapping. The linear solver PaStiX is integrated in FluidBox. A partitioning tool exists in the package and uses Scotch.

5.5.2. Plato

Participants: Hervé Guillard [contact], Boniface Nkonga, Giorgio Giorgiani, Afeintou Sangam, Elise Estibals.

PlaTo (A platform for Tokamak simulation) is a suite of data and softwares dedicated to the geometry and physics of Tokamaks. Plato offers interfaces for reading and handling distributed unstructured meshes, numerical templates for parallel discretizations, interfaces for distributed matrices and linear and non-linear equation solvers. Plato provides meshes and solutions corresponding to equilibrium solutions that can be used as initial data for more complex computations as well as tools for visualization using Visit or Paraview. The use of this platform for large scale simulation has been validated up to $\mathcal{O}(1000)$ CPU [14] [10]

The numerical schemes used in the platform are of finite element or finite volume type. To deal with the geometry of tokamaks, Plato uses curved prisms made of a tensor product of unstructured triangular meshes in the poloidal plane by 1D meshes in the toroidal direction. The numerical strategy uses 3D finite volume schemes for the first-order terms and P1 finite element for second-order terms. Several models (anisotropic diffusion, Grad-Shafranov equilibrium, reduced MHD model) have been validated and are presently available. In addition, a stabilized finite element method using a tensor product of C^1 (Powell-Sabin) triangular element by 1D cubic splines in the toroidal direction has been recently developed and is presently in a validation phase.

5.5.3. Jorek-Inria

Participants: Hervé Guillard, Boniface Nkonga, Emmanuel Franck [Tonus, Inria Nancy - Grand Est], Ahmed Ratnani [IPP Garching].

<https://gforge.inria.fr/projects/jorek/>

Jorek-Inria is a new version of the JOREK software, for MHD modeling of plasma dynamic in tokamaks geometries. The numerical approximation is derived in the context of finite elements where 3D basic functions are tensor products of 2D basis functions in the poloidal plane by 1D basis functions in the toroidal direction. More specifically, Jorek uses curved bicubic isoparametric elements in 2D and a spectral decomposition (sine, cosine) in the toroidal axis. Continuity of derivatives and mesh alignment to equilibrium surface fluxes are enforced. Resulting linear systems are solved by the PASTIX software developed at Inria-Bordeaux.

The new formulation of the Jorek-Inria code extends this approximation strategy by introducing more flexibility and a variety of finite elements used in the poloidal plane and in the toroidal direction. It also proposes a sparse matrix interface SPM (Sparse Matrix Manager) that allows to develop clean code without a hard dependency on any linear solver library (i.e. PetSc, Pastix, Mumps, ...). It is expected that the two developments PlaTo and Jorek-Inria will merge in the next years.

COATI Project-Team

5. New Software and Platforms

5.1. JourneyPlanner

Participant: Marco Biazzini [correspondant].

JourneyPlanner is a Java implementation of a recursive algorithm to solve a TSP problem on small dense graphs, where non-trivial constraints must be satisfied, that make commonly used paradigms (as dynamic programming) unfit to the task.

This work is done in collaboration with the R&D service of the "Train Transportation" division of Amadeus.

5.2. Software updates

Participants: David Coudert, Luc Hogue, Aurélien Lancin, Nicolas Nisse, Michel Syska.

During this year, we have maintained and augmented already existing softwares. In particular:

- DRMSim (<http://drmsim.gforge.inria.fr/>) : discrete-event simulation engine aiming at enabling the large-scale simulations of routing models.
- GRPH (<http://grph.inria.fr/>) : graph optimization library written in Java. This year, we have integrated the discrete-events simulation engine of DRMSim and some dynamic models (evolution of the connectivity with the mobility of nodes) to GRPH. Notice that we have identified more than 300 academic users of GRPH.
- Sage (<http://www.sagemath.org>) : open-source mathematics software initially created by William Stein (Professor of mathematics at Washington University). We contribute the addition of new graph algorithms along with their documentations and the improvement of underlying data structures.

5.3. Platforms

5.3.1. BigGraphs

Participants: Aurélien Lancin, Paul Bertot, Nicolas Chleq [SED-SOP], David Coudert, Luc Hogue, Fabrice Huet [Scale], Flavian Jacquot, Arnaud Legout [Diana], Eric Madelaine [Scale], Michel Syska [coordinator].

The objective of BigGraphs is to provide a distributed middleware for very large graphs processing. This new project has received a development grant (ADT) from Inria and is a joint work of three EPI from Inria: COATI, DIANA and SCALE.

The first phase of the project consists in the evaluation of the existing middlewares such as GraphX/Spark or Giraph/Hadoop with respect to the following criteria: ease of deployment, maintenance and use; variety of programming models (Map/Reduce, BSP, (a)synchronous message passing, centralized programming, mobile agent-based, etc.); overall efficiency and memory footprint; etc. One of the chosen use cases is a subgraph of the Twitter graph with 3 millions of nodes and 200 millions of edges. The experiments are run on the NEF cluster at Inria. We have implemented and tested the classic algorithms (using the BSP model): page rank, BFS, connected components as well as the iFUB algorithm for computing the diameter of large graphs.

In parallel, we are testing new ideas through the development of custom solutions for the deployment of application code in heterogeneous environments, for automatic discovery of cluster architecture, for the design of distributed object oriented applications, and techniques for distributed graph computing (asynchronous BSP, messaging, multi-core parallelism, etc.).

The next phase is to decide whether one framework is matching our needs (and use it as a basis for further developments) or if we have to produce our own.

COFFEE Project-Team

5. New Software and Platforms

5.1. NS2DDV

The code NS2DDV is developed jointly with the team SIMPAF, of the Inria Research Centre Lille Nord Europe. It is devoted to the simulation of non-homogeneous viscous flows, in two-dimensional geometries. The code is based on an original hybrid Finite Volume/Finite Element scheme; it works on unstructured meshes and can include mesh refinements strategies. Further details can be found in the research papers *J. Comput. Phys.*, 227, 4671–4696, 2008 and *J. Comput. Phys.*, 229 (17), 6027–6046, 2010. The code exists in two versions: a Matlab public version, a C++ prototype version allowing more ambitious simulations. Both versions are still subject to developments. The current versions is restricted to incompressible flows but ongoing progress are concerned with the simulation of avalanches. The source code of the public version is downloadable and several benchmarks tests can be reproduced directly.

5.2. Compass

for Computing Parallel Architecture to Speed up Simulation is a parallel code for the discretization of polyphasic flows by Finite Volumes methods. The code is mainly devoted to applications in porous media. It works on quite general polyhedral meshes. A first step in the code development has been made during the 2012 edition of CEMRACS and then pursued by C. Guichard, R. Masson and R. Eymard in 2013. A first version of the code has been deposited at the Agency for the Protection of Programs (APP). This current version of ComPASS has been tested on a gas storage two phase flow benchmark with GDFSuez using the Vertex Approximate Gradient spatial discretization. The results have shown a very good parallel scalability on the CICADA Cluster at UNS with a few millions of cells and up to 1024 cores. The objective is to develop a generic simulator for multiphase Darcy flows. This simulator will implement advanced finite volume methods on general 3D meshes and on heterogeneous anisotropic media, taking into account discrete fracture networks represented as interfaces of codimension one and coupled with the surrounding matrix. It will be able to treat a large range of multiphase Darcy flow models accounting for thermodynamical equilibrium and the coupling with an energy conservation equation. The simulator will run on massively parallel architectures with a few thousands of cores. It will be applied to several type of industrial applications starting with the simulation of high energy geothermal systems as a carbon-free source of power production.

5.3. SimBiof

We are developing numerical methods, currently by using Finite Differences approaches, for the simulation of biofilms growth. The underlying system of PDEs takes the form of multiphase flows equations with conservation constraints and vanishing phases. The numerical experiments have permitted to bring out the influence of physical parameters on the multidimensional growth dynamics.

5.4. AP_PartFlow

We are developing experimental codes, mainly based on Finite Differences, for the simulation of particulate flows. A particular attention is paid to guaranty the asymptotic properties of the scheme, with respect to relaxation parameters.

DEMAR Project-Team

4. New Software and Platforms

4.1. New Software and Platforms

4.1.1. RdP to VHDL tool

Participants: David Andreu, Thierry Gil, Robin Passama, Baptiste Colombani, Thibaut Possompes.

Our SENIS (Stimulation Electrique Neurale dIStribuee) based FES architecture relies on distributed stimulation units (DSU) which are interconnected by means of a 2-wire based network. A DSU is a complex digital system since it embeds among others a dedicated processor (micro-machine with a specific reduced instruction set), a monitoring module and a 3-layer protocol stack. To face the complexity of the units digital part and to ease its prototyping on programmable digital devices (e.g. FPGA), we developed an approach for high level hardware component programming (HILECOP). To support the modularity and the reusability of sub-parts of complex hardware systems, the HILECOP methodology is based on components. An HILECOP component has: a Petri Net (PN) based behavior, a set of functions whose execution is controlled by the PN, and a set of variables and signals. Its interface contains places and transitions from which its PN model can be interconnected as well as signals it exports or imports. The interconnection of those components, from a behavioral point of view, consists in the interconnection of places and/or transitions according to well-defined mechanisms: interconnection by means of oriented arcs or by means of the "merging" operator (existing for both places and transitions).

The Eclipse-based version of HILECOP (registered at the french Agence de Protection des Programmes (APP)) is regularly updated.

Undergoing work concerns the integration, in the HILECOP tool, of the formalism evolutions that allow behavior agregation as well as exception handling, both for analysis and implementation sides (H. Leroux PhD thesis).

Specification of GALS systems (Globally Asynchronous Locally Synchronous) is also an ongoing work, the aim being to take into account deployment properties like connecting different clocks to HILECOP components within a same FPGA, or on a set of interconnected FPGAs (and thus interconnecting them by means of asynchronous signals).

4.1.2. SENISManager

Participants: Robin Passama, David Andreu.

We developed a specific software environment called SENISManager allowing to remotely manage and control a network of DSUs, i.e. the distributed FES architecture. SENISManager performs self-detection of the architecture being deployed. This environment allows the manipulation of micro-programs from their edition to their remote control. It also allows the programming of control sequences executed by an external controller in charge of automatically piloting a stimulator.

SENISManager (registered at the french Agence de Protection des Programmes (APP) with the industrial partner) has been transferred to the industrial partner that develops a new version according to an Eclipse-based design.

4.1.3. Synergy Neurostimulation Software

Participants: David Andreu, Amandine Pantel, Arthur Haiarrassary.

We are developing a specific software environment called Synergy Neurostimulation Software allowing to remotely manage a stimulation architecture based on one controller piloting a set of distributed stimulation units, connected by means of a dedicated network. The controller embeds the set of FES functions according to which it controls stimulation units, in real-time.

This FES distributed architecture is based on our last version of stimulation units that embed stimulation sequencing and a more efficient modulation mechanism.

Synergy Neurostimulation Software will be soon registered at the french Agence de Protection des Programmes (APP).

4.1.4. MOS2SENS: Model Optimization and Simulation To Selective Electrical Neural Stimulation

Participants: Melissa Dali, Olivier Rossel, David Guiraud.

Multipolar electric stimulation of the nerve is a main issue, to access selective activation of organ or muscles. Knowing that electrodes configurations have to be specific to the type of nerve and to the organic or muscular targeted, we work on an accurate and flexible nerve modeling (work extension of Jérémy Laforêt PhD thesis, 2009), and we have developed new software MOS2SENS (from Model Optimization and Simulation To Selective Electrical Neural Stimulation) (fig.1). This model can predict nerve fiber activation through multipolar electrode stimulation. Furthermore the models provide an optimal current configuration to activate accurately the targeted muscle or organ (indeed a targeted group of fiber).

The new software MOS2SENS is an adjustment support tool for neuroprosthetics devices. It models and optimizes the current injected by multipolar CUFF electrodes inside the nerve in order to activate selective fiber targets in terms of spatial criterion.

There are two programs that perform the following functions:

- Generation of 3D geometric model
- Mathematical description of the link between stimulation currents and extracellular voltage present inside the nerve
- Nerve fiber activation prediction based on the current stimulation
- Optimization of the current injected according to the chosen target

The software has been implemented in Matlab with graphical user interface and use OpenMEEG open source software to compute electric fields from the electrode to the fibers.

MOS2SENS is filed in the Agency for the Protection of Programs (APP) under the identifier

IDDN.FR.001.490036.000.S.P.2014.000.31230

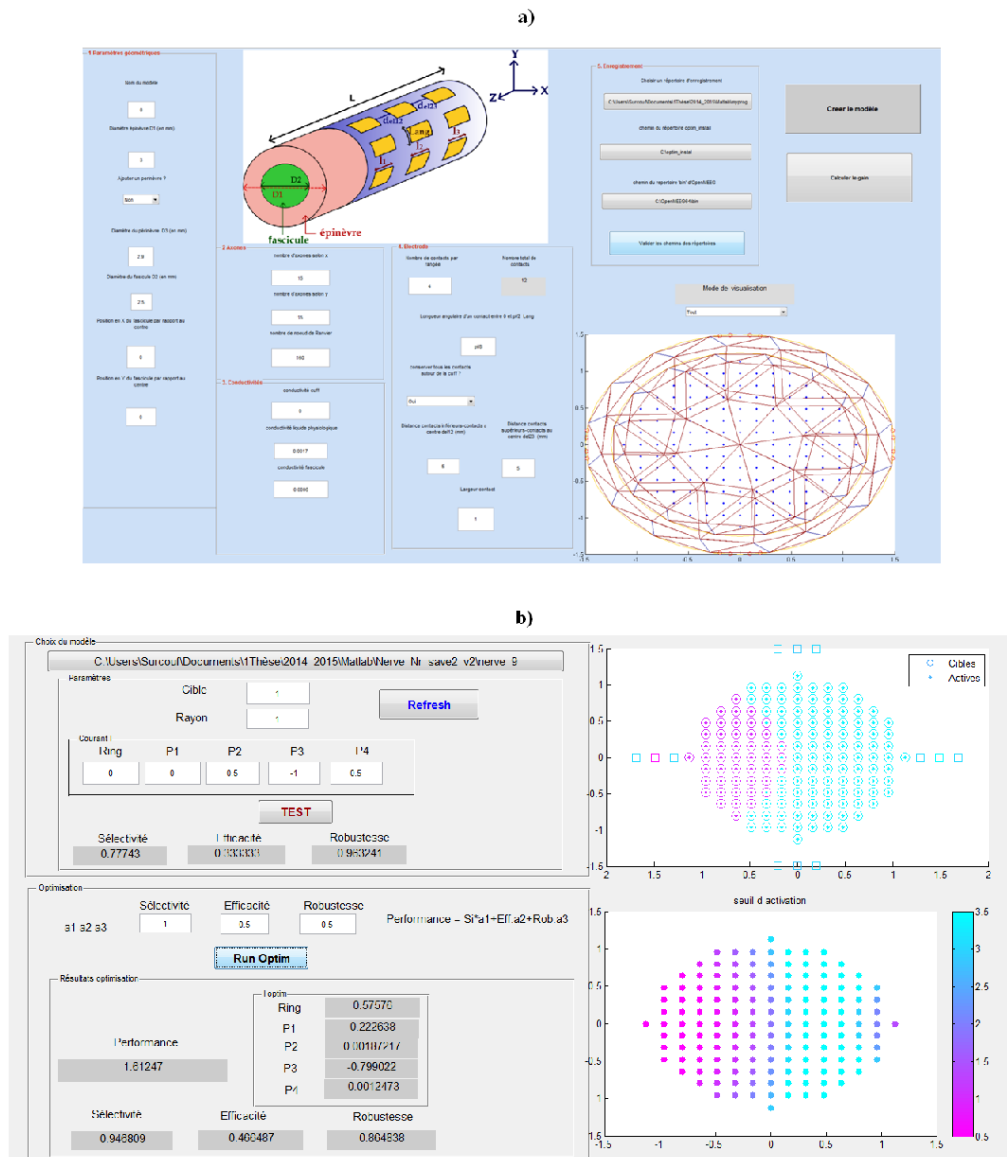
4.1.5. SENSBIOTK

As low cost and highly portable sensors, inertial measurements units (IMU) have become increasingly used in different topics, such as gait analysis, embodying an efficient alternative to motion capture systems. Meanwhile, being able to compute reliably accurate spatial parameters using few sensors remains a relatively complex problematic. The use of inertial data calls on various algorithms able to compute from raw sensors (accelerometer, magnetometer and gyrometer) different features (position, angle, etc...). SensbioTK (for Tool Kit) has been implemented using a Python programming environment. This opensource library provides to any IMU user a set of tools enabling the following functions :

- Conversion from inertial raw data to .csv file
- Computation of optimized scale and offset parameters for inertial sensors calibration (Gauss newton optimization)
- AHRS sensor fusion algorithms (Kalman filter and gradient descent based) : Madgwick, Mahony and Martin-Salaun implementation
- Stride length calculation from one shank located IMU
- 3D transformations and quaternion library

Many "ready to use" examples with relative data and scripts : goniometer, compass, pedometer, motion capture validation...

<https://github.com/sensbio/sensbiotk>



DIANA Team

4. New Software and Platforms

4.1. ns-3

Participants: Walid Dabbous [correspondant], Thierry Turletti.

ns-3 is a discrete-event network simulator for Internet systems, targeted primarily for research and educational use. ns-3 includes a solid event-driven simulation core as well as an object framework focused on simulation configuration and event tracing, a set of realistic 802.11 MAC and PHY models, an IPv4, UDP, and TCP stack and support for nsc (integration of Linux and BSD TCP/IP network stacks). ns-3 is free software, licensed under the GNU GPLv2 license, and it is publicly available for research, development, and use. Our team has been involved in ns-3 project since 2006 and we are founding member of the ns-3 consortium.

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

- Version: ns-3.21
- Keywords: networking event-driven simulation
- License: GPL (GPLv2)
- Type of human computer interaction: programmation C++/python, No GUI
- OS/Middleware: Linux, cygwin, osX
- Required library or software: standard C++ library: GPLv2
- Programming language: C++, python
- Documentation: doxygen

4.2. DCE

Participants: Thierry Turletti [correspondant], Walid Dabbous.

DCE enables developers and researchers to develop their protocols and applications in a fully controllable and deterministic environment, where tests can be repeated with reproducible results. It allows unmodified protocol implementations and application code to be tested over large and possibly complex network topologies through the ns-3 discrete-event network simulator. The single-process model used in the DCE virtualization core brings key features, such as the possibility to easily debug a distributed system over multiple simulated nodes without the need of a distributed and complex debugger. Examples of tested applications over DCE include Quagga, iperf, torrent, tthttpd, CCNx and various Linux kernel versions (from 2.6.36 to 3.12 versions). DCE was initially developed by Mathieu Lacage during his PhD thesis and is maintained by engineers in the team in collaboration with Hajime Tazaki from University of Tokyo. DCE/ns-3 is an important component of the Reproducible Research Lab. DCE is free software, licensed under the GNU GPLv2 license, and is publicly available for research, development, and use.

See also the web page <https://www.nsnam.org/overview/projects/direct-code-execution/>

- Version: DCE-1.2
- Keywords: emulation, virtualization, networking event-driven simulation
- License: GPL (GPLv2)
- Type of human computer interaction: programmation C/C++, No GUI
- OS/Middleware: Linux
- Required library or software: standard C++ library: GPLv2
- Programming language: C++, python
- Documentation: doxygen

4.3. NEPI

Participants: Thierry Turlatti [correspondant], Alina Quereilhac, Julien Tribino, Lucia Guevgeozian Odizzio.

NEPI, the Network Experimentation Programming Interface, is a framework to describe and orchestrate network experiments on a variety of network experimentation platforms, including simulators, emulators, live testbeds, and testbed federations. NEPI is capable of supporting arbitrary platforms through the use of a generic network experiment description model, based on abstracting network experiments as a collection of arbitrary resource objects, and through the generalization of the experiment life cycle for all resources. The common resource life cycle consist on the sequence of operations deploy, start, stop, and release. Different resource objects can implement specific versions of those operations to adapt to any platform. NEPI resolves experiment orchestration as an online scheduling problem that consists on executing the deploy, start, stop, and release operations for every resource in the correct order.

During the year 2013 we fully re-implemented NEPI's core libraries to adopt the scheduling-based experiment orchestration approach, improving the flexibility and extensibility of the framework compared to the previous static stage-based orchestration approach. By the end of 2013 the new NEPI framework supported describing and orchestrating experiments on live testbeds, including SSH enables Linux testbed, PlanetLab Internet testbed, and OMF wireless testbed (version 5.4).

In 2014 the framework was extended to support simulation and emulation, using the ns-3 simulator and its direct code execution (DCE) emulation extension. Additionally, automated translation of a same experiment scenario to different platforms, i.e. multi platform experimentation, was incorporated into the framework to provide an unified environment for the development and evaluation of production quality networking software, meant to be deployed on real networks. This unified development and evaluation environment simplifies the transition from a realistic live platform to a controlled emulation platform, and vice-versa, in order to take advantage of the complementary features offered by them. The environment was demonstrated at ACM ICN 2014 for the case of Content Centric Networking (CCNx) development, combining PlanetLab and DCE platforms.

Finally, NEPI now supports OMF experiment control protocol version 6.0, which is the new mainstream release of OMF control framework for testbeds, and SFA (Slice Federation Architecture), for resource discovery and provisioning across federated testbeds. The combination of OMF 6.0 and SFA was adopted as the standard for federated experiment orchestration in the European federation projects OpenLab and Fed4FIRE. NEPI's ability to support federated experiment orchestration was demonstrated at the OpenLab Final Review. Information about this demo is available at <http://nepi.inria.fr/UseCases/VLCCCNStreamingExperiment>.

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

- Version: 3.2
- ACM: C.2.2, C.2.4
- Keywords: networking experimentation, simulation, emulation
- License: GPL (3)
- Type of human computer interaction: python library
- OS/Middleware: Linux
- Required library or software: python – <http://www.python.org>
- Programming language: python

4.4. OpenLISP

Participant: Damien Saucez [correspondant].

Among many options tackling the scalability issues of the current Internet routing architecture, the Locator/Identifier Separation Protocol (LISP) appears as a viable solution. LISP improves a network's scalability, flexibility, and traffic engineering, enabling mobility with limited overhead. As for any new technology, implementation and deployment are essential to gather and master the real benefits that it provides. We propose a complete open source implementation of the LISP control plane. Our implementation is deployed in the worldwide LISP Beta Network and the French LISP-Lab testbed, and includes the key standardized control plane features. Our control plane software is the companion of the existing OpenLISP dataplane implementation, allowing the deployment of a fully functional open source LISP network compatible with any implementation respecting the standards.

See also the web page <http://www.lisp.ipv6.lip6.fr/a/Download.html>.

- Version: 3.2
- ACM: C.2.1, C.2.2, C.2.6
- Keywords: routing, LISP, control-plane
- License: BSD
- Type of human computer interaction: XML, CLI
- OS/Middleware: POSIX
- Required library or software: Expat 2
- Programming language: C
- Documentation: Unix man
- Deployment: ddt-root.org

4.5. ACQUA

Participants: Chadi Barakat [correspondant], Salim Afra, Damien Saucez.

ACQUA is an Application for Predicting User Quality of Experience at Internet Access. It was supported by the French ANR CMON project on collaborative monitoring. ACQUA presents a new way for the evaluation of the performance of Internet access. Starting from network-level measurements as the ones we often do today (bandwidth, delay, loss rates, etc), ACQUA targets the estimated quality of experience related to the different applications if run at the access. An application in ACQUA is a function that links the network-level measurements to its expected quality of experience. In its first version (the version available online), ACQUA was concentrating on delay measurements at the access and on the detection and estimation of the impact of delay anomalies (local problems, remote problems, etc). The current work is concentrating on using the ACQUA principle in the estimation and prediction of the quality of experience of main applications (see section 5.2 for more details).

See also the web page <https://team.inria.fr/diana/acqua/>.

- Version: 1.0
- ACM: C.2.2, C.2.3
- Keywords: Internet measurement, Internet Access, Quality of Experience
- License: GPL (3)
- Type of human computer interaction: C#
- OS/Middleware: MS Windows
- Required library or software: visual studio <http://www.visualstudio.com/en-us/products/visual-studio-express-vs.aspx>
- Programming language: C# for client, java for server

4.6. ElectroSmart

Participants: Arnaud Legout [correspondant], Inderjeet Singh, Maksym Gabielkov.

The ElectroSmart project is based on a large crowd sourcing collection of electromagnetic radiations measured by the ElectroSmart application running on real users smartphones. We target a large number of users and many scientific exploitation of the collected data, exploitation that we describe in the following.

Exposure of human beings to electromagnetic radiations is a growing worldwide health concern. While the biological impact of electromagnetic radiations is not fully understood, there are reports of hypersensitivity to such radiations and hints toward a possible correlation between high exposition and cancer. However, the biological impact of electromagnetic radiations is just one half of the problem, the other half is the exploration of the real exposure of the population to electromagnetic radiations. Indeed, whatever the biological impact, it will be function of the level of exposure, and this level of exposure is unknown.

Collecting the real exposure of human beings to electromagnetic radiations is a complex task. It is possible, but costly and time consuming, to ask auditing organizations to make one-shot measurements. However, there is no way accessible to the general audience to make long term measurements.

The goal of this project is to create the first long term measurement of the electromagnetic exposure of a large worldwide population. This project is supported by the Inria ADT ElectroSmart.

- Version: 1.0alpha
- Keywords: background electromagnetic radiations
- License: Inria proprietary licence
- Type of human computer interaction: Android application
- OS/Middleware: Android
- Required library or software: Android
- Programming language: Java
- Documentation: javadoc

4.7. Platforms

4.7.1. *Reproducible research laboratory* (R²LAB)

Scientific evaluation of network protocols requires that experiment results must be reproducible before they can be considered as valid. This is particularly difficult to obtain in the wireless networking domain, where characteristics of wireless channels are known to be variable, unpredictable and hardly controllable. Indeed, anechoic chambers with RF absorbers preventing radio waves reflections and with Faraday cage blocking external interferences represent an ideal environment for experiments reproducibility. This year witnessed the realization of such experimental platform (called R²LAB or Reproducible Research Laboratory) at Inria Sophia-Antipolis, in the context of the FIT 'Equipment of Excellence' project. The objectives of this platform are twofold : on the one hand, we need to achieve highly controllable wireless experiments (e.g. control plane for 5G), and to this end, the testbed features an anechoic chamber. On the other hand, we need to make it possible to deploy experiments that have demanding resource requirements, as this is typically the case with e.g. ICN-based research, or when involving simulation. For that reason, the platform features some powerful servers of its own; in addition, these experiments can be either hybrid-experiments (as NEPI will be deployed) or federated experiments through several testbeds such as PlanetLab. As the final objective is to provide an environment to easily run realistic and reproducible wireless experiments and simulations, it is important to be able to increase the testbed realism by injecting noise and interfering signals in a controllable way. Experimentation results done in R²LAB could also be used to augment the realism of propagation models in simulators, which are able to run large scale scenarios. We are currently deploying the wireless nodes. The next step will to extend the testbed to support software defined networking (SDN) and LTE experimentations and to install specific tools for operating the platform.

ECUADOR Project-Team

5. New Software and Platforms

5.1. AIRONUM

Participant: Alain Dervieux [correspondant].

Aironum is an experimental software that solves the unsteady compressible Navier-Stokes equations with $k - \epsilon$, LES-VMS and hybrid turbulence modelling on parallel platforms, using MPI. The mesh model is unstructured tetrahedrization, with possible mesh motion. See <http://www-sop.inria.fr/tropics/aironum>

- Version: v 1.0
- Programming language: Fortran95 (mostly). About 100,000 lines.

Aironum was developed by Inria and University of Montpellier. It is used by Inria, University of Montpellier and University of Pisa (I). Aironum is used as an experimental platform for:

- Numerical approximation of compressible flows, such as upwind mixed element volume approximation with superconvergence on regular meshes.
- Numerical solution algorithms for the implicit time advancing of the compressible Navier-Stokes equations, such as parallel scalable deflated additive Schwarz algorithms.
- Turbulence modelling such as the Variational Multiscale Large eddy Simulation and its hybridization with RANS statistical models.

5.2. TAPENADE

Participants: Laurent Hascoët [correspondant], Valérie Pascual, Ala Taftaf, Jan Hueckelheim [Queen Mary University of London].

Tapenade is an Algorithmic Differentiation tool that transforms an original program into a new program that computes derivatives of the original program. Algorithmic Differentiation produces analytical derivatives, that are exact up to machine precision. Adjoint-mode AD can compute gradients at a cost which is independent from the number of input variables. Tapenade accepts source programs written in Fortran77, Fortran90, or C. It provides differentiation in the following modes: tangent, vector tangent, adjoint, and vector adjoint. Documentation is provided on <http://www-sop.inria.fr/tropics/tapenade.html>, in Inria technical report RT-0300, and in [9].

- Version: v3.9, r5092, February 2014
- ACM: D.3.4 Compilers; G.1.0 Numerical algorithms; G.1.4 Automatic differentiation; I.1.2 Analysis of algorithms
- AMS: 65K10; 68N20
- APP: IDDN.FR.001.040038.002.S.P.2002.000.10600
- Keywords: algorithmic differentiation, adjoint, gradient, optimisation, inverse problems, static analysis, data-flow analysis, compilation
- Programming language: Java

Tapenade implements the results of our research about models and static analyses for AD. Tapenade can be downloaded and installed on most architectures. Alternatively, it can be used as a web server. Higher-order derivatives can be obtained through repeated application.

Tapenade performs sophisticated data-flow analysis, flow-sensitive and context-sensitive, on the complete source program to produce an efficient differentiated code. Analyses include Type-Checking, Read-Write analysis, and Pointer analysis. AD-specific analyses include:

- **Activity analysis:** Detects variables whose derivative is either null or useless, to reduce the number of derivative instructions.
- **Adjoint Liveness analysis:** Detects the source statements that are dead code for the computation of derivatives.
- **TBR analysis:** In adjoint-mode AD, reduces the set of source variables that need to be recovered.

Tapenade is not open-source. Academic usage is free for one year. Other usages require a paying license, as detailed on the web page. Ten industrial licences have been sold. Tapenade has been downloaded several hundred times, and the web tool served several thousands of true connections (robots and crawlers excluded) The tapenade-users mailing list is over one hundred registered users.

FOCUS Project-Team

5. New Software and Platforms

5.1. Jolie

Members of Focus have developed Jolie [8] (Java Orchestration Language Interpreter Engine, see <http://www.jolie-lang.org/>). Jolie is a service-oriented programming language. Jolie can be used to program services that interact over the Internet using different communication protocols. Differently from other Web Services programming languages such as WS-BPEL, Jolie is based on a user-friendly C/Java-like syntax (more readable than the verbose XML syntax of WS-BPEL) and, moreover, the language is equipped with a formal operational semantics. This language is used for the *proof of concepts* developed around Focus activities. For instance, contract theories can be exploited for checking the conformance of a Jolie program with respect to a given contract. A spin-off, called “Italiana Software”, has been launched around Jolie, its general aim is to transfer the expertise in formal methods for Web Services matured in the last few years onto Service Oriented Business Applications. The spin-off is a software producer and consulting company that offers service-oriented solutions (for instance, a “single sign-on” application) based on the Jolie language.

During 2014, the development of Jolie 1.1 has been completed (the release is due for the first half of January 2015). It is the result of about 600 commits, including more than 30 new standard library APIs, 100 bugfixes, and 100 improvements to the Jolie interpreter and libraries. Highlights:

- A new hierarchical semantics for handling sub-programs loaded in higher-order Jolie services.
- Support for “abstract locations”. This enables the writing of extensions that automatically fetch the bindings to the external services needed by a Jolie program. We plan to use this feature to develop binding procedures that ensure correctness.
- Introduction of a tracer option for the Jolie interpreter, which displays the execution trace of a Jolie program (useful for debugging).
- Substantial improvements to memory management of higher-order programs.
- Improved integration with web applications, which supports new techniques for handling the evolution of legacy web applications using the composition primitives of Jolie.

Moreover Jolie Enterprise has been released: this is an administrative tool that allows one to deploy Jolie services on remote nodes. Jolie Enterprise is able to manage services that run on different nodes on different machines, tracking all messages exchanged between services and viewing the log on GUI so that one can have a report of what happened in the system. Currently there are about 15 installations of Jolie Enterprise at SME in clothing, construction and manufacturing.

5.2. Others

Below we list some software that has been developed, or is under development, in Focus.

- *Deadlock analysis* (<http://df4abs.nws.cs.unibo.it/>).

We have prototyped a framework for statically detecting deadlocks in a concurrent object-oriented language with asynchronous method calls and cooperative scheduling of method activations (the language is inspired by the ABS language developed in the EU project HATS and currently extended with primitives for cloud-computing in the EU project ENVISAGE). Since this language features recursion and dynamic resource creation, deadlock detection is extremely complex and state-of-the-art solutions either give imprecise answers or do not scale. In order to augment precision and scalability we propose a modular framework that allows several techniques to be combined. The basic component of the framework is a front-end inference algorithm that extracts abstract behavioural descriptions of methods, called contracts, which retain resource dependency information. Then these contracts are analysed by a back-end that uses a fix-point technique to derive in a deterministic way the deadlock information.

- *CaReDeb* (<http://www.cs.unibo.it/caredeb>).

Reversible debugging provides developers with a way to execute their applications both forward and backward, seeking the cause of an unexpected or undesired event. We have developed CaReDeb, the first prototype of a causal-consistent reversible debugger. Causal consistent here means that independent actions are undone independently, while dependent actions are undone in reverse order. This allows the programmer to concentrate on the threads responsible of the bug, independently of the actual interleaving. CaReDeb provides primitives that given a misbehaviour, e.g., a variable has not the expected value, allow one to go back to the action responsible for it, e.g., the one that assigned the wrong value to the variable. Notably, the programmer has no need to know which thread the action belongs to, since this is found automatically by the debugger. The procedure can be iterated till the bug is found. CaReDeb targets a fragment of the language Oz, which is at the basis of Mozart. The considered fragment provides functional variables, procedures, threads, and asynchronous communication via ports.

- *AIO CJ* (<http://www.cs.unibo.it/projects/jolie/aio cj.html>).

AIO CJ is a framework for programming adaptive distributed systems based on message passing. AIO CJ comes as a plugin for Eclipse, AIO CJ-ecl, allowing to edit descriptions of distributed systems as adaptive interaction-oriented choreographies (AIO C). From interaction-oriented choreographies the description of single participants can be automatically derived. Adaptation is specified by rules allowing to replace predetermined parts of the AIO C with a new behaviour. A suitable protocol ensures that all the participants are updated in a coordinated way. As a result, the distributed system follows the specification given by the AIO C under all changing sets of adaptation rules and environment conditions. In particular, the system is always deadlock-free. AIO CJ can interact with external services, seen as functions, by specifying their URL and the protocol they support (HTTP, SOAP, ...). Deadlock-freedom guarantees of the application are preserved provided that those services do not block.

- *METIS* (<https://github.com/aeolus-project/metis>)

As partners of the Aeolus project we have developed a tool for the automatic synthesis of deployment plans. A deployment plan is a sequence of actions that, when performed, allows the deployment of a given configuration of components. METIS (Modern Engineered Tool for Installing Software systems) is a tool that enables one to automatically generate a deployment plan, starting from a description of the configuration following the Aeolus model. The software is open source. It is written entirely in OCaml and is about 3.5K lines of source code. The tool is based on theoretical results that guarantee its soundness and completeness, while maintaining polynomial computational complexity. METIS already showed its effectiveness in practice by handling synthesized problem instances with hundreds of components in less than a minute. We are currently validating Metis in a production environment by integrating it in Armonic, an infrastructure for cloud application deployment in OpenStack cloud systems developed by the Mandriva company.

- *SUNNY-CP* (<https://github.com/jacopoMauro/sunny-cp>)

Within the Constraint Programming (CP) paradigm, a portfolio approach enables to combine a number of different constraint solvers in order to create a globally better solver, dubbed a portfolio solver. After several empirical evaluations (e.g., [22], [23], [24]) we have decided to develop *SUNNY-CP*, a portfolio solver for solving both Constraint Satisfaction Problems and Constraint Optimization Problems. The goal of *SUNNY-CP* is to provide a flexible, configurable, and usable CP portfolio solver that can be set up and executed just like a regular individual CP solver. To the best of our knowledge, *SUNNY-CP* is the only sequential portfolio solver able to solve generic CP problems, and it was the only portfolio solver that attended the MiniZinc Challenge 2014 (i.e., the only active international competition to evaluate the performance of CP solvers). (*SUNNY-CP* performed very well, ranking 4th in the competition and receiving an honourable mention by the challenge organizers.) The application of *SUNNY-CP* in the optimization problems defined within the Aeolus project have lead to time improvements beyond an order of magnitude. *SUNNY-CP* is mainly written in Python, and we are currently enhancing the tool in order to make it more usable, flexible, and parallel (i.e., able to properly exploit multiple cores).

The software below have not undergone substantial modifications during 2014.

- *Croll-pi Interpreter* (<http://proton.inrialpes.fr/~mlienhar/croll-pi/implem/>). Croll-pi is a concurrent reversible language featuring a rollback operator to undo a past action (together with all the actions depending on it), and a compensation mechanism to avoid cycling by redoing the same action again and again. We have developed an interpreter for croll-pi using Maude.
- *IntML* is a functional programming language guaranteeing sublinear space bounds for all programs [51]. See the Activity Reports of previous years (in particular 2010) for more details.
- *Lideal* (<http://lideal.cs.unibo.it/>) is an experimental tool implementing type inference for dependently linear type systems. The tool reduces the problem of evaluating the complexity of PCF (i.e. functional programs with primitive integers and recursive definitions) to checking a set of first-order inequalities for validity. The latter can then be handled through SMT solvers or put in a form suitable for managing them with tools such as CoQ. See the Activity Reports of previous years (in particular 2010) for more details.

GALAAD2 Team

5. New Software and Platforms

5.1. Mathemagix, a free computer algebra environment

Participant: Bernard Mourrain.

<http://www.mathemagix.org/>

algebra, univariate polynomial, multivariate polynomial, matrices, series, fast algorithm, interpreter, compiler, hybrid software.

MATHEMAGIX is a free computer algebra system which consists of a general purpose interpreter, which can be used for non-mathematical tasks as well, and efficient modules on algebraic objects. It includes the development of standard libraries for basic arithmetic on dense and sparse objects (numbers, univariate and multivariate polynomials, power series, matrices, etc., based on FFT and other fast algorithms). These developments, based on C++, offer generic programming without losing effectiveness, via the parameterization of the code (*template*) and the control of their instantiations.

The language of the interpreter is imperative, strongly typed and high level. A compiler of this language is available. A special effort has been put on embedding of existing libraries written in other languages like C or C++. An interesting feature is that this extension mechanism supports template types, which automatically induce generic types inside Mathemagix. Connections with GMP, MPFR for extended arithmetic, LAPACK for numerical linear algebra are currently available in this framework.

The project aims at building a bridge between symbolic computation and numerical analysis. It is structured by collaborative software developments of different groups in the domain of algebraic and symbolic-numeric computation.

In this framework, we are working more specifically on the following components:

- REALROOT: a set of solvers using subdivision methods to isolate the roots of polynomial equations in one or several variables; continued fraction expansion of roots of univariate polynomials; Bernstein basis representation of univariate and multivariate polynomials and related algorithms; exact computation with real algebraic numbers, sign evaluation, comparison, certified numerical approximation.
- SHAPE: tools to manipulate curves and surfaces of different types including parameterized, implicit with different type of coefficients; algorithms to compute their topology, intersection points or curves, self-intersection locus, singularities, ...

These packages are integrated from the former library SYNAPS (SYmbolic Numeric APplicationS) dedicated to symbolic and numerical computations. There are also used in the algebraic-geometric modeler AXEL.

Collaborators: Grégoire Lecerf, Joris van der Hoeven and Philippe Trébuchet.

5.2. Axel, a geometric modeler for algebraic objects

Participants: Nicolas Douillet, Anaïs Ducoffé [contact], Valentin Michelet, Bernard Mourrain, Hung Nguyen, Meriadeg Perrinel.

<http://axel.inria.fr>

computational algebraic geometry, curve, implicit equation, intersection, parameterization, resolution, surface, singularity, topology

We are developing a software called AXEL (Algebraic Software-Components for gEometric modeLing) dedicated to algebraic methods for curves and surfaces. Many algorithms in geometric modeling require a combination of geometric and algebraic tools. Aiming at the development of reliable and efficient implementations, AXEL provides a framework for such combination of tools, involving symbolic and numeric computations.

The software contains data structures and functionalities related to algebraic models used in geometric modeling, such as polynomial parameterizations, B-splines, implicit curves and surfaces. It provides algorithms for the treatment of such geometric objects, such as tools for computing intersection points of curves or surfaces, for detecting and computing self-intersection points of parameterized surfaces, for implicitization, for computing the topology of implicit curves, for meshing implicit (singular) surfaces, etc.

The developments related to isogeometric analysis have been integrated as dedicated plugins. Optimization techniques and solvers for partial differential equations developed by R. Duvigneau (OPALE) have been connected.

The new version of the algebraic-geometric modelers based on the DTK platform is still developed in order to provide a better modularity and a better interface to existing computation facilities and geometric rendering interface. This software is intended to be multi-platform, and jobs are running nightly on the Continuous Integration platform <https://ci.inria.fr/> of Inria, performing builds and tests on Virtual Machines of different OS such as Fedora, Ubuntu, Windows.

AXEL is written in C++ and thanks to a wrapping system using SWIG, its data structures and algorithms can be integrated into C# programs, as well as Python and Java programs. This wrapper was used to integrate AXEL into the CAD software TopSolid, developed by Missler Company and written in C#. But it also enables AXEL to embed a Python interpreter.

Other functionalities were also added or improved: the scientific visualization was improved and it is now possible to create dynamic geometric model in AXEL.

The software is distributed as a source package, as well as binary packages for Linux, MacOSX and Windows. It is hosted at <http://dtk.inria.fr/axel> with some of its plugins developed on Inria's gforge server (<http://gforge.inria.fr>) The first version of the software has been downloaded more than 15000 times, since it is available. A new version, AXEL 2.3.1, was released at the end of this year.

Collaboration with Gang Xu (Hangzhou Dianzi University, China), Julien Wintz (Dream), Elisa Berrini (MyCFD, Sophia), Angelos Mantzaflaris (GISMO library, Linz, Austria) and Laura Saini (Post-Doc GALAAD/Missler, TopSolid).

GEOMETRICA Project-Team

5. New Software and Platforms

5.1. CGAL, the Computational Geometry Algorithms Library

Participants: Jean-Daniel Boissonnat, Olivier Devillers, Marc Glisse, Aymeric Pellé, Monique Teillaud, Mariette Yvinec.

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

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

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

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

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

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

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

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

CGAL is about 700,000 lines of code and supports various platforms: GCC (Linux, Mac OS X, Cygwin...), Visual C++ (Windows), Intel C++. A new version of CGAL is released twice a year, and it is downloaded about 10000 times a year. Moreover, CGAL is directly available as packages for the Debian, Ubuntu and Fedora Linux distributions.

More numbers about CGAL: there are now 12 editors in the editorial board, with approximately 20 additional developers. The user discussion mailing-list has more than 1000 subscribers with a relatively high traffic of 5-10 mails a day. The announcement mailing-list has more than 3000 subscribers.

5.1.1. High-dimensional kernel *Epick_d*

Participant: Marc Glisse.

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

5.1.2. Number type *Mpzf*

Participant: Marc Glisse.

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

5.1.3. *CGALmesh: a Generic Framework for Delaunay Mesh Generation*

Participants: Jean-Daniel Boissonnat, Mariette Yvinec.

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

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

5.1.4. Periodic Meshes

Participants: Aymeric Pellé, Monique Teillaud.

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

5.2. Gudhi library

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

With the collaboration of David Salinas

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

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

GRAPHIK Project-Team

5. New Software and Platforms

5.1. Cogui

Participants: Alain Gutierrez, Michel Leclère, Marie-Laure Mugnier, Michel Chein, Madalina Croitoru.

Cogui (<http://www.lirmm.fr/cogui>) is a tool for building and verifying knowledge bases. It is a freeware written in Java (version 1.6). Currently, it supports Conceptual Graphs and import/export in RDFS and Datalog⁺.

This year, we have particularly improved scripts, which are interpreted pieces of code allowing to freely manipulate objects of the KB. The main improvements are the following:

- script management with better bug tracking and error reporting;
- interoperability between scripts and objects of the knowledge base;
- embedding of a Java library, which allows to import java classes into scripts (a feature required in the application developed for Qualinca).

5.2. Graal

Participants: Clément Sipieter, Jean-François Baget, Marie-Laure Mugnier, Swan Rocher.

Graal is a new software platform written in java, built since March 2014 from the Alaska platform developed during Bruno Paiva Lima Da Silva's PhD thesis. It also integrates algorithms developed by various members of the team. It is developed by Clément Sipieter thanks to the Inria ADT QUASAR.

Graal is intended to be a generic platform for ontological query answering with existential rules. It will implement and allow to compare various paradigms that fall into that framework.

In its current state, Graal allows storage of data via a generic interface in different storage paradigms and systems. Currently, the relational database management systems MySQL, PostgreSQL, Sqlite, and InMemory graph and LinkedList structures are implemented. The triple store Jena TDB and the graph database system Sparksee are coming soon. Graal also allows us to query this database taking into account an ontology represented by a set of existential rules. It provides forward chaining and backward chaining algorithms (building up on the work of Mélanie König) and a tool for the analysis of the properties of a set of rules which is an integration of Swan Rocher's tool Kiabora <http://www2.lirmm.fr/~mugnier/graphik/kiabora/>. The input and output of this software can be expressed in our Datalog-inspired format DLGP or in the Semantic Web language OWL. This software is designed in a modular way, hence it is possible to use only a subpart of Graal without embedding it all or to easily replace an implementation of a module by another.

HEPHAISTOS Team

5. New Software and Platforms

5.1. Introduction

Software development is an essential part of the research done by HEPHAISTOS since a large part of our methods can only be validated experimentally (both for our numerical experiments and in robotics). Software developments follow various directions:

1. interval arithmetic: although we do not plan to work in this very specialized area (we generally rely on existing packages) interval arithmetic is an important part of our interval analysis algorithms and we may have to modify the existing packages so as to deal, in particular, with multi-precision and arithmetic extensions
2. interval analysis libraries: we daily use the ALIAS library that has been designed in the project and is still under development. A long term work is to develop a generic programming framework that allows for modularity and flexibility, with the objectives of testing new functionalities easily and building specific solvers by a simple juxtaposition of existing modules
3. interface to interval analysis: in our opinion interval analysis software must be available within general purpose scientific software (such as Maple, Mathematica) and not only as a stand-alone tool. Indeed most end-users are reluctant to learn a new programming language just to solve problems that are only small elements of a more general problem. Furthermore interval analysis efficiency may benefit from the functionalities available in the general purpose scientific software.

5.2. Interval analysis libraries

5.2.1. ALIAS

Participants: Jean-Pierre Merlet [correspondant], Odile Pourtallier.

The ALIAS library (*Algorithms Library of Interval Analysis for Systems*), whose development started in 1998, is a collection of procedures based on interval analysis for systems solving and optimization.

ALIAS is made of two parts:

- ALIAS-C++: the C++ library (87 000 code lines) which is the core of the algorithms
- ALIAS-Maple: the Maple interface for ALIAS-C++ (55 000 code lines). This interface allows one to specify a solving problem within Maple and get the results within the same Maple session. The role of this interface is not only to generate the C++ code automatically, but also to perform an analysis of the problem in order to improve the efficiency of the solver. Furthermore, a distributed implementation of the algorithms is available directly within the interface.

Although these libraries are intended to be used within the project-team they can be freely downloaded as a library file (but the user may introduce its own code in several part of the package) and has been used for example at LIRMM and IRCCyN.

5.3. Platforms

A large number of teams at Inria are developing hardware platforms whose development is quite different from pure software. In our case we have several of such platforms:

- *instrumented flat*: HEPHAISTOS benefits from its own experimental workplace with a simulated flat that includes all the basic home elements (kitchen, bedroom, toilets, relaxation and rehabilitation area)⁰

⁰see <http://www-sop.inria.fr/hephaistos/prototypes/main.html>

- *walking aids family* ANG: ANG-light (for walking analysis), ANG-II (a fully motorized rollator) and ANG-med (with adjustable friction brakes in the rear wheels).
- *cable-driven parallel robots family* MARIONET: MARIONET-ASSIST for transfer and manipulation, MARIONET-REHAB for rehabilitation purposes, MARIONET-VR for rehabilitation and training in an immersive room, MARIONET-SCHOOL for dissemination
- *miscellaneous robots and sensors*: mobile robots (Roomba, Wanny, PoBots), a motion base supporting up to 250 kg, a motion capture system with 12 cameras, force plates ...

INDES Project-Team

5. New Software and Platforms

5.1. Introduction

Most INDES software packages, even the older stable ones that are not described in the following sections, are freely available on the Web. In particular, some are available directly from the Inria web site:

<http://www.inria.fr/valorisation/logiciels/langages.fr.html>

Most software packages can be downloaded from the INDES web site:

<http://www-sop.inria.fr/teams/indes>

5.2. Language-based Security

Participants: José Fragoso Santos, Tamara Rezk [correspondant].

5.2.1. JavaScript Library *iflowtypes.js*

The JavaScript library *iflowtypes.js* is designed to type secure information flow in JavaScript. *iflowtypes.js* has two main modes of operation: fully static and hybrid. In the hybrid mode, the program to be typed is instrumented with runtime assertions that are verified at runtime. By deferring rejection to runtime, the hybrid type system is able to type more programs than fully static mechanisms. This library is available at the URL: <http://j3fsantos.github.io/PersonalPage/TypeSystem/>.

5.2.2. JavaScript Library *iflowsigs.js*

The JavaScript library *iflowsigs.js* is designed to inline an information flow monitor into JavaScript code. *iflowsigs.js* supports is able to track information flow even in programs that interact with arbitrary Web APIs. This library is available at the URL: <http://j3fsantos.github.io/PersonalPage/IFMonitor/>.

5.3. Web programming

Participants: Yoann Couillec, Vincent Prunet, Manuel Serrano [correspondant].

5.3.1. The HOP web programming environment

HOP is a higher-order language designed for programming interactive web applications such as web agendas, web galleries, music players, etc. It exposes a programming model based on two computation levels. The first one is in charge of executing the logic of an application while the second one is in charge of executing the graphical user interface. HOP separates the logic and the graphical user interface but it packages them together and it supports strong collaboration between the two engines. The two execution flows communicate through function calls and event loops. Both ends can initiate communications.

The HOP programming environment consists in a web *broker* that intuitively combines in a single architecture a web server and a web proxy. The broker embeds a HOP interpreter for executing server-side code and a HOP client-side compiler for generating the code that will get executed by the client.

An important effort is devoted to providing HOP with a realistic and efficient implementation. The HOP implementation is *validated* against web applications that are used on a daily-basis. In particular, we have developed HOP applications for authoring and projecting slides, editing calendars, reading RSS streams, or managing blogs.

HOP has won the software *open source contest* organized by the ACM Multimedia Conference 2007. It is released under the GPL license. It is available at <http://hop.inria.fr>.

5.4. Old software

5.4.1. Camloo

Camloo is a caml-light to bigloo compiler, which was developed a few years ago to target bigloo 1.6c. New major releases 0.4.x of camloo have been done to support bigloo 3.4 and bigloo 3.5. Camloo make it possible for the user to develop seamlessly a multi-language project, where some files are written in caml-light, in C, and in bigloo. Unlike the previous versions of camloo, 0.4.x versions do not need a modified bigloo compiler to obtain good performance. Currently, the only supported backend for camloo is bigloo/C. We are currently rewriting the runtime of camloo in bigloo to get more portability and to be able to use HOP and camloo together.

5.4.2. Skribe

SKRIBE is a functional programming language designed for authoring documents, such as Web pages or technical reports. It is built on top of the SCHEME programming language. Its concrete syntax is simple and looks familiar to anyone used to markup languages. Authoring a document with SKRIBE is as simple as with HTML or LaTeX. It is even possible to use it without noticing that it is a programming language because of the conciseness of its original syntax: the ratio *tag/text* is smaller than with the other markup systems we have tested.

Executing a SKRIBE program with a SKRIBE evaluator produces a target document. It can be HTML files for Web browsers, a LaTeX file for high-quality printed documents, or a set of *info* pages for on-line documentation.

5.4.3. Scheme2JS

Scm2JS is a Scheme to JavaScript compiler distributed under the GPL license. Even though much effort has been spent on being as close as possible to R5RS, we concentrated mainly on efficiency and interoperability. Usually Scm2JS produces JavaScript code that is comparable (in speed) to hand-written code. In order to achieve this performance, Scm2JS is not completely R5RS compliant. In particular it lacks exact numbers.

Interoperability with existing JavaScript code is ensured by a JavaScript-like dot-notation to access JavaScript objects and by a flexible symbol-resolution implementation.

Scm2JS is used on a daily basis within HOP, where it generates the code which is sent to the clients (web-browsers). Scm2JS can be found at <http://www-sop.inria.fr/indes/scheme2js>.

5.4.4. The FunLoft language

FunLoft (described in <http://www-sop.inria.fr/teams/indes/rp/FunLoft>) is a programming language in which the focus is put on safety and multicore.

FunLoft is built on the model of FairThreads which makes concurrent programming simpler than usual preemptive-based techniques by providing a framework with a clear and sound semantics. FunLoft is designed with the following objectives:

- provide a safe language, in which, for example, data-races are impossible.
- control the use of resources (CPU and memory), for example, memory leaks cannot occur in FunLoft programs, which always react in finite time.
- have an efficient implementation which can deal with large numbers of concurrent components.
- benefit from the real parallelism offered by multicore machines.

A first experimental version of the compiler is available on the Reactive Programming site <http://www-sop.inria.fr/teams/indes/rp>. Several benchmarks are given, including cellular automata and simulation of colliding particles.

5.4.5. The Bigloo compiler

The programming environment for the Bigloo compiler [7] is available on the Inria Web site at the following URL: <http://www-sop.inria.fr/teams/index/fp/Bigloo>. The distribution contains an optimizing compiler that delivers native code, JVM bytecode, and .NET CLR bytecode. It contains a debugger, a profiler, and various Bigloo development tools. The distribution also contains several user libraries that enable the implementation of realistic applications.

BIGLOO was initially designed for implementing compact stand-alone applications under Unix. Nowadays, it runs harmoniously under Linux and MacOSX. The effort initiated in 2002 for porting it to Microsoft Windows is pursued by external contributors. In addition to the native back-ends, the BIGLOO JVM back-end has enabled a new set of applications: Web services, Web browser plug-ins, cross platform development, etc. The new BIGLOO .NET CLR back-end that is fully operational since release 2.6e enables a smooth integration of Bigloo programs under the Microsoft .NET environment.

5.4.6. CFlow

The prototype compiler “CFlow” takes as input code annotated with information flow security labels for integrity and confidentiality and compiles to F# code that implements cryptography and protocols that satisfy the given security specification.

Cflow has been coded in F#, developed mainly on Linux using mono (as a substitute to .NET), and partially tested under Windows (relying on .NET and Cygwin). The code is distributed under the terms of the CeCILL-B license.

5.4.7. FHE type-checker

We have developed a type checker for programs that feature modern cryptographic primitives such as fully homomorphic encryption. The type checker is thought as an extension of the “CFlow” compiler developed last year on the same project. It is implemented in F#. The code is distributed under the terms of the CeCILL-B license.

5.4.8. Mashic compiler

The Mashic compiler is applied to mashups with untrusted scripts. The compiler generates mashups with sandboxed scripts, secured by the same origin policy of the browsers. The compiler is written in Bigloo and can be found at <http://www-sop.inria.fr/index/mashic/>.

5.4.9. IFJS compiler

The IFJS compiler is applied to JavaScript code. The compiler generates JavaScript code instrumented with checks to secure code. The compiler takes into account special features of JavaScript such as implicit type coercions and programs that actively try to bypass the inlined enforcement mechanisms. The compiler guarantees that third-party programs cannot (1) access the compiler internal state by randomizing the names of the resources through which it is accessed and (2) change the behaviour of native functions that are used by the enforcement mechanisms inlined in the compiled code.

The compiler is written in JavaScript and can be found at <http://www-sop.inria.fr/index/ifJS>.

LAGADIC Project-Team

5. New Software and Platforms

5.1. ViSP: a visual servoing and tracking software library

Participants: Fabien Spindler [correspondant], Aurélien Yol, Eric Marchand, François Chaumette.

Since 2005, we develop and release under the terms of the GPLv2 license, ViSP, an open source library available from <http://team.inria.fr/lagadic/visp>. It allows fast prototyping of visual tracking and visual servoing tasks. ViSP was designed to be independent with the hardware, to be simple to use, expandable and cross-platform.

ViSP allows to design vision-based tasks for eye-in-hand and eye-to-hand visual servoing that contains the most classical visual features that are used in practice. It involves a large set of elementary positioning tasks with respect to various visual features (points, segments, straight lines, circles, spheres, cylinders, image moments, pose...) that can be combined together, and image processing algorithms that allow tracking of visual cues (dots, segments, ellipses...), 3D model-based tracking of known objects or template tracking. Simulation capabilities are also available. ViSP and its full functionalities are presented in Fig. 1 and described in [5].

This year, we continued our efforts to improve the software by ensuring the compatibility with third-party libraries that evolves a lot like CMake 3.0.0 and OpenCV 3.0.0 and by enlarging the compatibility with exotic platforms like RaspberryPi. We also fixed some issues, allowed the model-based tracker to consider circles. We introduced new bar code and face detection but also tracking capabilities. Moreover, we completely re-factored the capabilities concerning keypoint detection and matching. We improved the documentation by providing new tutorials covering the main capabilities of the software. A new release was produced in February. The source code tarball was downloaded 1000 times. With the help of the community, this release was packaged for Debian and Ubuntu 14.04. A new release is in preparation.

Concerning ROS community, all the existing packages in “vision_visp” ROS stack (see http://wiki.ros.org/vision_visp) were updated and ported to indigo build system. To ease ViSP usage in the ROS framework, the last release was packaged for ROS.

ViSP is used in research labs in France, USA, Japan, Korea, India, China, Lebanon, Italy, Spain, Portugal, Hungary, Canada. For instance, it is used as a support in graduate courses at IFMA Clermont-Ferrand, University of Picardie in Amiens, Télécom Physique in Strasbourg and ESIR in Rennes.

5.2. DESlam software

Participant: Patrick Rives [correspondant].

The DESlam (Dense Egocentric Slam) software developed in collaboration with Andrew Comport from I3S in Sophia Antipolis was registered to the APP (“Agence de Protection des Programmes”) (IDDN.FR.001.320001.000.S.P.2012.000.21000). This software proposes a full and self content solution to the dense Slam problem. Based on a generic RGB-D representation valid for various type of sensors (stereovision, multi-cameras, RGB-D sensors...), it provides a 3D textured representation of complex large indoor and outdoor environments and it allows localizing in real time (45Hz) a robot or a person carrying out a mobile camera.

5.3. HandiViz software

Participants: Marie Babel [correspondant], François Pasteau.

The HandiViz software proposes a semi-autonomous navigation framework of a wheelchair relying on visual servoing. It has been registered to the APP (“Agence de Protection des Programmes”) as an INSA software (IDDN.FR.001.440021.000.S.P.2013.000.10000) and is under GPL license.

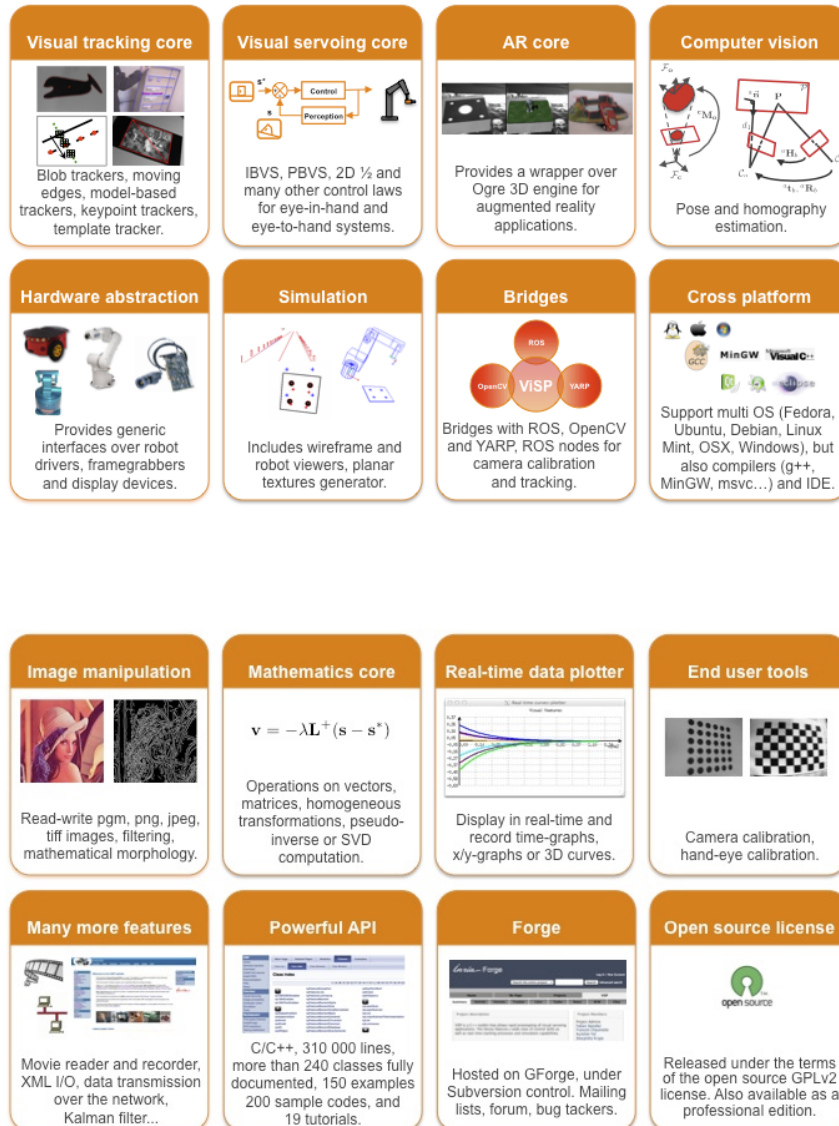


Figure 1. This figure highlights ViSP main capabilities for visual tracking, visual servoing, and augmented reality that may benefit from computer vision algorithms. ViSP allows controlling specific platforms through hardware abstraction or in simulation. ViSP provides also bridges over other frameworks such as OpenCV and ROS. All these capabilities are cross-platform. Moreover, for easing the prototyping of applications, ViSP provides tools for image manipulation, mathematics, data plotting, camera calibration, and many other features. ViSP powerful API is fully documented and available on Inria's forge as an open source software under GPLv2 license.

5.4. Platforms

5.4.1. Robot vision platforms

Participant: Fabien Spindler [correspondant].

We exploit two industrial robotic systems built by Afma Robots in the nineties to validate our researches in visual servoing and active vision. The first one is a Gantry robot with six degrees of freedom, the other one is a cylindrical robot with four degrees of freedom (see Fig. 2). These robots are equipped with cameras. The Gantry robot allows also to embed grippers on its end-effector.

Seven papers published by Lagadic in 2014 enclose results validated on this platform [12], [18], [21], [24], [47], [51], [52].

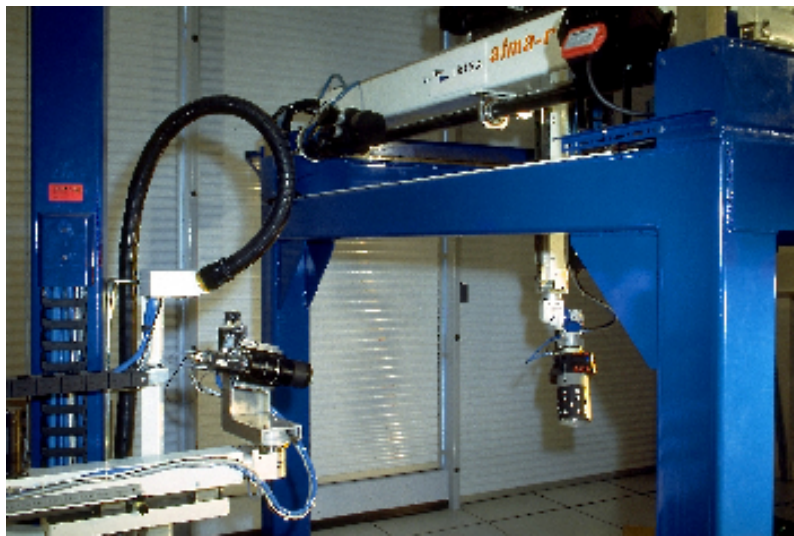


Figure 2. Lagadic robotics platforms for vision-based manipulation

5.4.2. Mobile robotics platforms

Participants: Fabien Spindler [correspondant], Erwan Demairy, Marie Babel, Patrick Rives.

5.4.2.1. Indoor mobile robots

For fast prototyping of algorithms in perception, control and autonomous navigation, the team uses Hannibal in Sophia Antipolis, a cart-like platform built by Neobotix (see Fig. 3 .a), and, in Rennes, a Robotino from Festo (see Fig. 3 .b) and Pioneer 3DX from Adept (see Fig. 3 .c). These platforms are equipped with various sensors needed for Slam purposes, autonomous navigation and sensor-based control.

Moreover, to validate the researches in personally assisted living topic (see 6.2.1), we have in Rennes a six wheel electric wheelchair from Penny and Giles Drives Technology (see Fig. 3 .d) and a five wheel electric wheelchair from You-Q (see Fig. 3 .e). The control of the wheelchair is performed using a plug and play system between the joystick and the low level control of the wheelchair. Such a system lets us acquire the user intention through the joystick position and control the wheelchair by applying corrections to its motion. The wheelchairs have been fitted with cameras and eleven ultrasound sensors to perform the required servoing for assisting handicapped people.

Note that eleven papers exploiting the indoors mobile robots were published this year [16], [29], [30], [31], [33], [37], [43], [41], [42], [56], [58].

5.4.2.2. *Outdoor mobile robots*

The team exploits also Cycab urban electrical cars (see Figs. 3 .f and 3 .g). Two vehicles in Sophia Antipolis and one in Rennes are instrumented with cameras and range finders to validate researches in the domain of intelligent urban vehicle. Cycabs were used as experimental testbeds in several national projects.

Two papers published by Lagadic in 2014 enclose experimental results obtained with these outdoor mobile robots [11], [14].

5.4.2.3. *Technological Development Action (ADT) P2N*

The ADT P2N aims at sharing existing and in development codes between the Lagadic and E-Motion teams in the field of autonomous navigation of indoor robots. These codes are also used in the platforms involved in the large-scale initiative action PAL (Personnally Assisted Living, see Section 8.2.6).

This year, the most notable activities for this ADT have been to:

- make the Slam module developed by Lagadic usable by the E-Motion navigation module;
- port the code on the wheelchairs used in PAL;
- develop the core architecture running under ROS supporting the different sensors and platforms available in Sophia-Antipolis.
- demonstrate the social based navigation methods on the Hannibal platform (see Section 6.2.3)

5.4.3. *Medical robotics platforms*

Participants: Fabien Spindler [correspondant], Alexandre Krupa.

This testbed is of primary interest for researches and experiments concerning ultrasound visual servoing applied to probe positioning, soft tissue tracking or robotic needle insertion tasks described in Section 6.5

This platform is composed by two Adept Viper six degrees of freedom arms (see Fig. 4 .a). Ultrasound probes connected either to a SonoSite 180 Plus or an Ultrasonix SonixTouch imaging system can be mounted on a force torque sensor attached to each robot end-effector.

We designed an experimental setup to test an autonomous robotic needle insertion method based on visual servoing 6.5.3 . The experimental setup is composed with a gelatin phantom simulating soft tissues, a flexible biopsy needle actuated by an Adept Viper arm and a 3D ultrasound probe held by the second Adept Viper arm (see Fig. 4 .b).

This year, six papers enclose experimental results obtained with this platform [13], [34], [35], [48], [49], [50].

5.4.4. *Humanoid robot*

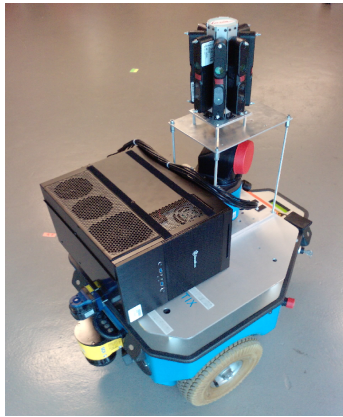
Participants: Giovanni Claudio, Fabien Spindler [correspondant].

Romeo is a humanoid robot from Aldebaran Robotics which is intended to be a genuine personal assistant and companion. In September, we were the first of the four European research laboratories that acquire a Romeo. For the moment only the upper part of the body (arms, head) is working. This research platform is now being used to validate our researches. We developed a first demonstration that make use of visual servoing and visual tracking approaches developed in the team to grasp a box and deliver it to a human (see Fig. 5).

5.4.5. *Unmanned Aerial Vehicles (UAVs)*

Participants: Fabrizio Schiano, Paolo Robuffo Giordano.

In 2014 the team also started some activities involving perception and control for single and multiple quadrotor UAVs, especially thanks to a grant from “Rennes Métropole” (see Section 8.1.4). To this end, we purchased two quadrotors from Mikrokopter GmbH, Germany (Fig. 6 .a), and one quadrotor from 3DRobotics, USA (Fig. 6 .b). These quadrotors will be used as robotic platforms for testing a number of single and multiple flight control schemes with a special attention on the use of onboard vision as main sensory modality.



(a)



(b)



(c)



(d)



(e)

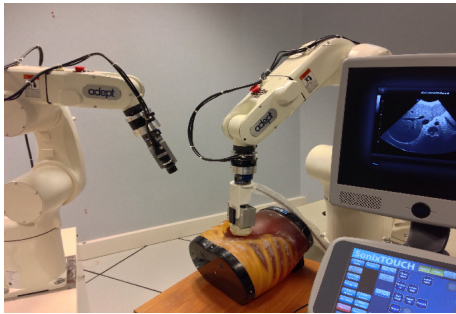


(f)

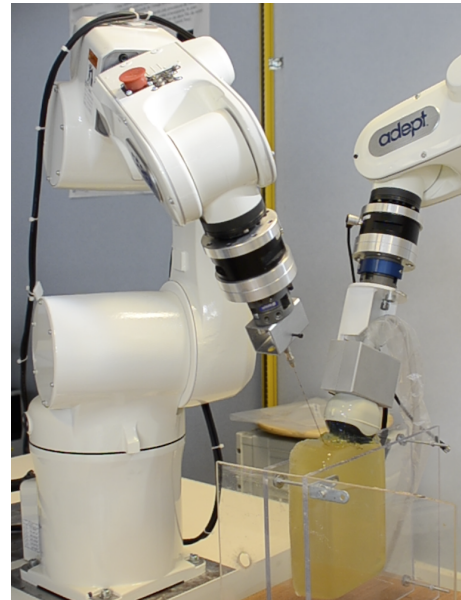


(g)

Figure 3. a) Hannibal platform, b) Robotino, c) Pioneer P3-DX robot, d) wheelchair from Penny and Giles Drives Technology, e) wheelchair from You-Q, f) Cycab available in Rennes, g) one of the Cycabs available in Sophia Antipolis.



(a)



(b)

Figure 4. a) Lagadic medical robotics platforms. On the right Viper S850 robot arm equipped with a SonixTouch 3D ultrasound probe. On the left Viper S650 equipped with a tool changer that allows to attach a classical camera or biopsy needles. b) Robotic setup for autonomous needle insertion by visual servoing.

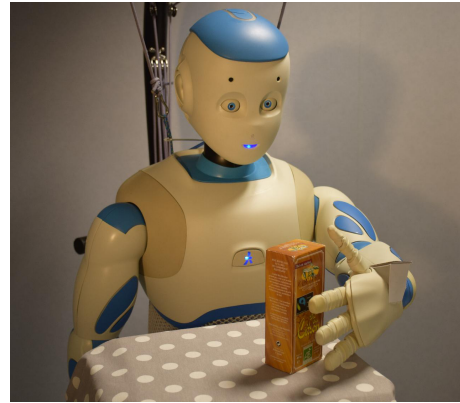


Figure 5. Romeo experimental platform.



(a)



(b)

Figure 6. a) Quadrotor XL1 from Mikrokopter, b) Quadrotor Iris from 3DRobotics

LEMON Team

5. New Software and Platforms

5.1. SW2D

Participant: Vincent Guinot.

Urban floods are usually simulated using two-dimensional shallow water models. A correct representation of the urban geometry and hydraulics would require that the average computational cell size be between 0.1 m and 1 m. The meshing and computation costs make the simulation of entire districts/conurbations impracticable in the current state of computer technology.

An alternative approach consists in upscaling the shallow water equations using averaging techniques. This leads to introducing storage and conveyance porosities, as well as additional source terms, in the mass and momentum balance equations. Various versions of porosity-based shallow water models have been proposed in the literature. The Shallow Water 2 Dimensions (SW2D) computational code embeds various finite volume discretizations of these models. It uses fully unstructured meshes with arbitrary numbers of edges. The key features of the models and numerical techniques embedded in SW2D are

- specific momentum/energy dissipation models that are active only under transient conditions. Such models, that are not present in classical shallow water models, stem from the upscaling of the shallow water equations and prove essential in modeling the features of fast urban flow transients accurately
- modified HLLC solvers for an improved discretization of the momentum source terms stemming from porosity gradients
- higher-order reconstruction techniques that allow for faster and more stable calculations in the presence of wetting/drying fronts.

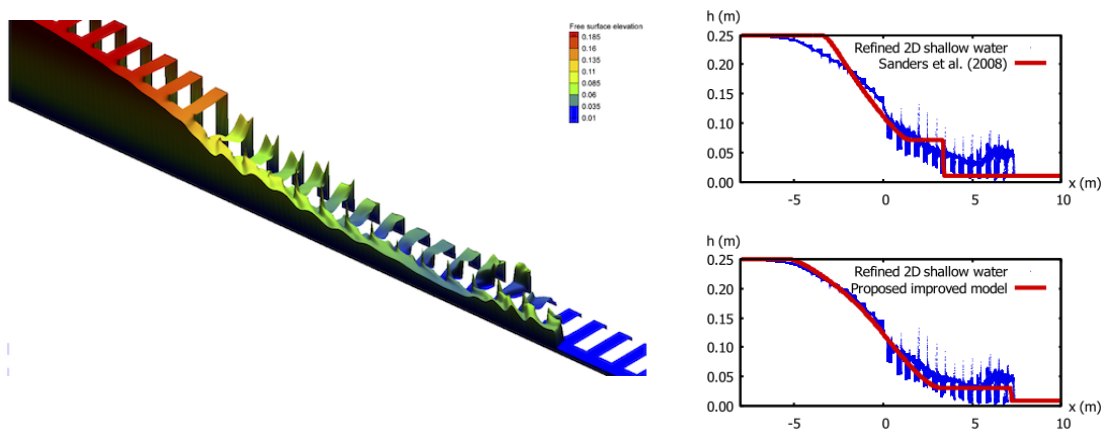


Figure 2. Propagation of a flood wave into a channel with lateral storage. Refined 2D simulation using the SW2D computational code

5.2. Stochastic Downscaling Method

Participant: Antoine Rousseau.

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 modeling 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 modeling of the Fundacion Inria Chile with the Windpos project. In France, its development is going on through the collaborative Modéol project on the evaluation of wind potential.

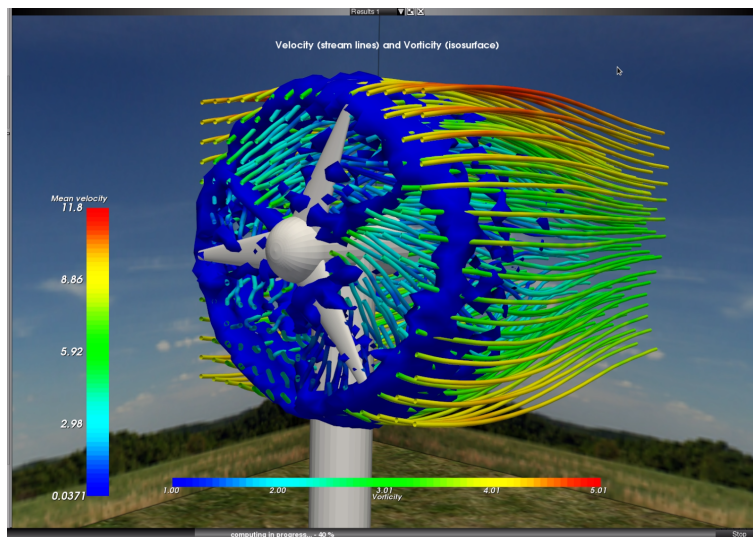


Figure 3. Velocity streamlines and vorticity around a wind mill (artistic view). WINDPOS Project.

This is a joint work with Mireille Bossy from the team TOSCA.

5.3. Action Dépollution

Participants: Antoine Rousseau, Alexis Pacholik.

Action Dépollution (see [website](#) in french) is a serious game made for learning how to purify fast and well a water reservoir, such as lakes. In the scope of the international initiative Mathematics of Planet Earth, this game shows an application of mathematics related to environmental education and sustainable development. The player can act as a researcher, that compares different strategies and looks for the best solution. The conception has been achieved in collaboration with the Inria project-team MODEMIC, and the realization with the help of the start-up Funkadelichik, sponsored by the french consortium Cap' Maths and Inria (Direction de la Communication).

This work is in connection with the INRA/Inria patent [19].

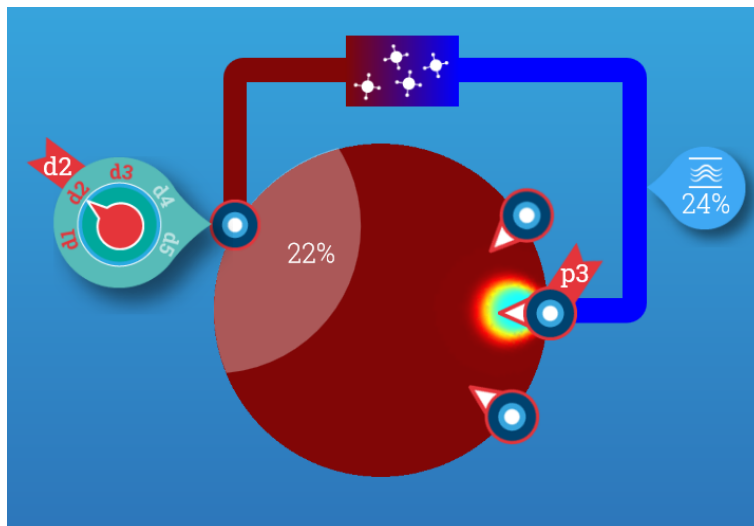


Figure 4. Player interface. Serious game Action Dépollution.

MAESTRO Project-Team

5. New Software and Platforms

5.1. New Software

5.1.1. *ns-3*

Participants: Sara Alouf, Abdulhalim Dandoush, Giovanni Neglia.

ns-3 is an open source, C++ based, GPL licensed and highly used discrete-event network simulator. It is targeted primarily for research and educational use. ns-3 is particularly suited for the goals of the research project with ALSTOM Transport (see §7.1.3). New modules have been developed to enable the simulation of the real antennas used by ALSTOM. Also, modules related to the handoff procedure were debugged and modified to fit the proprietary algorithm used by ALSTOM. Another new module allows to simulate the proprietary communication-based train control protocol used by ALSTOM.

5.2. Platforms

5.2.1. *Marmote*

Participants: Alain Jean-Marie, Issam Rabhi.

In the framework of the ANR MARMOTE, a new software platform dedicated to Markovian modeling is being built. The architecture has been defined so as to be compatible with the software previously developed by members of the project, principally PSI (from the team MESCAL, joint between Inria, Univ. Joseph Fourier (Grenoble) and Institut polytechnique de Grenoble) and XBORNE (from the MAGMAT team of the Univ. Versailles St Quentin). The platform will provide a user interface allowing the modeler to access to a large number of solution methods for generic Markov models as well as optimized methods for specific families of models.

MARELLE Project-Team

5. New Software and Platforms

5.1. Coq

Participants: Enrico Tassi, Benjamin Grégoire.

Coq is developed mainly in the project-team $\pi.r^2$ with contributions from many other individuals. Enrico Tassi and Benjamin Grégoire are regular contributors. In particular for 2014, Benjamin Grégoire provided advice on connecting virtual machine execution with other aspects of the Coq system and Enrico Tassi worked on a new interactive mode that supports a *document* view of the proof script, with faster user experience. Enrico Tassi also worked on improvements for the use of Coq on Windows.

5.2. Easycrypt

Participants: Gilles Barthe [IMDEA Software Institute], François Dupressoir [IMDEA Software Institute], Benjamin Grégoire [correspondant], César Kunz [IMDEA Software Institute], Benedikt Schmid [IMDEA Software Institute], Pierre-Yves Strub [IMDEA Software Institute].

EasyCrypt is a toolset for reasoning about relational properties of probabilistic computations with adversarial code. Its main application is the construction and verification of game-based cryptographic proofs. EasyCrypt can also be used for reasoning about differential privacy.

5.3. zoocrypt

Participants: Gilles Barthe [IMDEA Software Institute], François Dupressoir [IMDEA Software Institute], Benjamin Grégoire [correspondant], César Kunz [IMDEA Software Institute], Benedikt Schmid [IMDEA Software Institute], Pierre-Yves Strub [IMDEA Software Institute].

ZooCrypt (see <http://www.easycrypt.info/zoocrypt/>) is an automated tool for analyzing the security of padding-based public-key encryption schemes (i.e. schemes built from trapdoor permutations and hash functions). This year we extended the tool to be able to deal with schemes based on cyclic groups and bilinear maps.

5.4. CoqApprox

Participants: Nicolas Brisebarre [CNRS], Mioara Joldes, Érik Martin-Dorel, Micaela Mayero [Iut de Villela-neuse], Jean-Michel Muller, Ioana Paşca [Iut de Nimes], Laurence Rideau, Laurent Théry [correspondant].

We develop a formalization of rigorous polynomial approximation using Taylor models inside the Coq proof assistant, with a special focus on genericity and efficiency for the computations. In 2014, this library has been included in CoqInterval, distributed by the Toccata research team.

5.5. Ssreflect and Mathematical Components

Participants: Yves Bertot, Cyril Cohen, Laurence Rideau, Enrico Tassi [correspondant], Laurent Théry.

Most of the formal proofs developed in our team are integrated in the Ssreflect extension of the Coq system and the Mathematical Components library. Work this year has concentrated on providing new versions of ssreflect that are compatible with the evolutions of Coq (to prepare for the upcoming release) and integrating our results in the description of real numbers. We also laid the foundations for a book explaining the structure and principles at work in the Math-components library.

MCTAO Project-Team

5. New Software and Platforms

5.1. Hampath

Participants: Jean-Baptiste Caillau, Olivier Cots [corresponding participant], Joseph Gergaud.

Hampath is a software developed to solve optimal control problems but also to study Hamiltonian flow. It has been developed since 2009 by members of the APO team from Institut de Recherche en Informatique de Toulouse, jointly with colleagues from the Université de Bourgogne. It is now updated with McTAO team members. See more on <http://cots.perso.math.cnrs.fr/hampath/>.

MODEMIC Project-Team

5. New Software and Platforms

5.1. Action Depollution

Participant: Alain Rapaport.

Action Depollution is a “serious” game for learning how to purify fast and well a water reservoir, such as lakes. In the scope of the international initiative Mathematics of Planet Earth, this game shows an application of mathematics related to environmental education and sustainable development. The player can act as a researcher, that compares different strategies and looks for the best solution. The conception has been achieved with the Inria project-team LEMON, and the realization with the help of the start-up Funkadelichik, sponsored by the french consortium Cap’Maths.

This work is in connection with the INRA/Inria patent [47] that has been deposited jointly with LEMON Team.

5.2. VITELBIO (VIRtual TELluric BIOreactors)

Participants: Jérôme Harmand, Alain Rapaport.

Vitelbio is a simulator of the microbial activity in soils, for which the spatialization is represented as a network of interconnected reservoirs. The software allows to draw an interconnections graph, that respects the constraint of the maximum flow, and to choose the biological characteristics of various bacterial species in competition for a single nutrient. The simulator computes the time evaluations of the different populations in each compartment, and compares the overall yielding of the ecosystem in terms of bio-conversion of the substrate. This software has been developed in the framework of the INRA/Inria project VITELBIO (VIRtual TELluric BIOreactors), with the help of the company ITK. It is today mainly used for educational purposes (in MSC and PhD lectures).

5.3. Mass-structured chemostat simulators

Participants: Fabien Campillo, Coralie Fritsch.

We developed in Python two pieces of software. The first one aims at simulating a chemostat dynamics with a mass-structured bacterial population: first with an IBM approach, second with a integro-differential equation. The latter approach is using uncentered difference scheme; the former one is stochastic and so needs numerous runs to built empirical representation of the distribution law [27].

The second piece of software is a graphical user interface for the previous one, allowing for runs on remote number cruncher and graphical post-treatment of runs.

The need of reusability of these codes leads us to develop them in an oriented programming framework. This work was done with the help of MISTEA (P. Neveu) and I3M (P. Pudlo).

MORPHEME Project-Team

4. New Software and Platforms

4.1. New Software

4.1.1. Stracking

This software is developed within the ANR project MOTIMO. It allows to segment and track spermatozoons from confocal microscopy image sequences [12]. It has been transferred to IFMT, one of our partner of MOTIMO.

4.2. Platforms

4.2.1. Biological Image Platform (PIB)

This platform, based on the DTK meta-platform, aims at gathering the team software development, and at providing a visual development tool.

NACHOS Project-Team

5. New Software and Platforms

5.1. MAXW-DGTD

Participants: Alexandra Christophe-Argenvillier, Loula Fézoui, Stéphane Lanteri [correspondant], Raphaël Léger, Jonathan Viquerat.

MAXW-DGTD is a software suite for the simulation of time domain electromagnetic wave propagation. It implements a solution method for the Maxwell equations in the time-domain. MAXW-DGTD is based on a discontinuous Galerkin method formulated on unstructured triangular (2d case) or tetrahedral (3d case) meshes [19]. Within each element of the mesh, the components of the electromagnetic field are approximated by a arbitrary high order nodal polynomial interpolation method. This discontinuous Galerkin method combines a centered scheme for the evaluation of numerical fluxes at a face shared by two neighboring elements, with an explicit Leap-Frog time scheme. The software and the underlying algorithms are adapted to distributed memory parallel computing platforms thanks to a parallelization strategy that combines a partitioning of the computational domain with message passing programming using the MPI standard. Besides, a peripheral version of the software has been recently developed which is able to exploit the processing capabilities of a hybrid parallel computing system comprising multicore CPU and GPU nodes.

- AMS: AMS 35L50, AMS 35Q60, AMS 35Q61, AMS 65N08, AMS 65N30, AMS 65M60
- Keywords: Computational electromagnetics, Maxwell equations, discontinuous Galerkin, tetrahedral mesh.
- OS/Middleware: Linux
- Required library or software: MPI (Message Passing Interface), CUDA
- Programming language: Fortran 77/95

5.2. MAXW-DGFD

Participants: Stéphane Lanteri [correspondant], Ludovic Moya, Ronan Perrussel.

MAXW-DGFD is a software suite for the simulation of time-harmonic electromagnetic wave propagation. It implements a solution method for the Maxwell equations in the frequency domain. MAXW-DGFD is based on a discontinuous Galerkin method formulated on unstructured triangular (2d case) or tetrahedral (3d case) meshes. Within each element of the mesh, the components of the electromagnetic field are approximated by a arbitrary high order nodal polynomial interpolation method. The resolution of the sparse, complex coefficients, linear systems resulting from the discontinuous Galerkin formulation is performed by a hybrid iterative/direct solver whose design is based on domain decomposition principles. The software and the underlying algorithms are adapted to distributed memory parallel computing platforms thanks to a parallelization strategy that combines a partitioning of the computational domain with a message passing programming using the MPI standard. Some recent achievements have been the implementation of non-uniform order DG method in the 2d case and of a new hybridizable discontinuous Galerkin (HDG) formulation also in the 2d and 3d cases.

- AMS: AMS 35L50, AMS 35Q60, AMS 35Q61, AMS 65N08, AMS 65N30, AMS 65M60
- Keywords: Computational electromagnetics, Maxwell equations, discontinuous Galerkin, tetrahedral mesh.
- OS/Middleware: Linux
- Required library or software: MPI (Message Passing Interface)
- Programming language: Fortran 77/95

5.3. SISMO-DGTD

Participants: Nathalie Glinsky, Stéphane Lanteri [correspondant].

SISMO-DGTD is a software for the simulation of time-domain seismic wave propagation. It implements a solution method for the velocity-stress equations in the time-domain. SISMO-DGTD is based on a discontinuous Galerkin method formulated on unstructured triangular (2d case) or tetrahedral (3d case) meshes [4]. Within each element of the mesh, the components of the electromagnetic field are approximated by an arbitrary high order nodal polynomial interpolation method. This discontinuous Galerkin method combines a centered scheme for the evaluation of numerical fluxes at a face shared by two neighboring elements, with an explicit Leap-Frog time scheme. The software and the underlying algorithms are adapted to distributed memory parallel computing platforms thanks to a parallelization strategy that combines a partitioning of the computational domain with a message passing programming using the MPI standard.

- AMS: AMS 35L50, AMS 35Q74, AMS 35Q86, AMS 65N08, AMS 65N30, AMS 65M60
- Keywords: Computational geoseismics, elastodynamic equations, discontinuous Galerkin, tetrahedral mesh.
- OS/Middleware: Linux
- Required library or software: MPI (Message Passing Interface)
- Programming language: Fortran 77/95

NEUROMATHCOMP Project-Team

4. New Software and Platforms

4.1. Virtual Retina: A Large-Scale Simulator of Biological Retina

Participants: Bruno Cessac, Maria-Jose Escobar [Universidad Técnica Federico Santa María, Valparaiso, Chile], Christobal Nettle [Universidad Técnica Federico Santa María, Valparaiso, Chile], Pierre Kornprobst, Adrien Wohrer [Group for Neural Theory - ENS, Paris, France].

Virtual Retina is a simulation software developed by Adrien Wohrer during his PhD [85], [84] that allows large-scale simulations of biologically-plausible retinas.

Virtual Retina has a variety of biological features implemented such as (i) spatio-temporal linear filter implementing the basic center/surround organization of retinal filtering, (ii) non-linear contrast gain control mechanism providing instantaneous adaptation to the local level of contrast; (iii) spike generation by one or several layers of ganglion cells paving the visual field.

Virtual Retina is under Inria CeCill C open-source licence, so that one can download it, install it and run it on one's own image sequences. Virtual Retina also offers a web service (v 2.1), so that users may test directly the main software on their own data, without any installation. This webservice was developed in collaboration with Nicolas Debeissat (engineer, 2002).

We are now interested in the analysis of the collective behavior of ganglion cells responses. To take this collective behavior into account, Virtual Retina needs to be extended since in its current version, ganglion cells are independent. The goal is to produce better retinal models from experimental recordings obtained with our collaborators at the Institut de la Vision (Olivier Marre and Serge Picaud), Evelyne Sernagor (New Castle University) and Luca Berdondini (IIT) using e.g. multi-electrode arrays. This will allow us to better understand the correlations between retina spikes trains and to improve the Virtual Retina model [84] in such a way that it could reproduce the retinal response at the population level. Another application is to the electric stimulation of a retina with implanted multi-electrode arrays in collaboration with the Institut de la Vision and the INT (Frédéric Chavane). Other evolutions of Virtual Retina are also investigated by external partners like the role/implementation of starburst amacrine cells involved in direction selectivity (collaboration with Universidad Técnica Federico Santa María, Valparaiso, Chile, and Centro de Neurociencia de Valparaiso) (see also e.g., [74]).

- IDDN number: IDDN.FR.001.210034.000.S.P.2007.000.31235
- Version: v 2.2.2 (September 2011)
- Link: <http://www-sop.inria.fr/neuromathcomp/public/software/virtualretina>

4.2. Event Neural Assembly Simulation

Participants: Bruno Cessac, Sélim Kraria [Inria DREAM], Theodora Karvouniari, Hassan Nasser, Daniela Pamplona, Thierry Viéville [Inria Mnemosyne Bordeaux].

With the advent of new Multi-Electrode Arrays (MEA) techniques, the simultaneously recording of the activity of groups of neurons (up to several hundreds) over a dense configuration, supplies today a critical database to unravel the role of specific neural assemblies. Thus, the analysis of spike trains obtained from *in vivo* or *in vitro* experimental data, requires suitable statistical models. The Enas software offers new computational methods taking into account time constraints in neural networks (such as memory effects). It also offers several statistical model choices, some of these models already used in this community, and some others developed by us, and allows a quantitative comparison between these models. It also offers a control of finite-size sampling effects inherent to empirical statistics.

Compared to existing software ([Pandora](#); [Sigtool](#); [Spyke Viewer](#); [Orbital Spikes](#)) Enas offers new computational methods taking into account time constraints in neural networks (such as memory effects), based on theoretical methods rooted in statistical physics and applied mathematics. The algorithms used are based on linear programming, nonlinear parameter estimations, statistical methods.

EnaS allows interfaces with existing toolboxes used by this community such as Matlab.

EnaS is developed jointly by the Neuromathcomp, CORTEX/Mnemosyne, and DREAM Inria teams, under CeCILL-C licence, APP logiciel Enas : IDDN.FR.OO1.190004.000.S.P.2014.000.31235. It can be freely downloaded.

It has benefited from the support of an ADT Inria from 2011 to 2013.

The software is freely downloadable at <https://enas.inria.fr/#download>.

Website: <https://enas.inria.fr/>

OPALE Project-Team

5. New Software and Platforms

5.1. NUM3SIS

Participant: Régis Duvigneau [correspondant].

The Opale project-team has initiated a few years ago the development of NUM3SIS (<http://num3sis.inria.fr>), which is a modular platform devoted to scientific computing and numerical simulation. It is not restricted to a particular application field, but is designed to host complex multidisciplinary simulations. Main application fields are currently Computational Fluid Dynamics (by Opale project-team), Computational Electro-Magnetics (by Nachos project-team) and pedestrian traffic simulation (by Opale project-team). Some components of the platform are also used by the Tosca project-team for CO2 market simulation and wind simulation in collaboration with Ciric (Inria-Chile).

NUM3SIS provides innovative software tools to overcome some limitations encountered by classical monolithic simulation codes. In particular, the platform is based on abstract concepts commonly used in scientific computing, such as mesh, fields, finite-elements, linear solvers etc, that can be implemented in plugins. A fast prototyping of algorithms can be achieved using a visual programming interface. A component is dedicated to deployment on parallel architectures. Moreover, the platform relies on a "store" system to foster exchange of plugins, scripts or data.

This work is being carried out with the support of one engineer in the framework of an ADT (Action de Développement Technologique) program.

5.2. FAMOSA

Participant: Régis Duvigneau [correspondant].

Opale team is developing the software platform FAMOSA (C++), that is devoted to multidisciplinary design optimization in engineering. It integrates the following components:

- an optimization library composed of various algorithms : several descent methods from steepest-descent method to quasi-Newton BFGS method (deterministic, smooth), the Multi-directional Search Algorithm (deterministic, noisy), the Covariance Matrix Adaption Evolution Strategy (semi-stochastic, multi-modal) and the Efficient Global Optimization method (deterministic, multi-modal). It also contains the Pareto Archived Evolution Strategy to solve multi-objective optimization problems ;
- an evaluation library managing the performance estimation process (communication with external simulation tools) ;
- a metamodel library that contains tools to build a database and kriging models that are used to approximate the objective function for different purposes;
- a scenario library that allows to use the previous components to achieve various tasks:
 - Construct a design of experiments ;
 - Construct a metamodel ;
 - Find the design that minimizes a cost functional ;
 - Find the Pareto front for two cost functionals
 - Play a Nash game to find the equilibrium between two criteria ;
 - Apply a multiple gradient descent strategy to improve simultaneously two criteria.

The FAMOSA platform is employed by Opale project-team to test its methodological developments. The platform is also used by the Fluid Mechanics Laboratory at Ecole Centrale de Nantes for hydrodynamic design applications and ONERA for multidisciplinary design optimization (MDO). Moreover, it is presently tested by Peugeot Automotive industry for external aerodynamic design purpose.

5.3. Plugins for AXEL

Participant: Régis Duvigneau [correspondant].

Opale team is developing plugins in the framework of the algebraic modeler Axel, in collaboration with the Galaad project-team. These developments correspond to two research axes :

- isogeometric analysis and design. In particular, two simulation tools for heat conduction and compressible flows have been implemented, in conjunction with some deterministic and semi-stochastic optimization algorithms for optimum-shape design ;
- geometrical modeling for design optimization.

5.4. Integration platform for multidiscipline optimization applications

Participants: Toan Nguyen, Laurentiu Trifan.

A prototype software integration platform is developed and tested for multidiscipline optimization applications. It is based on a workflow management system called YAWL (<http://www.yawlfoundation.org>). The goal is to design, develop and assess high-performance distributed scientific workflows featuring resilience, i.e., fault-tolerance and exception-handling capabilities. The platform is used to experiment new resilience algorithms, including monitoring and management of application-level errors. Errors include time-outs and out of bounds data values. They can be added and modified by the users. The platform is tested against use-cases provided by the industry partners in the OMD2 project supported by the French Agence Nationale de la Recherche. For example, an optimization of a car air-conditioning pipe was implemented and deployed on the Grid5000 infrastructure. It also takes into account run-time errors related to resource consumption, e.g., memory overflow, to automatically and dynamically relocate the applications tasks involved on the various clusters. This work was Laurentiu Trifan's PhD thesis, defended in October 2013.

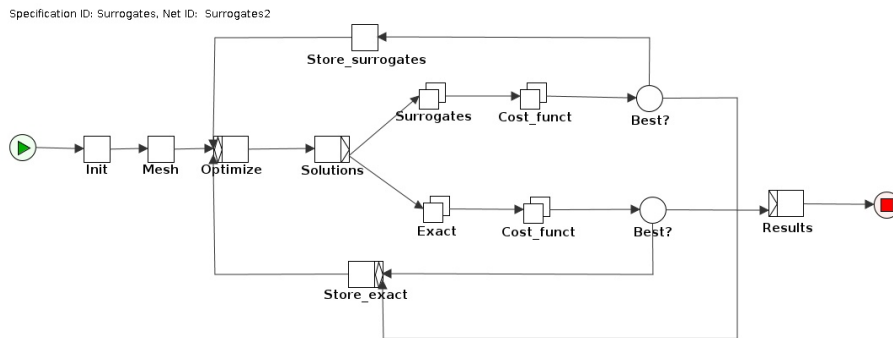


Figure 1. Testcase deployment on the Grid5000 infrastructure.

REVES Project-Team

5. New Software and Platforms

5.1. Multi-View Image-Based Relighting Suite

Participants: Clement Riant, Sylvain Duchêne, Adrien Bousseau, George Drettakis.

We have continued our development of a set of libraries for handling multi-view image-based relighting algorithms. These constitute the basis for the relighting methods developed for the EU projects VERVE and CR-PLAY.

This software package includes a set of modules for processing point clouds and meshes produced by automatic multi-view stereo computer vision solutions. It includes all file management, point cloud and mesh handling, as well as ray-tracing using the Intel Embree ray tracer to compute illumination properties on the mesh. An interactive viewer is also included. A new intrinsic image approach is included as well as a module for relighting and shadow movement, based on an image-driven approach to moving cast shadows.

5.2. IBR-Common

Participants: Jerome Esnault, Gaurav Chaurasia, George Drettakis.

This framework provides common tools, utilities and pieces of code to facilitate prototyping of new ideas related to image-based rendering algorithms. Common features include loading shaders, loading images and 3D reconstructions, setting OpenGL context, basic user interface. The factored architecture of the framework allows users to quickly instantiate custom image-based renderers and test them on common datasets. In addition, a CMake structure automates the handling of cross-platform third-party libraries, file systems and compilation. The framework also allowed us to create a version of image-based rendering dedicated to the Immersive Space, in the context of the VERVE EU project.

5.3. IBR in Unity

Participants: Jerome Esnault, Gaurav Chaurasia, George Drettakis.

We have ported our image-based rendering algorithm to the Unity game engine, in collaboration with the Testaluna game company. This technology transfer is in the context of the CR-PLAY EU project.

Our implementation offers important features to game developers:

- Automatic generation of IBR datasets (calibrated cameras and 3D reconstruction) from multiple images of a scene.
- Ability to use different structure-from-motion (Bundler or VisualSFM) and multiview-stereo algorithms (PMVS or MVE from our partner TU Darmstadt).
- Integration of the rendering algorithm in Unity for game prototyping. This port required us to translate the algorithm from C++ to C# and to adapt shaders to be compatible with Unity requirements.

Figure 3 shows a screenshot of our Unity package in use for the creation of a simple game.

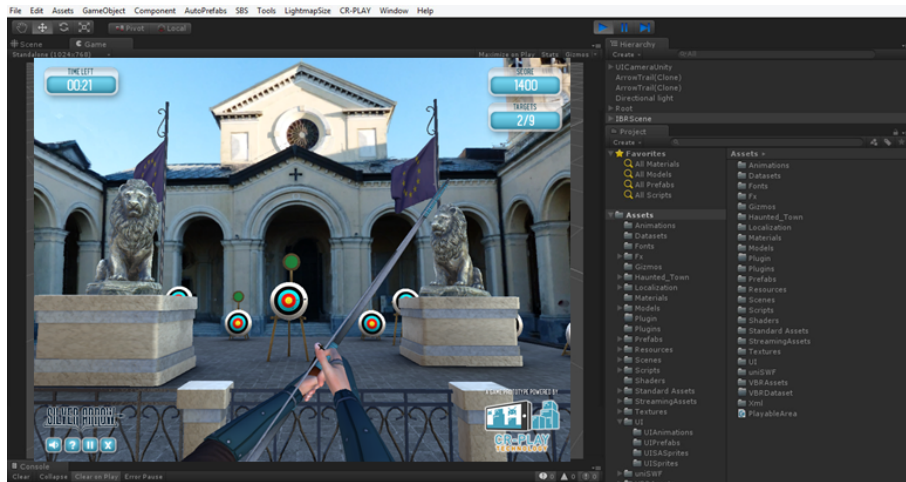


Figure 3. Screen capture of the Unity game development tool. The background buildings are rendered with our image-based rendering algorithm.

SCALE Team

5. New Software and Platforms

5.1. Platforms

5.1.1. EventCloud

Participants : Iyad Alshabani, Maéva Antoine, Françoise Baude, Fabrice Huet, Laurent Pellegrino.

The *EventCloud* is an open source middleware that aims to act as a distributed datastore for data fulfilling the W3C RDF specification (<http://www.w3.org/RDF/>). It allows to store and retrieve quadruples (RDF triples with context) through SPARQL but also to manage events represented as quadruples. The *EventCloud* architecture is based on a structured P2P overlay network targeting high-performance elastic data processing. Consequently it aims to be deployed on infrastructures like grids, clouds, i.e. whose nodes acquisition and relinquishment can be dynamic and subject to a pay-per-use mode. Each node participating in the overlay networks constituting EventCloud instances, is responsible for managing the storage of subsets of the events, and helps in matching potential looked up events and disseminating them in a collaborative manner. As such, each node is also potentially an event broker responsible for managing subscriptions and routing notifications.

The *EventCloud* provides a high level publish-subscribe API where users can register their interests using SPARQL. When matching RDF data are added, subscribers are automatically notified. Recent work around the *EventCloud* has focused on efficient algorithms for managing subscription and notification.

5.1.2. BtrPlace

Participants : Fabien Hermenier, Vincent Kherbache, Ludovic Henrio.

BtrPlace is an open source virtual machine (VM) scheduler for datacenters. *BtrPlace* has been designed to be extensible. It can be customized by plugins from third party developers to address new SLAs or optimization constraints. Its extensibility is possible thanks to a composable core scheduling algorithm implemented using Constraint Programming. *BtrPlace* is currently bundled with a catalog of more than 20 constraints to address performance, fault tolerance, isolation, infrastructure management or energy efficiency concerns. It is currently used inside the FSN project OpenCloudWare (<http://opencloudware.org/>) and the European project DC4Cities (<http://dc4cities.eu/>).

This year we first put an emphase on *BtrPlace* dissemination. *BtrPlace* has been frequently released and it is now available online on a dedicated Web site (<http://btrplace.org>). To increase its visibility and to ease its integration, we decided to made *BtrPlace* directly available from the central repository of maven, the standard system to manage Java projects. Finally, *BtrPlace* has been registered on the *Agence pour la Protection des Programmes*.

5.1.3. OSA

Participants : Olivier Dalle.

OSA stands for Open Simulation Architecture. OSA (<http://osa.inria.fr/>) is primarily intended to be a federating platform for the simulation community: it is designed to favor the integration of new or existing contributions at every level of its architecture. The platform core supports discrete-event simulation engine(s) built on top of the ObjectWeb Consortium's Fractal component model. In OSA, the systems to be simulated are modeled and instrumented using Fractal components. In OSA, the event handling is mostly hidden in the controller part of the components, which alleviates noticeably the modeling process, but also eases the replacement of any part of the simulation engine. Apart the simulation engine, OSA aims at integrating useful tools for modeling, developing, experimenting, and analysing simulations. OSA is also a platform for experimenting new techniques and approaches in simulation, such as aspect oriented programming, separation of concerns, innovative component architectures, and so on.

5.1.4. VerCors

Participants: Eric Madelaine, Ludovic Henrio, Bartłomiej Szejna, Nassim Jibai, Oleksandra Kulankhina, Siqi Li.

The VerCors tools (<http://www-sop.inria.fr/oasis/VerCors>) include front-ends for specifying the architecture and behaviour of components in the form of UML diagrams. We translate these high-level specifications, into behavioural models in various formats, and we also transform these models using abstractions. In a final step, abstract models are translated into the input format for various verification toolsets. Currently we mainly use the various analysis modules of the CADP toolset.

We have achieved this year a major version of the platform frontend, named VCE-v3, that is now distributed on our website, and used by some of our partners. It includes integrated graphical editors for GCM component architecture descriptions, UML classes, interfaces, and state-machines. The user diagrams can be checked using the recently published validation rules from [11]; then the corresponding GCM components can be executed using an automatic generation of the application ADL, and skeletons of Java files.

But VCE-v3 is using the Obeo-designer platform, which is a commercial product, and we have started a port to the newly available Sirius platform (<http://eclipse.org/sirius/>), with the goal to distribute the next major release of VCE, next year, under Sirius.

STARS Project-Team

5. New Software and Platforms

5.1. SUP

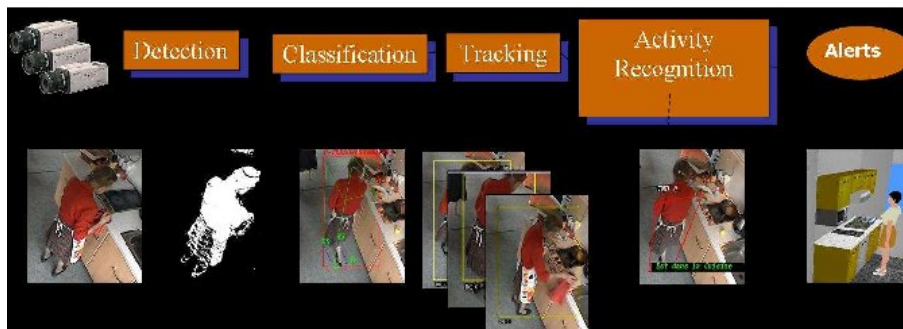


Figure 5. SUP workflow

5.1.1. Presentation

SUP is a Scene Understanding Software Platform (see Figure 5) written in C++ designed for analyzing video content. SUP is able to recognize events such as 'falling', 'walking' of a person. SUP divides the workflow of a video processing into several separated modules, such as acquisition, segmentation, up to activity recognition. Each module has a specific interface, and different plugins (corresponding to algorithms) can be implemented for a same module. We can easily build new analyzing systems thanks to this set of plugins. The order we can use those plugins and their parameters can be changed at run time and the result visualized on a dedicated GUI. This platform has many more advantages such as easy serialization to save and replay a scene, portability to Mac, Windows or Linux, and easy deployment to quickly setup an experimentation anywhere. SUP takes different kinds of input: RGB camera, depth sensor for online processing; or image/video files for offline processing.

This generic architecture is designed to facilitate:

1. integration of new algorithms into SUP;
2. sharing of the algorithms among the Stars team. Currently, 15 plugins are available, covering the whole processing chain. Some plugins use the OpenCV library.

Goals of SUP are twofold:

1. From a video understanding point of view, to allow the Stars researchers sharing the implementation of their algorithms through this platform.
2. From a software engineering point of view, to integrate the results of the dynamic management of vision applications when applying to video analytic.

The plugins cover the following research topics:

- algorithms : 2D/3D mobile object detection, camera calibration, reference image updating, 2D/3D mobile object classification, sensor fusion, 3D mobile object classification into physical objects (individual, group of individuals, crowd), posture detection, frame to frame tracking, long-term tracking of individuals, groups of people or crowd, global tacking, basic event detection (for example entering a zone, falling...), human behaviour recognition (for example vandalism, fighting,...) and event fusion; 2D & 3D visualisation of simulated temporal scenes and of real scene interpretation results; evaluation of object detection, tracking and event recognition; image acquisition (RGB and RGBD cameras) and storage; video processing supervision; data mining and knowledge discovery; image/video indexation and retrieval.
- languages : scenario description, empty 3D scene model description, video processing and understanding operator description;
- knowledge bases : scenario models and empty 3D scene models;
- learning techniques for event detection and human behaviour recognition;

5.1.2. Improvements

Currently, the OpenCV library is fully integrated with SUP. OpenCV provides standardized data types, a lot of video analysis algorithms and an easy access to OpenNI sensors such as the Kinect or the ASUS Xtion PRO LIVE.

In order to supervise the GIT update progress of SUP, an evaluation script is launched automatically everyday. This script updates the latest version of SUP then compiles SUP core and SUP plugins. It executes the full processing chain (from image acquisition to activity recognition) on selected data-set samples. The obtained performance is compared with the one corresponding to the last version (i.e. day before). This script has the following objectives:

- Check daily the status of SUP and detect the compilation bugs if any.
- Supervise daily the SUP performance to detect any bugs leading to the decrease of SUP performance and efficiency.

The software is already widely disseminated among researchers, universities, and companies:

- PAL Inria partners using ROS PAL Gate as middleware
- Nice University (Informatique Signaux et Systèmes de Sophia), University of Paris Est Créteil (UPEC - LISSI-EA 3956)
- EHPAD Valrose, Institut Claude Pompidou
- European partners: Lulea University of Technology, Dublin City University,...
- Industrial partners: Toyota, LinkCareServices, Digital Barriers

Updates and presentations of our framework can be found on our team website <https://team.inria.fr/stars/software> . Detailed tips for users are given on our Wiki website <http://wiki.inria.fr/stars> and sources are hosted thanks to the Inria software developer team SED.

5.2. ViSEvAI

ViSEvAI is a software dedicated to the evaluation and visualization of video processing algorithm outputs. The evaluation of video processing algorithm results is an important step in video analysis research. In video processing, we identify 4 different tasks to evaluate: detection, classification and tracking of physical objects of interest and event recognition.

The proposed evaluation tool (ViSEvAI, visualization and evaluation) respects three important properties:

- To be able to visualize the algorithm results.
- To be able to visualize the metrics and evaluation results.
- To allow users to easily modify or add new metrics.

The ViSEvAI tool is composed of two parts: a GUI to visualize results of the video processing algorithms and metrics results, and an evaluation program to evaluate automatically algorithm outputs on large amounts of data. An XML format is defined for the different input files (detected objects from one or several cameras, ground-truth and events). XSD files and associated classes are used to check, read and write automatically the different XML files. The design of the software is based on a system of interfaces-plugins. This architecture allows the user to develop specific treatments according to her/his application (e.g. metrics). There are 6 user interfaces:

1. The video interface defines the way to load the images in the interface. For instance the user can develop her/his plugin based on her/his own video format. The tool is delivered with a plugin to load JPEG image, and ASF video.
2. The object filter selects which objects (e.g. objects far from the camera) are processed for the evaluation. The tool is delivered with 3 filters.
3. The distance interface defines how the detected objects match the ground-truth objects based on their bounding box. The tool is delivered with 3 plugins comparing 2D bounding boxes and 3 plugins comparing 3D bounding boxes.
4. The frame metric interface implements metrics (e.g. detection metric, classification metric, ...) which can be computed on each frame of the video. The tool is delivered with 5 frame metrics.
5. The temporal metric interface implements metrics (e.g. tracking metric, ...) which are computed on the whole video sequence. The tool is delivered with 3 temporal metrics.
6. The event metric interface implements metrics to evaluate the recognized events. The tool provides 4 metrics.

The GUI is composed of 3 different parts:

1. The visualization of results windows dedicated to result visualization (see Figure 6):
 - Window 1: the video window displays the current image and information about the detected and ground-truth objects (bounding-boxes, identifier, type,...).
 - Window 2: the 3D virtual scene displays a 3D view of the scene (3D avatars for the detected and ground-truth objects, context, ...).
 - Window 3: the temporal information about the detected and ground truth objects, and about the recognized and ground-truth events.
 - Window 4: the description part gives detailed information about the objects and the events,
 - Window 5: the metric part shows the evaluation results of the frame metrics.
2. The object window enables the user to choose the object to be displayed (see Figure 7).
3. The multi-view window displays the different points of view of the scene (see Figure 8).

The evaluation program saves, in a text file, the evaluation results of all the metrics for each frame (whenever it is appropriate), globally for all video sequences or for each object of the ground truth.

The ViSEvAI software was tested and validated into the context of the Cofriend project through its partners (Akka, ...). The tool is also used by IMRA, Nice hospital, Institute for Infocomm Research (Singapore), ... The software version 1.0 was delivered to APP (French Program Protection Agency) on August 2010. ViSEvAI is under GNU Affero General Public License AGPL (<http://www.gnu.org/licenses/>) since July 2011. The tool is available on the web page : http://www-sop.inria.fr/teams/pulsar/EvaluationTool/ViSEvAI_Description.html

5.3. Clem

The *Clem Toolkit* [68](see Figure 9) is a set of tools devoted to design, simulate, verify and generate code for LE [18] [81] programs. LE is a synchronous language supporting a modular compilation. It also supports automata possibly designed with a dedicated graphical editor and implicit Mealy machine definition.

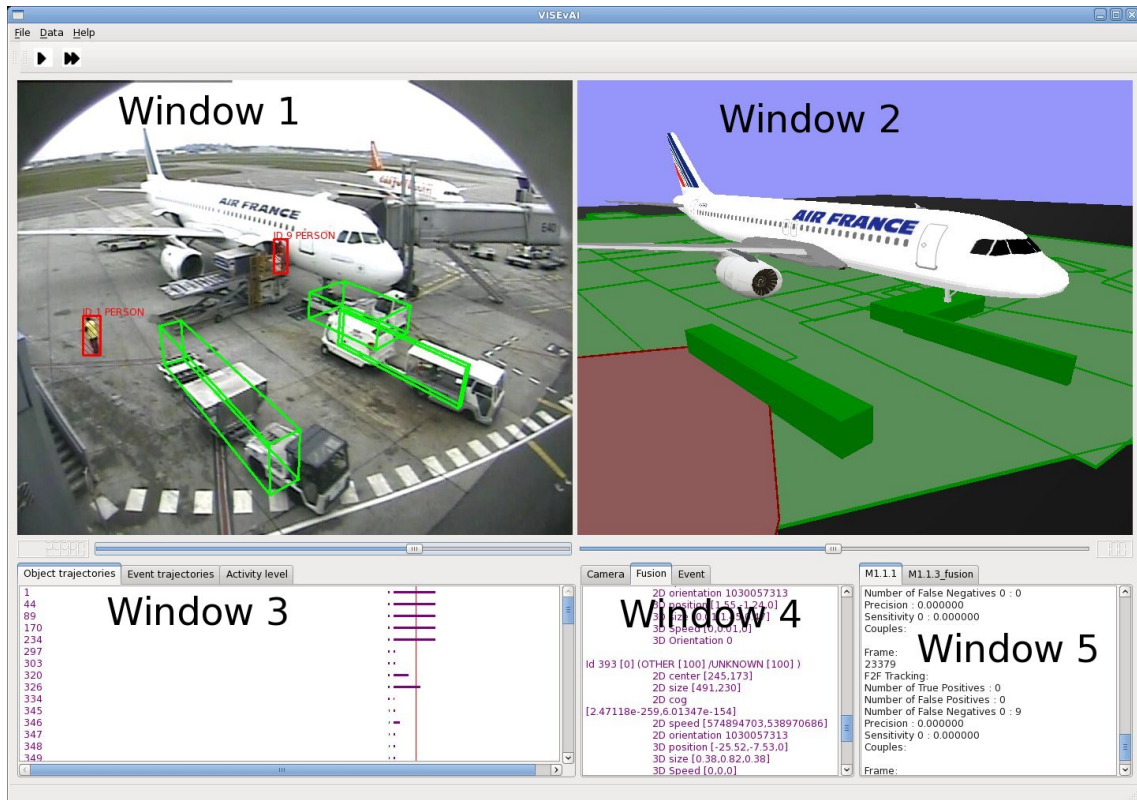


Figure 6. GUI of the ViSEvAI software

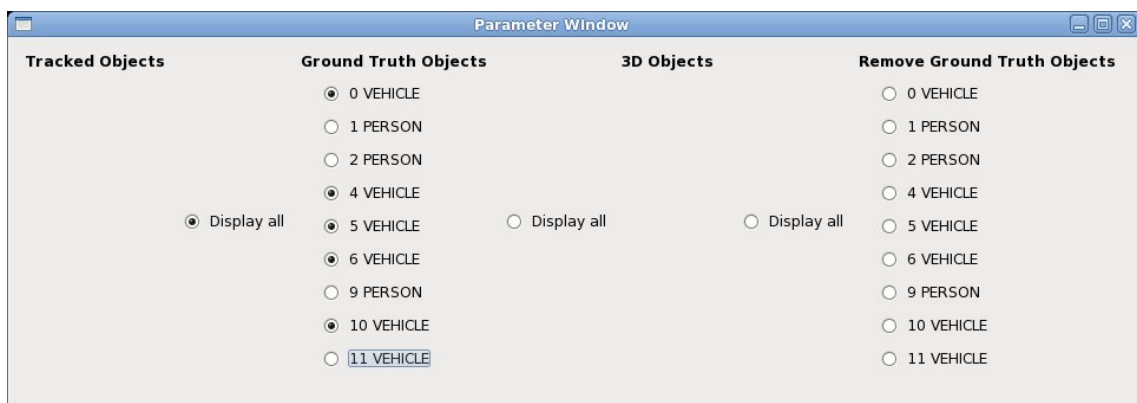


Figure 7. The object window enables users to choose the object to display

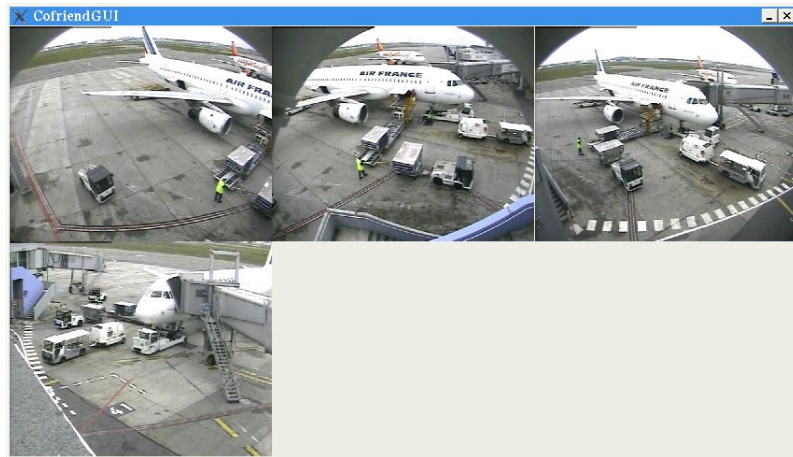


Figure 8. The multi-view window

Each LE program is compiled later into *lec* and *lea* files. Then when we want to generate code for different backends, depending on their nature, we can either expand the *lec* code of programs in order to resolve all abstracted variables and get a single *lec* file, or we can keep the set of *lec* files where all the variables of the main program are defined. Then, the *finalization* will simplify the final equations and code is generated for simulation, safety proofs, hardware description or software code. Hardware description (Vhdl) and software code (C) are supplied for LE programs as well as simulation. Moreover, we also generate files to feed the NuSMV model checker [65] in order to perform validation of program behaviors. In 2014, LE supports data value for automata and CLEM is used in 2 research axes of the team (SAM and SynComp). CLEM is registered at the APP since May 2014.

The work on CLEM was published in [68], [69], [18], [19].

Web page: <http://www-sop.inria.fr/teams/pulsar/projects/Clem/>

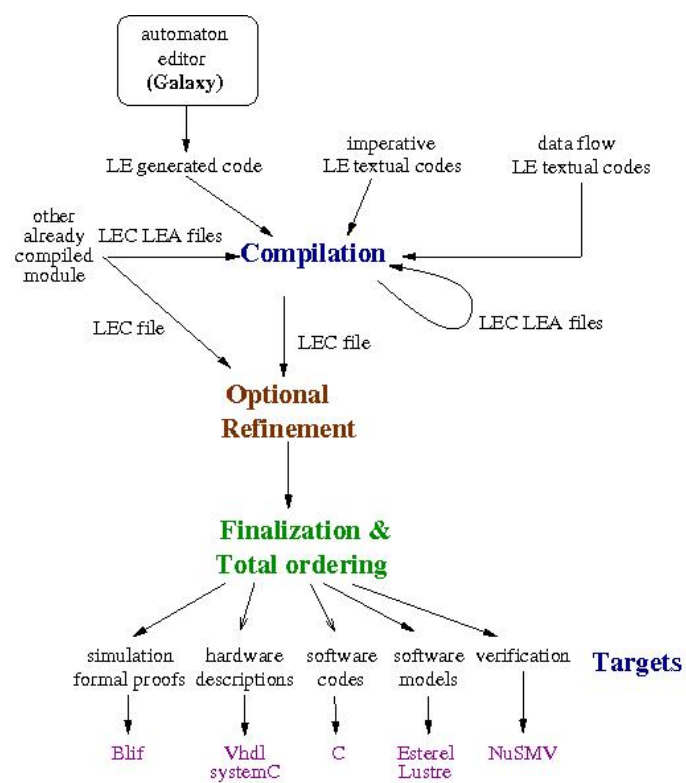


Figure 9. The Clem Toolkit

TITANE Project-Team

5. New Software and Platforms

5.1. CGAL, the Computational Geometry Algorithms Library

Participants: Pierre Alliez, Clement Jamin, Sven Oesau, Thijs Van Lankveld, Nicolas Douillet, David Bommes, Jingjing Shen.

CGAL is a C++ library of geometric algorithms and data structures. Our team is involved in several ongoing implementations: surface reconstruction, point set processing, shape detection in unstructured point sets, constrained 3D Delaunay triangulations, generalized barycentric coordinates (in collaboration with Dmitry Anisimov). Pierre Alliez is a member of the CGAL Editorial Board.

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 developement that includes a N players game approche on an auction carbon market.

- Version: 2.0

VIRTUAL PLANTS Project-Team

4. New Software and Platforms

4.1. OpenAlea

4.1.1. *OpenAlea 2.0*

Participants: Julien Coste, Guillaume Baty, Christophe Pradal, Christophe Godin, Frédéric Boudon, Christian Fournier.

Plant models are usually developed at different scales using various modeling paradigms: (i) imperative using a script or a compiled language, (ii) declarative to define a set of rewriting rules like in L-systems, (iii) interactive using a sketch-based interface for creating 3D models of plants, or (iv) visual programming to combine existing components.

However, all these computational paradigms have been developed in different software platforms in the plant modeling community, and, as of today, none of them provides all the modeling paradigms in an integrated software environment. However, the need to develop more complex and integrated models, often assembling many sub-models, led us to consider a modeling framework capable of supporting multiple design paradigms and models, and make them interoperable.

To address this problem we developed the OpenAlea platform. The Version 1.0 of the platform consisted of a middleware implementing a modular and component-based software architecture for assembling models written in different computer languages. *OpenAlea 2.0* adds to OpenAlea 1.0 a high-level formalism dedicated to the modeling of morphogenesis that makes it possible to use several modeling paradigms (Blackboard, L-systems, Agents, Branching processes, Cellular Automata) expressed with different languages (Python, L-Py, R, Visual Programming, ...) to analyse and simulate shapes and their development.

It offers an integrated modeling software environment *OpenAleaLab* that provides users with flexible and interactive tools to combine different modeling paradigms to support the computational investigation.

4.1.2. *OpenAleaLab*

Participants: Julien Coste, Guillaume Baty, Christophe Pradal, Christophe Godin, Frédéric Boudon, Christian Fournier.

This research theme is supported by the Inria ADT OpenAlea.

OpenAleaLab is a new integrated modeling environment (IME) for OpenAlea. This IME provides an IPython shell, a text editor, a project manager, a toolbox installer, a world data structure containing the objects and state variables shared by the different models and a 3D viewer window that makes it possible to observe the objects of the world. Different modelling paradigms, languages and tools for plant modelling are available as plug-ins, such as a visual programming environment, a L-system language, or a R editor and interpreter. OpenAleaLab is based on IPython architecture and is built using PyQt.

The core of the system is made up of a central data structure (the blackboard) called the world. This data structure may contain various computational objects that altogether define the state of the modeling system, and can be accessed (in read and write) by all the models. The investigation process can be seen as executing the system's models in turn to explore or change dynamically the world objects.

Models are knowledge sources that can modify the world when executed. A model can call for the execution of another model as a function. In this case the model passes an input value to the called model, that in turn returns an output value. In addition it may be possible that the called model changed the world as a side effect. The user launches the execution of a first model (then referred to as the master model), which then entails recursively the hierarchical execution of all the other models downstream of it. One can see that in this framework, the execution controller is then itself considered as a model (the master model).

4.1.3. Similarity and Provenance in OpenAlea workflows

Participants: Sarah Cohen-Boulakia, Christophe Pradal, Moussa Yattara [IBC], Patrick Valduriez [Inria].

This research theme is supported by IBC and Inria.

The number of available scientific workflows, designed in OpenAlea or in other workflow systems such as Galaxy or Taverna, is increasing over time. Methods to compare the scientific workflows become a necessity, to allow duplicate detection or similarity search. Scientific workflows are complex objects, and their comparison entails a number of distinct steps from comparing atomic elements to comparison of the workflows as a whole. Various studies have implemented methods for scientific workflow comparison and came up with often contradicting conclusions upon which algorithms work best. Comparing these results is cumbersome, as the original studies mixed different approaches for different steps and used different evaluation data and metrics.

We first contribute to the field [27] by (i) comparing in isolation different approaches taken at each step of scientific workflow comparison, reporting on an number of unexpected findings, (ii) investigating how these can best be combined into aggregated measures, and (iii) making available a gold standard of over 2000 similarity ratings contributed by 15 workflow experts on a corpus of 1500 workflows and re-implementations of all methods we evaluated.

Then, we introduced a novel and intuitive workflow similarity measure that is based on layer decomposition [39]. Layer decomposition accounts for the directed dataflow underlying scientific workflows, a property which has not been adequately considered in previous methods. We comparatively evaluate our algorithm using our gold standard and show that it a) delivers the best results for similarity search, b) has a much lower runtime than other, often highly complex competitors in structure-aware workflow comparison, and c) can be stacked easily with even faster, structure-agnostic approaches to further reduce runtime while retaining result quality.

Ongoing work includes considering *provenance* traces of executions in the similarity metrics and augmenting the number of workflows to be shared between scientists by working on the *provenance-equivalence* aspects between workflows and (Python) scripts. This work will be done in the context of the IBC Young researcher grant we obtained (co-led by S. Cohen-Boulakia and Ch. Pradal) in collaboration with members of Zenith and the INRA phenome platform.

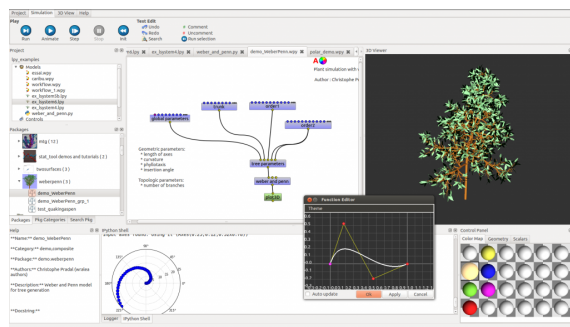


Figure 1. OpenAleaLab - A highly modular environment for modeling morphogenesis.

WIMMICS Project-Team

5. New Software and Platforms

5.1. Corese

Participants: Olivier Corby [correspondant], Alban Gaignard, Fabien Gandon, Fuqi Song.

Corese (COnceptual REsource Search Engine) is a Semantic Web Factory. It enables users to load and process RDFS schemas, RDF data and query and update the graph base thus created by using the SPARQL 1.1 Query & Update Language (figure 1).

Furthermore, Corese query language integrates original features such as approximate search, extended Property Path, SQL or XPath. It provides SPARQL Template Transformation Language for RDF graphs and a SPARQL based Inference Rule Language for RDF. Corese also provides distributed federated query processing, thanks to a collaboration with Alban Gaignard and Johan Montagnat from I3S.

Corese is a Semantic Web Factory that enables us to design and develop Semantic Web applications; it is available for download. In the past, Corese received two software development grants (ADT) from Inria and in 2014 we have a new grant for two more years. Corese is registered at the APP and in 2007 we decided to distribute it as open source software under license CeCILL-C.

Corese is used and has been used in more than 60 applications, 24 PhD Thesis and is used for education by several institutions. It has been used in European projects such as Ontorule, Palette, SevenPro, SeaLife and in ANR projects such as Kolflow, Ginseng, Neurolog, VIP, ISICIL, e-WOK Hub. Corese is the Semantic Web engine of Discovery Hub⁰ and of the Semantic Web Import Plugin for Gephi visualization⁰.

The work on Corese was published in [3], [2], [4], [58].

Web page: <http://wimmics.inria.fr/corese>

5.2. Question Answering Wikiframework-based System

Participant: Elena Cabrio.

The QAKiS system (figure 2) implements question answering over DBpedia. QAKiS allows end users to submit a query to an RDF triple store in English and obtain the answer in the same language, hiding the complexity of the non-intuitive formal query languages involved in the resolution process. At the same time, the expressiveness of these standards is exploited to scale to the huge amounts of available semantic data. Its major novelty is to implement a relation-based match for question interpretation, to convert the user question into a query language (e.g. SPARQL). English, French and German DBpedia chapters are the RDF data sets to be queried using a natural language interface.

Web page: <http://www.qakis.org>

5.3. French Chapter of DBpedia

Participants: Raphaël Boyer, Fabien Gandon.

DBpedia is an international crowd-sourced community effort to extract structured information from Wikipedia and make this information available on the semantic Web as linked open data. The DBpedia triple stores then allow anyone to solve sophisticated queries against Wikipedia extracted data, and to link the different data sets on these data. The French chapter of DBpedia was created and deployed by Wimmics and is now an online running platform providing data to several projects such as: QAKIS, Izipedia, zone47, Sépage, HdA Lab., JocondeLab, etc.

The platform can be found at: <http://www.dbpedia.fr>.

It is part of the Semanticpedia convention: <http://www.semanticpedia.org/>.

⁰<http://www.discoveryhub.co>

⁰<https://marketplace.gephi.org/plugin/semanticwebimport/>

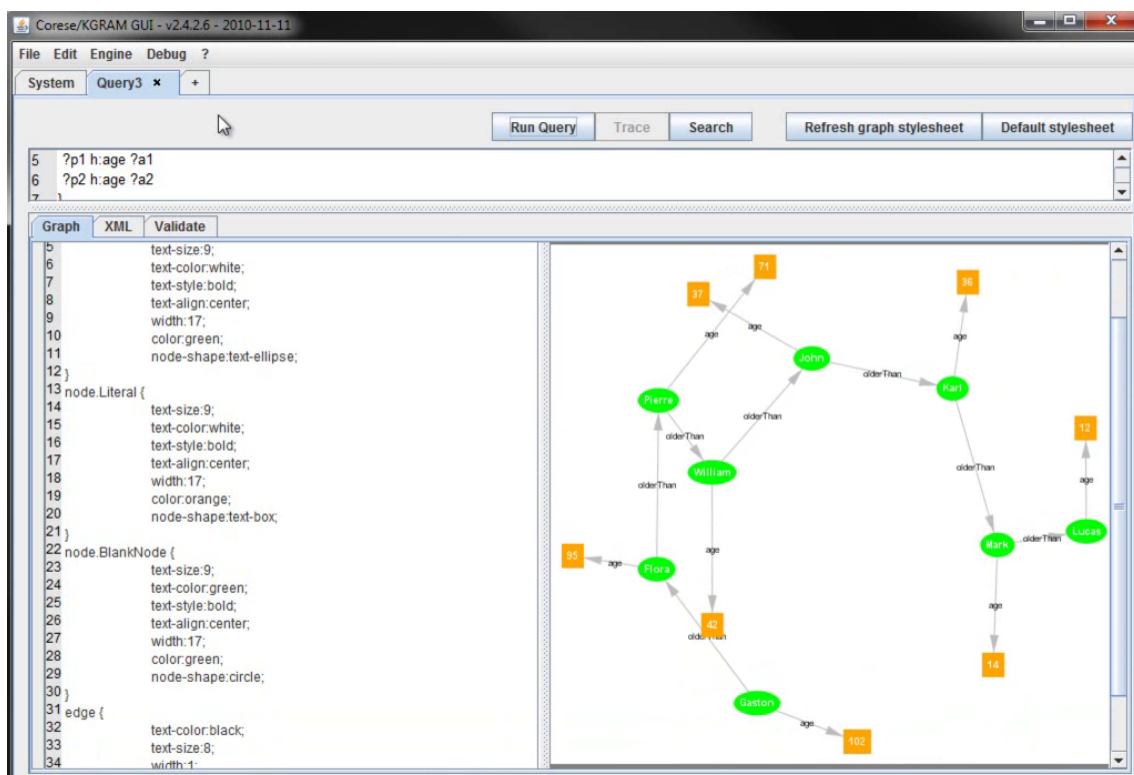


Figure 1. Corese

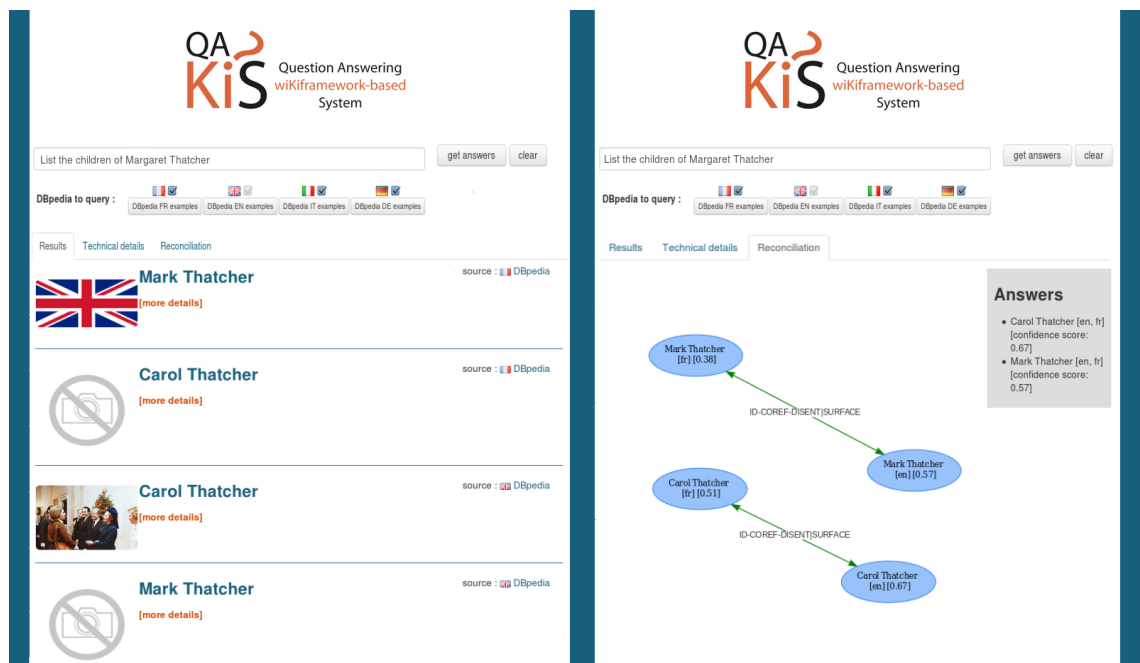


Figure 2. QAKIS

ZENITH Project-Team

5. New Software and Platforms

5.1. Hadoop_g5k

Participants: Reza Akbarinia, Miguel Liroz-Gistau, Patrick Valduriez.

URL: https://www.grid5000.fr/mediawiki/index.php/Hadoop_On_Execo

Apache Hadoop provides an open-source framework for reliable, scalable, parallel computing. It can be deployed and used in large-scale platforms such as Grid 5000. However, its configuration and management is very difficult, specially under the dynamic nature of clusters. Therefore, we built Hadoop_g5k (Hadoop easy deployment in clusters), a tool that makes it easier to manage Hadoop clusters and prepare reproducible experiments. Hadoop_g5k offers a set of scripts to be used in command-line interfaces and a Python interface. It is actually used by Grid5000 users, and helps them saving much time when doing their experiments with MapReduce.

5.2. LogMagnet

Participants: Julien Diener, Florent Masseglia.

URL: <https://team.inria.fr/zenith/?s=LogMagnet>

LogMagnet is a software for analyzing streaming data, and in particular log data. Log data usually arrive in the form of lines containing activities of human or machines. In the case of human activities, it may be the behavior on a Web site or the usage of an application. In the case of machines, such log may contain the activities of software and hardware components (say, for each node of a computing cluster, the calls to system functions or some hardware alerts). Analyzing such data is often difficult and crucial in the meanwhile. LogMagnet allows to summarize this data, and to provide a first analysis as a clustering. This summary may also be exploited as easily as the original data.

5.3. MultiSite-Rec

Participants: Mohamed Reda Bouadjenek, Florent Masseglia, Esther Pacitti.

URL: <https://code.google.com/p/multi-site-rec/>

Recommender systems are used as a mean to supply users with content that may be of interest to them. They have become a popular research topic, where many aspects and dimensions have been studied to make them more accurate and effective. In practice, recommender systems suffer from cold-start problems. However, users use many online services, which can provide information about their interest and the content of items (e.g. Google search engine, Facebook, Twitter, etc). These services may be valuable data sources, which supply information to help a recommender system in modeling users and items' preferences, and thus, make the recommender system more precise. Moreover, these data sources are distributed, and geographically distant from each other, which raise many research problems and challenges to design a distributed recommendation algorithm. MultiSite-Rec is a distributed collaborative filtering algorithm, which exploits and combine these multiple and heterogeneous data sources to improve the recommendation quality.

5.4. PlantRT: Gossip-Based Recommendation

Participants: Alexis Joly, Julien Champ, Miguel Liroz-Gistau, Esther Pacitti, Maximilien Servajean [contact].

URL: <http://www2.lirmm.fr/~servajean/prototypes/plant-sharing/plant-rt.html>

PlantRT is a distributed gossip-based platform for content sharing enabling plants observation keywords search and GPS position based recommendation. It combines advantages from centralized and P2P systems.

5.5. Pl@ntNet

Participants: Julien Champ, Hervé Goëau, Alexis Joly [contact].

URL: <http://goo.gl/CpSrr3>

Pl@ntNet is an image sharing and retrieval application for the identification of plants. It is developed in the context of the Pl@ntNet project that involves four French research organisations (Inria, Cirad, INRA, IRD) and the members of Tela Botanica social network. The key feature of the iOS and Android front ends is to help identifying plant species from photographs, through a server-side visual search engine based on several results of ZENITH team on content-based information retrieval. Since its first release in March 2013 on the apple store, the application was downloaded by around 300K users in more than 150 countries (between 500 and 5000 active users daily with peaks occurring during the week-ends). The collaborative training set that allows the content-based identification is continuously enriched by the users of the application and the members of Tela Botanica social network. At the time of writing, it includes about 100K images covering more than 5000 French plant species about 4/5 of the whole French flora (this is actually the widest identification tool built anytime).

5.6. SON (Shared-data Overlay Network)

Participants: Esther Pacitti, Didier Parigot [contact], Patrick Valduriez.

URL: <http://www-sop.inria.fr/teams/zenith/SON>

SON is an open source development platform for P2P networks using web services, JXTA and OSGi. SON combines three powerful paradigms: components, SOA and P2P. Components communicate by asynchronous message passing to provide weak coupling between system entities. To scale up and ease deployment, we rely on a decentralized organization based on a DHT for publishing and discovering services or data. In terms of communication, the infrastructure is based on JXTA virtual communication pipes, a technology that has been extensively used within the Grid community. Using SON, the development of a P2P application is done through the design and implementation of a set of components. Each component includes a technical code that provides the component services and a code component that provides the component logic (in Java). The complex aspects of asynchronous distributed programming (technical code) are separated from code components and automatically generated from an abstract description of services (provided or required) for each component by the component generator.

5.7. Snoop & SnoopIm

Participants: Alexis Joly [contact], Julien Champ, Jean-Christophe Lombardo.

URL: <http://otmedia.lirmm.fr/>

Snoop is a generalist C++ library dedicated to high-dimensional data management and efficient similarity search. Its main features are dimension reduction, high-dimensional feature vectors hashing, approximate k-nearest neighbors search and Hamming embedding. Snoop is a refactoring of a previous library called PMH developed jointly with the French National Institute of Audiovisual. It is based on the joined research work of Alexis Joly and Olivier Buisson. SnoopIm is a content-based image search engine built on top of Snoop and allowing to retrieve small visual patterns or objects in large collections of pictures. The software is being experimented in several contexts including a logo retrieval application set up in collaboration with the French Press Agency, an experimental plant identification tool mixing textual and visual information retrieval (in the context of the Pl@ntNet project) and a research project on high-throughput analysis of root architecture images.

5.8. SciFloware

Participants: Dimitri Dupuis, Didier Parigot [contact].

URL: <http://www-sop.inria.fr/members/Didier.Parigot/pmwiki/Scifloware>

SciFloware is an action of technology development (ADT Inria) with the goal of developing a middleware for the execution of scientific workflows in a distributed and parallel way. It capitalizes on our experience with SON and an innovative algebraic approach to the management of scientific workflows. SciFloware provides a development environment and a runtime environment for scientific workflows, interoperable with existing systems. We validate SciFloware with workflows for analyzing biological data provided by our partners CIRAD, INRA and IRD.

5.9. WebSmatch (Web Schema Matching)

Participants: Emmanuel Castanier, Patrick Valduriez [contact].

URL: <http://websmatch.gforge.inria.fr/>

In the context of an action of technology development (ADT) started in october 2010, WebSmatch is a flexible, open environment for discovering and matching complex schemas from many heterogeneous data sources over the Web. It provides three basic functions: (1) metadata extraction from data sources; (2) schema matching (both 2-way and n-way schema matching), (3) schema clustering to group similar schemas together. WebSmatch is being delivered through Web services, to be used directly by data integrators or other tools, with RIA clients. Implemented in Java, delivered as Open Source Software (under LGPL) and protected by a deposit at APP (Agence de Protection des Programmes). WebSmatch is being used by Datapublica and CIRAD to integrate public data sources.