RESEARCH CENTER
Sophia Antipolis - Méditerranée

FIELD

Activity Report 2013

Section Software

Edition: 2014-03-19

ABS Project-Team

5. Software and Platforms

5.1. Software

This section briefly comments on all the software distributed by ABS . On the one hand, the software released in 2013 is briefly described as the context is presented in the sections dedicated to new results. On the other hand, the software made available before 2013 is briefly specified in terms of applications targeted.

In any case, the website advertising a given software also makes related publications available.

5.1.1. addict: Stoichiometry Determination from Mass Spectrometry Data

Participants: Deepesh Agarwal, Frédéric Cazals, Noël Malod-Dognin.

Context. Given the individual masses of the proteins present in a complex, together with the mass of that complex, *stoichiometry determination* (SD) consists of computing how many copies of each protein are needed to account for the overall mass of the complex. Our work on the stoichiometry determination (SD) problem for noisy data in structural proteomics is described in [17]. The `addict` software suite not only implements our algorithms `DP++` and `DIOPHANTINE` , but also important algorithms to determine the so-called Frobenius number of a vector of protein masses, and also to estimate the number of solutions of a SD problem, from an unbounded knapsack problem.

Distribution. Binaries for the `addict` software suite are made available from http://team.inria.fr/abs/software/addict/.

5.1.2. vorpatch and compatch: Modeling and Comparing Protein Binding Patches

Participants: Frédéric Cazals, Noël Malod-Dognin.

Context. Modeling protein binding patches, i.e. the sets of atoms responsible of an interaction, is a central problem to foster our understanding of the stability and of the specificity of macro-molecular interactions. We developed a binding patch model which encodes morphological properties, allows an atomic-level comparison of binding patches at the geometric and topological levels, and allows estimating binding affinities—with state-of-the-art results on the protein complexes of the binding affinity benchmark. Given a binary protein complex, `vorpatch` identifies the binding patches, and computes a topological encoding of each patch, defined as an *atom shelling tree* generalizing the core-rim model. The program `compatch` allows comparing two patches via the comparison of their atom shelling trees, by favoring either a geometric or a topological comparison.

Distribution. Binaries for `VORPATCH` and `COMPATCH` are available from http://team.inria.fr/abs/software/vorpatch-compatch.

5.1.3. voratom: Modeling Protein Assemblies with Toleranced Models

Participants: Frédéric Cazals, Tom Dreyfus.

Context. Large protein assemblies such as the Nuclear Pore Complex (NPC), chaperonin cavities, the proteasome or ATP synthases, to name a few, are key to numerous biological functions. Modeling such assemblies is especially challenging due to their plasticity (the proteins involved may change along the cell cycle), their size, and also the flexibility of the sub-units. To cope with these difficulties, a reconstruction strategy known as Reconstruction by Data Integration (RDI), aims at integrating diverse experimental data. But the uncertainties on the input data yield equally uncertain reconstructed models, calling for quantitative assessment strategies.

To leverage these reconstruction results, we introduced TOleranced Model (TOM) framework, which inherently accommodates uncertainties on the shape and position of proteins represented as density maps — maps from cryo electron-microscopy or maps stemming from reconstruction by data integration. In a TOM, a fuzzy molecule is sandwiched between two union of concentric balls, the size of the region between these two unions conveying information on the uncertainties.

The corresponding software package, `VORATOM` , includes programs to (i) perform the segmentation of (probability) density maps, (ii) construct toleranced models, (iii) explore toleranced models (geometrically and topologically), (iv) compute Maximal Common Induced Sub-graphs (MCIS) and Maximal Common Edge Sub-graphs (MCES) to assess the pairwise contacts encoded in a TOM.

Distribution. Binaries for the software package `VORATOM` are made available from http://team.inria.fr/abs/software/voratom/.

5.1.4. intervor: Modeling Macro-molecular Interfaces

Participant: Frédéric Cazals.

In collaboration with S. Loriot (The GEOMETRY FACTORY*)*

Context. Modeling the interfaces of macro-molecular complexes is key to improve our understanding of the stability and specificity of such interactions. We proposed a simple parameter-free model for macro-molecular interfaces, which enables a multi-scale investigation —from the atomic scale to the whole interface scale. Our interface model improves the state-of-the-art to (i) identify interface atoms, (ii) define interface patches, (iii) assess the interface curvature, (iv) investigate correlations between the interface geometry and water dynamics / conservation patterns / polarity of residues.

Distribution. The following website http://team.inria.fr/abs/software/intervor serves two purposes: on the one hand, calculations can be run from the website; on the other hand, binaries are made available. To the best of our knowledge, this software is the only publicly available one for analyzing Voronoi interfaces in macro-molecular complexes.

5.1.5. vorlume: Computing Molecular Surfaces and Volumes with Certificates

Participant: Frédéric Cazals.

In collaboration with S. Loriot (The GEOMETRY FACTORY*, France)*

Context. Molecular surfaces and volumes are paramount to molecular modeling, with applications to electrostatic and energy calculations, interface modeling, scoring and model evaluation, pocket and cavity detection, etc. However, for molecular models represented by collections of balls (Van der Waals and solvent accessible models), such calculations are challenging in particular regarding numerics. Because all available programs are overlooking numerical issues, which in particular prevents them from qualifying the accuracy of the results returned, we developed the first certified algorithm, called `vorlume`. This program is based on so-called certified predicates to guarantee the branching operations of the program, as well as interval arithmetic to return an interval certified to contain the exact value of each statistic of interest—in particular the exact surface area and the exact volume of the molecular model processed.

Distribution. Binaries for `Vorlume` is available from http://team.inria.fr/abs/software/vorlume.

5.1.6. ESBTL: the Easy Structural Biology Template Library

Participant: Frédéric Cazals.

In collaboration with S. Loriot (The GEOMETRY FACTORY*, France) and J. Bernauer (Inria AMIB, France).*

Context. The ESBTL (Easy Structural Biology Template Library) is a lightweight C++ library that allows the handling of PDB data and provides a data structure suitable for geometric constructions and analyses, such as those proposed by `INTERVOR`, `VORPATCH` and `COMPATCH`.

Distribution. The C++ source code is available from http://esbtl.sourceforge.net/http://esbtl.sourceforge.net/.

AOSTE Project-Team

5. Software and Platforms

5.1. TimeSquare

Participants: Charles André, 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 four 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...).

In practice TimeSquare is a plug-in developed with Eclipse modeling tools. 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 other such actions are under way.

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: Maxence Guesdon, Yves Sorel [correspondant], Cécile Stentzel, 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, executives 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, Zhen Zhang, Dumitru Potop Butucaru [correspondant].

The Lopht (Logical to Physical Time Compiler) has been designed as an implementation of the AAA methodology. Lopht is similar to SynDEx by relying on off-line allocation and scheduling techniques to allow real-time implementation of dataflow synchronous specifications onto multiprocessor systems. But it has two significant originality points: a stronger focus on efficiency (but without compromising correctness), and a focus on novel target architectures (many-core chips and time-triggered embedded systems).

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 optimisations 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).

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 IRT SystemX/FSF project, in the collaboration with Astrium Space Transportation (Airbus Defence and Space), and in the collaboration with Kalray SA. 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 additionaly 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 commun 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. 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 $\overline{H}^{2^{m\times l}}$, 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 rather high so that the choice of an initial point for the optimization can play a crucial role. 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 and Brussels. 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 provides satisfactory results in the source recovery problem and 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 provides no guarantee to obtain a single realization that satisfies the prescribed constraints. The software Dedale-HF (see Section 5.4), which is the successor of RGC, solves with guarantees 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), a rational approximation engine developed within the team. 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 strong 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 airborn systems and a license agreement has been recently negotiated with TAS-Espagna. 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).

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 users of the filtering community. For a given 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. problem of the user, *i.e.* the one corresponding to user-specified filter characteristics. The reference files are computed off-line using Groebner basis techniques or numerical techniques based on the exploration of a monodromy group. The use of such a continuation technique combined with an efficient implementation of the integrator produces a drastic reduction, by a factor of 20, of 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 of this software has been sold end of 2011, to TAS-Espagna, in order for it to tune filters with topologies having multiple solutions. The use of Dedale-HF is here coupled with that of 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, and 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 [10].

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).

FindSources3D [2] 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 [8] and Section 4.2 , it makes use of the software RARL2 (Section 5.1) for the rational approximation step in plane sections. The data transmission preliminary step ("cortical mapping") is solved using boundary element methods through the software OpenMEEG (its CorticalMapping features) developed by the Athena Team (see http://www-sop.inria.fr/athena/software/OpenMEEG/). A new release of FindSources3D is now available, which is being demonstrated and distributed to the medical team we maintain contact with (hosp. la Timone, Marseille). A further release is currently under development, due to the strong interest for this software by the German firm BESA GmbH (see http://www.besa.de/), involved in EEG software for research and clinical applications, and a deeper collaboration with this company has been started this year. Figure 4 shows the good results of a two sources distribution recovered by FindSources3D from potential values at electrodes on a sphere (scalp) generated by BESA's simulator, and then back to a more realistic head geometry. There, the achieved localization error is small enough, and FindSources3D provides suitable initial guess to heavier dedicated recovery tools, along with an estimation of the number of sources which may be incorporated to the software as an additional functionality (at the moment, the user is still involved in this estimation). Taking into account several time instants will be considered next.

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).

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

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

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.

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. It also 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. Software and Platforms

4.1. SOFA

Participants: Hervé Delingette [correspondant], Brina Goyette, Federico Spadoni, Stéphanie Marchesseau, Hugo Talbot.

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 Shaman, 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: GPL
- License: LGPL
- Type of human computer interaction: console, opengl, qt
- OS/Middelware: 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 [Correspondant], Florian Vichot, Moulay Fadil, Loïc Cadour.

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, 2.1.2, was made in September 2013. See also the web page http://med.inria.fr.

- Version: 2.1.2
- 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/Ubutu), Mac OS X (10.6-10.9)

ATHENA Project-Team

5. Software and Platforms

5.1. OpenMEEG

Participants: Théodore Papadopoulo, Maureen Clerc, 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 [57], [58]. 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 several more indirect downloads. Work is under progress to integrate it in a commercial package (BESA).

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. 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. 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 vizualization) 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.

5.4. 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

5.5. ImplicitFEM

Participants: Théodore Papadopoulo, Sylvain Vallaghé.

ImplicitFEM is a software to simulate the forward EEG/MEG problem. It uses a volumic finite element approach (FEM) that allows the modeling of anisotropic conductivities (which OpenMEEG cannot). Its main originality is to avoid the need of meshes that can be very complicated to build for the head. Instead, it uses directly representations of tissue interfaces as levelsets (that can be provided directly by some segmentation program based on levelsets or can be generated from other representations). It also uses non-differentiable elements so as to properly model continuity of both potential and normal current across the tissues interfaces (which correspond to conductivity discontinuities). This tool is currently used only internally by students and researchers.

- Version: 0.5
- Programming language: C++

5.6. External Stimulator for OpenViBE

Participants: Maureen Clerc, Loïc Mahé, Dieter Devlaminck.

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.

- Version: 1.0
- Keywords: Brain Computer Interfaces
- Multiplatform: Windows - Linux - MacOSX
- Programming language: C++

AXIS Project-Team

5. Software and Platforms

5.1. Introduction

From its creation, AxIS has proposed new methods and software validated experimentally on various applications: Data Mining, Web Usage Mining, Information Retrieval, Activity Modeling. See Sections from 5.3 until 5.6 and our 2013 results.

In the context of the CPER Télius contract (2010-2013), AxIS has proposed to provide a Focus platform (renamed FocusLab) aiming the community of Living Labs or any researcher/actor involving in experimental project with users.

5.2. FocusLab Platform

Participant: Brigitte Trousse [co-correspondent].

Between 2010-2012 in the context of CPER Télius (cf. Section 7.1.1), we bought various hardware (eye-trackers, physiologic sensors, equilibrium platform, tablets, Arduino components, etc.) and software (Sphinx for questionnaires, Story Board for usage scenarios, Interface prototyping tools such as JustInMind, etc.) in order to observe and analyse user behaviours in supporting the design and evaluation of ICT-based services or products within a living lab approach and also to mine data. FocusLab hardware and software (under licences) were used since 2011 by Inria teams and external collaborators with success, supporting experiments or training.
Our goal was also to provide a web-based application for reserving FocusLab material (hardware, software and documentation) and to prepare the access/download of some software issued from Inria research. We started with AxIS software as a first step of the mutualised software part of the platform.
The development process of the web-based FocusLab platform started slowly in 2011, after finding some ways to fund human resources. We started by transforming some AxIS KDD methods into web services. Such a work was pursued this year (cf. Section 6.6) linked to Elliot purposes. This platform (http://focuslab.inria.fr) is based on a Service oriented Architecture.

5.3. Data Mining

5.3.1. Classification and Clustering Methods

Participants: Marc Csernel, Yves Lechevallier [co-correspondent], Brigitte Trousse [co-correspondent].

We developed and maintained a collection of clustering and classification software, written in C++ and /or Java:

Supervised methods

- a Java library (Somlib) that provides efficient implementations of several SOM(Self-Organizing Map) variants [44], [43], [69], [68], [73], especially those that can handle dissimilarity data (available on Inria's Gforge server (public access) Somlib, developed by AxIS Rocquencourt and Brieuc Conan-Guez from Université de Metz.
- a functional Multi-Layer Perceptron library, called FNET, that implements in C++ supervised classification of functional data [64], [67], [66], [65] (developed by AxIS Rocquencourt).

Unsupervised methods : partitioning methods

- Two partitioning clustering methods on the dissimilarity tables issued from a collaboration between AxIS Rocquencourt team and Recife University, Brazil: CDis and CCClust [77]. Both are written in C++ and use the "Symbolic Object Language" (SOL) developed for SODAS. And one partitioning method on interval data (Div).
- Two standalone versions improved from SODAS modules, SCluster and DIVCLUS-T [41] (AxIS Rocquencourt).

Unsupervised methods : agglomerative methods

- a Java implementation of the 2-3 AHC (developed by AxIS Sophia Antipolis). The software is available as a Java applet which runs the hierarchies visualization toolbox called HCT for Hierarchical Clustering Toolbox (see [3] and [42]).

A Web interface developed in C++ and running on our Apache internal Web server .is available for the following methods: SCluster, Div, Cdis, CCClust.
Previous versions of the above software have been integrated in the SODAS 2 Software [61] which was the result of the european project ASSO [6] (2001-2004). SODAS 2 supports the analysis of multidimensional complex data (numerical and non numerical) coming from databases mainly in statistical offices and administration using Symbolic Data Analysis [39]. This software is registrated at APP (Agence de la Protection des Programmes). For the latest version of the SODAS 2 software, see [60], [79].

In 2013, a new release of MND (Dynamic Clustering Method for Mixed Data) algorithm has been done based on [80] (cf. section 6.2.5) and used on clustering the user profiles and analysing user behaviour change (cf. Section 6.5.4).

5.3.2. Extracting Sequential Patterns with Low Support

Participant: Brigitte Trousse [correspondent].

Two methods for extracting sequential patterns with low support have been developed by D. Tanasa in his thesis (see Chapter 3 in [72] for more details) in collaboration with F. Masseglia and B. Trousse :

- **Cluster & Divide**,
- and **Divide & Discover** [8].

These methods have been successfully applied from 2005 on various Web logs.

5.3.3. Mining Data Streams

Participant: Brigitte Trousse [correspondent].

In Marascu's thesis (2009) [57], a collection of software have been developed for knowledge discovery and security in data streams. Three **clustering methods for mining sequential patterns (Java) in data streams** method have been developed in Java:

- SMDS compares the sequences to each others with a complexity of _$O(n^2)$.
- SCDS is an improvement of SMDS, where the complexity is enhanced from $O(n^2)$ to $O(n.m)$ with n the number of navigations and m the number of clusters.
- ICDS is a modification of SCDS. The principle is to keep the clusters' centroids from one batch to another.

Such methods take batches of data in the format "Client-Date-Item" and provide clusters of sequences and their centroids in the form of an approximate sequential pattern calculated with an alignment technique.

In 2010 the Java code of one method called SCDS has been integrated in the MIDAS demonstrator and a C++ version has been implemented by F. Masseglia for the CRE contract with Orange Labs with the deliverability of a licence) with a visualisation module (in Java).

[6] ASSO: Analysis System of Symbolic Official data.

It has been tested on the following data:

- Orange mobile portal logs (100 million records, 3 months) in the context of Midas project (Java version) and the CRE (Orange C++ version)
- Inria Sophia Antipolis Web logs (4 million records, 1 year, Java version)
- Vehicle trajectories (Brinkhoff generator) in the context of MIDAS project (Java version).

In 2012 within the context of the ELLIOT contract, SCDS has been integrated as a Web service (Java version) in the first version of FOCUSLAB platform: a demonstration was made on San Rafaelle Hospital media use case at the first ELLIOT review at Brussels. We applied SCDS web service on data issued from two other use cases in Logistics (BIBA) and Green Services (Inria) [38].

The three C++ codes done for the CRE (Orange Labs) have been deposit at APP. The java code will be deposit in 2014 at APP.

5.4. Web Usage Mining

5.4.1. AWLH for Pre-processing Web Logs

Participants: Yves Lechevallier [co-correspondent], Brigitte Trousse [co-correspondent].

AWLH (AxIS Web Log House) for Web Usage Mining (WUM) is issued from AxISLogMiner software which implements the mult-site log preprocessing methodology and extraction of sequential pattern with low support developed by D. Tanasa in his thesis [72] , [15] for Web Usage Mining (WUM). In the context of the Eiffel project (2008-2009), we isolated and redesigned the core of AxISlogMiner preprocessing tool (we called it AWLH) composed of a set of tools for pre-processing web log files. The web log files are cleaned before to be used by data mining methods, as they contain many noisy entries (for example, robots requests). The data are stored within a database whose model has been improved.

So AWLH offers:

- Processing of several log files from several servers,
- Support of several input formats (CLF, ECLF, IIS, custom, etc.),
- Incremental pre-processing,
- Java API to help integration of AWLH in external application.

5.4.2. ATWUEDA for Analysing Evolving Web Usage Data

Participants: Yves Lechevallier [co-correspondent], Brigitte Trousse [co-correspondent].

ATWUEDA for Web Usage Evolving Data Analysis [52] [4] was developed by A. Da Silva in her thesis [52] under the supervision of Y. Lechevallier. This tool was developed in Java and uses the JRI library in order to allow the application of R which is a programming language and software environment for statistical computing functions in the Java environment.

ATWUEDA is able to read data from a cross table in a MySQL database. It splits the data according to the user specifications (in logical or temporal windows) and then applies the approach proposed in the Da Silva's thesis in order to detect changes in dynamic environment. The proposed approach characterizes the changes undergone by the usage groups (e.g. appearance, disappearance, fusion and split) at each time-stamp. Graphics are generated for each analysed window, exhibiting statistics that characterizes changing points over time.

Version 2. of ATWUEDA (September 2009) is available at Inria's gforge website.

The efficiency of ATWUEDA [46] has been demonstrated by applying it on real case studies such as on condition monitoring data streams of an electric power plant provided by EDF.

ATWUEDA is used by Telecom Paris Tech and EDF [4].

5.5. Information Retrieval

5.5.1. CBR*Tools for Managing and Reusing Past Experiences based on Historical Data

Participant: Brigitte Trousse [correspondent].

CBR*Tools [53], [54] is an object-oriented framework [55], [50] for Case-Based Reasoning which is specified with the UMT notation (Rational Rose) and written in Java. It offers a set of abstract classes to model the main concepts necessary to develop applications integrating case-based reasoning techniques: case, case base, index, measurements of similarity, reasoning control. It also offers a set of concrete classes which implements many traditional methods (closest neighbours indexing, Kd-tree indexing, neuronal approach based indexing, standards similarities measurements). CBR*Tools currently contains more than 240 classes divided in two main categories: the core package for basic functionality and the time package for the specific management of the behavioural situations. The programming of a new application is done by specialization of existing classes, objects aggregation or by using the parameters of the existing classes.

CBR*Tools addresses application fields where the re-use of cases indexed by behavioural situations is required. The CBR*Tools framework was evaluated via the design and the implementation of several applications such as Broadway-Web, Educaid, BeCKB, Broadway-Predict, e-behaviour and Be-TRIP.

CBR*Tools is concerned by two past contracts: EPIA and MobiVIP.

CBR*Tools is available on demand for research, teaching and academic purpose via the FocusLab platform. The user manual can be downloaded at the URL: http://www-sop.inria.fr/axis/cbrtools/manual/.

See also the web page http://www-sop.inria.fr/axis/cbrtools/manual/.

5.5.2. Broadway*Tools for Building Recommender Systems on the Web

Participant: Brigitte Trousse [correspondent].

Broadway*Tools is a toolbox supporting the creation of adaptive recommendation systems on the Web or in a Internet/Intranet information system. The toolbox offers different servers, including a server that computes recommendations based on the observation of the user sessions and on the re-use of user groups' former sessions. A recommender system created with Broadway*tools observes navigations of various users and gather evaluations and annotations, to draw up a list of relevant recommendations (Web documents, keywords, etc).

Based on Jaczynski's thesis [53], different recommender systems have been developed for supporting Web browsing, but also browsing inside a Web-based information system or for query formulation in the context of a meta search engine.

5.6. Activity Modeling

5.6.1. K-MADe for Describing Human Operator or User Activities

Participant: Dominique Scapin [correspondent].

K-MADe tool (Kernel of Model for Human Activity Description Environment). The K-MADe is intended for people wishing to describe, analyze and formalize the activities of human operators, of users, in environments (computerized or not), in real or simulated situation, in the field, or in the laboratory. Although all kinds of profiles of people are possible, this environment is particularly intended for ergonomics and HCI (Human Computer Interaction) specialists. It has been developed through collaboration between ENSMA (LISI XSlaboratory) and Inria.

This year we participated in the AFIHM Working Group on Task Models (http://www.gt-mdt.fr/fr/) "Groupe de Travail de l'AFIHM sur les Modèles de Tà¢ches"). Since the early work on MAD, domain modeling task is the subject of much research (particularly in the French-speaking community), in particular the definition of formalisms and tool construction, three of which are now operational and maintained: K-MADe, eCOMM and hAMSTERS, posing an alternative to CTT. Many teams use these formalisms in a variety of goals and task models occupy a place in the field of Model Driven Engineering, and support the teaching of HCI. The WG goals are to serve as a forum between research approaches, development teams and potential users, especially for non-IT users; fostering collaboration to validate approaches; encourage feedback in teaching task models; provide the French-speaking community and eventually the international community a set of centralized, shared resources about the notion of modeling tasks.

AYIN Team

5. Software and Platforms

5.1. Deposits

- The software SAMD (Semi-Automatic Melanoma Detection) V1.0 was deposited with the APP in December 2013. It has been tested on public databases.
- The software SAAD (Semi-Automatic Acne Detection) V1.0 was deposited with the APP in December 2013. It has been tested on public databases as well as on data sets provided by CHU Nice and Galderma.

BIOCORE Project-Team

5. Software and Platforms

5.1. Supervision software

We are developing a software for the supervision of bioreactors: this platform, named ODIN, has been built for the smart management of bioreactors (data acquisition, fault diagnosis, automatic control algorithm,...). This software was developed in C++ and uses a Scilab engine to run the advanced algorithms developed within BIOCORE. It has been implemented and validated with four different applications.

CASTOR Team

5. Software and Platforms

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.

5.2. PlaTo

Participant: Hervé Guillard [contact].

PlaTo (Platform for Tokamak simulation) is a specialized set of softwares dedicated to the geometry of Tokamaks whose main objective is to provide the researchers of the CASTOR team a common development tools. The platform integrates Fortran90 modules using the MPI communication library for parallel computations and some python and C codes. The PlaTo platform has been developed thanks to a 2010 ADT of Inria and the ANR ESPOIR. The construction of this platform integrates the following developments :

- The set-up of a (small) database corresponding to axisymmetric solutions of the equilibrium plasma equations for realistic geometrical and magnetic configurations (JET, ITER and the Tore-Supra upgrade WEST).The construction of meshes is always an important and time consuming task. PlaTo provides meshes and solutions corresponding to equilibrium solutions that will be used as initial data for more complex computations.
- A set of interfaces (PlaTo ToolBox) allowing easy transfer between different solution and mesh formats.
- Numerical templates allowing the use of 3D discretization schemes using finite element/volume methods. At present, several applications (reduced MHD, Euler equations, two fluid Euler model) are available in PlaTo .

5.3. PaMPA

Participants: Cécile Dobrzynski [Bacchus], Hervé Guillard, Laurent Hascoët [Ecuador], Cédric Lachat, François Pellegrini [Bacchus].

PaMPA ("Parallel Mesh Partitioning and Adaptation") is a middleware library dedicated to the management of distributed meshes. Its purpose is to relieve solver writers from the tedious and error prone task of writing again and again service routines for mesh handling, data communication and exchange, remeshing, and data redistribution. An API of the future platform has been devised, and the coding of the mesh handling and redistribution routines is in progress. PaMPA will be used as a base module for the PLATO solvers, to balance dynamically, refine and coarsen its distributed mesh.

5.4. Cedres++

Participants: Jacques Blum, Cédric Boulbe, Blaise Faugeras, 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 equilibirum equations inside the plasma and the Maxwell equations outside. Moreover, the magnetic field is often supposed not to depend on the azimutal 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 vaccuum 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.

5.5. 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.

COATI Project-Team

5. Software and Platforms

5.1. Grph

Participants: David Coudert, Luc Hogie [correspondant], Aurélien Lancin, Issam Tahiri, Michel Syska.

Around 20,000 lines of code, developed in Java, and licensed under LGPL. See http://grph.inria.fr.

The objective of GRPH is to provide researchers and engineers a suitable graph library for graph algorithms experimentation and network simulation. GRPH is primarily a software library, but it also comes with a set of executable files for user interaction and graph format conversion; as such, it can be used autonomously. Performance and accessibility are the primary targets of the GRPH library. It allows manipulating large graphs (millions of nodes). Its model considers mixed graphs composed of directed and undirected simple- and hyper-edges. GRPH comes with a collection of graph algorithms which is regularly augmented.

GRPH includes bridges to other graph libraries such as JUNG, JGraphT, CORESE (a software developed by the WIMMICS team Inria-I3S), LAD (Christine Solnon, LIRIS), Nauty (Brendan D. McKay), SageMath, as well as specific algorithms developed by Matthieu Latapy and Jean-Loup Guillaume (LIP6), etc.

In 2013, we have added several graph algorithms to GRPH (e.g., subgraph isomorphism, subgraph search as sets or regular expressions, transitive closure, etc.). In particular, a significant effort has been put on the support for paths with multiple data-structures for more efficient in-memory representation of paths, and the implementation of algorithms for the enumeration of paths, the characterization of paths, the computation of the k-shortest paths, etc. Furthermore, we have improved the support of weigths in graphs and developed software bridges to SageMath and OGDF. We have also added several models (link-failures, node mobility) for graph dynamics using the discrete-event simulator included in GRPH, as well as models for the development of decentralized algorithms (useful for instance for the simulation of routing schemes). Finally, we have redesigned the website which now includes a forum gathering the community of users.

5.2. SageMath

Participant: David Coudert.

Sagemath is a free open-source mathematics software aiming at becoming an alternative to Maple and Matlab. Initially created by William Stein (Professor of mathematics at Washington University), Sagemath is currently developed by more than 180 contributors around the world (mostly researchers) and its source code, developed in Python, Cython, and C++, has reached 350 MB.

It is of interest for COATI members because it combines a large collection of graph algorithms with various libraries in algebra, calculus, combinatorics, linear programming, statistics, etc. We use SageMath for quickly testing algorithms, analyzing graphs, and disseminating algorithms. We also use it for teaching purposes in the Master 2 IFI, stream UBINET.

In 2013, David Coudert has contributed to the development of the SageMath releases 5.0 to 6.0 with 10 patches (from bug fix to advanced graph algorithms) and participated to the reviewing process of more than 20 patches that are now part of the standard distribution.

5.3. DRMSim

Participants: David Coudert, Luc Hogie [correspondant], Aurélien Lancin, Nicolas Nisse, Issam Tahiri.

Around 45,000 lines, developed in Java, collaboration between COATI, LaBRI, and Alcatel-Lucent Bell labs.

DRMSim relies on a discrete-event simulation engine aiming at enabling the large-scale simulations of routing models. DRMSim is developped in the framework of the FP7 EULER project. It proposes a general routing model which accommodates any network configuration. Aside to this, it includes specific models for Generalized Linear Preference (GLP), and k-chordal network topologies, as well as implementations of routing protocols, including a previously defined routing protocol and lightweight versions of BGP (Border Gateway Protocol).

The metric model takes measures along a discrete-event simulation which can be performed in many ways.

Commonly, a simulation campaign consists in iterating over the set of combinations of parameter values, calling the simulation function for every combination. These combinations are most often complex, impeding their description by a set of mathematical functions. Thus DRMSim provides a simulation methodology that describes (programmatically) the way a simulation campaign should be conducted.

DRMSim stores on disk every step of the execution of a simulation campaign. In a simulation campaign, simulation runs are independent (no simulation depends on the result computed by another simulation). Consequently they can be executed in parallel. Because one simulation is most likely to use large amount of memory and to be multi-threaded, parallelizing the simulation campaign on one single computer is a poor parallelization scheme. Instead, we currently work at enabling the remote parallel execution of several simulation runs, with the same distribution framework that is used in the GRPH library.

DRMSim relies on the Mascsim abstract discrete-event simulation framework, the GRPH library and the Java4Unix integration framework.

In 2013, the work on DRMSim consisted (1) in the implementation of a full support for dynamic networks, including topological modifications and evolving transfer loads in the simulated network. The implementation of the BGP protocol was updated so as to support these dynamic properties. (2) This implementation of BGP was also augmented with a framework enabling its dynamic profiling. (3) Finally DRMSim does no longer relies on the Dipergrafs library. Instead it now uses GRPH, which brings better performance, stability and a broader set of graph algorithms.

See also the web page http://drmsim.gforge.inria.fr/.

5.4. Utilities

5.4.1. P2PVSim

Participant: Remigiusz Modrzejewski [correspondant].

Around 12,000 lines, developed in Python.

P2PVSim is a discrete-event simulator created for analyzing theoretical properties of peer-to-peer live video streaming algorithms. Implemented in Python it was designed with clarity and extensibility in mind from the beginning. It is capable of simulating overlays of a few thousands of peers. Multiple control protocols have been implemented. At the same time, a lot of work was put into the performance and scalability aspects of the software. Currently it is meant for simulating overlays of a few thousand peers running multiple control protocols that have been implemented. And in 2012, a distributed version of P2PVSim was developed running on an arbitrary number of computers. It has been so far used with success on a dozen computers with multiple cores all located in the same LAN.

5.4.2. Papareto

Participant: Luc Hogie [correspondant].

About 500 lines, developed in Java.

PAPARETO is a Java framework for the development of evolutionary solutions to computational problems. The primary motivation for developing an evolutionary framework was to give the GRPH library the ability to generate particular graph instances. Papareto differs from other evolutionary frameworks (ECJ, WatchMaker, JGAP, etc) in the following ways:

- it is *multi-objective*;
- it is *not a genetic algorithms (GAs)* framework because it manipulates objects *as is*. It does not consider their chromosomes representation. Performance consequently is no longer impacted by the computational cost of encoding/decoding;
- it *parallelizes* the creation and the evaluation of a new generation, adaptively to the evolving load of the computer;
- it is *self-adaptive* in the sense that it dynamically evaluates the performance of the crossover and mutation operators, then gives greater priority to most efficient ones;
- it is *easy to use*, by exposing the cleanest and more natural API possible and the minimal set of functionality enabling researchers and engineers to perform evolutionary computing;

See also the webpage http://www.i3s.unice.fr/~hogie/papareto/.

5.4.3. Toools

Participants: Luc Hogie [correspondant], Aurélien Lancin.

Around 3,000 lines, developed in Java.

TOOOLS is a general purpose Java toolbox which, much like Google Guava and Apache Commons, aims at providing classes useful in daily programming tasks. It focuses on the following topics:

- runtime (threads, control of parallel executions of SIMD code, execution of external processes, management of I/O operations, piping);
- input/output files (a complete, easier to use and more complete new model for files on disk is provided) and streams;
- reflection, including dynamic loading of classes, classpath management, Java beans, and access to the source code at runtime;
- application configuration files (parsing, querying, saving);
- plain text, XML;
- collections, including Java collection utilities and efficient sets of primitive integers;
- mathematical and statistical operations.

See also the webpage http://www-sop.inria.fr/members/Luc.Hogie/toools/.

5.4.4. Other software

We ensure the maintenance of various tools developped in the past:

Java4unix a software glue for the integration of Java applications intro the UNIX environment; http://www-sop.inria.fr/members/Luc.Hogie/java4unix/;

Jalinopt a Java toolkit for linear optimization; http://www-sop.inria.fr/members/Luc.Hogie/jalinopt/;

JavaFarm a minimal middleware infrastructure for practical distributed computing; see http://www-sop.inria.fr/members/Luc.Hogie/javafarm/;

Macsim a discrete event simulation engine use in the DRMSIM routing model simulator; http://www-sop.inria.fr/mascotte/software/mascsim/;

Jaseto a Java toolkit for the XML (de)serialization of Java objects; http://www-sop.inria.fr/members/Luc.Hogie/jaseto/;

COFFEE Project-Team

5. 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.

COPRIN Project-Team

5. Software and Platforms

5.1. Introduction

Software development is an essential part of the research done by COPRIN 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`, `Scilab`) 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: David Daney, 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.

DEMAR Project-Team

5. Software and Platforms

5.1. Software and Platforms

5.1.1. RdP to VHDL tool

Participants: Gregory Angles, David Andreu, Thierry Gil, Robin Passama.

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 its 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 unit's 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 inter-connected as well as signals it exports or imports. The interconnection of those components, from a behavioral point out 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).

GALS (Globally Asynchronous Locally Synchronous) systems can be specified, connecting different clocks to HILECOP components, and interconnecting them by means of asynchronous signals.

Undergoing work includes the modification of the formalism in order to allow behavior agregation as well as exception handling, both for analysis and implementation sides.

The Eclipse-based version of HILECOP is regulary updated. The last version of HILECOP (registered at the french Agence de Protection des Programmes (APP)) is accessible to the academic community (http://www.lirmm.fr/~gil/Temp/).

5.1.2. 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 has been transferred to the industrial partner and a new version is under development according to an Eclipse-based design. This new version should be available by the end of 2014.

DIANA Team

5. Software and Platforms

5.1. FIT platform

We have started, since 2011, the procedure of building a new experimental platform at Sophia-Antipolis, in the context of the FIT Equipment of Excellence project. This platform has two main goals : the first one is to enable highly controllable experiments due to its anechoic environment. These experiments can be either hybrid-experiments (as NEPI will be deployed, see section 5.4) or federated experiments through several testbeds. The second goal is to make resource consuming experiments (like CCNx) possible due to some powerful servers that will be installed and connected to the PlanetLab testbed. During 2013, a first call for bids has been made during March/April and has been unfortunately declared unsuccessful due to an overestimation of the building's price. As some premises became vacant at the same time, a second call for bids has been launched during September/October. This latter was a success because three interesting offers have been received and negotiated during the end of the year. The notification is planned for the 14th of January 2014.

5.2. ns-3

Participants: Thierry Turletti [correspondant], Daniel Camara, Walid Dabbous.

ns-3 is a discrete-event network simulator for Internet systems, targeted primarily for research and educational use. ns-3 is free software, licensed under the GNU GPLv2 license, and is publicly available for research, development, and 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 solid 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).
See also the web page http://www.nsnam.org.

- Version: ns-3.19
- 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

5.3. DCE

Participants: Emilio Mancini [correspondant], Daniel Camara, Walid Dabbous, Thierry Turletti.

Direct Code Execution (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, thttpd, CCNx and various Linux kernel versions (from 2.6.36 to 3.12 versions).

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

5.4. NEPI

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

NEPI stands for Network Experimentation Programming Interface. NEPI is a generic framework to manage network experiments, which allows to describe experiments in a simple way and automates experiment execution and result collection in a variety of experimentation environments, including simulators and live testbeds. NEPI was designed to be extensible for potentially any experimentation environment, which can be done by extending its well defined description and execution API's.

During the year 2013 we fully re-implemented NEPI in order to support new important requirements that arose from our participation in the FED4fire and OpenLab European projects. Among those requirements we can enumerate the ability to dynamically provision experiment resources during the experiment run time (e.i. Add new resources to the experiment at any moment), the interactive experiment execution (i.e. NEPI can be used as an interactive experiment management tool), and the description of experiment work-flows (e.i. Allow to include execution/deployment dependencies across resources).

This re-implementation of NEPI gave place to an improved, more extensible and user friendly framework which was officially released as NEPI 3.0 in December 2013. This new version of NEPI was shipped with support for new testbeds, including any testbeds supporting SSH key authentication and OMF (cOntrol and Management Framework) testbeds. Support for OMF technology is a very important features since it is a key part of the federation control management framework proposed in the FED4Fire project. The version of OMF currently supported by NEPI is 5.4. OMF version 6.0, which is the new mainstream release, will be supported during 2014. In order to comply with the requirements of FED4Fire for a federation framework, work is undergo in NEPI to fully support SFA (Slice Federation Architecture), in its latest version, for resource discovery and provisioning across federated testbeds.

Additional improvements to the NEPI framework during 2013 include out-of-the box support for Future Internet technologies such as CCN and OpenFlow for certain testbeds.

Finally, during 2013 a new NEPI web site was released, including an improved look&feel (http://nepi.inria.fr), user manual and code reference pages, detailed experiment examples and a issue tracking page.

- Version: 3.0
- ACM: C.2.2, C.2.4
- Keywords: networking experimentation
- License: GPL (3)
- Type of human computer interaction: python library, QT GUI
- OS/Middelware: Linux
- Required library or software: python – http://www.python.org – http://rpyc.sourceforge.net
- Programming language: python

5.5. Bake

Participants: Daniel Camara [correspondant], Walid Dabbous, Thierry Turletti.

Bake is an integration tool, which can be used by software developers to automate the reproducible build of a number of projects which depend on each other and which might be developed, and hosted by unrelated parties. Bake was developed to automate the reproducible build of ns-3 taking into account the particular needs of it. However, Bake is not specific for ns-3, it can be used by any open source project composed of a number of interdependent sub projects and that needs to simplify and automatize the assembly of these pieces of software in a coherent and useful way.

Bake is free software, licensed under the GNU GPLv2 license, and is publicly available for research, development, and use.

See also the web page http://planete.inria.fr/software/bake/index.html

- Version: Bake-0.1
- Keywords: Integration tool, distributed project, build and installation version control
- License: GPL (GPLv2)
- Type of human computer interaction: command line tool, No GUI
- OS/Middleware: Linux, Mac Os
- Required library or software: Python, GNU C++, Mercurial, CVS, GIT, Bazaar, Tar, Unzip, Unrar, 7z, XZ, Make, cMake, patch, autoreconf
- Programming language: python
- Documentation: doxygen

5.6. Com4Innov Network Testing Platform

Participants: Emilio Mancini [correspondant], Walid Dabbous.

Developed for the Com4Innov project, the platform integrates a set of tools in a virtual appliance, in order to conduct network experiments, store and share their results, and collect network diagnostic information. The platform's core is developed as a Web Application in the Apache Tomcat application server. It has a web user interface, and a public API accessible to external tools. Its architecture is designed to be integrated with SQL or NoSQL databases, and with HTTP or REST client. The core platform is completed by measurement tools and mobile phone clients.

The implementation is available at the url: http://gforge.inria.fr/projects/com4innov. It is currently in advanced development stage.

- Version: 0.5
- Keywords: network, diagnostic
- License: not yet public
- Type of human computer interaction: Web Interface
- OS/Middelware: Linux
- Required library or software: Apache Tomcat Java, Linux Distribution and standard development environment, GPL.
- Programming languages: Java, C++, C#
- Documentation: latex, javadoc

5.7. Mobile Devicess Network Analyzer Tool

Participants: Emilio Mancini [correspondant], Arnaud Legout.

This tool has been developed as a client for the Com4Innov Network Testing Platform, but it evolved in an autonomous one. It samples the signal's strength of wireless networks, correlating it with the GPS position, and then produces coverage maps. Once the sampling is done, it allows the user to upload such data, completed with diagnostic information to the platform.

The implementation is available at the url: http://gforge.inria.fr/projects/com4innov.

- Version: 1.0
- Keywords: network, diagnostic
- License: not yet public
- Type of human computer interaction: Android Application
- OS/Middelware: Android
- Required library or software: Android, GPL.
- Programming languages: Java
- Documentation: latex, javadoc

5.8. SfaWrap

Participants: Thierry Parmentelat [correspondant], Mohamed Larabi.

The SfaWrap is a reference implementation of the Slice-based Federation Architecture (SFA), the emerging standard for networking experimental testbed federation. We are codeveloping the SfaWrap with Princeton University, and during 2013, we have focused on:

- Implementing the Aggregate Manager (AM) API v3.
- Implementing a compatibility layer between AM API v2 and AM API v3.
- Supporting testbed providers in exposing their testbeds through SFA in order to enlarge the federation of testbeds.
- Maintaning the codebase of SFAWrap.
- Releasing SFAWrap software packages for the latest versions of different Linux Distributions (Namely: Fedora, Debian and Ubuntu).
- Version: sfa-3.1-1, myplc-5.3.1
- Keywords: networking testbed federation
- License: Various Open Source Licenses
- Type of human computer interaction: Web-UI, XMLRPC-based API, Qt-based graphical client
- OS/Middelware: Linux
- Required library or software: python2.7 or superior
- Programming languages: python
- Documentation: http://svn.planet-lab.org/#SFAUser-leveldocumentation
- Codebase: http://git.onelab.eu/?p=sfa.git;a=summary

5.9. Experimentation Software

MonLab Monitoring Lab is a platform for the emulation and monitoring of traffic in virtual ISP networks. It was supported by the FP7 ECODE project and is available for download at the web page of the tool http://planete.inria.fr/MonLab/ under the terms of the GPL licence. MonLab presents a new approach for the emulation of Internet traffic and for its monitoring across the different routers of the emulated ISP network. In its current version, the traffic is sampled at the packet level in each router of the platform, then monitored at the flow level. We put at the disposal of users real traffic emulation facilities coupled to a set of libraries and tools capable of Cisco NetFlow data export, collection and analysis. Our aim is to enable running and evaluating advanced applications for network wide traffic monitoring and optimization. The development of such applications is out of the scope of this research. We believe that the framework we are proposing can play a significant role in the systematic evaluation and experimentation of these applications' algorithms. Among the direct candidates figure algorithms for traffic engineering and distributed anomaly detection. Furthermore, methods for placing monitors, sampling traffic, coordinating monitors, and inverting sampling traffic will find in our platform a valuable tool for experimentation.

ACQUA ACQUA is an Application for Collaborative Estimation of QUality of Internet Access. It was supported by the French ANR CMON project on collaborative monitoring. ACQUA is based on the principle of active measurements to a predefined set of landmarks and on the estimation of the Internet access quality by correlating these measurements together. This correlation will point to the importance of observed problems and to estimates to the quality the end user should expect from its access when running his applications over the Internet (one can see the measurements to the landmarks as samples of the global set of possible paths). In its first version (the version available online), ACQUA was concentring on delay measurements at the access and on the detection and estimation of the impact of delay anomalies (local problems, far away problems, etc). The current work is concentring on using the ACQUA principle in the estimation and prediction of the quality of experience of main applications. More details and code can be found at http://planete.inria.fr/acqua/.

ElectroSmart We are developing the application ElectroSmart as part of the Inria ADT ElectroSmart. The ElectroSmart application will enable crowd sourcing of electromagnetic exposures based on the electromagnetic radiations measured by smartphones.

FOCUS Project-Team

5. 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.

In 2013 the development of the Jolie language has mainly focused on tools for the programming environment and on the integration of the language with cloud infrastructure. More in detail, we have produced the following software.

- PaaSSOA. This is a prototype for the deployment of Jolie services in a cloud infrastructure. We have also worked on the integration of Jolie with the Drools engine. Drools is used in PaaSSOA for storing and managing monitor events from services.
- JEye. This is web GUI prototype for designing Jolie graphical workflows which can be deployed into PaaSSOA.
- WSOA. We have developed a SaaS (Software as a Service) layer for the publication of Jolie APIs using different formats (http, JSON, SOAP, XML, and so on). We have also enhanced some libraries for the integration between Jolie and GWT technology.

All the server side code of PaaSSOA, JEye and WSOA has been developed by using Jolie.

5.2. Others

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

- *Deadlock analysis* 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).

  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 behavioral descriptions of methods, called contracts, which retain resource dependency information. Then this algorithm is integrated with a number of possible different back-ends that analyze contracts and derive deadlock information. We have prototyped two such back-ends:

  1. an evaluator that computes a fixpoint semantics, and
  2. an evaluator using abstract model checking.

  The evaluator (1) is available at http://www.cs.unibo.it/~laneve/deadlock/index.html

The evaluator (2) is available at http://www.cs.unibo.it/~giachino/siteDat/index.php

- *CaReDeb* (http://proton.inrialpes.fr/~mezzina/deb/).

  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 misbehavior, 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.

- *AIOCJ* (http://www.cs.unibo.it/projects/jolie/aiocj.html).

  AIOCJ is a framework for programming adaptive distributed systems based on message passing. AIOCJ comes as a plugin for Eclipse, AIOCJ-ecl, allowing to edit descriptions of distributed systems as adaptive interaction-oriented choreographies (AIOC). 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 AIOC 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 AIOC under all changing sets of adaptation rules and environment conditions. In particular, the system is always deadlock-free.

- *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. Experimental results are encouraging as METIS looks quite effective in practice by handling problem instances with hundreds of components in less than a minute. This is a key ingredient in the solution to the automation problem addressed by the Aeolus project. The paper [48] is dedicated to the description of the tool, while [41] addresses the formal aspects of the technique.

The sofware below have not undergone substantial modifications during 2013

- *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.

  We used the interpreter to test the expressive power of croll-pi on various problems, including the 8-queen problem, error handling in an automotive scenario from the EU project Sensoria, and constructs for distributed error handling such as stabilizers.

- *IntML* is a functional programming language guaranteeing sublinear space bounds for all programs [50]. 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.

GALAAD Project-Team

5. 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 the embeding 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 Ducoffe [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 Continous Integration platform `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. Software and Platforms

5.1. CGAL, the Computational Geometry Algorithms Library

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

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

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

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

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

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

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

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

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

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

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

GRAPHIK Project-Team

5. 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, the following features have been developed:

- we have introduced the concept of scripted rule which associates more fluidly the editable graphical objects with scripts that perform operations on knowledge graphs. These features have been tested and improved in various projects this year (see *e.g.* Qualinca in Section 8.1 or CTFC in Section 7.2).
- we have implemented an interface for quick and assisted creation of graphs. It is based upon Datalog+/- language and provides tools for automatic completion.
- finally, default conceptual graphs rules were implemented in Cogui. An editing interface is available as well as the operation to find all extensions of a rule set. This feature is required by the CTFC project (see Section 7.2).

5.2. Cogui/Capex

Participants: Alain Gutierrez, Patrice Buche, Awa Diattara, Jérôme Fortin.

Cogui/Capex is a platform for expert knowledge management. It has been developed in order to propose a simple and useful interface to applicative domain experts. This will allow us to validate the integration of our theoretical tools into a real-world application and strengthen GraphIK's involvement in agronomy applications (see the projects with CTFC in Section 7.2 and Panzani in Section 7.3).

5.3. Alaska

Participants: Bruno Paiva Lima Da Silva, Jean-François Baget, Madalina Croitoru.

Alaska (http://alaska.bplsilva.com/) is a java library dedicated to the storage and querying of large knowledge bases. It intends to be the foundation layer of our OBDA (Ontology Based Data Access) software developments. It has been built, first as part of a master thesis, and now of Bruno Paiva Lima da Silva' PhD (that will be defended in Jan. 2014).

In Alaska, facts and queries are defined via a generic interface that favors a logical view of these objects. Implementations of this interface allow for the storage of facts w.r.t. different storage paradigms and systems (e.g., relational databases *MySQL* and *Sqlite*; triple stores *Sesame* and graph databases *Neo4J*, *DEX*, *HyperGraphDB* and *OrientDB*). For the time being, we can store 10^7 to 10^8 atoms. In the same way, logical queries can be evaluated through different methods, be it the native querying mechanism of the considered database (*e.g.SPARQL* or *SQL*), or specifically designed algorithms (from a simple backtrack to a full constraint solver based upon *Choco*http://www.emn.fr/z-info/choco-solver/ for hard problem instances). Note that all these methods provide the same answer set to queries.

This library already allows for testing our OBDA algorithms on large instances. The ADT Quasar (that will start in March 2014) will involve the integration of Alaska with other tools developped in the team (see also Section 5.4), and its improvement from a research library to a distributable tool.

5.4. Tools for Rule-Based Reasoning

Participants: Mélanie König, Michel Leclère, Marie-Laure Mugnier, Swan Rocher, Michaël Thomazo.

Kiabora has been designed to analyze an existential rule base (see Section 6.1) and determine if it allows for finite query answering, i.e., if any conjunctive query evaluated over any fact base while taking this set of rules into account will be answered in a finite time. This year, we fixed some bugs and added some specific options. In addition, a presentation and a demo of Kiabora were made at RR 2013 [38].

Besides, the algorithms presented in [38], [37], [41] were implemented and let to experiments. These algorithms are still under development since new improvements have to be integrated.

INDES Project-Team

5. 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 other software packages can be downloaded from the INDES Web site:

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

5.2. Functional programming

Participants: Cyprien Nicolas, Bernard Serpette, Manuel Serrano [correspondant].

5.2.1. 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/indes/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.

5.3. Language-based Security

Participants: Tamara Rezk [correspondant], José Santos.

5.3.1. 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/indes/ifJS.

5.4. Web programming

Participants: Gérard Berry, Cyprien Nicolas, Manuel Serrano [correspondant].

5.4.1. 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.

5.5. Old software

5.5.1. 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.

5.5.2. 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.

5.5.3. 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.

5.5.4. 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.

5.5.5. 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.

5.5.6. 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.

5.5.7. 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/indes/mashic/.

LAGADIC Project-Team

5. 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 licence, ViSP, an open source library available from http://team.inria.fr/lagadic/visp/visp.html. 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 allows tracking of visual cues (dots, segments, ellipses,...) or 3D model-based tracking of known objects. Simulation capabilities are also available. ViSP and its full functionalities are presented in Fig. 1 and described in [6].

This year, we continued our efforts to improve the software by increasing the compatibility with exotic platforms, fixing issues, and by introducing an hybrid scheme in the model-based tracker to take advantage of texture. We also improved the documentation by providing tutorials covering the main capabilities of the software. Two releases were produced, one in February downloaded 1000 times and the other in July downloaded 730 times. With the help of the community, the last release was also packaged for Ubuntu 13.10. A new template tracker developed during A. Dame's Ph.D. was recently introduced and will be available in the next release.

Concerning ROS community, all the existing packages in "`vision_visp`" ROS stack (see http://www.ros.org/wiki/vision_visp) were updated and ported to catkin 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

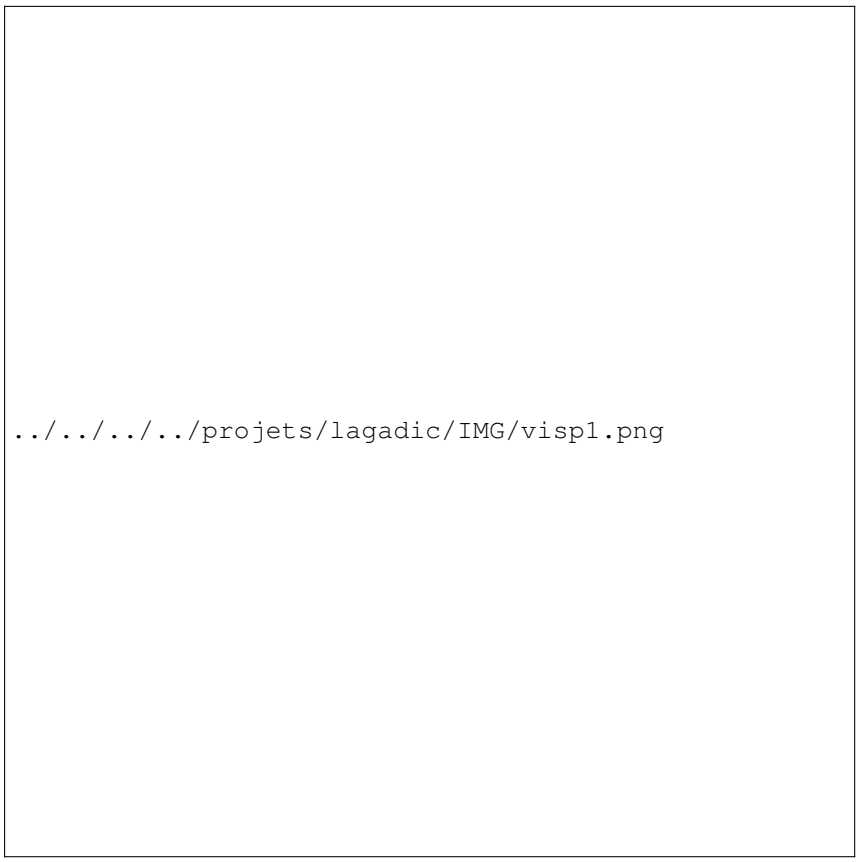
Participant: Patrick Rives [correspondant].

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 indoors or outdoors environments and it allows to localize in real time (45Hz) a robot or a person carrying out a mobile camera.

5.3. 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.


../../../../projets/lagadic/IMG/visp1.png

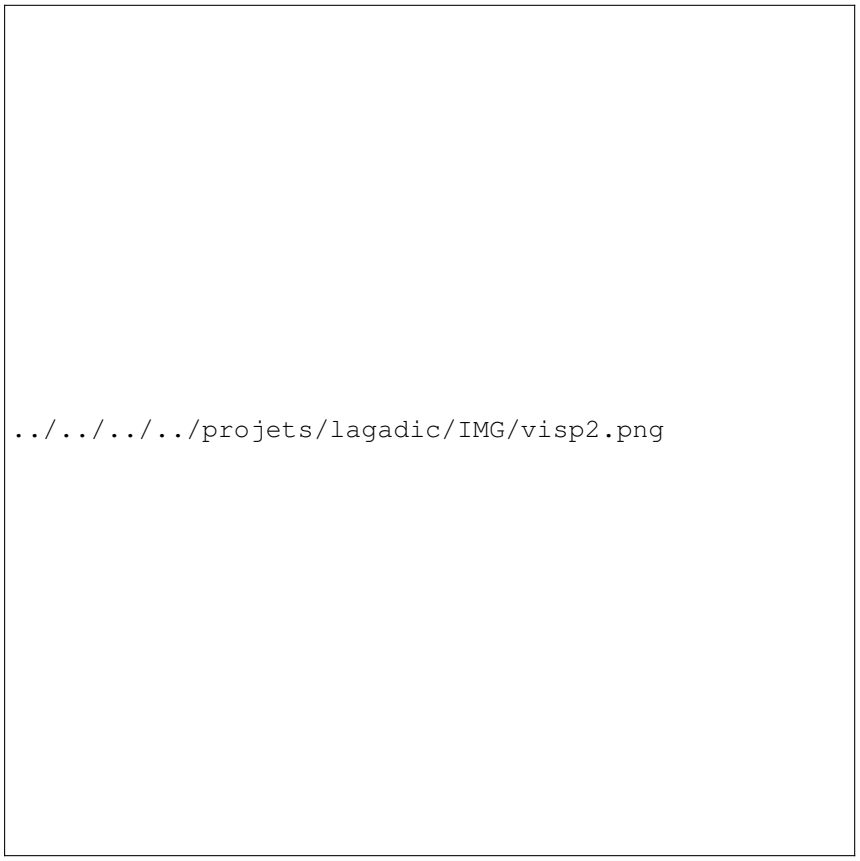

../../../../projets/lagadic/IMG/visp2.png

Figure 2. Lagadic robotics platforms for vision-based manipulation

Three papers published by Lagadic in 2013 enclose results validated on this platform.

5.4. 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 positioning or tracking tasks described in Section 6.4 .

This platform is composed by two Adept Viper six degrees of freedom arms (see Fig. 3). 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 started experimentation to validate needle detection and tracking under ultrasound imaging (see Section 6.4.1).

This year, two papers enclose experimental results obtained with this platform.

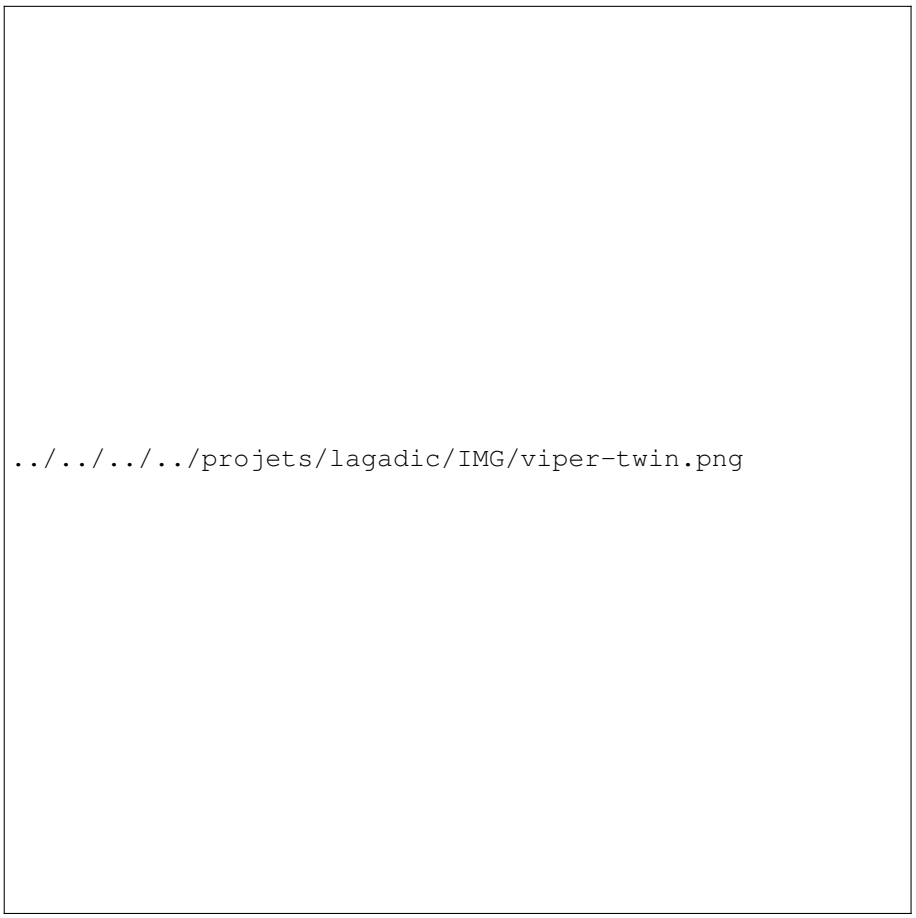


Figure 3. Lagadic medical robotics platforms. On the right Viper S850 robot arm equipped with a SonixTouch 3D ultrasound probe. On the left Viper S650 equipped with a tool changer that allows to attach a classical camera.

5.5. Mobile robotics platforms

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

5.5.1. Indoors 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. 4 .a), and a Pioneer 3DX from Adept in Rennes (see Fig. 4 .b) as well as a Robotino from Festo. 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 6.3.4), we bought in Rennes a six wheel electric wheelchair from Penny and Giles Drives Technology (see Fig. 4 .c). 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 let us acquire the user intention through the joystick position and control the wheelchair by applying corrections to its motion. The wheelchair has been fitted with cameras to perform the required servoing for assisting handicapped people. Moreover, to ensure the direct security of the user, seven infrared proximity sensors have been installed all around the wheelchair.

Note that three papers exploiting the indoors mobile robots were published this year.

5.5.2. Outdoors mobile robots

The team exploits also Cycab urban electrical cars (see Figs. 4 .d and 4 .e). 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.

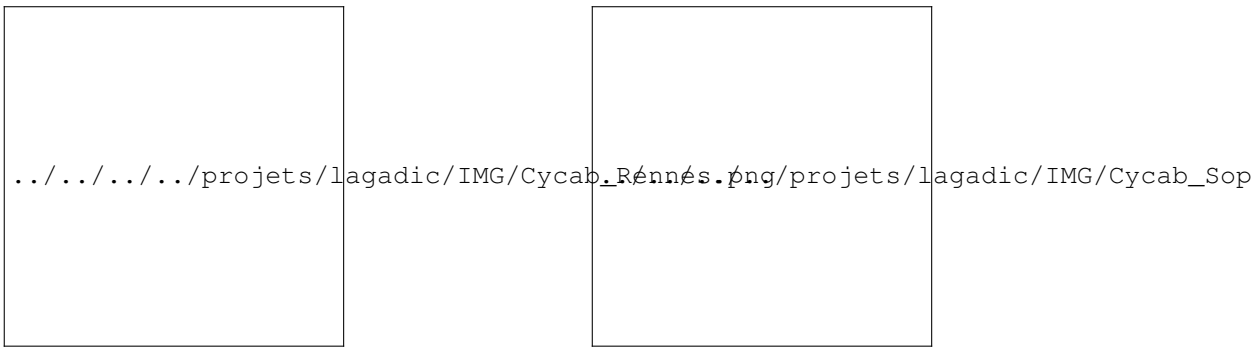
Three papers published by Lagadic in 2013 enclose experimental results obtained with these outdoors mobile robots.

5.5.3. 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 indoors 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:

- adapt a navigation module developed by E-Motion to the mobile platform used at Sophia-Antipolis;
- 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.

(a) (b) (c)



(d) (e)

Figure 4. a) Hannibal platform, b) Pioneer P3-DX robot, c) six wheel electric wheelchair, d) Cycab available in Rennes, e) one of the Cycabs available in Sophia Antipolis.

LOGNET Team

5. Software and Platforms

5.1. myMed

Our flagship software is called myMed. myMed is a highly innovative project in which three main orthogonal components are brought together:

- a software development kit, SDKmyMed, with which we can build social networks in “rush time”;
- a novel distributed hosting cloud, CLOUDmyMed, with which the social applications (developed by us and by third parties) can be hosted and run;
- a pull of 5-10 social network applications, aka “sociapps” developed in our team to test the SDKmyMed.

The sociapp can be enjoyed in almost all platforms, from web browsers, to mobile web, until IOS and Android devices.

5.2. myMed backbone

Participants: Luigi Liquori [contact], The Mymed Engineer Team.

Figure 4. The myMed backbone and the myMed LaunchPad

We have implemented a "backbone" for the myMed social network using a nosql database called Cassandra http://cassandra.apache.org, the latter used also by social networks like Facebook and Twitter. The backbone relies on 50 PC quad code HP400, equipped with 2Tb of hard drive each.

5.3. myMed frontend

Participants: Luigi Liquori [contact], The Mymed Engineer Team.

We have implemented a front-end with which all the social application can be used and downloaded via a "store" mechanism similar to the ones of Apple and Google stores. Social applications can be chosen, voted for via a reputation system, and uninstalled (including all personal data) if the user wants. We have also implemented a "template" allowing to build "proofs-of-concept" of social networks in a very short time.

5.4. Synapse simulator in Oversim

Participants: Vincenzo Ciancaglini [contact], Luigi Liquori.

Synapse-Oversim is an implementation of the Synapse overlay interconnection protocol in the Oversim overlay simulator. The software presents two main contributions: first of all, a fork of the original Oversim simulator has been implemented in order to support running multiple protocol modules in a single instance of Oversim, a necessary feature in order to simulate a set of heterogeneous interconnected networks. Secondly, the whole Synapse protocol has been implemented on top of Oversim, in order to allow for the efficient inter-routing of messages between heterogeneous overlays. The Synapse code has been developed in C++, by running in Oversim, its correctness and its performances can be evaluated, while then the code can be easily ported to a real-world application.

5.5. Synapse model Erlang validator

Participant: Vincenzo Ciancaglini [contact].

During the work on the Synapse protocol, we devised a mathematical model which would allow us to estimate performance indexes of an interconnected system without having to deploy a full-scale experiment. In order to be validated, however, the model results needed to be verified against some simulation results, run under simplified conditions, but with the highest possible number of nodes. To achieve this, a dedicated simulator has been developed using Erlang, a programming language dedicated to parallel and distributed applications, which allow for the simulation of extreme systems, with a number of nodes beyond one million, in the fastest way achievable, by fully exploiting the multicore architecture of modern machines. The simulator instantiates a lightweight thread for each node, and the communication are rendered by message passing between the different node threads, thus keeping the simulation conditions as close as possible to a real world behavior.

5.6. CCN-TV Omnet++ simulator

Participants: Vincenzo Ciancaglini [contact], Riccardo Loti, Luigi Liquori.

CCN-TV-SIM is a software, based on the network simulation framework Omnet++, which simulates a real time video broadcast system over content-centric networks. The system is able to manage multiple streams of video at different rates, using real video traces, simulate different caching policies, different channels being transmitted concurrently, background network traffic, and different channel switch rates. Furthermore it can exploits network topologies taken from real networks, like the Deutsche Telecom network, or the Geant.

5.7. Java implementation of the OGP protocol and the experiment controller

Participants: Giang Ngo Hoang [contact], Luigi Liquori.

OGP-Experiment contains Java implementation of the OGP protocol (OGP stands for overlay gateway protocol) which is used for inter-routing between heterogeneous overlay networks, and a Java implementation of the experiment controller, which is responsible for scheduling, managing and monitoring the statistics of the experiments. The software supports experiments in churn and no-churn environments. Performance metrics of the OGP protocol, such as the latency, the successful rate of data lookup and the traffic generated by a peer are reported. The experiments are performed on the Grid 5000 platform. Heterogeneous overlays which are connected by OGP can be easily plugged into the software.

5.8. Java implementation of the Synapse protocol and the experiment controller

Participants: Hoang Giang Ngo [contact], Luigi Liquori.

Synapse-Experiment contains Java implementation of the Synapse overlay interconnection protocol and Java implementation of the experiment controller which is responsible for scheduling, managing and monitoring the statistics of the experiments. The software supports experiments in churn and no-churn environments. Performance metrics of the Synapse protocol, such as the latency, the successful rate of data looking up and the traffic generated by a peer are reported. The experiments are performed on the Grid 5000 platform.

5.9. Reputation Computation Engine for Social Web Platforms

Participant: Thao Nguyen [contact].

Among the three components of a Trust and Reputation System, information gathering is most dependent on the application system, followed by the decision support component and then by the building of a robust Reputation Computation Engine and an experimental GUI, showing how bad users are segregated by the engine. To simulate the working of the reputation engine, we set up a population of Nu users, providing the same service, and undertaking Nt transactions. In each transaction, a random consumer is assigned to request the service. Other users will then be candidate providers for this request. When a user plays the role of a consumer, his behavior is modeled in the raterType attribute. Three types of raters include HONEST, DISHONEST and COLLUSIVE. HONEST raters share their personal experience honestly, i.e. Rr = Ep. DISHONEST raters provide ratings 0:5 different from their true estimation, i.e. Rr = Ep +- 0:5. COLLUSIVE raters give the highest ratings (Rr = 1) to users in their collusion and the lowest ratings (Rr = 0) to the rest. Similarly, when a user acts as a provider, he can be one of the following types of providers: GOOD, NORMAL, BAD, or GOODTURNBAD. This type is denoted in providerType attribute. The QoS of the service provided by a BAD, NORMAL, or GOOD provider has a value in the interval (0; 0:4], (0:4; 0:7], or (0:7; 1] respectively. A GOODTURNBAD provider will change the QoS of his service when 50% of Nt transactions have been done in the simulation. To get a transaction done, a consumer obtains a list of providers, computes reputation scores for them, chooses a provider to perform the transaction, updates his private information, and publishes his rating for the provider. The quality of service that the consumer will experience depends on the providerType of the chosen provider. The difference between the consumer's rating for the provider and his observation depends on the consumer's raterType.

To run a simulation, the user must specify 10 parameters as described above: Simulation(Nu, Nt, %G, %N, %B, %GTB, %H, %D, %C, %dataLost). The simulator has been published in [22].

In 2013, the simulator has been improved and made more robust: it would one of the output of the Ph.D. work of Thao Nguyen whose defense is envisaged in the first half of 2014.

5.10. Ariwheels

Participants: Luigi Liquori [contact for the *Ariwheels* simulator], Claudio Casetti [Politecnico di Torino, Italy], Diego Borsetti [Politecnico di Torino, Italy], Carla-Fabiana Chiasserini [Politecnico di Torino, Italy], Diego Malandrino [Politecnico di Torino, Italy, contact for the *Ariwheels* client].

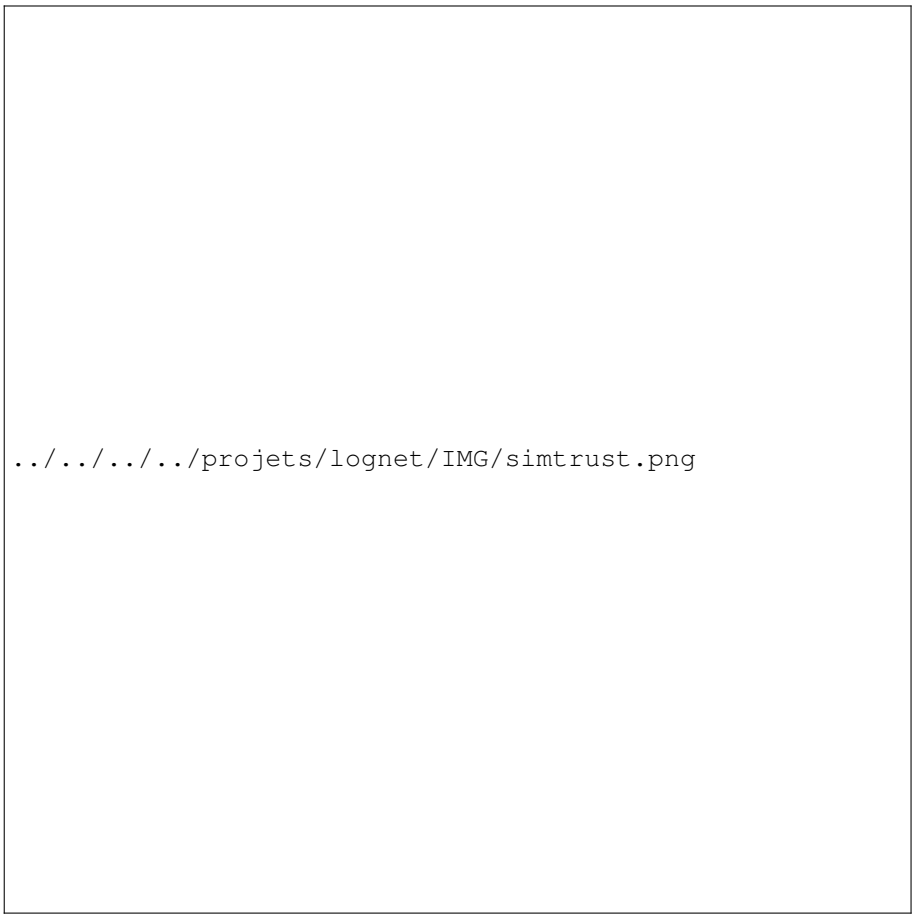


Figure 5. A prototype graphical GUI for the Reputation Computation Engine

Ariwheels is an info-mobility solution for urban environments, with access points deployed at both bus stops (forming thus a wired backbone) and inside the buses themselves. Such a network is meant to provide connectivity and services to the users of the public transport system, allowing them to exchange services, resources and information through their mobile devices. *Ariwheels* is both:

- a protocol, based on *Arigatoni* and the publish/subscribe paradigm;
- a set of applications, implementing the protocol on the different types of nodes;
- a simulator, written in OMNET++ and recently ported to the ns2 simulator, see Fig 6 .

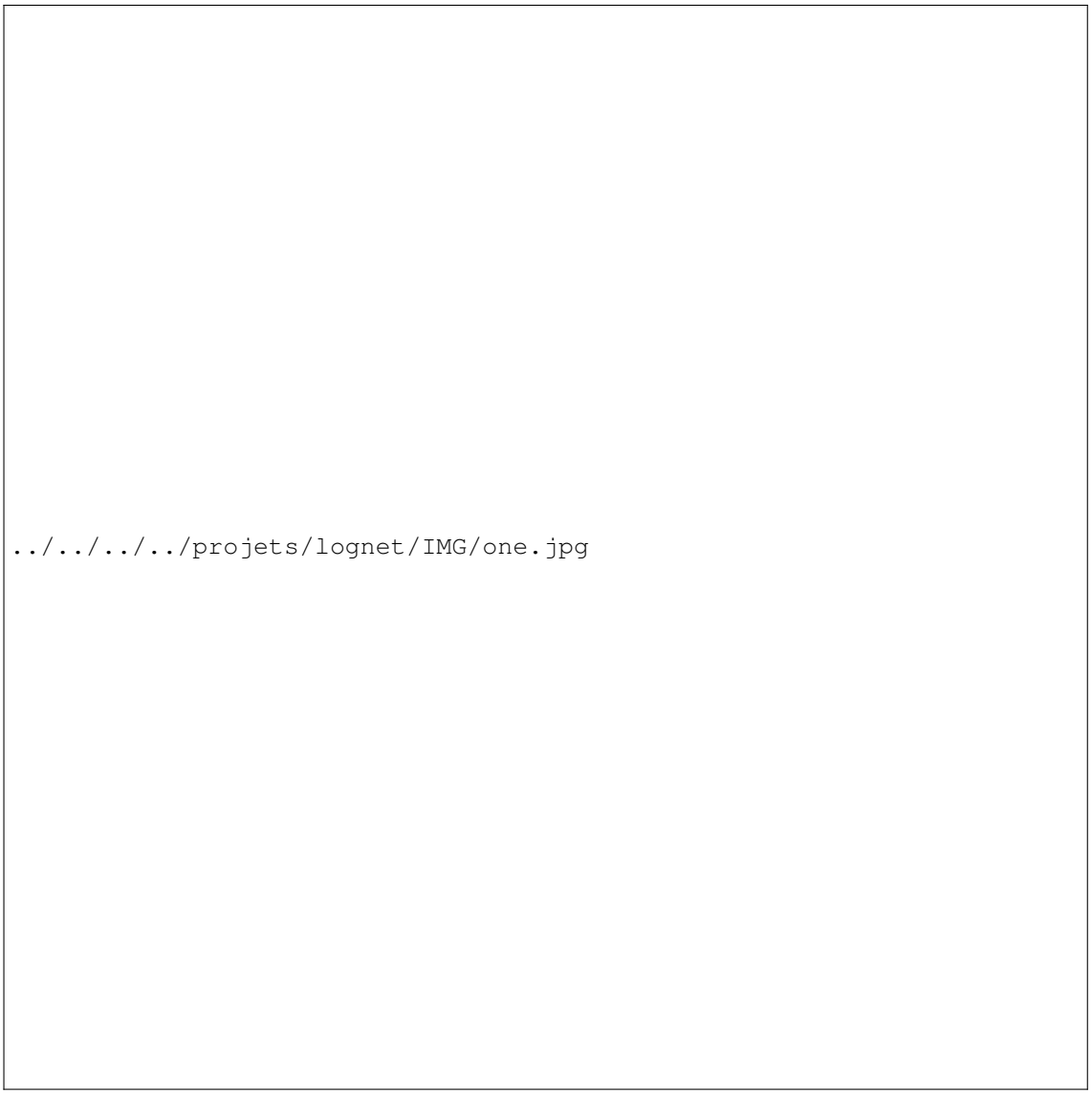


Figure 6. The Ariwheels simulator in Omnet

See the web page http://www-sop.inria.fr/members/Luigi.Liquori/ARIGATONI/Ariwheels.htm and http://arigtt.altervista.org.

5.11. Arigatoni simulator

Participants: Luigi Liquori [contact], Raphael Chand [Université de Geneva, Switzerland].

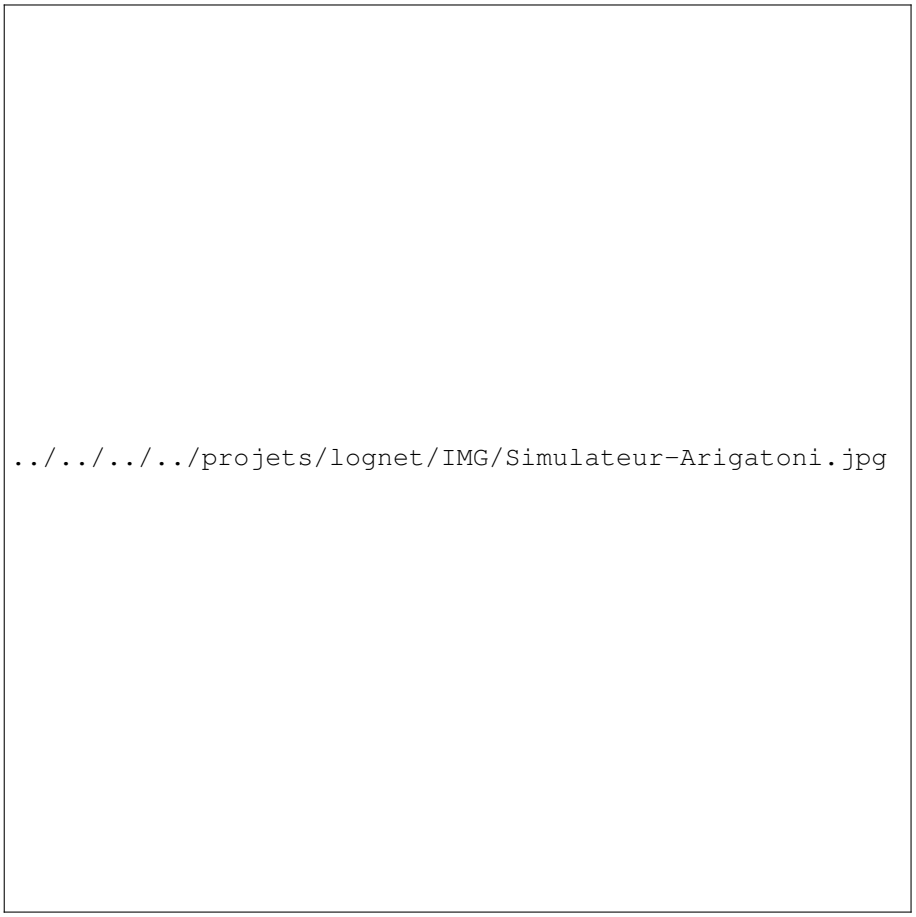


Figure 7. The Arigatoni simulator

We have implemented in C++ ($\sim$2.5K lines of code) the Resource Discovery Algorithm and the Virtual Intermittent Protocol of the Arigatoni Overlay Network. The simulator was used to measure the load when we issued n service requests at Global Computers chosen uniformly at random. Each request contained a certain number of instances of one service, also chosen uniformly at random. Each service request was then handled by the Resource Discovery mechanism of Arigatoni networks.

5.12. Synapse client

Participants: Laurent Vanni [contact], Luigi Liquori, Cédric Tedeschi, Vincenzo Ciancaglini.

In order to test our Synapse protocol [21] on real platforms, we have initially developed JSynapse, a Java software prototype, which uses the Java RMI standard for communication between nodes, and whose purpose is to capture the very essence of our Synapse protocol. It is a flexible and ready-to-be-plugged library which can interconnect any type of overlay networks. In particular, JSynapse fully implements a Chord-based inter-overlay network. It was designed to be a lightweight and easy-to-extend software. We also provided some

practical classes which help in automating the generation of the inter-overlay network and the testing of specific scenarios. We have experimented with JSynapse on the Grid'5000 platform connecting more than 20 clusters on 9 different sites. Again, Chord was used as the intra-overlay protocol. See, http://www-sop.inria.fr/teams/lognet/synapse-net2012/.

5.13. Open Synapse client

Participant: Bojan Marinkovic [contact].

Opensynapse is an open source implementation of [21]. It is available for free under the GNU GPL. This implementation is based on Open Chord (v. 1.0.5) - an open source implementation of the Chord distributed hash table implementation by Distributed and Mobile Systems Group Lehrstuhl fuer Praktische Informatik Universitaet Bamberg, see http://www-sop.inria.fr/teams/lognet/synapse-net2012/.

Opensynapse is implemented on top of an arbitrary number of overlay networks. Inter-networking can be built on top of Synapse in a very efficient way. Synapse is based on co-located nodes playing a role that is reminiscent of neural synapses. The current implementation of Opensynapse in this precise case interconnects many Chord overlay networks. The new client currently can interconnect an arbitrary number of Chord networks. This implementation follows the notation presented in [20], and so, each new Chord network is called a *Floor*.

5.14. Husky interpreter

Participants: Marthe Bonamy [contact], Luigi Liquori.

```
../../../../projets/lognet/IMG/husky.jpg
```

Figure 8. Launching the Husky interpreter

Husky is a variable-less language based on lambda calculus and term rewriting systems. Husky is based on the version 1.1 of *Snake* It was completely rewritten in CAML by Marthe Bonamy, ENSL (new parser, new syntactic constructions, like, *e.g.*, guards, anti-patterns, anti-expressions, exceptions and parametrized pattern matching). In *Husky*, all the keywords of the language are ASCII-symbols. It could be useful for teaching basic algorithms and pattern-matching to children.

5.15. myTransport Gui

Participants: Laurent Vanni [contact], Vincenzo Ciancaglini, Liquori Liquori.

myTransport is a GUI built on top of the Synapse protocol and network. Its purpose is to be a proof of concept of the future service of info-mobility to be available in the myMed social Network, see Figure 9 . The GUI is written in Java and it is fully functional in the Nokia N800 Internet tablet devices. myTransport has been ported to the myMed social network.

5.16. myDistributed Catalog for Digitized Cultural Heritage

Participants: Vincenzo Ciancaglini [contact], Bojan Marinkovic [MISANU, Serbia], Liquori Liquori.

Figure 9. myTransport on the Nokia N800 Internet tablet

Peer-to-peer networks have emerged recently as a flexible decentralized solution to handle large amount of data without the use of high-end servers. We have implemented a distributed catalog built up on an overlay network called "Synapse". The Synapse protocol allows interconnection of different overlay networks each of them being an abstraction of a "community" of virtual providers. Data storage and data retrieval from different kind of content providers (i.e. libraries, archives, museums, universities, research centers, etc.) can be stored inside one catalog. We illustrate the concept based on the Synapse protocol: a catalog for digitized cultural heritage of Serbia, see Figure 10 .

5.17. myStreaming P2P

Participants: Vincenzo Ciancaglini [contact], Rossella Fortuna [Politech Bari], Salvatore Spoto [Univ. Turin], Liquori Liquori, Luigi Alfredo Grieco [Politech Bari].

We have implemented, in Python, a fork of Goalbit http://goalbit.sourceforge.net, an open source video streaming platform peer-to-peer software streaming platform capable of distributing high-bandwidth live video content to everyone preserving its quality. We have aligned with the classical gossip-based distribution protocol a *DHT* that distribute contents according to a content-based strategy.

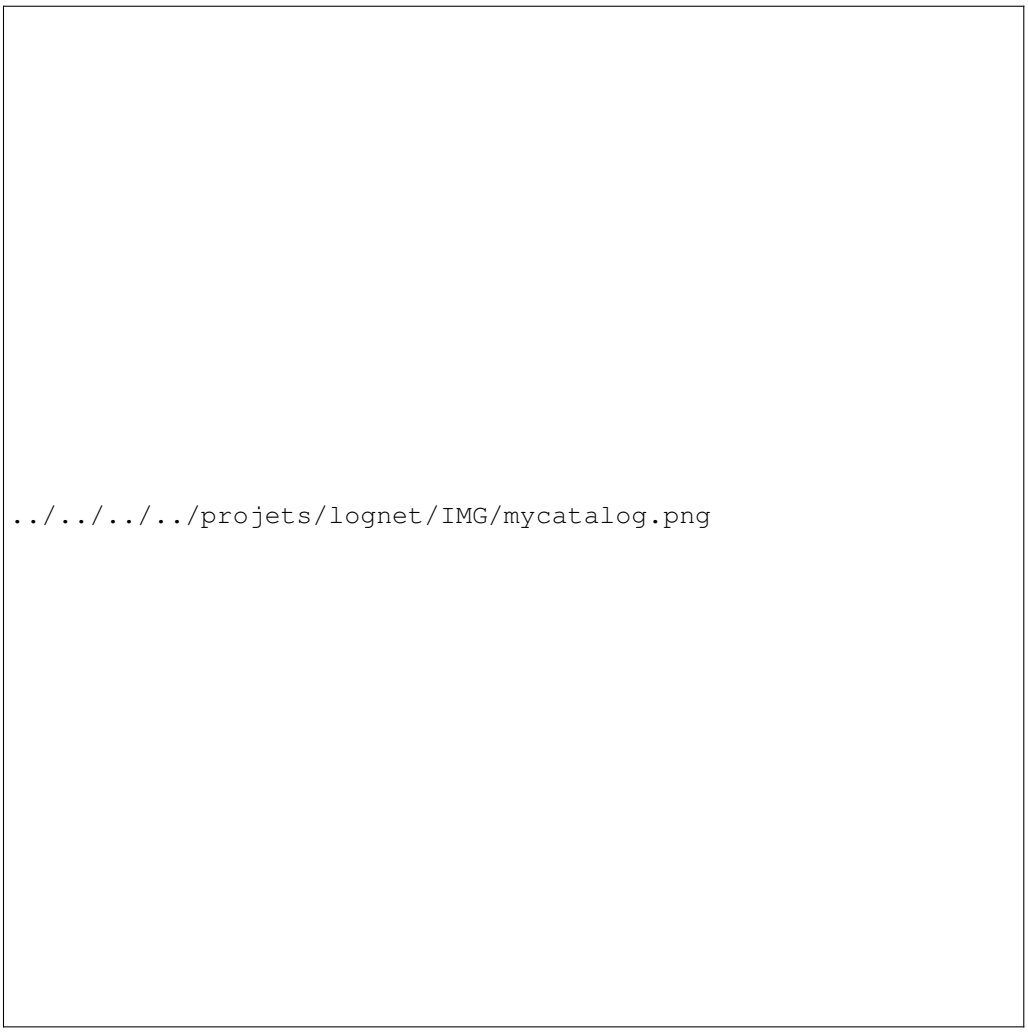


Figure 10. myDistributed Catalog

MAESTRO Project-Team (section vide)

MARELLE Project-Team

5. Software and Platforms

5.1. Tralics

Participant: José Grimm [correspondant].

Tralics is a Latex-to-XML translator available at http://www-sop.inria.fr/marelle/tralics. Version 2.15 has been released in 2012.

5.2. Semantics

Participant: Yves Bertot [correspondant].

This is a library for the Coq system, where the description of a toy programming language is presented. The value of this library is that it can be re-used in classrooms to teach programming language semantics or the Coq system. The topics covered include introductory notions to domain theory, pre and post-conditions, abstract interpretation, and the proofs of consistency between all these point of views on the same programming language. Standalone tools for the object programming language can be derived from this development.
See also the web page http://coq.inria.fr/pylons/pylons/contribs/view/Semantics/v8.4.

- ACM: F3.2 F4.1
- AMS: 68N30
- Programming language: Coq

5.3. 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.

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). ZooCrypt includes an experimental mechanism to generate EasyCrypt proofs of security of analyzed schemes.

5.4. CoqEAL

Participants: Maxime Dénès, Yves Bertot [correspondant].

CoqEAL is a library of certified algorithms for linear algebra to be used in Coq. It provides a collection of algorithms to compute efficiently on matrices and polynomials. These algorithms are designed to run efficiently directly in the Coq system and take the best advantage of the internal execution capabilities of the this system (virtual machine execution of native code execution after compilation).

5.5. CoqApprox

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

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.

5.6. CoqHensel

Participants: Érik Martin-Dorel, Laurent Théry, Micaela Mayero [Iut de Villetaneuse], Guillaume Hanrot [ENS Lyon].

The CoqHensel library provides a Coq formalization of Hensel's lemma for both univariate and bivariate cases, with some effective and modular certificate checkers for the univariate small integral roots problem, the bivariate small integral roots problem, as well as the integer small value problem (ISValP), with the ultimate goal to provide a fully formally verified chain for solving the Table Maker's Dilemma.

MCTAO Project-Team

5. Software and Platforms

5.1. Hampath

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

Hampath is a software developped to solve optimal control problems but also to study Hamiltonian flow. It has been developped 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. Software and Platforms

5.1. SMC Demos (Sequential Monte Carlo demos)

Participant: Fabien Campillo.

SMC Demos proposes a set of demonstration Matlab procedures for nonlinear filtering approximation via particle filtering (sequential Monte Carlo): bearing-only tracking with obstacles, tracking in digital terrain model, track-before-detect in a sequence of digital picture, mobile phone tracking based on the signal strength to nearby antenna. This software is deposited with the "Agence pour la Protection des Programmes" (APP, 7/7/2009) [[2]].

5.2. IBM Cellulose

Participant: Fabien Campillo.

In the context of the DISCO/ANR and MnMs/RNSC projects (see Sections 7.2.1 and 7.2.2), in collaboration with Ariane Bize (Irstea), the team has developed an individual-based model for the degradation of one cellulose bead (dozens of micrometers in diameter) by cellulolytic bacteria [[3]].

5.3. VITELBIO (VIrtual TELluric BIOreactors)

Participants: Jérôme Harmand, Alain Rapaport.

VITELBIO is a simulation tool for studying networks of interconnected chemostat models with the objective of mimicking microbial activities in heterogeneous media, such as the soil. This software, that has been developed with the help of ITK Company, is accessible on a server from any web navigator [[4]] and makes use of Flex for the user interface and Octave for the numerical integration. It is is no longer maintained but serves as a teaching support.

[2] http://www-sop.inria.fr/members/Fabien.Campillo/software/smc-demos/
[3] http://www-sop.inria.fr/members/Fabien.Campillo/software/ibm-cellulose/
[4] https://sites.google.com/site/vitelbio/logiciel

MORPHEME Project-Team (section vide)

NACHOS Project-Team

5. Software and Platforms

5.1. MAXW-DGTD

Participants: Stéphane Lanteri [correspondant], Loula Fezoui, Ludovic Moya, 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 [16]. 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 muticore 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/Middelware: Linux
- Required library or software: MPI (Message Passing Interface), CUDA
- Programming language: Fortran 77/95

5.2. MAXW-DGFD

Participants: Stéphane Lanteri [correspondant], 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 paralleization 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/Middelware: Linux
- Required library or software: MPI (Message Passing Interface)
- Programming language: Fortran 77/95

5.3. SISMO-DGTD

Participants: Nathalie Glinsky [correspondant], Stéphane Lanteri.

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 [6]. 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 paralleization 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/Middelware: Linux
- Required library or software: MPI (Message Passing Interface)
- Programming language: Fortran 77/95

5.4. NUM3SIS

Participants: Nora Aissiouene, Thibaud Kloczko [SED [1] team], Régis Duvigneau [OPALE project-team], Thibaud Kloczko [SED team], Stéphane Lanteri, Julien Wintz [SED team].

NUM3SIS http://num3sis.inria.fr is a modular platform devoted to scientific computing and numerical simulation. It is designed to handle complex multidisciplinary simulations involving several fields such as Computational Fluid Dynamics (CFD), Computational Structural Mechanic (CSM) and Computational ElectroMagnetics (CEM). In this context, the platform provides a comprehensive framework for engineers and researchers that speeds up implementation of new models and algorithms. From a software engineering point of view, num3sis specializes and extends some layers of the meta-platform dtk, especially its core and composition layers. The core layer enables the user to define generic concepts used for numerical simulation such as mesh or finite-volume schemes which are then implemented through a set of plugins. The composition layer provides a visual programming framework that wraps these concepts inside graphical items, nodes. These nodes can then be connected to each other to define data flows (or compositions) corresponding to the solution of scientific problems. NUM3SIS provides a highly flexible, re-usable and efficient approach to develop new computational scenarios and takes advantage of existing tools. The team participates to the development of the NUM3SIS platform through the adaptation and integration of the MAXW-DGTD simulation software. This work is being carried out with the support of two engineers in the framework of an ADT (Action de Développement Technologique) program.

5.5. Medical Image Extractor

Participants: Stéphane Lanteri, Julien Wintz [SED team].

[1] Service d'Experimentation et de Développement

Medical Image Extractor http://num3sis.inria.fr/software/apps/extractor provides functionalities needed to extract meshes from labeled MR or PET-CT medical images. It puts the emphasis on consistence, by generating both boundary surfaces, and volume meshes for each label (ideally identifying a tissue) of the input image, using the very same tetreahedrization. As this process requires user interaction, images and meshes are visualized together with tools allowing navigation and both easy and accurate refinement of the generated meshes, that can then be exported to serve as an input for other tools, within a multidisciplinar software toolchain. Using both DTK http://dtk.inria.fr and NUM3SIS SDKs, Medical Image Extractor comes within NUM3SIS'framework. Using cutting edge research algorithms developped by different teams at Inria, spread among different research topics, namely, visualization algorithms from medical image processing, meshing algorithms from algorithmic geometry, it illustrates the possibility to bridge the gap between software that come from different communities, in an innovative and highly non invasive development fashion.

NEUROMATHCOMP Project-Team

4. 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 developped by Adrien Wohrer during his PhD [73], [72] 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 [72] 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 starbust amacrine cells involved in direction selectivity (collaboration with Universidad Técnica Federico Santa María, Valparaiso, Chile, and Centro de Neurociencia de Valaparaiso) (see also e.g., [64]).

- 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], Gaia Lombardi, Hassan Nasser, Wahiba Tahouali.

Enas is a library providing numerical tools for the simulation of neural networks and the analysis of spike trains either coming from neural simulators or from biological experiments. The goal is to provide statistical methods allowing to estimate a spatio-temporal statistical model of spike train statistics (including thus pairwise spatio-temporal correlations, but also higher order correlations) from experimental rasters. More precisely, the algorithms are based on our theoretical results on spike trains statistical analysis via Gibbs distributions. We estimate a parametric Gibbs potential optimally characterizing the statistics of empirical spike trains (by minimisation of the Kullback-Leibler divergence between the empirical measure and the Gibbs measure). From this, classical statistical indicators such as firing rate, correlations, higher order moments and statistical entropy are obtained. Also, the form of the Gibbs potential provides essential informations on the underlying neural network and its structure. This method does not only allows us to estimate the spikes statistics but also to compare different models, thus answering such questions about the neural code as: are correlations (or time synchrony or a given set of spike patterns,...) significant with respect to rate coding?

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. The C/C++ code has been organized as "bean java" to ease its use by programmers non specialized in advanced object programming. As a consequence the code is distributed in the form of an include source for the lightest and the most universal integration into users codes.

Event neural assembly simulation is developed under CeCILL C licence
APP logiciel Enas: IDDN.FR.OO1.360008.000.S.P.2009.000.10600.

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

The software is freely downloadable at http://enas.gforge.inria.fr/v3/download.html.

Website: `http://enas.gforge.inria.fr/`

OASIS Project-Team

5. Software and Platforms

5.1. ProActive

Participants: Françoise Baude, Denis Caromel, Ludovic Henrio, Fabrice Huet [correspondant], Bastien Sauvan.

ProActive (Proactive Parallel Suite) is a Java library (Source code under AGPL license) for parallel, distributed, and concurrent computing, also featuring mobility and security in a uniform framework. With a reduced set of simple primitives, ProActive provides a comprehensive API to simplify the programming of applications that are distributed on a Local Area Network (LAN), on cluster of workstations, Clouds, or on Internet Grids.

The library is based on an Active Object pattern that is a uniform way to encapsulate:

- a remotely accessible object,
- a thread,
- an actor with its own script,
- a server of incoming requests,
- a mobile and potentially secure agent.

and has an architecture to inter-operate with (de facto) standards such as Web Service, HTTP transport, ssh, Globus, etc.

ProActive is only made of standard Java classes, and requires **no changes to the Java Virtual Machine**, no preprocessing or compiler modification; programmers write standard Java code. Based on a simple Meta-Object Protocol, the library is itself extensible, making the system open for adaptations and optimisations. ProActive currently uses the RMI Java standard library as default portable transport layer, but others such as Ibis or HTTP can be used instead, in an adaptive way.

ProActive is particularly well-adapted for the development of applications distributed over the Internet, thanks to reuse of sequential code, through polymorphism, automatic future-based synchronisations, migration of activities from one virtual machine to another. The underlying programming model is thus innovative compared to, for instance, the well established MPI programming model.

In order to cope with the requirements of large-scale distributed and heterogeneous systems like the Grid, many features have been incorporated into ProActive, including support for many transport and job submission protocols, GCM component support, graphical visualization interface, object migration, distributed and non-functional exception handling, fault-tolerance and checkpointing mechanisms; file transfer capabilities, a job scheduler, a resource manager able to manage various hosting machines, support for JMX and OSGi capabilities, web service object exposition, an SCA personality, etc.

ProActive is a project of the former ObjectWeb, now OW2 Consortium. OW2 is an international consortium fostering the development of open-source middleware for cutting-edge applications: EAI, e-business, clustering, grid computing, managed services and more. For more information, refer to [39], [37] and to the web pages http://www.objectweb.org and http://proactive.inria.fr/.

ProActive management, distribution, support, and commercialisation is now ensured by the start-up company ActiveEon (http://www.activeeon.com), in the context of a collaboration with Inria and UNS.

This year, the OASIS team made the following extensions to the ProActive library:

- Implementations related to the multi-active object programming model: multi-active components, declarative request service priority.
- Extension of the support of non-functional aspects for component systems: scripting language for reconfiguration, interceptor components.

5.2. Vercors platform

Participants: Eric Madelaine, Ludovic Henrio, Bartlomiej Szejna, Alexandra Savu, Oleksandra Kulankhina, Dongqian Liu.

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 finished conducting experiments within the Papyrus environment, aiming at the definition of a graphical specification environment combining some of the standard UML formalisms with a dedicated graphical formalism for the architecture of GCM components. We have concluded that Papyrus is not an appropriate environment for our purpose due to the fact that the software is very unstable and badly-documented.
- We have achieved this year a major port of the frontend of the Vercors, namely VCE, the Vercors Component Editor, that is now based on the Obeo Designer (Eclipse) platform (http://www.obeodesigner.com). The main motivation, and achievement of this port was to integrate editors for some existing UML formalisms (Class and State-machines) with our GCM architecture editor. The new version of Vercors Component Editor (VCE v.3) has an editor for the GCM Components diagrams integrated with the UML class and state-machines diagrams editors. It also includes a generator of the ADL v.2 specification of the GCM-based architecture and a diagrams validation module.

5.3. Open Simulation Architecture (OSA)

Participant: Olivier Dalle.

OSA stands for Open Simulation Architecture. OSA 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.

5.4. BtrPlace

Participants: Fabien Hermenier, Vincent Kherbache, Huynh Tu Dang.

Btrplace (http://btrp.inria.fr) is an open source virtual machine (VM) placement algorithm 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 reconfiguration 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, the catalog of constraints has been augmented according to the needs expressed inside OpenCloudWare. It as also been upgraded to a support for *continuous constraints* [11], a safer restriction mode to ensure the constraints can be satisfied at any moment, even during a reconfiguration process. A significant effort has also been made to make BtrPlace more usable and visible thanks to frequent releases, software documentation, tutorials, and live demo.

5.5. EventCloud

Participants: Françoise Baude, Fabrice Huet, Laurent Pellegrino, Bastien Sauvan, Iyad Alshabani, Maeva Antoine, Amjad Alshabani, Justine Rochas, Michel Jackson de Souza.

EventCloud (http://eventcloud.inria.fr) is an open source middleware that aims to act as a distributed datastore for data fulfiling the W3C RDF specification (www.w3.org/RDF/). It allows storing and retrieving quadruples (RDF triples with context) through SPARQL but also managing events represented as quadruples. The EventCloud architecture is based on a structured P2P overlay network targetting 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 middleware has been developed using the GCM/ProActive library embedding the most recent advances from the Multi-active Object model (see Section 6.1.1) and its implementation. Interactions from end user applications with an EventCloud instance can happen directly using Java APIs along the GCM/ProActive model, or they can be achieved through GCM interfaces exposed following the Web Services Notification specification. Web Services Notification (WSN) is a set of specifications from the OASIS consortium (www.oasis-open.org) that standardises the way Web Services interact using "Notifications" or "Events". They form the foundation for Event Driven Architectures built using Web Services.

The EventCloud middleware is currently used as a component within the platforms developed within ANR SocEDA (Section 7.1.1) and FP7 PLAY (Section 7.2.1.1) projects. A significant effort has also started to apply it in application domains from the BigData area, as in Intelligent Transportation Systems 7.2.2 .

OPALE Project-Team

5. Software and Platforms

5.1. NUM3SIS

Participants: Régis Duvigneau [correspondant], Nora Aïssiouene, Babett Lekouta.

The Opale project-team has initiated a few years ago the development of NUM3SIS (http://num3sis.inria.fr), which is 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 programing 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 two engineers 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 appplications tasks involved on the various clusters. This work was Laurentiu Trifan's PhD thesis, defended in October 2013 [37]. (See Fig. 1 .)

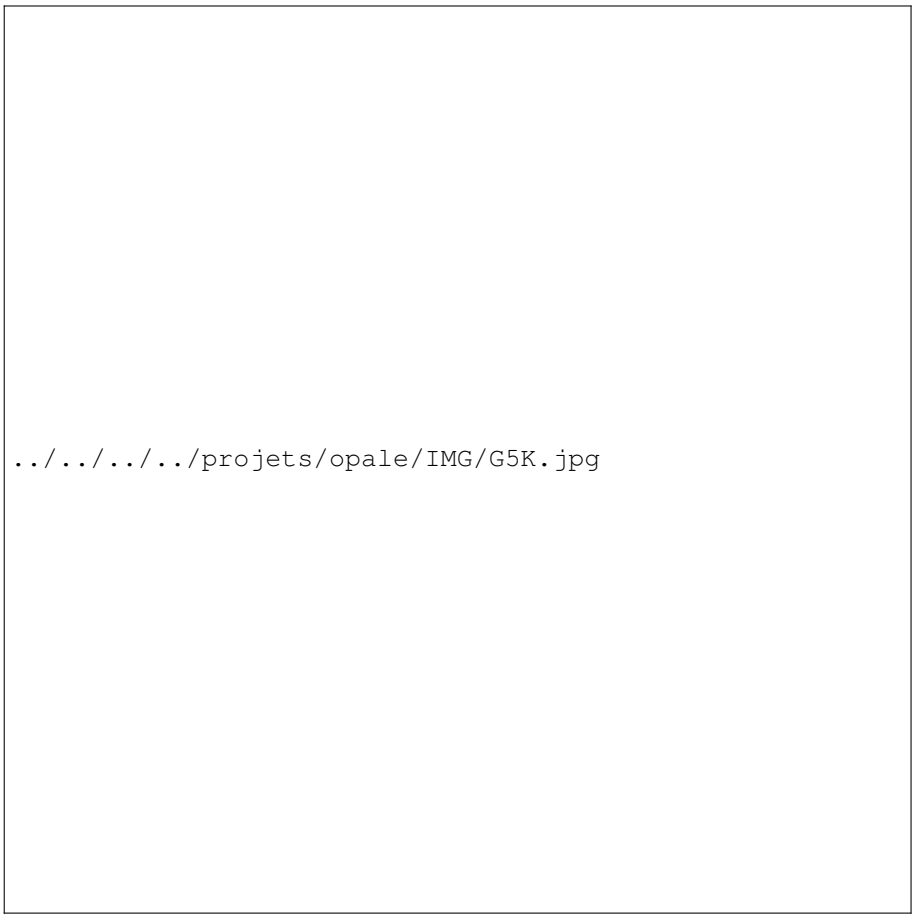


Figure 1. Testcase deployment on the Grid5000 infrastructure.

REVES Project-Team

4. Software and Platforms

4.1. Multi-View Image-Based Rendering and Relighting Suite

Participants: Clement Riant, Sylvain Duchêne, Pierre-Yves Laffont, Adrien Bousseau, George Drettakis.

We have designed and implemented a set of libraries for handling multi-view image-based rendering and relighting algorithms. These constitute the basis for the software developed for the EU projects VERVE and CR-PLAY.

4.1.1. RID: Rich Intrinsic-image Decomposer

We developed a software platform to perform rich intrinsic decomposition methods from photographs of outdoor scenes, as described in [13]. It includes main scripts and functions in Matlab for treatment of the input data, interfaces to software for multi-view reconstruction (Bundler, PMVS) and meshing from point clouds (method developed by Julie Digne, a postdoc in the GEOMETRICA project team). We then interface software for image matting using the Matting Laplacian, and User-Assisted Intrinsic Images. The system also includes an interface with Adobe Photoshop, for visualization and demonstration of our results in end-user image editing software. The method performs the computation of sun, sky and indirect lighting received at 3D points of an automatically reconstructed scene, using a modified version of the PBRT stochastic raytracer. Finally, there is a scene calibration module and an OpenGL viewer.

4.1.2. ROSSE: Relighting Outdoor Scenes with Shadow Editing

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 computer 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.

4.1.3. SWARPI: Superpixel Warp for Image-based rendering

Depth Synthesis and Warped-Based Superpixel Image-Based Rendering. This software package is the implementation of the publication [12]. The main software consists of two components: the depth synthesis step and the image-based rendering step:

a) The depth synthesis step is a Matlab package that reads 3D points coming from an automated 3D reconstruction pipeline, together with images and calibrated cameras, and produces the superpixel decomposition and the depth synthesis algorithm. The current version uses the open source packages bundler and PMVS (GPL v3 license), but other 3D reconstruction approaches could be used instead. b) The rendering step is in C++, and takes the result of the first step as input to allow interactive navigation. The code uses multi-pass deferred shading with geometry shaders (OpenGL 4.0 or above) to perform the rendering.

In addition to the implementation of [12], we have developed a Matlab interface for manual depth correction ("depth painting") An APP (Agency for the Protection of Programs) registration of this software is pending.

4.2. APF: state-of-the-art 3D audio library

Participants: Adrien David, George Drettakis.

This work was performed in collaboration with Jean-Christophe Lombardo of the DREAM group (i.e., the research support development group of our Inria center). REVES has several audio research publications over the last 10 years, which correspond to a class of functionalities such as clustering, masking, progressive processing etc.. The first component is the masking or culling algorithm, which aims at removing all the inaudible audio sources from a virtual scene based on perceptual metrics. The second component, called clustering, aims at grouping audio sources that are spatially close to each other and premix them to a representative cluster source, so that all spatialization related processing can be applied only on the representative premixed source [9]. Other audio topics were also considered and developed, like progressive and scalable frequency domain mixing, sound propagation, scalable reverberation, modal sound synthesis and contact sounds generation [1].

In order to maintain all the knowledge in the group and re-use these technologies in the Immersive Space, a previous young engineer (David Grelaud) wrote a fully documented audio library (APF) which gathers about 10 audio publications and 1 US patent. APF is a cross-platform, object oriented C++ API available on GForge. All the code has been re-implemented and a completely new software architecture resulted in a twofold increase in the speed of our algorithms. APF runs in the Immersive Space and uses the tracking system to spatialize virtual audio sources around the listener. It can also exploit personal Head Related Transfer Functions (HRTF).

We have implemented a network communications layer to create an audio rendering server on a separate machine, and the library is fully integrated into the osgVR platform.

APF has been critical in establishing collaborations in the context of various grant proposals (EU and national).

4.3. GaborNoise Software

Participants: Ares Lagae, George Drettakis.

We proposed a new procedural noise function last year, Gabor noise [6]. In the context of this project, we have developed a software package, which includes a CPU reference implementation of the 2D noise, and a complete GPU implementation of the 2D noise, surface noise, and 3D noise. This software package has been filed for APP protection and is in the process of being transferred to industrial partners.

This work is a collaboration with Sylvain Lefebvre, former member of the team, now in the ALICE project-team, Inria Nancy - Grand Est.

4.4. Gabor Noise By Example

Participant: George Drettakis.

In collaboration with B. Galerne, S. Lefebvre and A. Lagae (KU Leuven) we have released to code for the 2012 SIGGRAPH paper Gabor Noise By Example (see http://www-sop.inria.fr/reves/Basilic/2012/GLLD12/). This includes a matlab code for the analysis and C++/cuda code for the synthesis.

SCIPORT Team

5. 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 with MPI as parallel programming concept. The mesh model is unstructured tetrahedrization, with possible mesh motion. See also 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 Hascoet [correspondant], Valérie Pascual, Ala Taftaf.

TAPENADE is an Automatic Differentiation tool that transforms an original program into a new program that computes derivatives of the original program. Automatic 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, ans vector adjoint. Documentation is provided on the web site of the reserch team http://www-sop.inria.fr/tropics/, in Inria technical report RT-0300, and in [13]. TAPENADE runs under most operating systems and requires installation of Java jdk1.6 or upward.

- Version: v3.8, r4996, November 2013
- 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: automatic 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. TAPENADE differentiates computer programs according to the model described in section 3.1 and in [13] Higher-order derivatives can be obtained through repeated application of tangent AD on tangent- and/or adjoint-mode AD.

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 analysis 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. Industrial or commercial usage require a paying license, as detailled on the team's web page. The software has been downloaded several hundred times, and the web tool served several thousands of true connections (not counting robots). The tapenade-users mailing list is over one hundred registered users.

STARS Project-Team

5. Software and Platforms

5.1. SUP

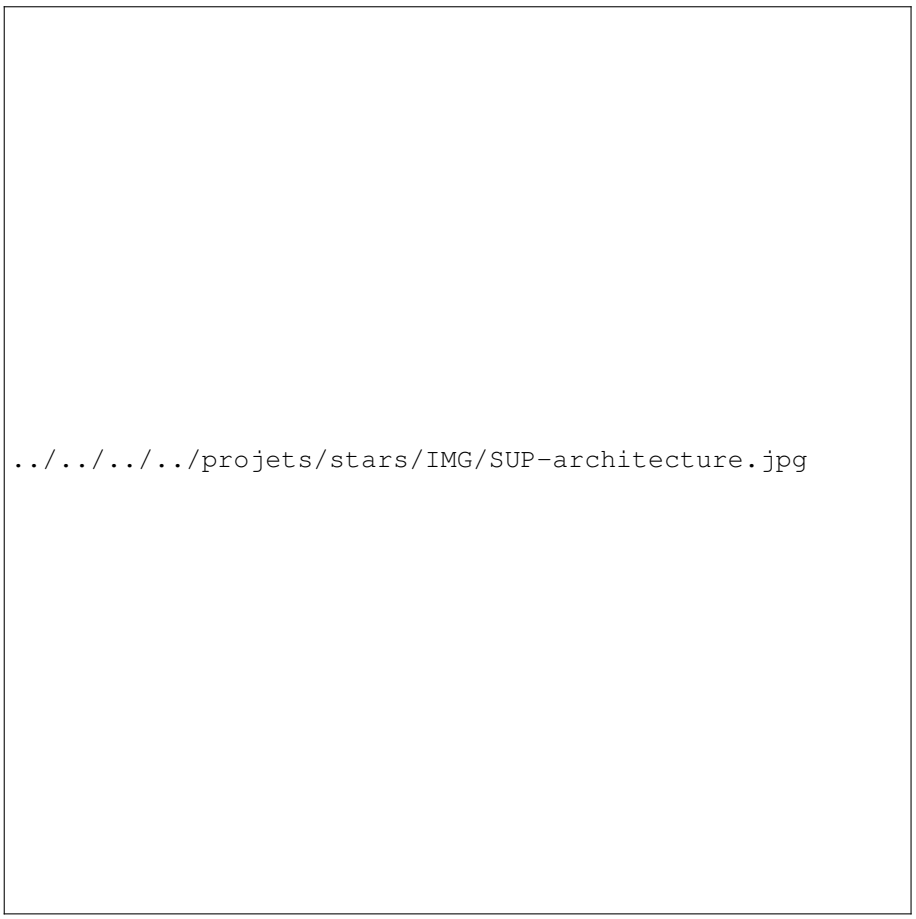


Figure 5. Tasks of the Scene Understanding Platform (SUP).

SUP is a Scene Understanding Software Platform written in C and C++ (see Figure 5). SUP is the continuation of the VSIP platform. SUP is splitting the workflow of a video processing into several modules, such as acquisition, segmentation, etc., up to activity recognition, to achieve the tasks (detection, classification, etc.) the platform supplies. Each module has a specific interface, and different plugins implementing these interfaces can be used for each step of the video processing. This generic architecture is designed to facilitate:

1. integration of new algorithms in SUP;
2. sharing of the algorithms among the Stars team.

Currently, 15 plugins are available, covering the whole processing chain. Several plugins are using the Genius platform, an industrial platform based on VSIP and exploited by Keeneo.

Goals of SUP are twofold:

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

5.2. ViSEval

ViSEval 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 (ViSEvAl, 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.
- For users to easily modify or add new metrics.

The ViSEvAl 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 amount 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 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 widows 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).

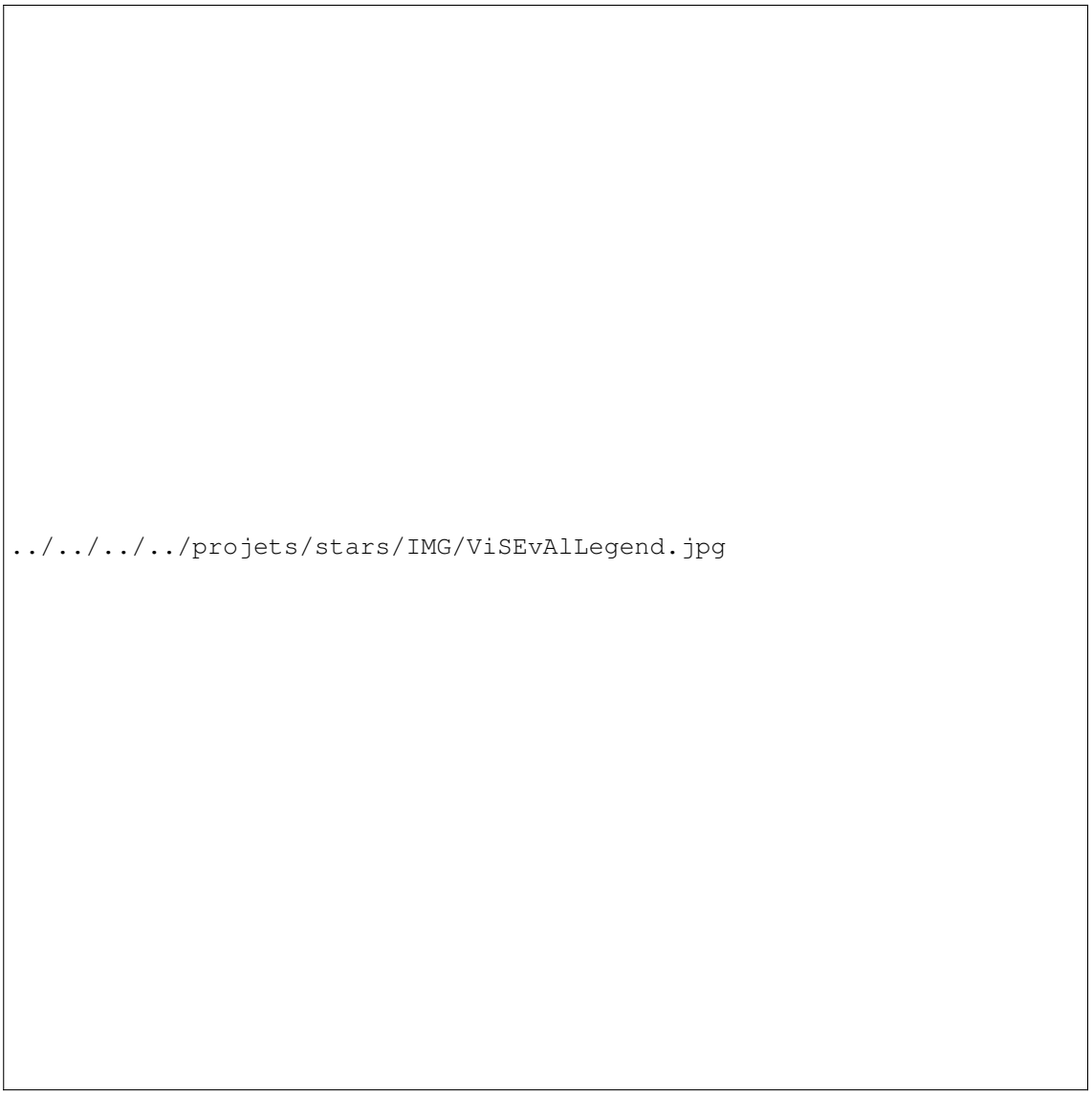


Figure 6. GUI of the ViSEvAl software

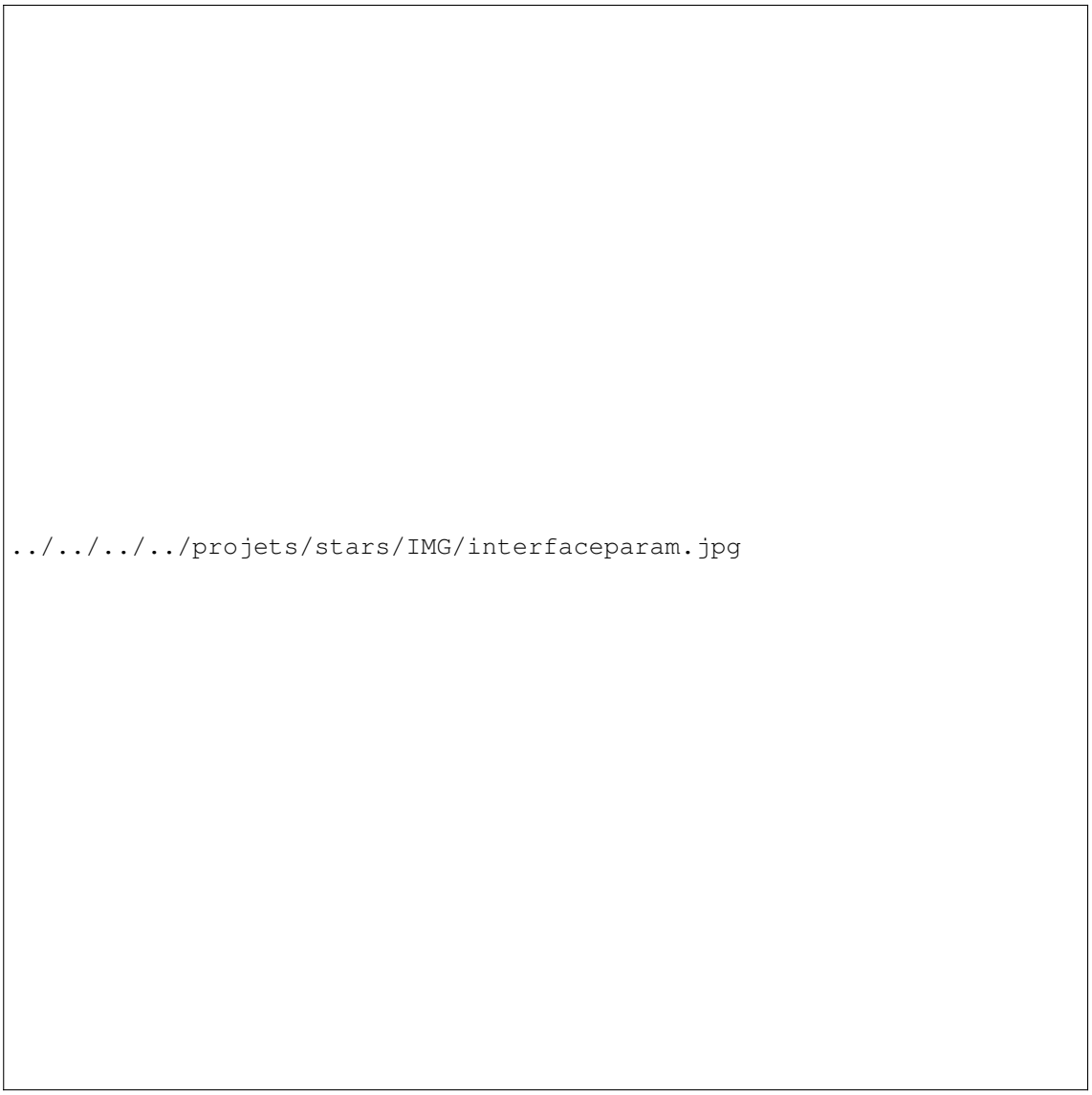


Figure 7. The object window enables users to choose the object to display

Figure 8. The multi-view window

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 ViSEvAl 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. ViSEvAl 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/ViSEvAl_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 [19] [82] programs. LE is a synchronous language supporting a modular compilation. It also supports automata possibly designed with a dedicated graphical editor.

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.

Figure 9. The Clem Toolkit

TITANE Team

5. Software and Platforms

5.1. CGAL, the Computational Geometry Algorithms Library

Participants: Pierre Alliez, Clement Jamin, Florent Lafarge, Sven Oesau, David Bommes.

CGAL is a C++ library of geometric algorithms and data structures. Our team is involved in several ongoing implementations: parallelization of mesh generation and triangulations, shape detection in unstructured point sets, geodesic distances on surface meshes and barycentric coordinates (in collaboration with Dmitry Anisimov). Pierre Alliez is a member of the CGAL Editorial Board.

5.2. APP deposits

5.2.1. MeshMantics

Participants: Yannick Verdie, Florent Lafarge, Pierre Alliez.

MeshMantics is a software for segmenting 2-manifold surface meshes in an urban context. Four classes of interest are considered: ground, vegetation, roof and facades.

TOSCA Project-Team

5. Software and Platforms

5.1. SDM

Participant: Mireille Bossy [correspondant].

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 project-team MOISE.

- Version: 2.0

5.2. CarbonQuant

Participants: Mireille Bossy [correspondant], Selim 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.

Starting in September 2011, CarbonQuant is an ADT [1] Inria.

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

- Version: 2.0

[1] Technology Development Action

VIRTUAL PLANTS Project-Team

4. Software and Platforms

4.1. V-Plants

Participants: Frédéric Boudon, Christophe Godin [coordinator], Yann Guédon, Christophe Pradal [software architect], Jean-Baptiste Durand, Pascal Ferraro, Julien Coste, Guillaume Baty.

Computer algorithms and tools developed by the Virtual Plants team are integrated in a common software suite *V-Plants*, dedicated to the modeling and analysis of plant development at different scales (e.g. cellular tissue, whole plant, stand). The VPlants packages are integrated in OpenAlea as Python components. Several components are distributed and usable through the visual programming environment (see figure 1):

- Multi-scale geometric modeling and visualization. VPlants.PlantGL is a geometric library which provides a set of graphical tools and algorithms for 3D plant modeling at different scales [7]. It is used by many other components to represent the geometry of biological shapes from 3D meristems, plant architectures to plant populations. VPlants.PlantGL is built around a scene-graph data structure and provides efficient algorithms and original geometrical shapes (parametric surfaces, dedicated envelops), that are useful for plant modeling.
- Statistical sequence and tree analysis. Different statistical packages (i.e. VPlants.StatTool, VPlants.SequenceAnalysis, VPlants.TreeMatching and VPlants.TreeAnalysis) are now available in OpenAlea. They provide different models and algorithms for plant architecture analysis and simulation.
- Meristem functioning and development. A first set of components has been created in the last 4-years period to model meristem development in OpenAlea. These tools are currently being integrated thoroughly in the platform so that modelers and biologists can use them, and reuse components easily (for meristem 3D reconstruction, cell tracking, statistical analysis of tissues, creating and manipulating atlases, creating or loading models of growth that can further be run on digitized structures, etc).
- Standard data structure for plants. A new implementation of the MTG formalism for representing and manipulating multiscale plant architecture has been developed. It provides a central data-structure to represent plants in a generic way in OpenAlea. This implementation is available through the packages OpenAlea.MTG. These components make it possible to share plant representations between users and fosters the interoperability of new models.
- Simulation system. The study of plant development requires increasingly powerful modeling tools to help understand and simulate the growth and functioning of plants. In the last decade, the formalism of L-systems has emerged as a major paradigm for modeling plant development. Previous implementations of this formalism were made based on static languages, i.e. languages that require explicit definition of variable types before using them. These languages are often efficient but involve quite a lot of syntactic overhead, thus restricting the flexibility of use for modelers. We developed L-Py [2] an adaptation of L-systems to the Python language (basis of OpenAlea). Thanks to its dynamic typing property, syntax is simple, code execution is made easy and introspection property of the language makes it possible to parameterize and manipulate simply complex models. Independent L-systems can be composed to build-up more complex modular models. MTG structures (that are a common way to represent plants at several scales) can be translated back and forth into L-system data-structure and thus make it easy to reuse in L-systems tools for the analysis of plant architecture based on MTGs. Extensions to integrate multiscale dynamic models are currently being developed in collaboration with P. Prusinkiewicz and his team from the University of Calgary.

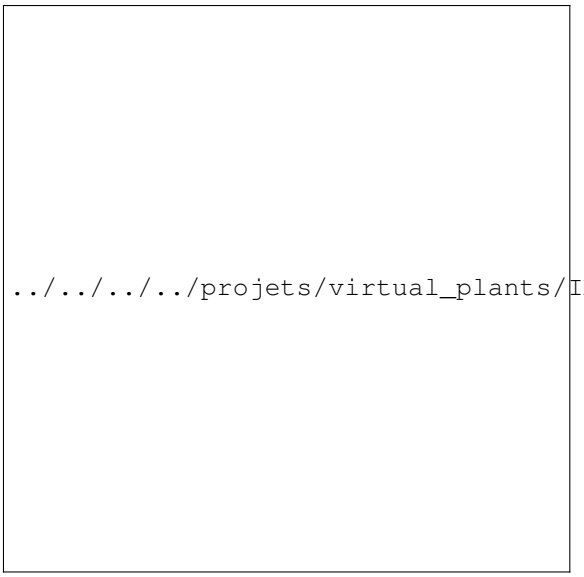


Figure 1. V-Plants components of the OpenAlea platform: simulating plant processes at different scales. Top Left: Reconstruction of a virtual meristem, analysis and simulation of the auxin fluxes inside the meristem. Top Right: Reconstruction of a virtual apple tree from digitized data. Bottom: Simulation of an ecosystem (A beech "Fagus Sylvatica L." trees forest) with a multi level approaches. On the left, explicit representation of the crown volumes that serves as input to generate the detailed representation, on the right.

4.2. OpenAlea

Participants: Frédéric Boudon, Christophe Godin, Yann Guédon, Christophe Pradal [coordinator], Christian Fournier, Julien Coste.

This research theme is supported by the Inria ADT Grant OpenAlea 2.0 and by an Agropolis RTRA Grant named OpenAlea.

OpenAlea[8] is an open source and collaborative software project primarily dedicated to the plant research community. It is designed as a component framework to dynamically glue together models from different plant research labs, and to enhance re-usability of existing models in the plant research community.

The architecture of OpenAlea is based on a component architecture. It provides a set of standard components (OpenAlea.Stdlib), a package manager to dynamically add and retrieve new components, and a port graph data-structure to compose models by interconnecting components into a data-flow.

Visualea provides a visual programming environment, used by scientists to build new model interactively by connecting available components together through an easy-to-use graphical user interface.

In 2013, the following progresses were accomplished:

1. Development and extension of OpenAlea and Visualea:
   - The standard library of components has been extended with useful scientific packages such as a flexible data plotting package (Openalea.Pylab), 2D and 3D image manipulation (Openalea.Image) and linear algebra operations (Openalea.Numpy).
   - Several models of computation have been implemented on the data-flow data-structure to enable discrete event simulation and control flow inside OpenAlea.
2. Animation and diffusion
   - A scientific board has been defined to manage the development and diffusion of OpenAlea. It is composed by 12 scientists.
   - StandAlone binary installers have been released on Windows and Mac to ease the installation of a large number of packages without relying on a web server. A Ubuntu repository has been set up on Launchpad.
   - The OpenAlea project is hosted at the Inria gforge (link http://openalea.gforge.inria.fr). The web site is visited by more than 370 unique visitor each month; 650000 web pages have been visited and the different available components of OpenAlea have been downloaded more than 540,000 times during the last two years. OpenAlea is the first project at Inria Gforge in term of number of downloads and of page views.

4.3. Alinea

Participants: Christian Fournier, Christophe Pradal, Frédéric Boudon.

Other participants : Bruno Andrieu, Michael Chelle, Gaëtan Louarn, Benoit de Solan, Mariem Abichou, Liqi Han, Elmer Ccopa-Rivera, Frédéric Baret, Rafaële Casa, Guillaume Garin, Corinne Robert, Sébastien Saint-Jean, Didier Combes, Camille Chambon, Romain Barillot, Jean-Christophe Soulie, Delphine Luquet.

The Alinea software suite is distributed as a meta-package of the OpenAlea Platform. It is produced by a consortium of modelers from INRA , and consists of various ecophysiological and biophysical models of simulation (radiative transfer, interaction between plant and pest, circulation of hydric fluxes, and dispersion). The project is suported by 5 INRA teams and the Inria Virtual Plants project. This project has been homologated as strategic ressource for INRA, and is integrated in the CATI IUMA (Centre Automatisé de Traitement de l'Information, : Informatisation et Utilisation des Modèles dédiés aux Agro-Ecosystemes). The five following components are distributed with the OpenAlea platform:

- Alinea.Adel is a module to simulate the 3D architectural development of gramineous crops. In 2013, a new parameterisation procedure was developed for wheat, that allows to use the model for the simulation of agronomic experiments with a minimal set of measurements [28].

- Alinea.Alep is a generic model developed in 2013 to simulate pathosystems from the scale of leaf to the scale of the canopy
- Alinea.Caribu is a modeling suite for lighting 3D virtual scenes, especially designed for the illumination of virtual plant canopies such as virtual crop fields. It uses a special algorithm, the nested radiosity, that allows for a precise estimation of light absorption at the level of small canopy elements.
- Alinea.pyRATP allows to simulate the light interception, photosynthesis and transpiration of a plant canopy.
- Alinea.TopVine is a component to reconstruct grapevine canopy structure. Other components are in developmental state and will be released after publication by their autors.

WIMMICS Project-Team

5. Software and Platforms

5.1. Corese

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

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 a SPARQL based pretty printing 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 CNRS 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 2013 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 Hub and of the Semantic Web Import Plugin for Gephi visualization.

The work on Corese was published in [2], [1], [3], [4].

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

5.2. Semantic Web Import Plugin for Gephi visualization

Participants: Erwan Demairy, Fabien Gandon, Olivier Corby.

The SemanticWebImport [2] plugin is intended to allow the import of semantic data into Gephi open graph vizualisation platform (figure 2). Gephi is an interactive visualization and exploration platform for all kinds of networks and complex systems, dynamic and hierarchical graphs. The imported data are obtained by processing a SPARQL request on the semantic data. The data can be accessed following three manners:

1. by accessing local RDF & RDFS files and using the embedded Corese engine to apply the SPARQL request;
2. by accessing a remote REST SPARQL endpoint. In that case, the SPARQL request is applied remotely and the graph is built locally by analyzing the result sent by the endpoint;
3. by accessing a remote SOAP SPARQL endpoint. As for the REST endpoint, the resulting graph is built from the result returned by the endpoint.

The software is released under version 1.0. It has received a development grant (ADT) from Inria.

Web pages: http://wiki.gephi.org/index.php/SemanticWebImport
https://gforge.inria.fr/projects/segviz-public

5.3. Datalift Linked Open Data Platform

Participants: Luca Costabello, Fabien Gandon, Serena Villata.

The Datalift platform aims at easing and automating publication of raw structured datasets on the Web of data. The platform proposes an extensible architecture and comes with modules enabling: data selection, schema selection and mapping; format and vocabulary conversion; storage, publication querying and access control; interlinking with other sources; visualization. The latest version of the code is maintained on the public forge of Inria [3].

[2] http://wiki.gephi.org/index.php/SemanticWebImport

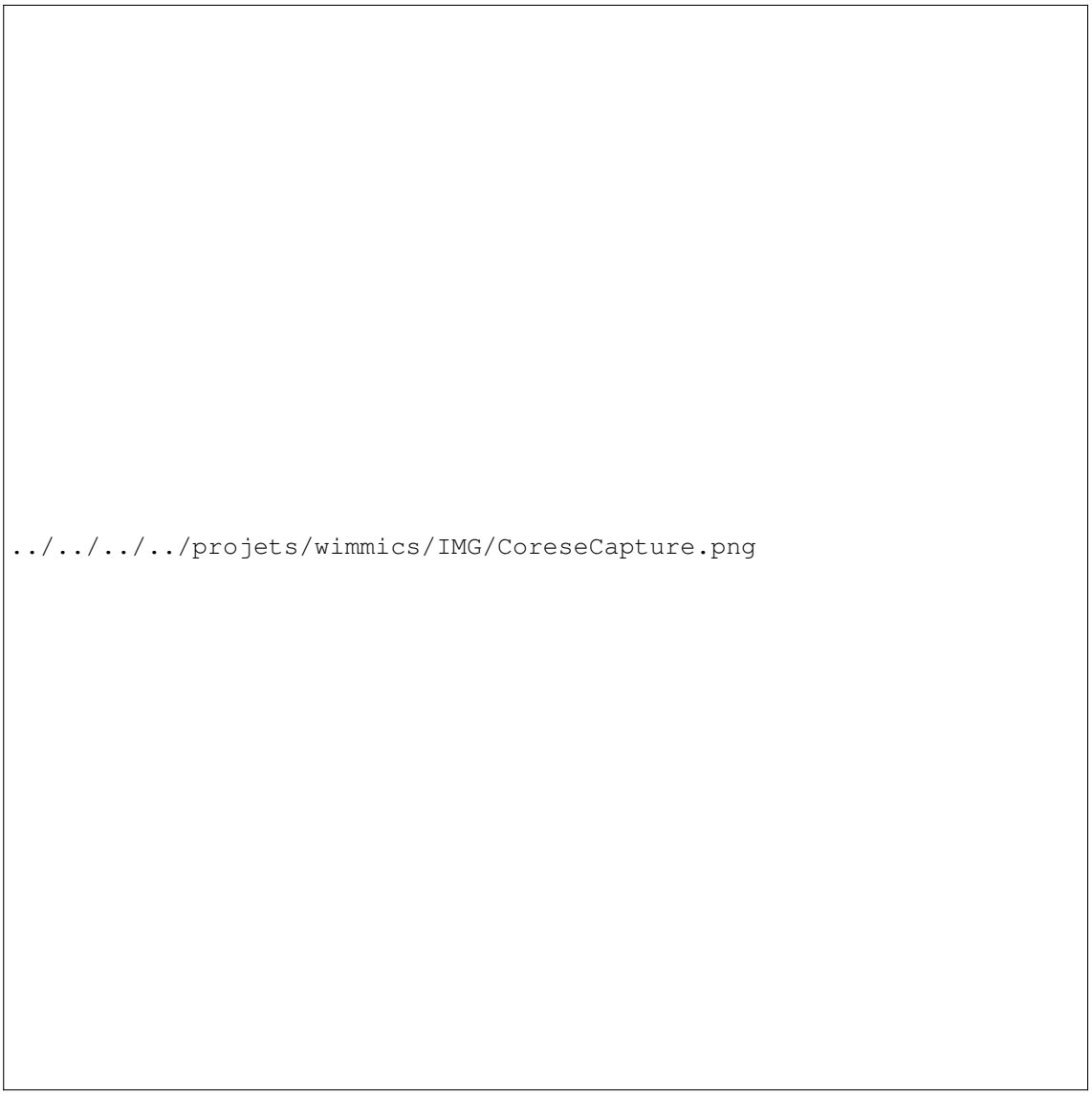


Figure 1. Corese

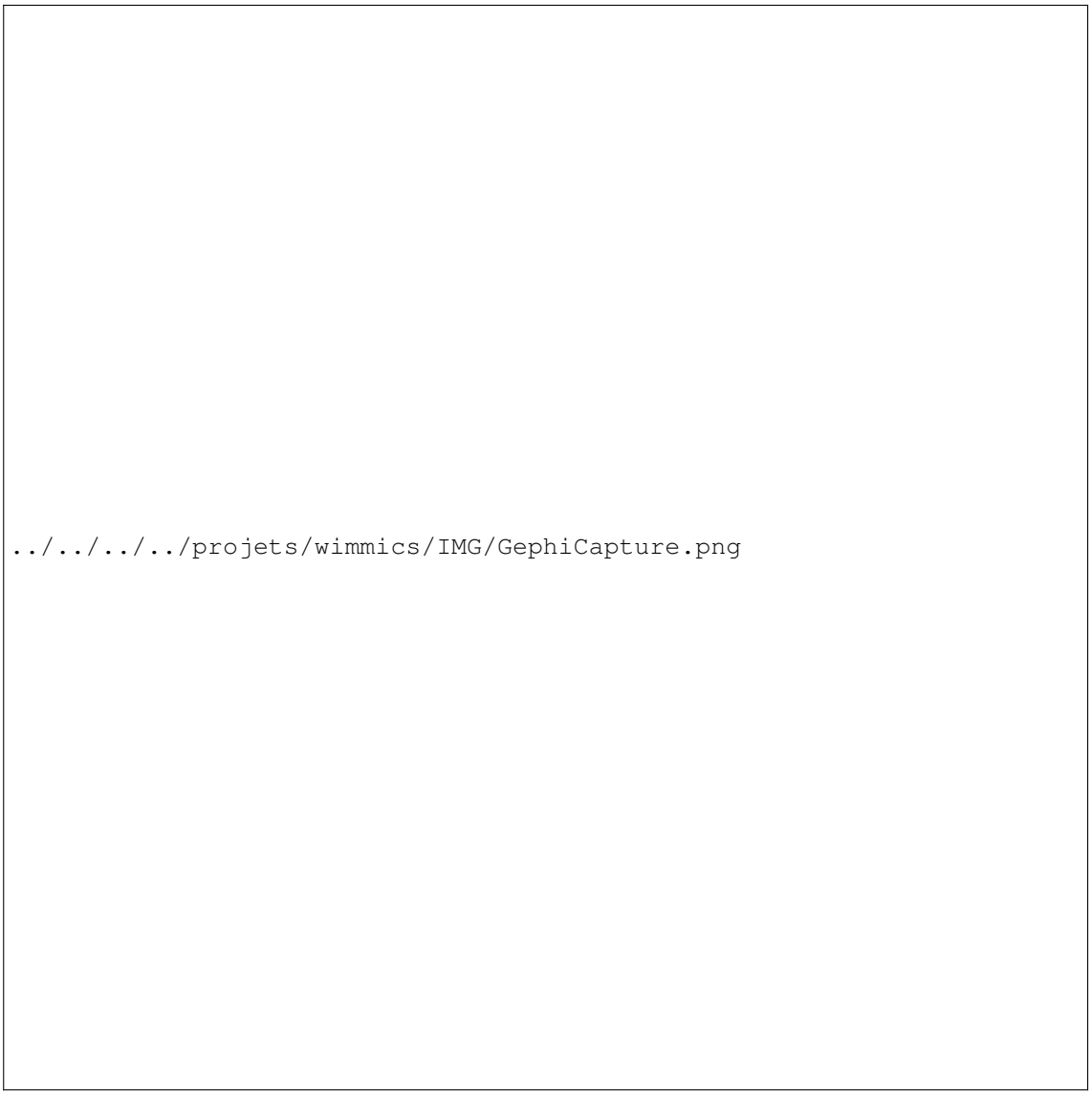


Figure 2. Gephi

Web page: http://www.datalift.org

5.4. Question Answering Wikiframework-based System

Participant: Elena Cabrio.

The QAKiS system (figure 3) 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

5.5. French Chapter of DBpedia

Participants: Julien Cojan, 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/.

5.6. Semantic Wiki

Participants: Pavel Arapov, Michel Buffa.

WikiNEXT is a semantic wiki prototype (figure 4) written in JavaScript, from database to server and client code. It is not in competition with wikis like Semantic Media Wiki, but more a test bed for new ideas. Every wiki page is an application that keeps a Web Socket open with the server, enabling incremental saves or collaborative editions using Google wave like algorithms. Using JavaScript on the whole chain of operations avoids data transformation from/to different formats like in traditional approaches (Objects, JSON/XML, and SQL). WikiNEXT uses JavaScript distributed objects and includes an IDE to write JS applications within wiki pages.

Web page: http://wikinext.gexsoft.com

5.7. ISICIL

Participants: Nicolas Delaforge, Fabien Gandon [resp.].

In the context of the ISICIL ANR project, we have developped a Semantic Web server which provides core services to manage simple tagging of resources (internal or from the Web) and to assist the semantic enrichment of the folksonomy of our communities of users. This server's implementation is based on the ISICIL main framework. The tagging model combines already existing ontologies such as SIOC [4], SCOT, and Newman's Tag Ontology[5]. SRTag, the model of folksonomy enrichment, is based on a named graph mechanism in order to maintain diverging statements made between tags using SKOS (for thesaurus like relation between tags) or SCOT (for spelling variant relations), and is shown in figure 5 .

[3] https://gforge.inria.fr/projects/datalift/
[4] http://sioc-project.org
[5] http://www.holygoat.co.uk/owl/redwood/0.1/tags

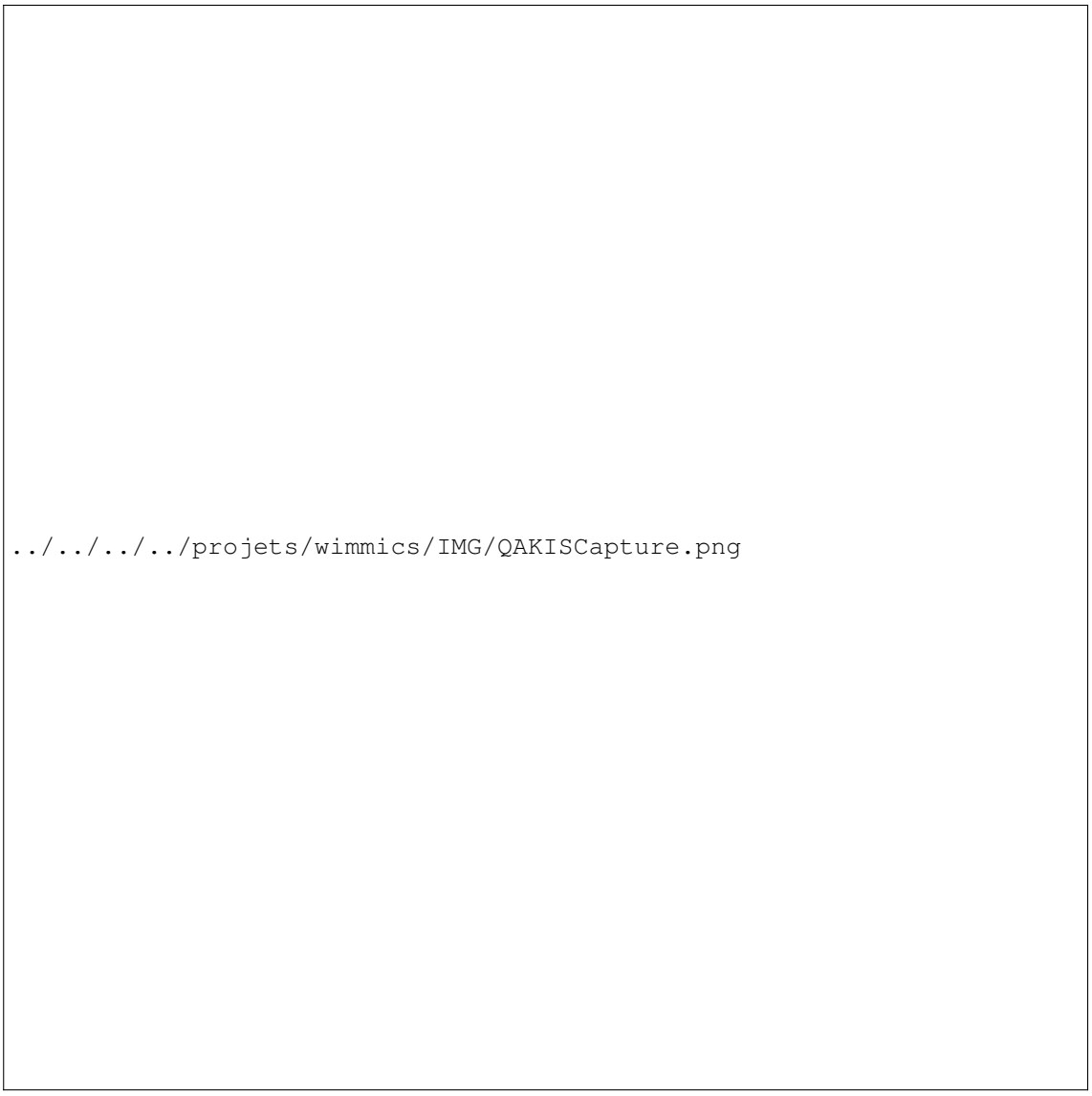


Figure 3. QAKIS

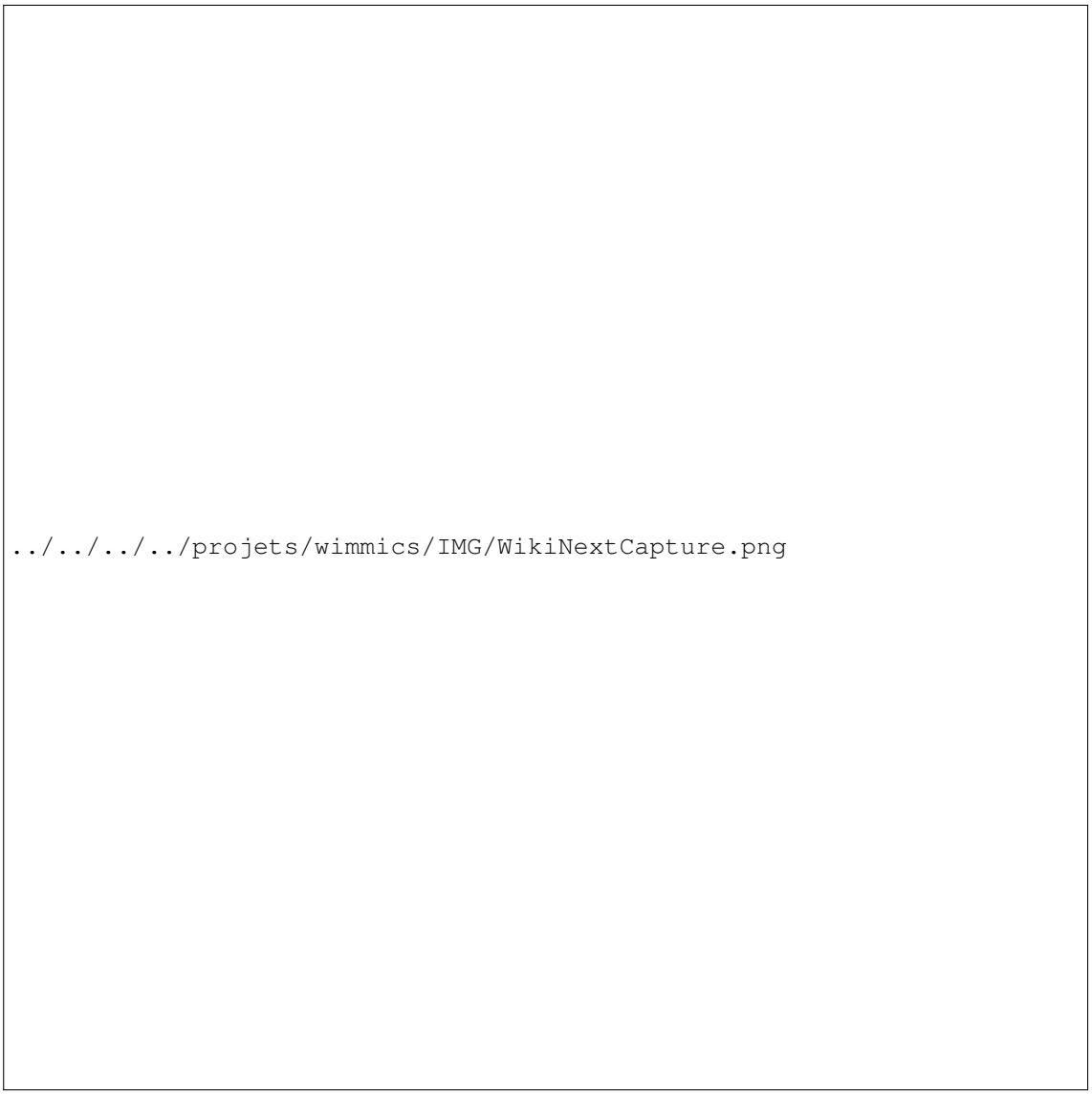


Figure 4. WikiNEXT

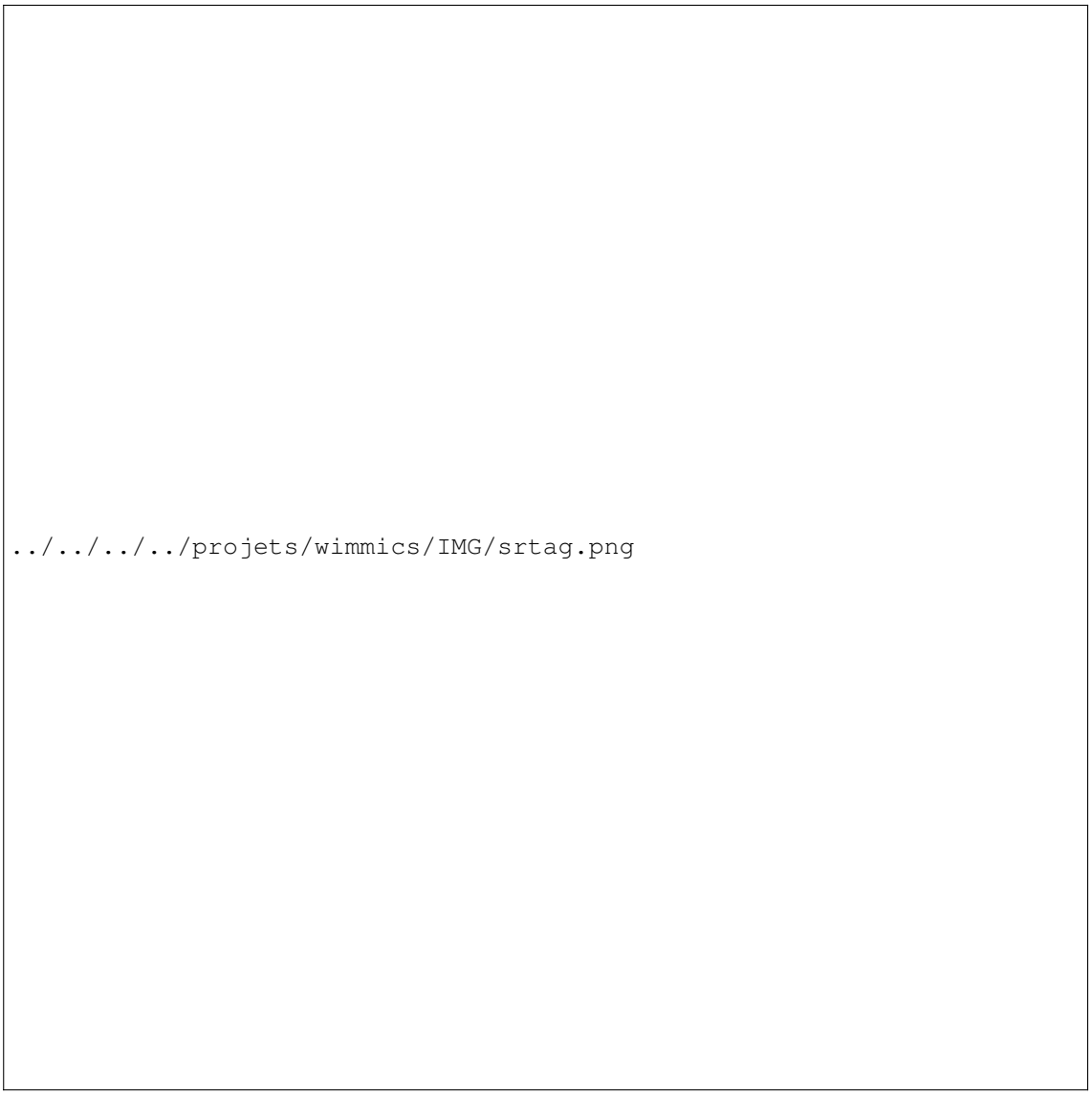


Figure 5. Folksonomy enrichment model

Web page: https://gforge.inria.fr/projects/isicil/

The code is now being refactored by the company Mnemotix, a SCOOP created as a spin-off of the project: http://www.mnemotix.com.

5.8. ZONE-project

Participant: Christophe Desclaux.

ZONE-project provides a new, innovative way to follow news (figure 6). At its core, the system is aggregating news items from various RSS feeds. Using the power of Semantic Web we are able to efficiently tag & annotate each news.Those tags are the basis of filters. Filters allow users to see only news that are relevant. For instance users can retrieve all news containing a tag, or on the contrary never see news containing specific tags. Basically it means that each user can create custom news feeds according to his interests. Though it may be tedious for John Doe to build its own filters, thus it will be possible to exchange filters with other users, or read specific news feeds built by other users. This will enable users to create news group feed focused on specific topics such as technology, heath, industry, transport, agriculture, communication, environment, etc. This project won the Inria BoostYourCode 2012 contest which was created in order to promote free & open source software.

Web page: http://www.zone-project.org/

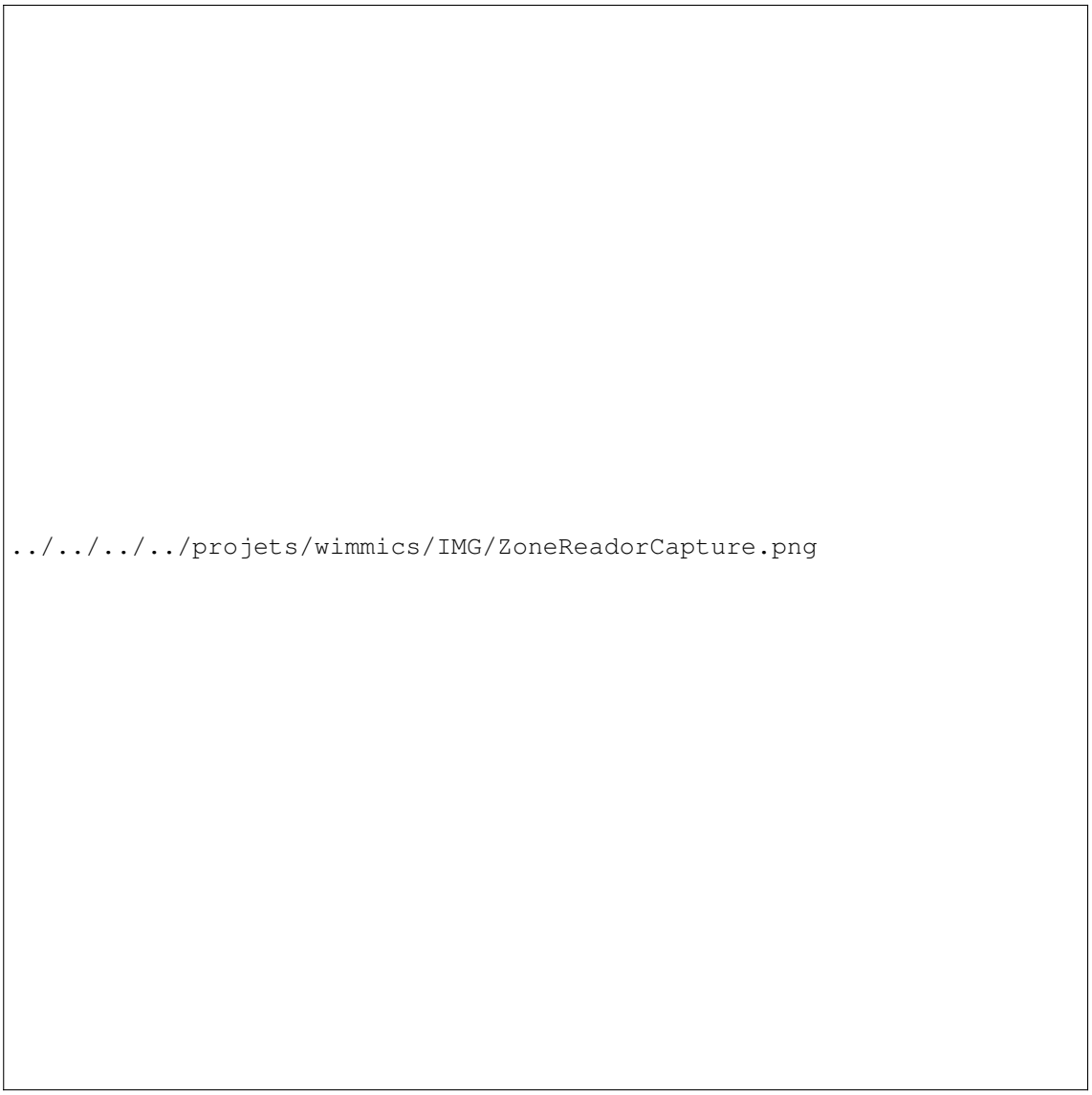


Figure 6. ZoneReador

ZENITH Project-Team

5. Software and Platforms

5.1. WebSmatch (Web Schema Matching)

Participants: Emmanuel Castanier, Rémi Coletta, Patrick Valduriez [contact].

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

In the context of the Action de Développement Technologique (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.

5.2. 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.

5.3. P2Prec (P2P recommendation service)

Participants: Esther Pacitti [contact], Didier Parigot, Maximilien Servajean.

URL: http://p2prec.gforge.inria.fr

P2Prec is a recommendation service for P2P content sharing systems that exploits users social data. To manage users social data, we rely on Friend-Of-A-Friend (FOAF) descriptions. P2Prec has a hybrid P2P architecture to work on top of any P2P content sharing system. It combines efficient DHT indexing to manage the users FOAF files with gossip robustness to disseminate the topics of expertise between friends. P2Prec is implemented in java using SON.

5.4. ProbDB (Probabilistic Database)

Participants: Reza Akbarinia [contact], Patrick Valduriez.

URL: http://probdb.gforge.inria.fr

ProbDB is a probabilistic data management system to manage uncertain data on top of relational DBMSs. One of the main features of the prototype is its portability; that means with a minimum effort it can be implemented over any DBMS. In ProbDB, we take advantage of the functionalities provided by almost all DBMSs, particularly the query processing functions. It is implemented in Java on top of PostgreSQL.

5.5. Pl@ntNet-mobile

Participants: Vera Bakic, Souheil Selmi, Hervé Goëau, Alexis Joly [contact].

URL: http://goo.gl/CpSrr3

Pl@ntNet-mobile is an image sharing and retrieval application for the identification of plants built in the continuity of the former web application Pl@ntNet-Identify [2] (presented in last year activity report). 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 this free app 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 80K users in about 150 countries (between 200 and 2000 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 80K images covering more than 3500 French plant species about 2/3 of the whole French flora (this is actually the widest identification tool built anytime).

5.6. Pl@ntNet-DataManager

Participants: Mathias Chouet [contact], Alexis Joly.

Pl@ntNet-DataManager [3] is a software dedicated to managing and sharing distributed heterogeneous botanical data. It is developed jointly by Zenith, the AMAP UMR team (CIRAD) and Telabotanica non profit organization. It allows scientists to define data structures dedicated to their own datasets, and share parts of their structures and data with collaborators in a decentralized way. Pl@ntNet DataManager offers innovative features like partial or complete P2P synchronization between distant databases (master-master), and a user friendly data structure editor. It also provides full text search, querying, CSV import/export, SQL export, image management, and geolocation. DataManager is built on NoSQL technology (CouchDB database), Javascript (Node.js), HTML5 and CSS3, and may be deployed on a server or run on a local machine (standalone version for Linux, Windows, Mac). It is being used by researchers and engineers of the Pl@ntNet Project (CIRAD, INRA, Inria, IRD, Tela-Botanica) to manage taxonomical referentials, herbarium data and geolocated plant observations.

5.7. SnoopIm

Participants: Julien Champ [contact], Alexis Joly, Pierre Letessier.

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

SnoopIm is a content-based search engine allowing to discover and retrieve small visual patterns or objects in large collections of pictures and to derive statistics from them (frequency, visual cover, size variations, etc.). It is implemented in Javascript on top of a C++ library developed in collaboration with INA [4]. The software is used at INA by archivists and sociologists in the context of the Transmedia Observatory project. It is also 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.

[2] http://identify.plantnet-project.org
[3] http://data.plantnet-project.org/
[4] http://www.ina-sup.com/

5.8. SciFloware

Participants: Dimitri Dupuis, Didier Parigot [contact], Patrick Valduriez.

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 will validate SciFloware with workflows for analyzing biological data provided by our partners CIRAD, INRA and IRD.