



RESEARCH CENTER
Sophia Antipolis - Méditerranée

FIELD

Activity Report 2012

Section Software

Edition: 2013-04-24

1. ABS Project-Team	4
2. AOSTE Project-Team	7
3. APICS Project-Team	10
4. ASCLEPIOS Project-Team	15
5. ATHENA Project-Team	16
6. AXIS Project-Team	17
7. AYIN Team	21
8. BIOCORE Project-Team	22
9. CASTOR Team	23
10. COFFEE Project-Team	25
11. COPRIN Project-Team	26
12. DEMAR Project-Team	28
13. FOCUS Project-Team	29
14. GALAAD Project-Team	31
15. GEOMETRICA Project-Team	33
16. GRAPHIK Project-Team	35
17. INDES Project-Team	37
18. LAGADIC Project-Team	40
19. LOGNET Team	45
20. MAESTRO Project-Team (section vide)	53
21. MARELLE Project-Team	54
22. MASCOTTE Project-Team	55
23. MCTAO Team (section vide)	58
24. MODEMIC Project-Team	59
25. MORPHEME Team	60
26. NACHOS Project-Team	61
27. NEUROMATHCOMP Project-Team	63
28. OASIS Project-Team	65
29. OPALE Project-Team	67
30. PLANETE Project-Team	69
31. REVES Project-Team	76
32. SCIPORT Team	78
33. STARS Team	80
34. TOSCA Project-Team	85
35. VIRTUAL PLANTS Project-Team	86
36. WIMMICS Team	90
37. ZENITH Project-Team	94

ABS Project-Team

4. Software

4.1. Software

This section briefly comments on all the software distributed by ABS. On the one hand, the software released in 2012 is briefly described as the context is presented in the sections dedicated to new results. On the other hand, the software made available before 2012 is briefly specified in terms of applications targeted.

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

4.1.1. *addict: Stoichiometry Determination for Mass Spectrometry Data*

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

Context. Our work on the stoichiometry determination (SD) problem for noisy data in structural proteomics is described in section 5.2.1. 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/voratom/>.

4.1.2. *vorpatch and compatch: Modeling and Comparing Protein Binding Patches*

Participants: Frédéric Cazals, Noël Malod-Dognin.

Context. Modeling protein binding patches 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 protein complex, *vorpatch* compute the binding patches, while the program *compatch* allows comparing two patches.

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

4.1.3. *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. 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/>.

4.1.4. *wsheller: Selecting Water Layers in Solvated Protein Structures*

Participants: Frédéric Cazals, Christine Roth.

Context. Given a snapshot of a molecular dynamics simulation, a classical problem consists of *quenching* that structure—minimizing the potential energy of the solute together with selected layers of solvent molecules. The program `wsheller` provides a solution to the water layer selection, and incorporates a topological control of the layers selected.

Distribution. Binaries for `wsheller` are available from <http://team.inria.fr/abs/software/wsheller>.

4.1.5. *intervor: Modeling Macro-molecular Interfaces*

Participant: Frédéric Cazals.

In collaboration with S. Lorient (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.

4.1.6. *vorlume: Computing Molecular Surfaces and Volumes with Certificates*

Participant: Frédéric Cazals.

In collaboration with S. Lorient (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>.

4.1.7. *ESBTL: the Easy Structural Biology Template Library*

Participant: Frédéric Cazals.

In collaboration with S. Lorient (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.

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

4.1.8. *A_purva*: Comparing Protein Structure by Contact Map Overlap Maximization

Participant: Noël Malod-Dognin.

In collaboration with N. Yanev (University of Sofia, and IMI at Bulgarian Academy of Sciences, Bulgaria), and R. Andonov (Inria Rennes - Bretagne Atlantique, and IRISA/University of Rennes 1, France).

Context. Structural similarity between proteins provides significant insights about their functions. Maximum Contact Map Overlap maximization (CMO) received sustained attention during the past decade and can be considered today as a credible protein structure measure. The solver *A_purva* is an exact CMO solver that is both efficient (notably faster than the previous exact algorithms), and reliable (providing accurate upper and lower bounds of the solution). These properties make it applicable for large-scale protein comparison and classification.

Distribution. The software is available from <http://apurva.genouest.org><http://apurva.genouest.org>.

AOSTE Project-Team

5. Software

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 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>), or UML2/MARTE models (http://www.omg.org/technology/documents/profile_catalog.htm).

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

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

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

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

5.4. SAS

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

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

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

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

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

APICS Project-Team

5. Software

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-eng.html>.

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

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

It thus performs model reduction in case 1) and 2) and frequency data identification in case 3). In the case of band-limited frequency data, it could be necessary to infer the behavior of the system outside the bandwidth before performing rational approximation (see 3.2.2). An appropriate Moebius 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 4.3) 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 modelled by an electrical circuit that was developed by the team (see section 4.3) 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” [53] 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/personnel/Fabien.Seyfert/Presto_web_page/presto_pres.html. 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 airborne systems and a license agreement has been recently negotiated with TAS-España. XLim (University of Limoges) is a heavy user of Presto-HF among the academic filtering community and some free license agreements are currently being considered with the microwave department of the University of Erlangen (Germany) and the Royal Military College (Kingston, Canada).

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 (see section 6.3). 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).

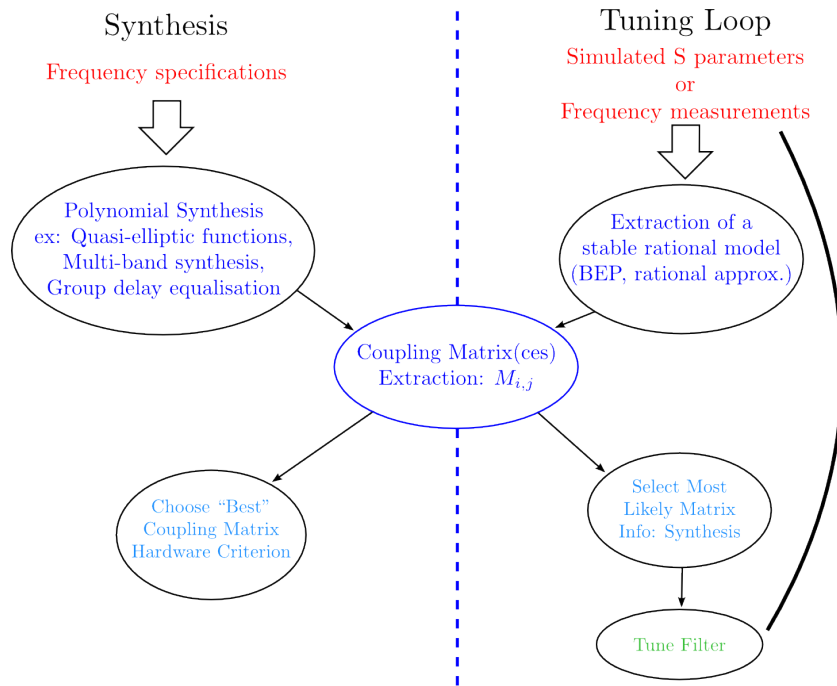


Figure 3. Overall scheme of the design and tuning process of a microwave filter.

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 2011, to TAS-Espagna to tune filter, with topologies with multiple solutions. The usage of Dedale-HF is here considered together with Presto-HF.

5.5. easyFF

Participant: Fabien Seyfert.

Status: A stable version is maintained.

This software has been developed by Vincent Lunot (Taiwan Univ.) during his Ph.d. 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 [11].

5.6. FindSources3D

Participant: Juliette Leblond [corresponding participant].

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

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

FindSources3D is a software dedicated to source recovery for the inverse EEG problem, in 3-layer spherical settings, from pointwise data (see <http://www-sop.inria.fr/apics/FindSources3D/>). Through the algorithm described in [16] 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 first release of FindSources3D is now available, which will be demonstrated and distributed within the medical teams we are in contact with (see Figure 4, CeCILL license, APP version 1.0: IDDN.FR.001.45009.S.A.2009.000.10000).

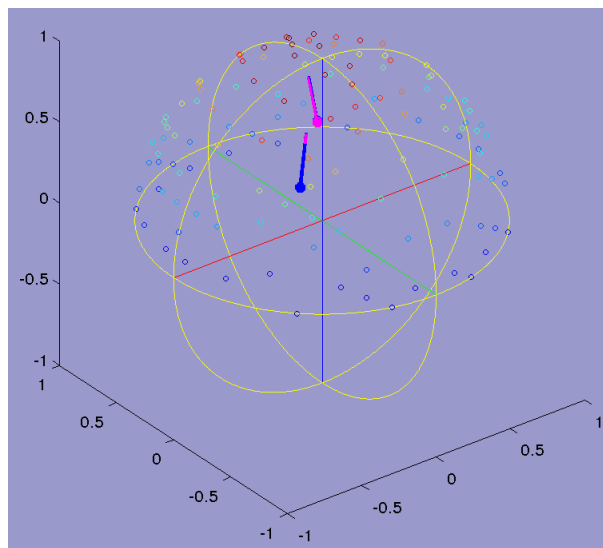


Figure 4. Potential values at electrodes on a sphere (scalp), recovered 2 sources (FindSources3D).

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 (Uppsala University, Sweden).

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.

Amongst 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

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 newer algorithms, but can also be used as an efficient prototyping tool. based on an advanced software architecture, it allows to:- create complex and evolving simulations by combining new algorithms with algorithms already included in SOFA- modify most parameters of the simulation (deformable behavior, surface representation, solver, constraints, collision algorithm, etc.) by simply editing an XML file- build complex models from simpler ones using a scene-graph description- efficiently simulate the dynamics of interacting objects using abstract equation solvers- reuse and easily compare a variety of available methods. It is mainly developed 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/Middleware: linux, windows, mac
- Required library or software: Qt - GPL - GLEW - BSD/MIT - Tinyxml - zlib
- Programming language: C/C++
- Documentation: - each function of the core API and each class in the SOFA modules - doxygen
- ACM: J.3
- Programming language: C/C++

4.2. MedInria

Participants: Benoît Bleuzé, Florian Vichot, Hakim Fadil, Loic Cadour, Agata Krason, Maxime Sermesant [correspondant], Nicolas Toussaint.

MedInria is a free collection of softwares developed by the Asclepios research project in collaboration with the Athena, Parietal and Visages Inria research projects. It aims at providing to clinicians state-of-the-art algorithms dedicated to medical image processing and visualization. Efforts have been made to simplify the user interface, while keeping high-level algorithms. MedInria is available for Microsoft windows XP/Vista/7, Linux Fedora Core, MacOSX, and is fully multithreaded.

The first release of Medinria 2.0 was done in April 2012.

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

- Version: 2.0
- Keywords: Medical Image Processing
- License: Proprietary Licence
- Type of human computer interaction: QT
- OS/Middleware: Windows - Linux - MacOSX
- Required library or software: DTI Track (Proprietary), vtkInria3D (CeCillB), Baladin (Proprietary)
- Programming language: C++

ATHENA Project-Team

5. Software

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 [5]. 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.

OpenMEEG is multiplatform (Linux, MacOS, Windows) and it is distributed under the French opensource license CeCILL-B. See also the web page <http://www-sop.inria.fr/athena/software/OpenMEEG/>.

5.2. Diffusion MRI

Participants: Aurobrata Ghosh, Rachid Deriche.

The algorithms previously developed within the ODYSSEE Project team and related to the Diffusion Tensor and Q-Ball imaging are available upon request from the Inria source forge (<https://gforge.inria.fr>). One can use all the estimation and visualization tools developed, ranging from estimation, regularization, segmentation to Q-ball estimation, fiber ODF estimation and tractography algorithms. New visualization tools for Q-Ball images represented by spherical harmonic decomposition have also been developed.

The software library comprises geometric and variational methods devised to estimate, regularize, segment and perform tractography in DT (Diffusion Tensor) and HARDI (High Angular Resolution) MRI images. The library is multi-platform (Linux, Windows and OS X) and is embedded into two open-source high level languages, TCL and Python.

5.3. medInria

Participants: Jaime Garcia Guevara, Théodore Papadopoulo.

The Athena team is involved along with the research teams Asclepios, Parietal and Visages in the development of medInria a free software platform dedicated to medical data visualization and processing.

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 contribution so far consists in various improvements on the core application as well as several plugins which will be available in the next version: advanced dMRI visualization and processing (integration of the Diffusion MRI library depicted in the previous section), M/EEG signal visualisation (by integrating code from the software AnyWave developed at by Bruno Colombet and J.-M. Badier [Inserm UMR 1106 and Aix-Marseille University](#)).

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

- Version: 2.0.1
- Keywords: Medical Image Processing and Visualization
- License: Proprietary Licence (soon open source for the core application)
- Multiplatform: Windows - Linux - MacOSX
- Programming language: C++

AXIS Project-Team

4. Software

4.1. Introduction

From its creation, AxIS has proposed new methods, approaches and **software** validated experimentally on various applications: Data Mining, Web usage Mining, Information Retrieval, Activity Modeling.

Some of our results are under process to be part of the FocusLab platform (CPER Télius 5.6) which is based on a Service oriented Architecture. The development process of the software part has started in 2011, finding ways to fund human resources. Such a platform aims the community of Living Labs domain. In [70], we report the usage of the FocusLab platform (hardware and software components) inside various regional and european projects..

4.2. Data Mining

4.2.1. Classification and Clustering Methods

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

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 [77], [76], [101], [100], [104], 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 [96], [99], [98], [97] (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 [84]. 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 [74] (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 [75]).

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 [95] which was the result of the european project ASSO⁶ (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 [71]. This software is registered at APP (Agence de la Protection des Programmes). The latest executive version of the SODAS 2 software, with its user manual can be downloaded at <http://www.info.fundp.ac.be/asso/sodaslink.htm> [78], [85].

⁶ASSO: Analysis System of Symbolic Official data

As a 2012 result, a release of MND (Dynamic Clustering Method for Multi-Nominal data) algorithm based on previous AxIS research (2003) has been done (cf. section 5.6).

4.2.2. *Extracting Sequential Patterns with Low Support*

Participant: Brigitte Trousse [correspondant].

Two methods for extracting sequential patterns with low support have been developed by D. Tanasa in his thesis (see Chapter 3 in [103] for more details) in collaboration with F. Masseglia and B. Trousse :

- **Cluster & Divide,**
- and **Divide & Discover** [11].

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

4.2.3. *Mining Data Streams*

Participants: Brigitte Trousse [correspondant], Mohamed Gaieb.

In Marascu's thesis (2009) [91], 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).

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 2011, in the context of the ELLIOT contract [cf. Section 6.3.1.1), SCDS has been integrated as a Web service (Java version) in the first version of FocusLab platform (cf. section 5.6) in the ELLIOT context: a demonstration was made on San Raffaele Hospital media use case at the first ELLIOT review at Brussels.

In 2012 we applied SCDS web service on data issued from co-creation step of two use cases in Logistics (BIBA) and Green Services (ICT Usage Lab). More data are needed to show the relevance of this method, it is planned in 2013 with the experimentation step of Green Services.

The three C++ codes done for the CRE (Orange Labs) have been deposited at APP.

4.3. Web Usage Mining

4.3.1. *AWLH for Pre-processing Web Logs*

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

AWLH (AxIS Web Log House) for Web Usage Mining (WUM) is issued from AxISlogminersoftware which implements the multi-site log preprocessing methodology and extraction of sequential pattern with low support developed by D. Tanasa in his thesis [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, ...),
- Incremental pre-processing,
- Java API to help integration of AWLH in external application.

An additional tool has been developed for capturing user actions in real time based on an open source project called "OpenSymphony ClickStream". An extension version of AWLH called **AWLH-Debate** has been developed for recording and structuring data issued from annotated documents inside discussion forums.

4.3.2. *ATWUEDA for Analysing Evolving Web Usage Data*

Participants: Yves Lechevallier [correspondant], Brigitte Trousse, Mohamed Gaieb, Yves Lechevallier [correspondant].

ATWUEDA for Web Usage Evolving Data Analysis [80] was developed by A. Da Silva in her thesis [79] 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 timestamp. Graphics are generated for each analyzed window, exhibiting statistics that characterizes changing points over time.

Version 2. of ATWUEDA (september 2009) is available at Inria's gforce website.

In 2011 we have demonstrated the efficiency of ATWUEDA [82] by applying it on another real case study on condition monitoring data streams of an electric power plant provided by EDF.

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

This year we studied how to transform the code of ATWUEDA as a web service for the version 1.2 of FocusLab: in fact we gave up this objective, which would require more resource than we have.

4.4. Information Retrieval

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

Participant: Brigitte Trousse [correspondant].

CBR*Tools [87], [88] is an object-oriented framework [89], [86] 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 neighbors 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 behavioral 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 behavioral 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 will be available 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/>.

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

Participant: Brigitte Trousse [correspondant].

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 [87], 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.

4.5. Activity Modeling

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

Participant: Dominique Scapin [correspondant].

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 a new version v1.2 of K-MAD was released in december. Its history, documentation and tool are available at: <http://kmade.sourceforge.net/index.php>. This work follows up the findings from the work of Caffiau and al. [73].

AYIN Team

5. Software

5.1. Software

5.1.1. Transfers

- The software MAD V2.0 was transferred to Galderma R&D in November 2012.
- The software Scombo v1.1 was transferred to Cutis laboratory (Galderma R& D, Sophia Antipolis) in May 2012, and to the French-Singaporean laboratory IPAL (Image and Pervasive Access Lab) in November 2012.

5.1.2. Deposits

- The software MAD (Melasma Automatic Detector) V2.0 was deposited with the APP in November 2012. A patent has also been deposited jointly by Galderma R& D and Inria during the same month. It deals with the melasma severity scoring from multi-spectral imaging.
- The software Scombo (Supervised Classifier of MultiBand Optical images) v1.1 was deposited with the APP in April 2012. It deals with the supervised classification of multiband optical images by using Markov random fields. It was developed with Aurélie Voisin, Vladimir Krylov and Josiane Zerubia.

BIOCORE Project-Team

5. Software

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

4. Software

4.1. 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. At present the software is only a private project but some parts of FluidBox are expected to be in the public domain by the end of the year.

4.2. PlaTo

Participants: Hervé Guillard [contact], Laure Combe.

The development of PlaTo (A platform for Tokamak simulation) (<http://www-sop.inria.fr/pumas/plato.php>) has been supported by an ADT action of the D2T an by the ANR ESPOIR. PlaTo is a suite of data and softwares dedicated to the geometry and physics of Tokamaks and its main objective is to provide the Inria large scale initiative “FUSION” teams working in plasma fluid models with a common development tool.

4.3. PaMPA

Participants: Cécile Dobrzynski [Bacchus], Hervé Guillard, Laurent Hascoët [Tropics], 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.

4.4. Cedres++

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

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

4.5. Equinox

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.

COFFEE Project-Team

5. Software

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. FV_PM

We are developing codes based on Finite Volume discretization, for the (2d and 3d) simulations of multiphase flows in porous media. For instance these methods apply to the simulation of problems motivated by CO2 storage, oil recovery or nuclear waste depository. A preliminary version, the code ComPASS (Computing Parallel Architecture to Speed up Simulations), which includes parallel procedures, has been recently developed, through a successful CEMRACS project.

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

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.

5.2.2. Int4Sci : a Scilab interface for interval analysis

Participants: David Daney, Gilles Trombettoni, Bertrand Neveu.

In 2006, we have started the development of a Scilab interface to C++ Bias/Profil interval arithmetic package and to the library ALIAS. The first version of Int4Sci has been released in 2008 – see <http://www-sop.inria.fr/coprin/logiciels/Int4Sci/> for linux, MacOS and Windows. A second version, compatible with Scilab 5.3 is in preparation . This interface provides an interval arithmetic, basic interval manipulation tools as well as the solving of linear interval systems. All functions are documented and a tutorial is available. Int4Sci is used in several universities for teaching the basis of interval analysis in place of using Rump's INTLAB for Matlab. We however lack the manpower to further enhance this software.

5.2.3. *Mathematica Interface to Interval Analysis*

Participants: Yves Papegay [correspondant], Jean-Pierre Merlet.

Since 2006, we have been implementing in Mathematica a high-level modular interface to the ALIAS library. Lack of manpower has slowed down this development.

DEMAR Project-Team

5. Software

5.1. Software

5.1.1. *RdP to VHDL tool*

Participants: Gregory Angles, David Andreu, Thierry Gil.

Our SENIS (Stimulation Electrique Neurale dIStribuee) based FES architecture relies on distributed stimulation units (DSU) which are interconnected by means of a 2-wire based network. A DSU is a complex digital system since it embeds among others a dedicated processor (micro-machine with a specific reduced instruction set), a monitoring module and a 3-layer protocol stack. To face the complexity of the 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 of view, consists in the interconnection of places and/or transitions according to well-defined mechanisms: interconnection by means of oriented arcs or by means of the "merging" operator (existing for both places and transitions).

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 aggregation as well as exception handling.

The Eclipse-based version of HILECOP is regularly 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.

A new version of SENISManager is under development according to an Eclipse-based design. This new version should be available in 2013.

FOCUS Project-Team

5. Software

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 2012 the development of Jolie has continued. The main activities have been:

- We have enhanced the correlation mechanism in Jolie to handle multiparty sessions with concurrent interactions with multiple participants.
- We have developed a compiler that projects choreography-based programs in Chor (<http://www.chor-lang.org>) to Jolie.
- We have developed a new website for Jolie (using Jolie itself), significantly updating its documentation.
- We have improved Jolie’s compatibility with the Java RMI technology.
- We have developed a first experimental implementation of a monitoring layer for Jolie services.

As last year, so in 2012 Jolie has been used for teaching, in a master course at the IT University of Copenhagen (ITU, Denmark) and in a master course at the Technical University of Denmark (DTU, Denmark).

5.2. Others

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

- *IntML* is a functional programming language guaranteeing sublinear space bounds for all programs [53]. See the Activity Reports of previous years (in particular 2010) for more details. During 2012 no substantial modifications have been made.
- *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.
- We have implemented a technique for the deadlock analysis of a concurrent object oriented language (ABS, designed within the European project HATS). The technique consists of
 - an inference system for contracts to be associated to methods. Contracts are terms that retain information about resource dependencies;
 - a fixpoint algorithm for solving contract definitions, which are recursive and may introduce new resource names.

The release of the software is planned for early 2013.

- *Croll-pi Interpreter* (<http://proton.inrialpes.fr/~mlienhar/croll-pi/implem/>). We have developed an interpreter for croll-pi using Maude. 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 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.

For other software, such as PiDuce, see the activity reports for Focus of previous years.

GALAAD Project-Team

5. Software

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 embedding of existing libraries written in other languages like C or C++. An interesting feature is that this extension mechanism supports template types, which automatically induce generic types inside Mathemagix. Connections with GMP, MPFR for extended arithmetic, LAPACK for numerical linear algebra are currently available in this framework.

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

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

- REALROOT: a set of solvers using subdivision methods to isolate the roots of polynomial equations in one or several variables; continued fraction expansion of roots of univariate polynomials; Bernstein basis representation of univariate and multivariate polynomials and related algorithms; exact computation with real algebraic numbers, sign evaluation, comparison, certified numerical approximation.
- SHAPE: tools to manipulate curves and surfaces of different types including parameterised, 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: Anaïs Ducoffe, Bernard Mourrain, Meriadeg Perrinel.

<http://axel.inria.fr>.

computational algebraic geometry, curve, implicit equation, intersection, parameterisation, 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 parameterisation, B-Spline, 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, detecting and computing self-intersection points of parameterized surfaces, 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. Optimisation techniques and solvers for partial differential equations developed by R. Duvigneau (OPALE) have been connected.

A new version of the algebraic-geometric modelers is developed by Meriadeg Perinnet to connect it to the platform Dtk in order to provide a better modularity and a better interface to existing computation facilities and geometric rendering interface.

The package is distributed as binary packages for Linux as well as for MacOSX. It is hosted at Inria's gforge (<http://gforge.inria.fr>) and referenced by many leading software websites such as <http://apple.com>. The first version of the software has been downloaded more than 15000 times, since it is available.

Collaboration with Gang Xu (Hangzhou Dianzi University, China), Julien Wintz (Dream).

GEOMETRICA Project-Team

5. Software

5.1. CGAL, the Computational Geometry Algorithms Library

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

With the collaboration of 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, Menelaos Karavelas, Sébastien Lorient, Abdelkrim Mebarki, Naceur Meskini, Andreas Meyer, Sylvain Pion, Marc Pouget, François Rebufat, Laurent Rineau, Laurent Saboret, Stéphane Tayeb, Jane Tournois, Radu Ursu, and Camille Wormser. <http://www.cgal.org>

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

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 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 meshes... Three 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 14 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

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 1.2, 2005–2010 GPL Licence). Currently, it supports Conceptual Graphs and import/export in RDFS.

Here are the major evolutions of the version delivered this year:

- *Cogui* now allows import/export in the Datalog+/- language thanks to a new Datalog+/- parser (see Sect. 5.4).
- Scripted rules were introduced. It is a new type of object that allows users to attach a script to a traditional rule in order to modify or control its behavior.
- A new interface ensures connectivity to a NoSQLdatabase (MongoDB).
- Large graphs can now be stored. In a near future, we will be able to perform queries on data too big to fit in central memory (see Sect. 5.2).

5.2. 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's thesis, and now of the PhD of Bruno Paiva Lima da Silva [34].

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 database used (e.g. *SPARQL* or *SQL*), or specifically designed algorithms (from a simple backtrack to a full constraint solver based upon *Choco* 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 (it is already used by other PhD students for their experiments), and will soon be ready to be distributed to a broader audience. Our generic approach will ease this dissemination to different research domains.

5.3. Kiabora

Participants: Swan Rocher [first year master internship], Michel Leclère, Marie-Laure Mugnier.

<http://www2.lirmm.fr/~mugnier/graphik/kiabora/index.html>

Kiabora is a tool dedicated to the analysis of a set of existential rules. It can check if this set belongs to a known *decidable* class of rules, either directly or by means of its *Graph of Rule Dependencies (GRD)*. *Kiabora* analyzes the properties of the strongly connected components in the GRD, which allows to determine properties of the rule set with respect to decidability as well as the kind of paradigm (forward or backward chaining) ensuring decidability.

Besides, *Kiabora* also provides format conversion and rule decomposition services. It is written in Java.

5.4. DLGP

Participants: Jean-François Baget, Michel Leclère, Marie-Laure Mugnier, Alain Gutierrez, Swan Rocher [first year master internship], Clément Sipieter [first year master internship].

http://www2.lirmm.fr/~mugnier/graphik/kiabora/downloads/datalog-plus_en.pdf

DLGP (for Datalog Plus) is a textual exchange format at once human-friendly, concise and easy to parse. This format can be seen as an extension of the commonly used format for plain Datalog. A file may contain four kinds of knowledge elements: facts, existential rules, negative constraints and conjunctive queries. This format will allow us to easily exchange data and ontologies with groups working on the equivalent Datalog+/- formalism, developed in Oxford.

A DLGP parser is now available.

For this section, participants are listed in alphabetical order.

INDES Project-Team

5. Software

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: Frédéric Boussinot [Inria], Cyprien Nicolas [Inria], Bernard Serpette [Inria], Manuel Serrano [correspondant].

5.2.1. *The Bigloo compiler*

The programming environment for the Bigloo compiler [5] 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.

5.2.2. *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.

5.3. Web programming

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

5.3.1. The HOP web programming environment

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

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

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

HOP has won the software *open source contest* organized by the ACM Multimedia Conference 2007 <http://mmc36.informatik.uni-augsburg.de/acmmm2007/>. It is released under the GPL license. It is available at <http://hop.inria.fr>.

5.4. Language-based security

Participants: Zhengqin Luo [Inria], Tamara Rezk [correspondant].

5.4.1. 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 <http://www.msr-inria.inria.fr/projects/sec/cflow/index.html>.

5.4.2. 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 <http://www.msr-inria.inria.fr/projects/sec/cflow/index.html>.

5.4.3. 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/>.

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>.

LAGADIC Project-Team

5. Software

5.1. ViSP: a visual servoing platform

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

Since 2005, we develop and release under the terms of the GPLv2 licence, ViSP, an open source library that 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].

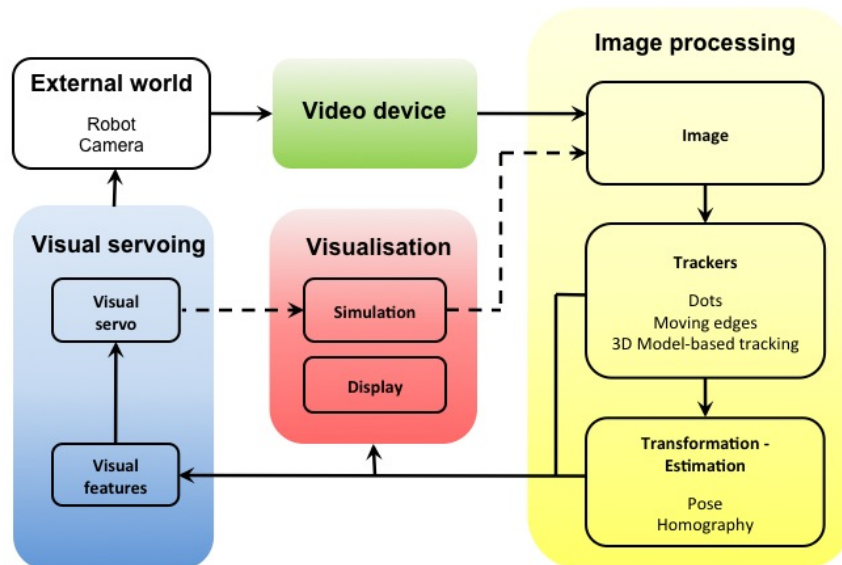


Figure 1. ViSP software architecture.

This year, we continued our efforts to improve the software and documentation quality. A new version available at <http://www.irisa.fr/lagadic/visp/visp.html> was released in July 2012. To ease ViSP installation, we provide also precompiled ViSP SDK including pre-built ViSP library and headers.

This last release under deposit to the APP (“Agence de Protection des Programmes”) has been downloaded 887 times since its availability. It 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 a graduate course delivered at MIT, at IFMA Clermont-Ferrand and ESIR Rennes engineer schools. ViSP is now also part of “vision_visp” ROS stack (see http://www.ros.org/wiki/vision_visp) and ViSP 3D model-based tracker has been proposed by colleagues from Laas in Toulouse as a ROS package. This encouraged us to enhance “vision_visp” stack by proposing new ROS packages to calibrate intrinsic and extrinsic camera parameters, and a new 3D model-based tracker with automatic initialisation and reinitialisation after tracking loss (with help of specific textured patterns on the object).

5.2. DESlam

Participants: Patrick Rives [correspondant], Maxime Meilland.

The DESlam (Dense Egocentric Slam) software developed in collaboration with Andrew Comport from I3S in Sophia Antipolis was deposited 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. Development work: 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.

Two papers published by Lagadic in 2012 enclose results validated on this platform. Note that it is also opened to researcher from other labs. For example, this year an associate professor from LSIIT in Strasbourg did experiments on the Gantry robot.

5.4. Development work: Medical robotics platforms

Participants: Fabien Spindler [correspondant], Alexandre Krupa.

This tesbed 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 a six degrees of freedom Adept Viper S850 arm (see Fig. 3). This year we bought a new Adept Viper S650 arm to replace our eight year old Hippocrate medical arm designed by the Sinters company. 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 plan to exploit the two Viper robots for demonstrating needle insersion under ultrasound imaging to precisely guide the needle toward a target while optimizing its visibility (see Section 6.4.4).

Note that four papers published by Lagadic in 2012 enclose experimental results obtained with this platform.

5.5. Development work: Mobile robotics platforms

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

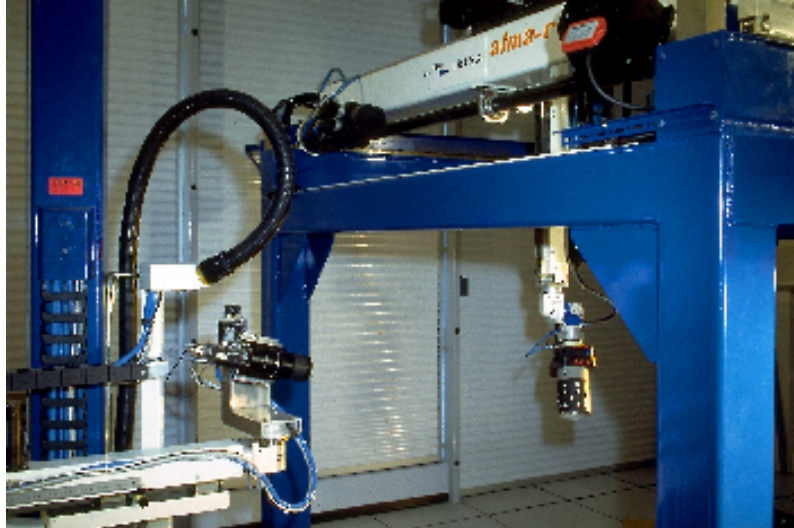


Figure 2. Lagadic robotics platforms for vision-based manipulation



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.

5.5.1. Indoors mobile robots

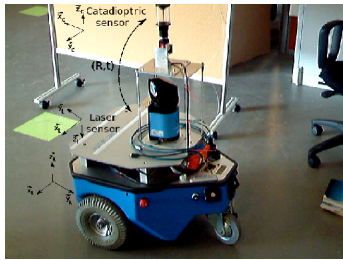
For fast prototyping of algorithms in perception, control and autonomous navigation, the team uses Hannibal in Sophia Antipolis, a cart-like platform built by Neobotix (see Fig. 4 .a), and a Pioneer 3DX from Adept in Rennes (see Fig. 4 .b) as well as a Robotino from Festo. These platforms are equipped with various sensors needed for Slam purposes, autonomous navigation and sensor-based control.

Moreover, to validate the researches in personally assisted living topic (see 6.3.6), 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 three 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.

5.5.2. Outdoors mobile robots

The team exploit 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.

Note that 5 papers published by Lagadic in 2012 enclose experimental results obtained with these mobile robotics platforms.



(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

4. Software

4.1. 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.

4.2. myMed backbone

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

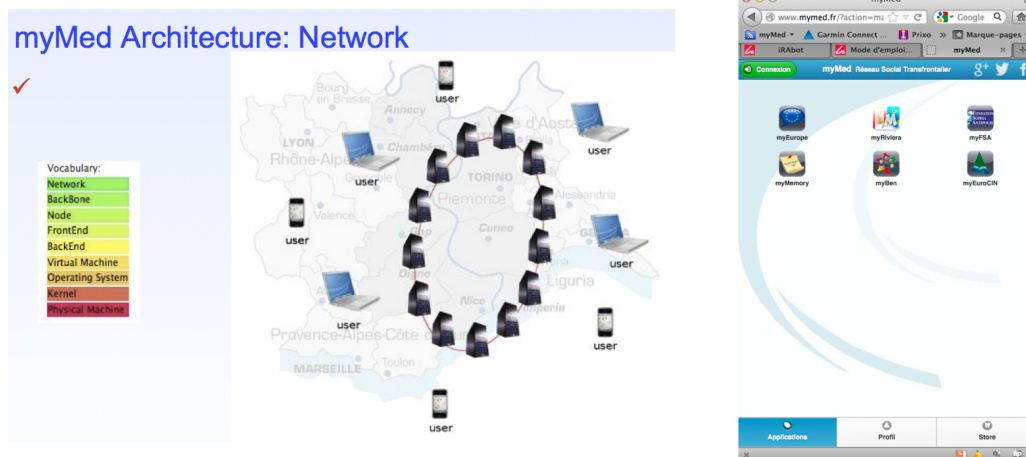


Figure 4. The myMed backbone and the myMed LaunchPad

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.

4.3. 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.

4.4. Synapse simulator in Oversim

Participant: Vincenzo Ciancaglini [contact].

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.

4.5. 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.

4.6. CCN-TV Omnet++ simulator

Participant: Vincenzo Ciancaglini [contact].

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.

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

Participant: Hoang Giang Ngo [contact].

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.

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

Participant: Hoang Giang Ngo [contact].

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.

4.9. Reputation Computation Engine for Social Web Platforms

Participants: Thao Nguyen [contact], Laurent Vanni.

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 N_u users, providing the same service, and undertaking N_t 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. $R_r = E_p$. DISHONEST raters provide ratings 0:5 different from their true estimation, i.e. $R_r = E_p \pm 0:5$. COLLUSIVE raters give the highest ratings ($R_r = 1$) to users in their collusion and the lowest ratings ($R_r = 0$) 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 N_t 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].

4.10. 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].

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 .

See the web page <http://www-sop.inria.fr/members/Luigi.Liquori/ARIGATONI/Ariwheels.htm> and <http://arigt.altevista.org>.

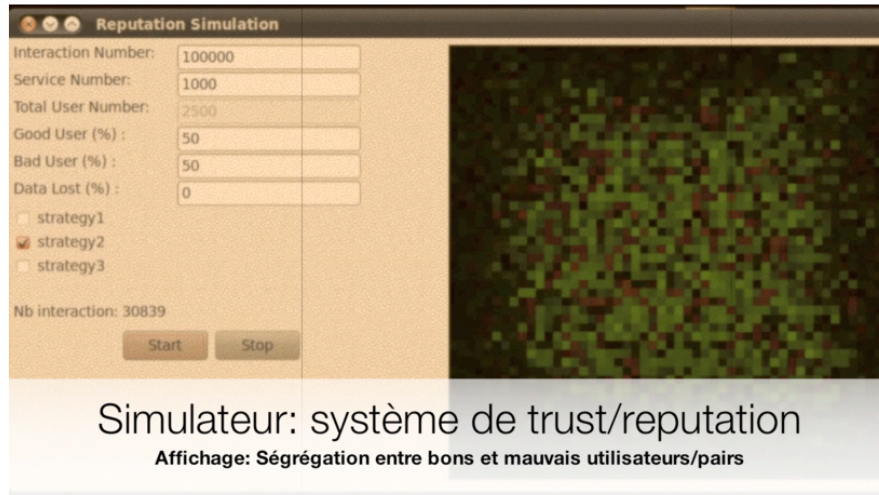


Figure 5. A prototype graphical GUI for the Reputation Computation Engine

4.11. Arigatoni simulator

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

We have implemented in C++ (~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.

4.12. Synapse client

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

In order to test our Synapse protocol [17] 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/>.

4.13. Open Synapse client

Participant: Bojan Marinkovic [contact].

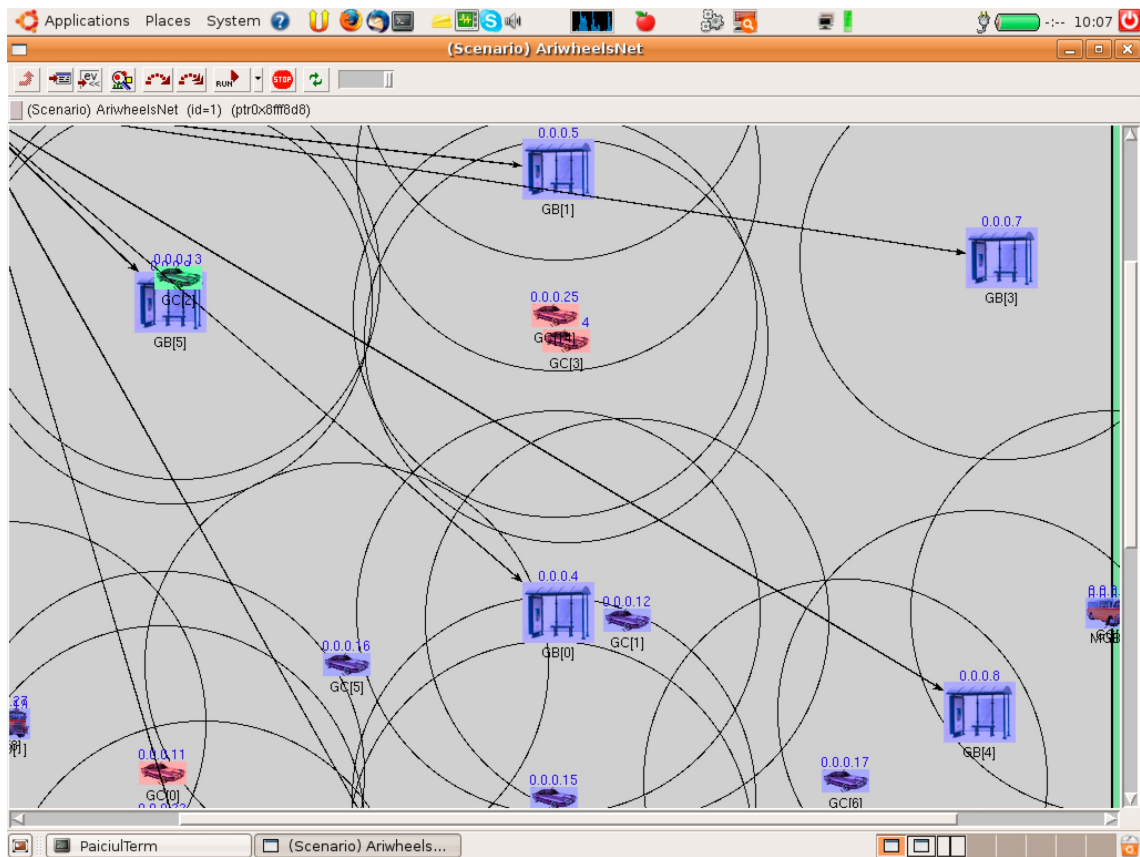


Figure 6. The Ariwheels simulator in Omnet

```

mascotte:~/ARIGATONI/CODE - Konsole
Session Edit View Bookmarks Settings Help

./a.out: /usr/local/lib/libstdc++.so.6: no version information available (required by ./a.out)
./a.out: /usr/local/lib/libstdc++.so.6: no version information available (required by ./a.out)
-t topology_1.net -s 3 -x 0 -r 0 -a 750 -S 128 -p 1234 -N 10000 -H 1 -g 100 -I 10
100%
ALL CORRECT
AVERAGE # PEERS INVOLVED PER REQUEST (GBUs ONLY | GBUs & GCUs): 19.7828 26.6448
AVERAGE # REQ MESSAGES PER REQUEST (GBUs ONLY | GBUs & GCUs): 20.6098 27.4718
AVERAGE # MESSAGES (ALL TYPES) PER REQUEST (GBUs ONLY | GBUs & GCUs): 68.0267 74.8887
MEAN # REQUESTS FOR DEPTH 0: 10000
MEAN # REQUESTS FOR DEPTH 1: 4131
MEAN # REQUESTS FOR DEPTH 2: 2067.71
MEAN # REQUESTS FOR DEPTH 3: 868.589
MEAN # REQUESTS FOR DEPTH 4: 506.055
MEAN # REQUESTS FOR DEPTH 5: 404.25
MEAN # REQUESTS FOR DEPTH 6: 313.842
MEAN # REQUESTS FOR DEPTH 7: 188.875
MEAN # REQUESTS FOR DEPTH 8: 0
MEAN: 951.096 MIN: 0 MAX: 10000 (ID2) STD DEV: 1625.7 MEDIAN: 43
ACCEPT RATE: 0.513175 REJECT RATE: 0.486825 UNARY ACCEPT RATE: 0.991244 UNARY REJECT RATE: 0.00875567
$

```

Figure 7. The Arigatoni simulator

Opensynapse is an open source implementation of [17]. 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 [16], and so, each new Chord network is called a *Floor*.

4.14. Husky interpreter

Participants: Marthe Bonamy [contact], Luigi Liquori.

Husky is a variable-less language based on lambda calculus and term rewriting systems. Husky is based on the version 1.1 of *Snake* [13]. 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.

4.15. 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.

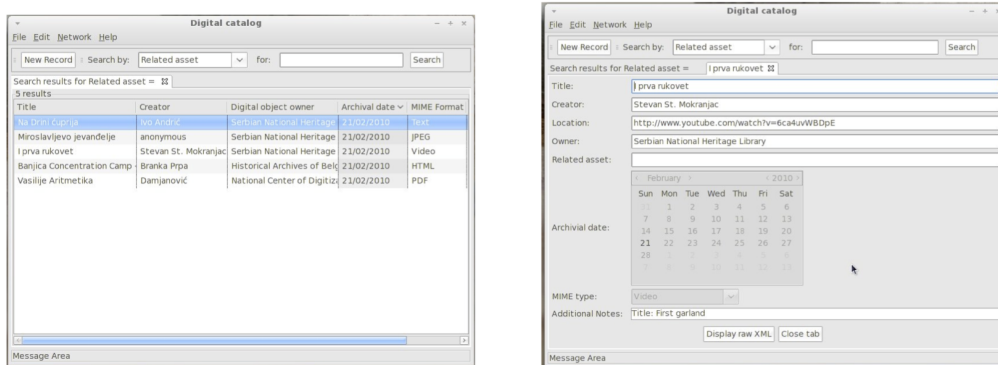


Figure 10. myDistributed Catalog

4.16. myDistributed Catalog for Digitized Cultural Heritage

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

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.

4.17. 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.

MAESTRO Project-Team (section vide)

MARELLE Project-Team

4. Software

4.1. 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 this year. Some features have been added, and some bugs corrected.

4.2. 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/v8.3>.

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

4.3. Certicrypt and Easycrypt

Participants: Gilles Barthe [IMDEA Software Institute], Juan Manuel Crespo [IMDEA Software Institute], Benjamin Grégoire [correspondant], Sylvain Heraud, César Kunz [IMDEA Software Institute], Federico Olmedo [IMDEA Software Institute], Santiago Zanella Béguelin [IMDEA Software Institute].

CertiCrypt takes a language-based approach to cryptography: the security of a cryptographic scheme and the cryptographic assumptions upon which its security relies are expressed by means of probabilistic programs, called games; in a similar way, adversarial models are specified in terms of complexity classes, e.g. probabilistic polynomial-time programs. This code-centric view leads to statements that are amenable to formalization and tool-assisted verification. CertiCrypt instruments a rich set of verification techniques for probabilistic programs, including equational theories of observational equivalence, relational Hoare logic, data-flow analysis-based program transformations, and game-based techniques such as eager/lazy sampling and failure events.

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

MASCOTTE Project-Team

5. Software

5.1. Grph

Participants: David Coudert, Luc Hogue [correspondant], Aurélien Lancin, Grégory Morel, Issam Tahiri.

Around 20,000 lines of code, developed in Java.

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 base graph algorithms which are regularly augmented.

So far, known users of the GRPH library include people at Mascotte and others involved in the FP7 EULER project. It got some contribution from the Inria team GANG who contributed GRPH with an implementation of the four-sweep algorithm which provides accurate lower bound on the diameter in linear time. It has a number of other academic users including research students at Bergamo University (Italy), and University of Southern Denmark (students supervised by Jørgen Bang-Jensen).

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), as well as specific algorithms developed by Matthieu Latapy and Jean-Lou Guillaume (LIP6), etc.

GRPH is distributed under the terms of a license defined by its contributors and is available for download. This license allows free usage and access to the source code. See <http://www-sop.inria.fr/mascotte/software/grph>.

In 2012, numerous graph algorithms have been added to GRPH, such as maximum matching, minimum vertex cover (brute force, branching, Niedermeier), maximum independent set (Fomin/Grandoni/Kratsch). Furthermore, to answer a number of issues about the generation of graph instances with particular properties, a framework for evolutionary computing dedicated to graphs was integrated to GRPH. Moreover, a reworked version of Mascsim was integrated in GRPH.

On-going works concern the distributed execution of graph algorithms, and a bridge to Sage.

See also the web page <http://www-sop.inria.fr/mascotte/software/grph/>.

5.2. DRMSim

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

Around 45,000 lines, developed in Java, collaboration between MASCOTTE and LaBRI.

DRMSim relies on a discrete-event simulation engine aiming at enabling the large-scale simulations of routing models. DRMSim is developed 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 the routing protocol proposed in [37] and lightweight versions of BGP (Border Gateway Protocol).

The system model considers the dynamic evolution of the simulated network. This model takes as its input parameter the distribution of failure probability for both routers and links.

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.

Finally, from an object-oriented point of view of its conception model, DRMSim manipulates graph abstractions, allowing the user to force the use of a library different from the default one, i.e. GRPH.

See also the web page <http://www-sop.inria.fr/mascotte/projets/DCR/>.

5.3. SageMath

Participants: David Coudert, Leonardo Sampaio.

Developed in Python, Cython, and C++. MASCOTTE members have already contributed to the development of more than 180 patches and to the reviewing process of more than 200 patches that are now part of the standard distribution.

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 has reached 350 MB. It is of interest for Mascotte 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 IFI, stream UBINET.

In 2012, David Coudert has contributed to the development of the Sage releases 5.0 to 5.6 with 15 patches (from bug fix to advance graph algorithms) and participated to the reviewing process of more than 30 patches.

5.4. Utilities

5.4.1. Java4unix

Participant: Luc Hogie [correspondant].

More than 5,000 lines, developed in Java.

Java4unix proposes a development and distribution framework which simplifies the use of Java for UNIX software programming/distribution. Until now, Java could hardly be used for the development UNIX applications because invoking Java applications from the UNIX shell must be done through an explicit call to the Java virtual machine and writing simple things in Java often requires long coding. Java4unix aims at filling those two gaps by providing a UNIX installer for java applications, turning them to standard UNIX application and a framework that UNIX programmers may use to manipulate files/text, etc.

Java4unix includes a module which enables the reporting and automatic releasing of Eclipse Java projects.

See also the web page <http://www-sop.inria.fr/members/Luc.Hogie/java4unix/>.

5.4.2. Jalinopt

Participants: Luc Hogie [correspondant], Grégory Morel.

Developed in Java.

Jalinopt is a Java toolkit for building and solving linear programs. It consists of a straightforward object-oriented model for linear programs, as well as a bridge to most common solvers, including GLPK and CPLEX. It is an interface to many LP solvers allowing users to code independently of the solver effectively. Although Jalinopt is inspired by Mascopt and JavaILP, it provides a significantly different model and an utterly different approach to connecting to the solver. In particular this approach, based in inter-process piping, offers better portability, and the possibility to connect (via SSH) to solvers on remote computers.

In 2012, we refined the object-oriented model of Jalinopt and improved its portability by making it working with LPSolve as its default native solver.

See also the web page <http://www-sop.inria.fr/members/Luc.Hogie/jalinopt/>.

5.4.3. *JavaFarm*

Participant: Luc Hogie [correspondant].

More than 1,500 lines, developed in Java.

JavaFarm is a middleware enabling the distribution of Java applications across farms of servers.

Its workflow basically enables an application to locally aggregate code and data into an object, called job, that will migrate to another computer where it will be computed. When a job completes, its result is transferred back to the caller. Among other features, JavaFarm supports futures (asynchronous job executions), thereby enabling parallelization of the distributed code. The design objectives of JavaFarm are to make distribution and parallelism as transparent and easy as possible.

See also the web page <http://www-sop.inria.fr/members/Luc.Hogie/javafarm/>.

5.4.4. *Mascsim*

Participants: Luc Hogie [correspondant], Aurélien Lancin, Issam Tahiri.

Around 12,000 lines, developed in Java.

Mascsim is a distributed discrete event simulator whose main target is to be easy to use. Unlike most discrete-event simulators, the researcher who is using Mascsim is required to provide only the bare minimum material needed for the simulation: a model for the system, a set of events describing what is going on in the system, as well as a set of metrics of interest. The simulation process is then entirely automatized.

In 2012, Mascsim was adapted and integrated to GRPH.

See also the web page <http://www-sop.inria.fr/mascotte/software/mascsim/>.

5.4.5. *P2PVSIm*

Participant: Remigiusz Modrzejewski [correspondant].

Around 12,000 lines, developed in Python.

P2PVSIm is a simple 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.

In 2012, a distributed version of P2PVSIm was developed. The objectives for developing a distributed version was to fasten the simulation of large campaigns, that would be too long to run on one single computer. The distributed P2PVSIm runs 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.

MCTAO Team (section vide)

MODEMIC Project-Team

5. Software

5.1. VITELBIO

Participants: Jérôme Harmand, Alain Rapaport.

VITELBIO (Virtual TELluric BIOreactors) is a simulation tool for studying networks of interconnected chemostats with the objective of mimicking microbial activities in soil. The software, developed with the help of ITK Company, is accessible on a server from any web navigator and make use of Flex for the user interface and Octave for the numerical integration. An important effort has been made for obtaining a pleasant and easy interface that is appealing for microbiologists: the network can be drawn graphically on the screen and simulation results can be easily compared between (virtual) experiments, superposing trajectories curves.

This software is used by several researchers, from LBE (INRA Narbonne), UMR Eco & Sols (Montpellier), UREP (Unité de Recherche sur l'Ecosystème Prairial, INRA Theix), Biomeco (Paris-Grignon), UMR EGC (Environnement et grandes cultures, Paris-Grignon)... and also as a teaching support. Vitelbio is presented at <http://sites.google.com/site/vitelbio/> and it is accessible at <http://vitelbio.itkweb.fr>.

5.2. SMC DEMOS

Participant: Fabien Campillo.

SMC DEMOS (Sequential Monte Carlo 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), available at <http://www-sop.inria.fr/members/Fabien.Campillo/software/smc-demos/>.

MORPHEME Team

4. Software

4.1. Software

4.1.1. Deposits

The software MAD V2.0 was deposited with the APP in November 2012. It deals with the melasma severity scoring from multi-spectral imaging.

4.1.2. Transfers

The software MAD V2.0 was transferred to Galderma R&D.

NACHOS Project-Team

5. Software

5.1. MAXW-DGTD

Participants: Stéphane Lanteri [correspondant], Loula Fezoui.

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 [14]. 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 [27].

- 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

Participant: Stéphane Lanteri [correspondant].

MAXW-DGFD is a software suite for the simulation of time harmonic electromagnetic wave propagation. It implements a solution method for the Maxwell equations in the frequency domain. MAXW-DGFD is based on a discontinuous Galerkin method formulated on unstructured triangular (2D case) or tetrahedral (3D case) meshes. Within each element of the mesh, the components of the electromagnetic field are approximated by a arbitrary high order nodal polynomial interpolation method. The resolution of the sparse, complex coefficients, linear systems resulting from the discontinuous Galerkin formulation is performed by a hybrid iterative/direct solver whose design is based on domain decomposition principles. The software and the underlying algorithms are adapted to distributed memory parallel computing platforms thanks to a parallelization strategy that combines a partitioning of the computational domain with a message passing programming using the MPI standard. Some recent achievements have been the implementation of non-uniform order DG method in the 2D case [18] and of a new hybridizable discontinuous Galerkin (HDG) formulation also in the 2D case [22].

- 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 [5]. Within each element of the mesh, the components of the electromagnetic field are approximated by an arbitrary high order nodal polynomial interpolation method. This discontinuous Galerkin method combines a centered scheme for the evaluation of numerical fluxes at a face shared by two neighboring elements, with an explicit Leap-Frog time scheme. The software and the underlying algorithms are adapted to distributed memory parallel computing platforms thanks to a parallelization strategy that combines a partitioning of the computational domain with a message passing programming using the MPI standard.

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

5.4. NUM3SIS

Participants: Nora Aissiouene, Thibaud Kloczko [SED³ 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].

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 tetrahedrization. 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 multidisciplinary software toolchain. Using both DTK <http://dtk.inria.fr> and NUM3SIS SDKs, Medical Image Extractor comes within NUM3SIS' framework. Using cutting edge research algorithms developed 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.

³Service d'Experimentation et de Développement

NEUROMATHCOMP Project-Team

4. Software

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

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

Virtual Retina is a simulation software developed by Adrien Wohrer during his PhD [79], [78] 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 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. Other evolutions of Virtual Retina are also investigated by external partners like the role/implementation of starburst amacrine cells involved in direction selectivity (collaboration with Universidad Técnica Federico Santa María, Valparaíso, Chile, and Centro de urociencia de Valaparaiso) (see also e.g., [70]).

- 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 [correspondent], Sélim Kraria [Inria DREAM], Olivier Marre [Institut de la vision, Paris], Hassan Nasser, Thierry Viéville [Inria Mnemosyne Bordeaux].

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.

It is designed mainly as

- An existing simulator plug-in (e.g. MVASpike or other simulators via the NeuralEnsemble meta-simulation platform),
- Additional modules for computations with neural unit assembly on standard platforms (e.g. Python, Matlab or the Scilab platform).
- Original modules for the analysis of spike train statistics intended to be used by the neuroscientists community.

Achievements include:

- Spike trains statistical analysis via Gibbs distributions. They are based on the estimation of 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?
- Spiking network programming for exact event's sequence restitution;
- Discrete neural field parameters algorithmic adjustments and time-constrained event-based network simulation reconciling clock and event based simulation methods.

Compared to existing libraries 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. The standard algorithms are based on the best free libraries in the domain such as gsl <http://www.gnu.org/software/gsl>.

Event neural assembly simulation is developed in gForge. It is under CeCILL C licence
APP logiciel Enas: IDDN.FR.001.360008.000.S.P.2009.000.10600.

Its development as a friendly software designed for the neuroscience community is our present purpose. This is done with the support of an ADT Inria.

Website: <http://enas.gforge.inria.fr/>

OASIS Project-Team

5. Software

5.1. ProActive

Participants: F. Baude, D. Caromel, L. Henrio, F. Huet [correspondant], F. Viale, O. Smirnov, B. Sauvan, A. Bourdin.

Proactive Parallel Suite

ProActive 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 exportation (Apache Axis2 and CXF),
- HTTP transport,
- ssh, rsh, RMI/ssh tunnelling,
- Globus: GT2, GT3, GT4, gsi, Unicore, ARC (NordGrid)
- LSF, PBS, Sun Grid Engine, OAR, Load Leveler

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 [5] [55] and to the web pages <http://www.objectweb.org> and <http://proactive.inria.fr/> which list several white papers.

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.

5.2. Vercors platform

Participants: E. Madelaine, L. Henrio, A. Savu, M. Alexe.

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 pursued last year experiments in distributed model-checking, and were able to generate explicit state-spaces of (sub-systems) for a new distributed use-case of several billion states. The challenges here lie in the structure of the verification workflow, and in finding strategies for separating the sub-systems in an intelligent way.
- We have also conducted intensive experiments within the Papyrus environment, aiming at the definition of a graphical specification environment combining some of the standard UML formalisms (typically class diagrams and state-machines), with a dedicated graphical formalism for the architecture of GCM components.

5.3. Open Simulation Architecture (OSA)

Participants: O. Dalle, V.D. Nguyen.

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

Participant: F. Hermenier.

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 objectives. Its extensibility is possible thanks to a composable core reconfiguration algorithm implemented using Constraint Programming.

Btrplace is a part of the OW2 project Entropy. It has been originally developed by Fabien Hermenier during its PhD at the Ecole des Mines of Nantes. BtrPlace is now a standalone project that is currently used to address fault tolerance, isolation, infrastructure management, performance, and energy efficiency concerns inside the national project OpenCloudWare (<http://opencloudware.org/>) and the European project Fit4Green.

This year, our development has been guided by our collaborations. The Fit4Green project chose to rely on BtrPlace to compute an energy-efficient placement for their VMs while some partners inside OpenCloudWare required new placement constraints. The inferring capabilities of BtrPlace and its catalog of placement constraint have then been upgraded accordingly.

OPALE Project-Team

5. Software

5.1. NUM3SIS

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

NUM3SIS (<http://num3sis.inria.fr>) 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 (CFD), Computational Electro-Magnetics (CEM, in collaboration with Nachos Project-Team) and pedestrian traffic simulation.

The most important concept in NUM3SIS is the concept of node. It is a visual wrapper around derivatives of fundamental concepts such as data, algorithm or viewer. Atomic nodes are provided for convenience in order to manipulate computational data (such as grids or fields), apply computational methods (such as the building of a finite-element matrix or the construction of a finite-volume flux) and visualize computational results (such as vector or tensor fields, on a screen or in an immersive space). For a given abstract node, different implementations can be found, each of them being embedded in a plugin system that is managed by a factory.

The second important concept in NUM3SIS is the concept of composition. It consists of the algorithmic pipeline used to link the nodes together. The use of these two concepts, composition and nodes, provides a highly flexible, re-usable and efficient approach to develop new computational scenarii and take benefit from already existing tools. This is a great advantage with respect to classical monolithic softwares commonly used in these fields.

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 in multidisciplinary design optimization (MDO). The platform is also used by the Fluid Mechanics Laboratory at Ecole Centrale de Nantes and by the K-Epsilon company (<http://www.k-epsilon.com>) for hydrodynamic design applications. Moreover, it is presently tested by Peugeot Automotive industry for external aerodynamic design purpose.

5.3. Plugins for AXEL

Participants: Régis Duvingneau [correspondant], Louis Blanchard.

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

- methods for 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 ;
- methods for geometrical modeling of bow shapes for trawler ships.

5.4. Integration platform for multidiscipline optimization applications

Participants: Toan Nguyen, Laurentiu Trifan.

A prototype software integration platform is developed and tested for multidiscipline optimization applications. It is based on a workflow management system called YAWL (<http://www.yawlfoundation.org>). The goal is to design, develop and assess high-performance distributed scientific workflows featuring resilience, i.e., fault-tolerance and exception-handling capabilities. The platform is used to experiment new resilience algorithms, including monitoring and management of application-level errors. Errors include time-outs and out of bounds data values. They can be added and modified by the users. The platform is tested against use-cases provided by the industry partners in the OMD2 project supported by the French Agence Nationale de la Recherche. For example, an optimization of a car air-conditioning pipe was implemented and deployed on the Grid5000 infrastructure. It also takes into account run-time errors related to resource consumption, e.g., memory overflow, to automatically and dynamically relocate the applications tasks involved on the various clusters. This work is part of Laurentiu Trifan's PhD thesis that is to be defended in 2013. (See Fig. 1 .)

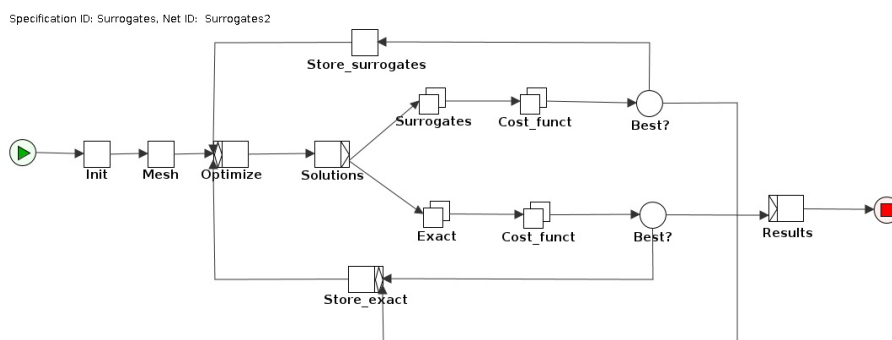


Figure 1. Testcase deployment on the Grid5000 infrastructure.

PLANETE Project-Team

5. Software

5.1. ns-3

Participant: Daniel Camara [correspondant].

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-3.7
- 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

5.2. EphPub

Participants: Mohamed Ali Kaafar [correspondant], Claude Castelluccia.

EphPub (Ephemeral Publishing) (previously called EphCom) implements a novel key storage mechanism for time-bounded content, that relies on the caching mechanism of the Domain Name System (DNS). Features of EphPub include: EphPub exploits the fact that DNS servers temporarily cache the response to a recursive DNS query for potential further requests. EphPub provides higher security than Vanish, as it is immune to Sybil attacks. EphPub is easily deployable and does not require any additional infrastructure, such as Distributed Hash Tables. EphPub comes with high usability as it does not require users to install and execute any extra additional software. EphPub lets users define data lifetime with high granularity. We provide EphPub as an Android Application to provide ephemeral exchanged SMS, emails, etc. and as a Firefox or Thunderbird extensions so as to support ephemeral publication of any online document.

For more details about the different software products, see <http://planete.inrialpes.fr/projects/ephemeral-publication/>.

- Version: v0.1.2-beta
- ACM: K.4.1
- AMS: 94Axx
- Keywords: Ephemeral communications, Right to Forget, Future Internet Architecture, Privacy
- Software benefit: We provide a Firefox Extension that easily allows users to manage disappearing emails. We also provide a command-line tool to manage disappearing files.
- APP: Under APP deposit internal process
- License: GPL
- Type of human computer interaction: Firefox extension + Unix Console
- OS/Middelware: Firefox under any OS
- Required library or software: Python Ext
- Programming language: Python
- Documentation: No detailed documentation has been released so far. A detailed howto can be consulted however at: http://code.google.com/p/disappearingdata/source/browse/wiki/EphCOM_Firefox_Extension.wiki?r=77

5.3. Username Tester

Participants: Claude Castelluccia [correspondant], Mohamed Ali Kaafar, Daniele Perito.

Usernames are ubiquitous on the Internet. Almost every web site uses them to identify its users and, by design, they are unique within each service. In web services that have millions or hundreds of millions of users, it might become difficult to find a username that has not already been taken. For instance, you might have experienced that a specific username you wanted was already taken. This phenomenon drives users to choose increasingly complex and unique usernames.

We built a tool to estimate how unique and linkable usernames are and made it available on this page for you to check. For example, according to our tool, “ladygaga” or “12345678” only carry 24 and 17 bits of entropy, respectively. They are therefore not likely to be unique on the Internet. On the other hand, usernames such as “pdjkwelr!” or “yourejerky” carry about 40 bits of entropy and are therefore very good identifiers.

Type your username (for example “zorro1982” or “dan.perito”) to discover how unique it is. This tool can help you to select an username that has low entropy and can’t be used to track you on the Internet.

Alternatively, try typing two usernames separated by a space. The tool will give an estimation on whether the two usernames are linkable. The tool is accessible here: <http://planete.inrialpes.fr/projects/how-unique-are-your-usernames/>

5.4. DroidMonitor

Participants: Claude Castelluccia [correspondant], Mohamed Ali Kaafar.

In nowadays world the technological progress evolves very quickly. There are more and more new devices, fully equipped with the latest innovations. The question is: do we adopt our main privacy concerns according to these new technologies as quickly as they grow and become widely available for us?...

We developed a novel tool, private data leakage monitoring tool, DroidMonitor. It aims to serve as an educational tool for regular Android Smartphones users to make them aware of existing privacy threats while they are using Location-Based Services. It can be downloaded here: <http://planete.inrialpes.fr/android-privacy/>

5.5. NEPI

Participants: Thierry Turletti [correspondant], Alina Quereilhac.

NEPI stands for Network Experimentation Programming Interface. NEPI implements a new experiment plane used to perform ns-3 simulations, planetlab and emulation experiments, and, more generally, any experimentation tool used for networking research. Its goal is to make it easier for experimenters to describe the network topology and the configuration parameters, to specify trace collection information, to deploy and monitor experiments, and, finally, collect experiment trace data into a central datastore. NEPI is a python API (with an implementation of that API) to perform all the above-mentioned tasks and allows users to access these features through a simple yet powerful graphical user interface called NEF.

During the year 2012 we improved support for PlanetLab experiments in NEPI, adding the ability to create customized routing overlays on top of PlanetLab. Details on these improvements can be found in [48]. We also included the ability to easily conduct CCNx <http://www.ccnx.org/> experiments using PlanetLab nodes. This work was presented at the CCNx 2012 community meeting [73], and has had a good impact on the number of NEPI users.

Additionally, ongoing work on the context of the Openlab, Fed4Fire and Simulbed projects, has lead to a number of interesting extensions to NEPI. We are currently developing support to conduct experiments on OMF wireless testbeds (<http://mytestbed.net/>). We are also working to support DCE enabled experimentation, using the ns-3 simulator, in NEPI. Furthermore, recent work on improving NEPI’s experiment control architecture, to enable both easier extension to new experimentation platforms and improve the user ability to control of experiment tasks, was presented at the CoNEXT’12 Students Workshop (see [61]).

For more information, see also the web page <http://nepi.inria.fr>.

- Version: 2.0
- ACM: C.2.2, C.2.4
- Keywords: networking experimentation
- License: GPL (2)
- Type of human computer interaction: python library, QT GUI
- OS/Middleware: Linux
- Required library or software: python – <http://www.python.org> – <http://rpyc.sourceforge.net>
- Programming language: python

5.6. Reference implementation for SFA Federation of experimental testbeds

Participants: Thierry Parmentelat [correspondant], Julien Tribino.

We are codevelopping with Princeton University a reference implementation for the Testbed-Federation architecture known as SFA for Slice-based Federation Architecture. During 2011 we have focused on the maturation of the SFA codebase, with several objectives in mind, better interoperability between the PlanetLab world and the EmuLab, a more generic shelter that other testbeds can easily leverage in order to come up with their own SFA-compliant wrapper and support for 'reservable' mode, which breaks the usual best-effort PlanetLab model. For more details about this contribution see section

See also the web page <http://planet-lab.eu>

- Version: myplc-5.0-rc26
- Keywords: networking testbed virtual machines
- License: Various Open Source Licences
- Type of human computer interaction: Web-UI, XMLRPC-based API, Qt-based graphical client
- OS/Middleware: Linux-Fedora
- Required library or software: Fedora-14 for the infrastructure side; the software comes with a complete software suite for the testbed nodes
- Programming languages: primarily python, C, ocaml
- Documentation: most crucial module plcapi is self-documented using a local format & related tool. See e.g. <https://www.planet-lab.eu/db/doc/PLCAPI.php>
- Codebase: <http://git.onelab.eu>

5.7. 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 codevelopping the SfaWrap with Princeton University, and during 2012, we have focused on:

- Participating in the discussions about the future and evolutions of the architecture of SFA, as part of the architecture working group of the GENI project.
- Turning this initially Planet-Lab specific implementation into a generic one, that testbed providers can easily leverage for bringing SFA-compliance to their own testbeds.
- Supporting the allocation and provisioning of both 'Exclusive' and 'Shared' testbed resources.
- Enlarging the federation scheme by federating various testbeds with heterogeneous resources, in order to allow researchers to combine all available resources and run advanced networking experiments of significant scale and diversity.

- Version: sfa-2.1-22, myplc-5.0-rc33
- Keywords: networking testbed federation
- License: Various Open Source Licenses
- Type of human computer interaction: Web-UI, XMLRPC-based API, Qt-based graphical client
- OS/Middleware: Linux
- Required library or software: python2.5 or superior
- Programming languages: python
- Documentation: <http://svn.planet-lab.org/#SFAUser-leveldocumentation>
- Codebase: <http://git.onelab.eu/?p=sfa.git;a=summary>

5.8. MultiCast Library Version 3

Participant: Vincent Roca [correspondant].

MultiCast Library Version 3 is an implementation of the ALC (Asynchronous Layered Coding) and NORM (NACK-Oriented Reliable Multicast Protocol) content delivery Protocols, and of the FLUTE/ALC file transfer application. This software is an implementation of the large scale content distribution protocols standardized by the RMT (Reliable Multicast Transport) IETF working group and adopted by several standardization organizations, in particular 3GPP for the MBMS (Multimedia Broadcast/Multicast Service), and DVB for the CBMS (Convergence of Broadcast and Mobile Services). Our software is used in operational, commercial environments, essentially in the satellite broadcasting area and for file delivery over the DVB-H system where FLUTE/ALC has become a key component. See <http://planetec-bcast.inrialpes.fr/> for more information.

5.9. OpenFEC.org: because open, free AL-FEC codes and codecs matter

Participants: Vincent Roca [correspondant], Jonathan Detchart [engineer], Ferdaouss Mattoussi [PhD student].

The goals of the OpenFEC.org <http://openfec.org> are:

1. to share IPR-free, open, AL-FEC codes,
2. to share high performance, ready-to-use, open, free, C-language, software codecs
3. to share versatile and automated performance evaluation environments.

This project can be useful to users who do not want to know the details of AL-FEC schemes but do need to use one of them in the software they are designing, or by users who want to test new codes or new encoding or decoding techniques, and who do know what they are doing and are looking for, or by users who need to do extensive tests for certain AL-FEC schemes in a given use-case, with a well defined channel model.

5.10. BitHoc

Participants: Chadi Barakat [correspondant], Thierry Turletti.

BitHoc (BitTorrent for wireless ad hoc networks) enables content sharing among spontaneous communities of mobile users using wireless multi-hop connections. It is an open source software developed under the GPLv3 licence. A first version of BitHoc has been made public. We want BitHoc to be the real testbed over which we evaluate our solutions for the support and optimization of file sharing in a mobile wireless environment where the existence of an infrastructure is not needed. The proposed BitHoc architecture includes two principal components: a membership management service and a content sharing service. In its current form it is composed of PDAs and smartphones equipped with WIFI adapters and Windows Mobile 6 operating system.

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

- Version: 1.2
- Keywords: Tracker-less BitTorrent for mobile Ad Hoc networks
- License: GPL (GPLv3)
- Type of human computer interaction: Windows Mobile 6 GUI
- OS/Middleware: Windows Mobile 6
- Required library or software: OpenSSL(<http://www.openssl.org/>, GPL), C++ Sockets, GPL)
- Programming languages: C++, C#
- Documentation: doxygen

5.11. TICIP

Participant: Chadi Barakat [correspondant].

TICIP is a TCP-friendly reliable transport protocol to collect information from a large number of network entities. The protocol does not impose any constraint on the nature of the collected information: availability of network entities, statistics on hosts and routers, quality of reception in a multicast session, weather monitoring, etc. TICIP ensures two main things: (i) the information to collect arrives entirely and correctly to the collector where it is stored and forwarded to upper layers, and (ii) the implosion at the collector and the congestion of the network are avoided by controlling the rate of sending probes. The congestion control part of TICIP is designed with the main objective to be friendly with applications using TCP. Experimental results show that TICIP can achieve better performance than using parallel TCP connections for the data collection. The code of TICIP is available upon request, it is an open source software under the GPLv3 licence.

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

- Version: 1.0
- Keywords: Information Collection, Congestion and Error Control
- License: GPL (GPLv3)
- Type of human computer interaction: XML file
- OS/Middleware: Linux/Unix
- Required library or software: C/C++ Sockets
- Programming languages: C/C++
- Documentation: Text

5.12. Private Data Publication

Participants: Gergely Acs, Claude Castelluccia.

We are developing a set of tools to privately publish different types of datasets. For example, we are developing a software that can be used to sanitize sequential data (described in our CCS paper [41]). The code generates the set of noisy n-grams and generate a synthetic, and private, dataset. We are also developing a tool that implement the histogram sanitization algorithm described in our ICDM paper [33].

These tools are accessible here: <http://planete.inrialpes.fr/projects/p-publication/>

5.13. Experimentation Software

ACQUA

ACQUA stands for Application for Collaborative Estimation of the Quality of Internet Access. It has been developed within the French National project ANR CMON on Collaborative Monitoring in conjunction with Grenouille.com. ACQUA consists of a tool that lets the user have an estimation of the anomalies of the Internet based on active measurements of end-to-end delay metrics among a predefined set of landmarks (i.e. test points). When an anomaly is detected it is expressed in terms of how many destinations are affected by this anomaly, and how important in terms of delay variation is this anomaly for these affected destinations. See also <http://planete.inria.fr/acqua/> for more information and for a java version of the code.

WisMon

WisMon is a Wireless Statistical Monitoring tool that generates real-time statistics from a unified list of packets, which come from possible different probes. This tool fulfills a gap on the wireless experimental field: it provides physical parameters on realtime for evaluation during the experiment, records the data for further processing and builds a single view of the whole wireless communication channel environment. WisMon is available as open source under the Cecill license, at <http://planete.inria.fr/software/WisMon/>.

WEX Toolbox

The Wireless Experimentation (WEX) Toolbox aims to set up, run and make easier the analysis of wireless experiments. It is a flexible and scalable open-source set of tools that covers all the experimentation steps, from the definition of the experiment scenario to the storage and analysis of results. Sources and binaries of the WEX Toolbox are available under the GPLv2 licence at <https://twiki-sop.inria.fr/twiki/bin/view/Projets/Planete/WEXToolkit>. WEX Toolbox includes the CrunchXML utility, which aims to make easier the running and the analysis of wireless experimentations. In a nutshell, it implements an efficient synchronization and merging algorithm, which takes XML (or PDML) input trace files generated by multiple probes, and stores only the packets fields that have been marked as relevant by the user in a MySQL database –original pcap traces should be first formatted in XML using Wireshark. These operations are done in a smart way to balance the CPU resources between the central server (where the database is created) and the different probes (i.e., PC stations where the capture traces are located). CrunchXML is available under the GNU General Public License v2 at <http://twiki-sop.inria.fr/twiki/bin/view/Projets/Planete/CrunchXML>.

WiMAX ns-3

This simulation module for the ns-3 network simulator is based on the IEEE 802.16-2004 standard. It implements the PMP topology with TDD mode and aims to provide detailed and standard compliant implementation of the standard, supporting important features including QoS scheduling services, bandwidth management, uplink request/grant scheduling and the OFDM PHY layer. The module is available under the GNU General Public License at <http://code.nsnam.org/jamine/ns-3-wimax>. It will be included in the official 3.8v release of ns-3.

MonLab

Monitoring Lab is a platform for the emulation and monitoring of traffic in virtual ISP networks. It is 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.

MobiTrade

MobiTrade is the ns-3 and Android implementation of our solution for trading content between wireless devices. The application provides a utility driven trading system for efficient content dissemination on top of a disruption tolerant network. While simple tit-for-tat (TFT) mechanisms can force nodes to *give one to get one*, dealing with the inherent tendency of peers to take much but give back little, they can quickly lead to deadlocks when some (or most) of interesting content must be somehow fetched across the network. To resolve this, MobiTrade proposes a trading mechanism that allows a node (*merchant*) to buy, store, and carry content for other nodes (its *clients*) so that it can later trade it for content it is personally interested in. To exploit this extra degree of freedom, MobiTrade nodes continuously profile the type of content requested and the collaboration level of encountered devices. An appropriate utility function is then used to collect an optimal inventory that maximizes the expected value of stored content for future encounters, matched to the observed mobility patterns, interest patterns, and collaboration levels of encountered nodes. See also <http://planete.inria.fr/MobiTrade>.

REVES Project-Team

4. Software

4.1. RID: Rich Intrinsic Decomposer

Participants: Pierre-Yves Laffont, Adrien Bousseau, George Drettakis.

We developed a software platform to perform rich intrinsic decomposition methods from photographs of outdoor scenes, as described in [18]. 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.

4.2. Imerse: Inria Multi-Environment Realistic Simulation Engine

Participants: Adrien David, George Drettakis.

In the context of the ADT Interact3D and the ARC NIEVE, we developed Imerse, a middleware to be used as a VR engine, helping in the implementation of realistic simulations for immersive installations. Imerse provides a wrapper to OSG's (OpenSceneGraph) deep scene graph and its traversals abilities into an abstracted collection of high level objects which directly represent realistic entities (such as indoor elements, machines and realistic characters). It provides capacities such as skeletal animations or spatialized audio by interfacing with APF, while its clear composite pattern allows implementing more behaviors easily.

Finally, a generic design based on triggers and functors lets the final user implement complex scenarios of VR applications with the feeling of writing a script in C++. Applications developed on top of Imerse plug transparently into osgVR developed in the DREAM group (i.e., the research support development group of our Inria center). We are using osgVR to render OSG's scene graph in a distributed manner, since rendering clusters are available in an increasing number of installations. osgVR is a software layer developed by the DREAM research support group, ensuring synchronization and events/inputs distribution among a list of rendering slaves. These two libraries are available on GForge.

4.3. 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. 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 pre-mix them to a representative cluster source, so that all spatialization related processing can be applied only on the representative pre-mixed 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, a previous 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 also been critical in establishing collaborations in the context of various grant proposals (EU and national).

4.4. 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.

SCIPORT Team

5. Software

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 Hascoët [correspondant], Valérie Pascual.

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, and adjoint. Documentation is provided on the web site of the reserch team and as the Inria technical report RT-0300. TAPENADE runs under most operating systems and requires installation of Java jdk1.6 or upward. See also <http://www-sop.inria.fr/tropics/>

- Version: v3.6, r4343, February 2012
- 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 [19] 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 detailed 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 Team

5. Software

5.1. SUP

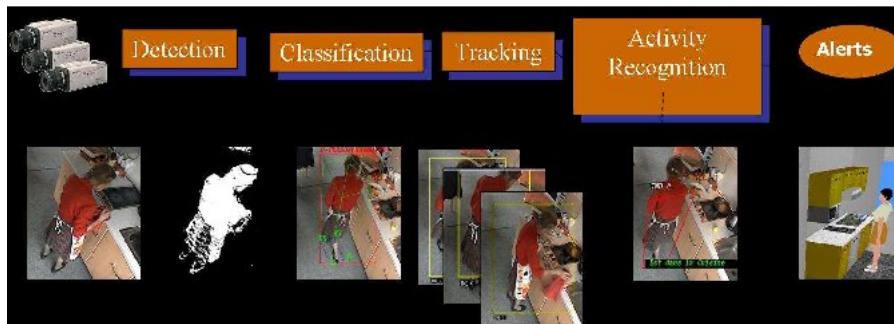


Figure 5. Tasks of the Scene Understanding Platform (SUP).

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 ViSEvAI tool is composed of two parts: a GUI to visualize results of the video processing algorithms and metrics results, and an evaluation program to evaluate automatically algorithm outputs on large 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).

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

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

5.3. Clem

The *Clem Toolkit* [63](see Figure 9) is a set of tools devoted to design, simulate, verify and generate code for LE [19] [77] programs. LE is a synchronous language supporting a modular compilation. It also supports automata possibly designed with a dedicated graphical editor.

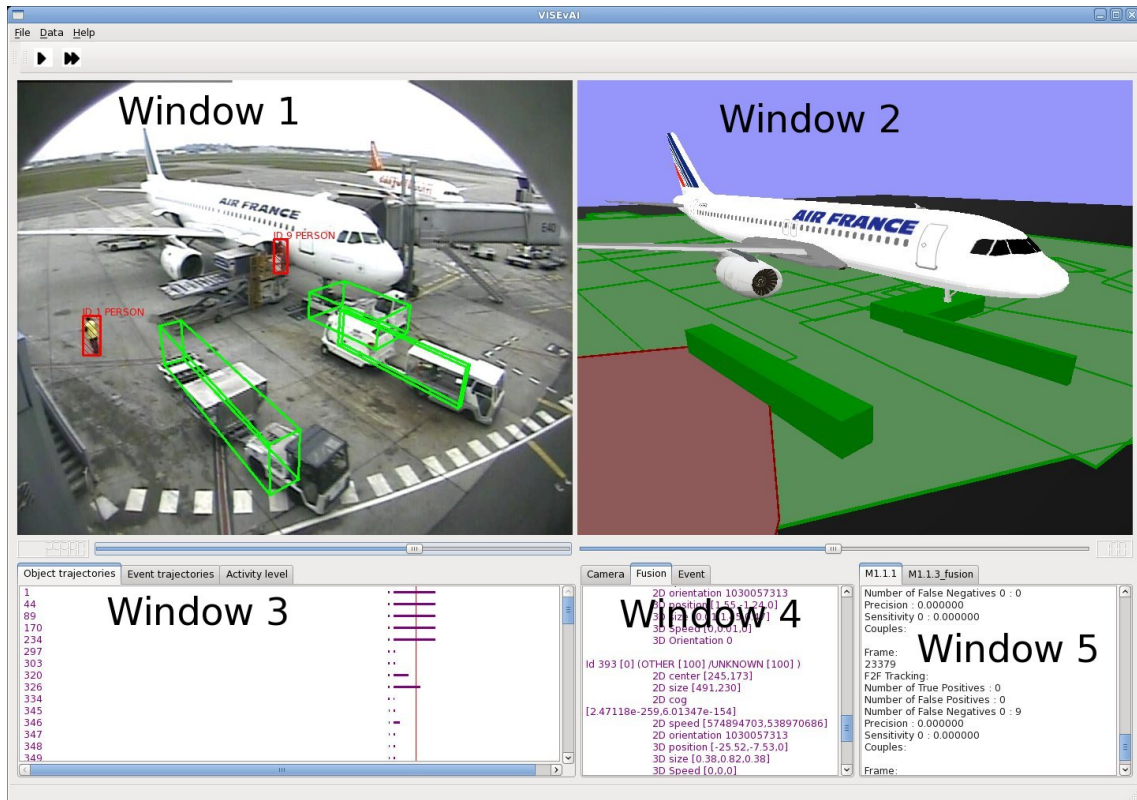


Figure 6. GUI of the ViSEvAI software

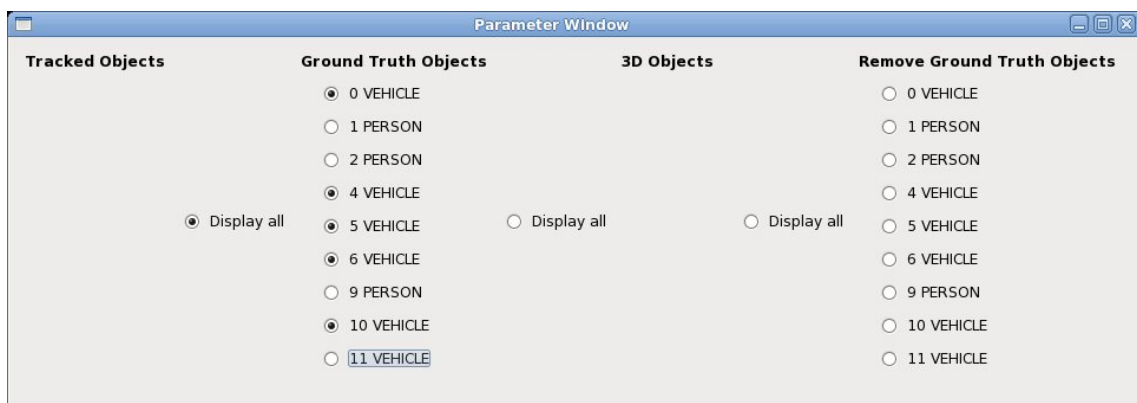


Figure 7. The object window enables users to choose the object to display

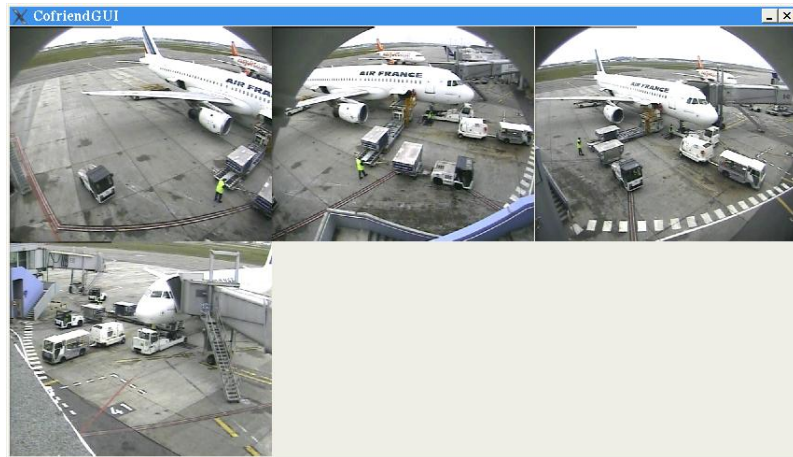


Figure 8. The multi-view window

Each LE program is compiled later into lec and lea files. Then when we want to generate code for different backends, depending on their nature, we can either expand the lec code of programs in order to resolve all abstracted variables and get a single lec file, or we can keep the set of lec files where all the variables of the main program are defined. Then, the *finalization* will simplify the final equations and code is generated for simulation, safety proofs, hardware description or software code. Hardware description (Vhdl) and software code (C) are supplied for LE programs as well as simulation. Moreover, we also generate files to feed the NuSMV model checker [61] in order to perform validation of program behaviors.

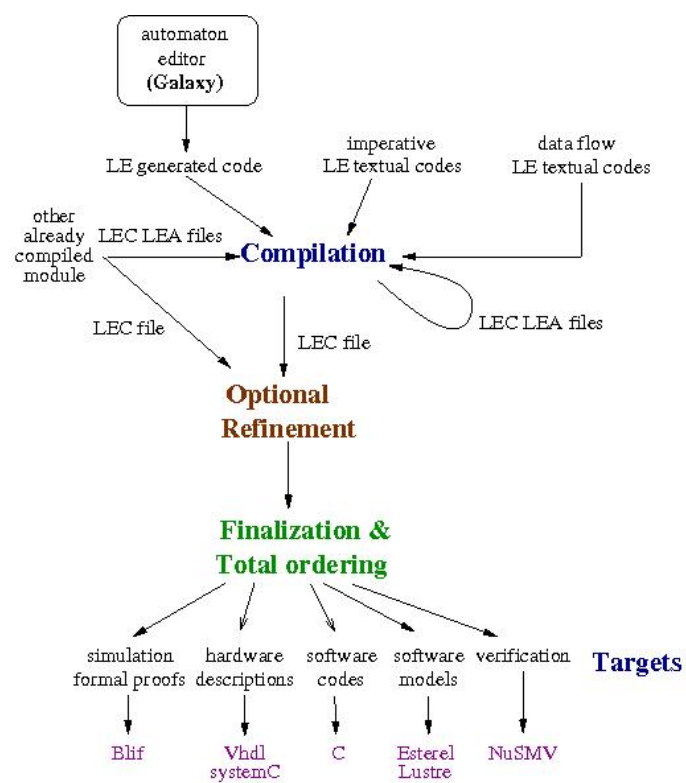


Figure 9. The Clem Toolkit

TOSCA Project-Team

5. Software

5.1. CarbonQuant

Participant: Mireille Bossy [correspondant].

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>.

- Version: 0.1

¹Technology Development Action

VIRTUAL PLANTS Project-Team

4. Software

4.1. V-Plants

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

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 2):

- 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 [9]. 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.StafTool, 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 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. A paper presenting L-Py [14] has been published to *Frontiers in Technical Advances in Plant Sciences*.

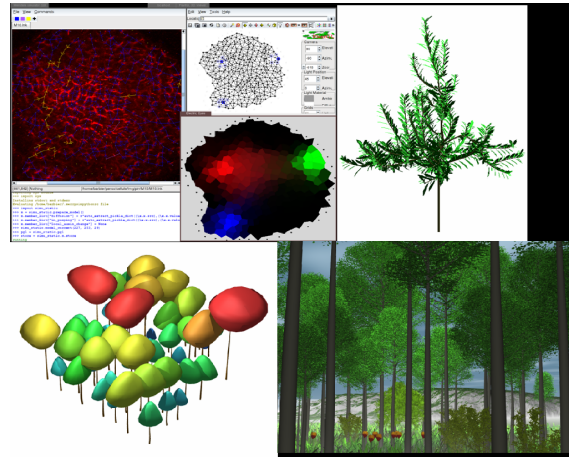


Figure 1. V-Plants components of the OpenAlea platform: simulating plant processes at different scales. Top Left: Reconstruction of a virtual meristem, analysis and simulation of the auxin fluxes inside the meristem. Top Right: Reconstruction of a virtual apple tree from digitized data. Bottom: Simulation 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 a Agropolis RTRA Grant named OpenAlea.

OpenAlea [10] 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 2012, one major release was done : Openalea 1.0. The following progresses were accomplished:

1. Develop and extend 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
 - The first OpenAlea Workshop have been held in Montpellier and has been attended by more than 60 scientists. 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.
- A continuous integration server has been set up To test the reliability of all the components after every commit.
- 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 520,000 times during the last two years. OpenAlea is the first project at Inria Gforge in term of number of downloads and of page views.

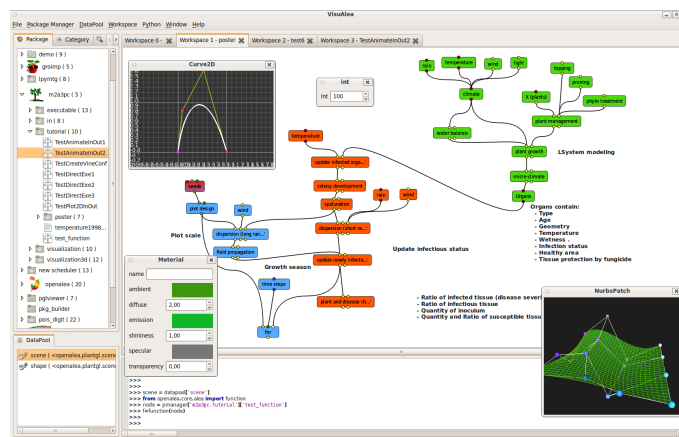


Figure 2. *OpenAlea.Visualea*: Visual programming interface. The package manager shows the available components. The components can be interconnected on a workspace to form a data-flow. The python interpreter allows low level interaction with the system.

4.3. Alinea

Participants: Christian Fournier, Christophe Pradal, Frédéric Boudon, Christophe Godin.

Other participants : Bruno Andrieu, Michael Chelle, Gaetan Louarn, Benoit de Solan, Mariem Abichou, Liqi Han, Elmer Ccopa-Rivera, Frederic Baret, Rafaele Casa, Youcef Mammeri, Didier Combes, Camille Chambon, Romain Barillot, Pierre Huynh, Jean-Christophe Soulie, Delphine Luquet.

The aim of this Action Ciblée Incitative of INRA is to constitute a consortium of modelers from INRA around the OpenAlea platform, and to integrate various ecophysiological models of simulation in OpenAlea (radiative transfer, interaction between plant and pest, circulation of hydric fluxes, and dispersion). The project includes 3 INRA teams and the Inria Virtual Plants project.

Different components have been integrated into the OpenAlea platform:

- Alinea.Adel is a module to simulate the 3D architectural development of gramineous crops.
- 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.TopVine is a component to reconstruct grapevine canopy structure.

- Ecomeristem is a crop growth, eco-physiological model that was designed for rice (model plant for cereals) to account for plant morphogenesis and its plasticity depending on genetic potential and sensitivity to the environment (water, temperature, radiation).
- Alinea.Nema is a module used for modeling of nitrogen dynamics between leaves.
- MAppleT is a FSPM model of an apple tree taking into account stochastic models for the topological development, a biomechanical model for branch bending, physiological laws as well as light interception.
- M2A3PC is a generic model to simulate spread of a pathogen on a growing plant like vine/powdery mildew and apple tree/apple scab.

In 2012, a 3D model of gramineous leave has been developed and presented at the PMA conference [28]. This dynamic leaf model is used to simulate different species of annual plants such as rice, wheat and maize.

WIMMICS Team

5. Software

5.1. Corese

Participants: Olivier Corby [correspondant], Fabien Gandon.

Corese ² (COnceptual REsource Search Engine) is a Semantic Web Factory. It enables users to load and process RDFS schemas, RDF metadata and to query the base of annotations thus created, by using the SPARQL Query Language.

Corese implements RDF, RDFS and SPARQL 1.1 Query Language & Update. Furthermore, Corese query language integrates original features such as approximate search, SQL or XPath. Approximate search consists of searching the best approximate answers to a query according to the ontology types. Corese also integrates a SPARQL-based Rule Language for RDF.

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 benefited from Inria software development support (ADT) with two software engineers. 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 50 applications, 24 PhD Thesis and is used for education by several institutions. It has been used as a Semantic Web Factory in such projects as Ontorule, Palette, SevenPro and SeaLife european projects, in e-WOK Hub, Neurolog, ISICIL and Kolflow ANR projects, BioMarker and KmP projects, Semantic Web Import Plugin for Gephi visualization and ECCO ontology editor. The work on Corese was published in [95], [96], [97], [94], [1], [5], [3], [2], [4].

This year we completed the KGRAM SPARQL 1.1 Query & Update interpreter.

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 ³ plugin is intended to allow the import of semantic data into Gephi open graph visualization platform. 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>

²<http://wimmics.inria.fr/corese>

³<http://wiki.gephi.org/index.php/SemanticWebImport>

5.3. ISICIL

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

The ISICIL software platform is made of several software components:

- XUL (XML-based User interface Language) extensions for the Firefox browser to assist the technology watch and business intelligence tasks by collecting relevant metadata according to the navigation context of the user.
- An application server based on Tomcat publishes services using the REST protocol to process requests of the users' applications and in particular the navigation extensions.

This architecture is summarized in Figure 1. Its major interest lies in the flexibility introduced by the loose coupling between REST services and navigators extensions or other applications.

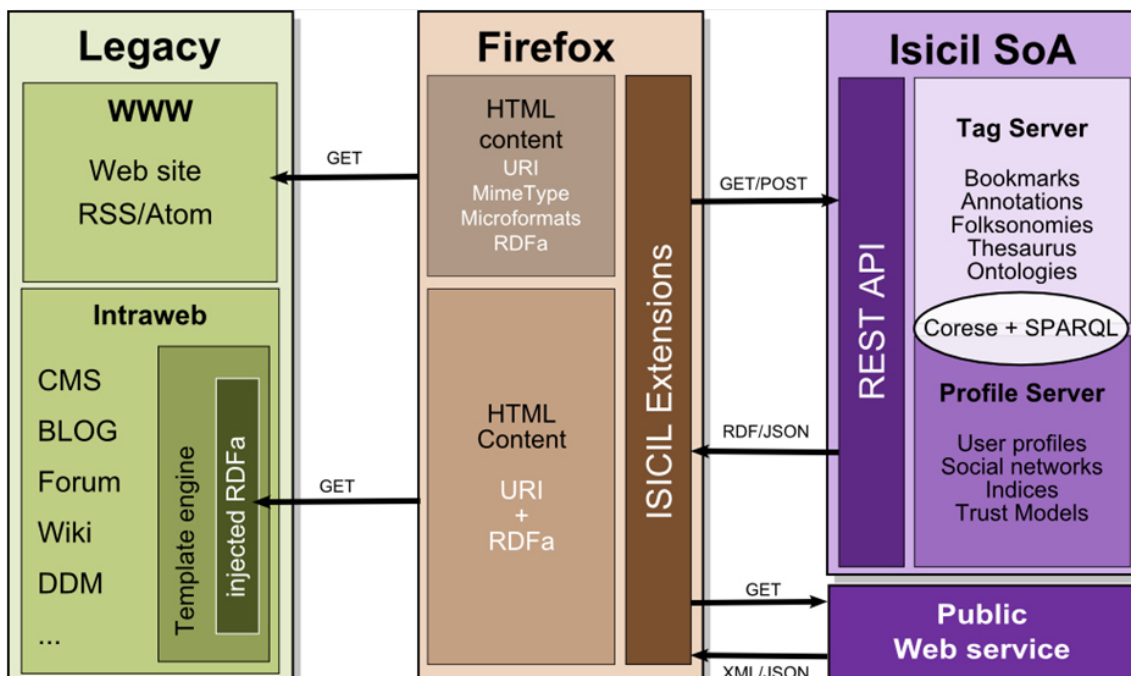


Figure 1. ISICIL Platform Architecture

In the context of the ISICIL ANR project, we have developed 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 SIOC⁴, SCOT, and Newman's Tag Ontology⁵ as shown in Figure 2. 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 3.

⁴<http://sioc-project.org>

⁵<http://www.holygoat.co.uk/owl/redwood/0.1/tags>

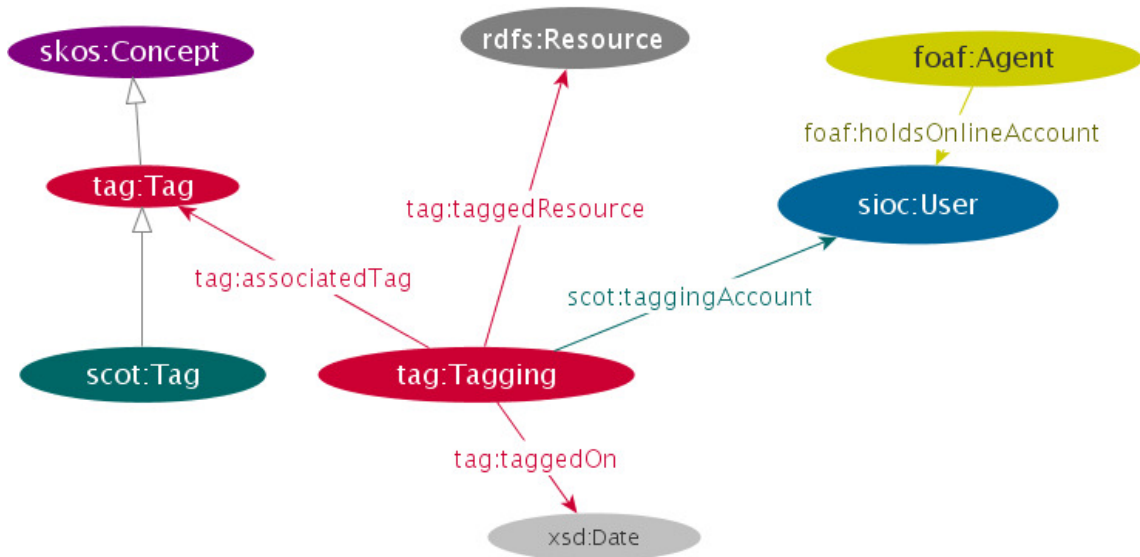


Figure 2. Model of tagging used in the Semantic Tag Server

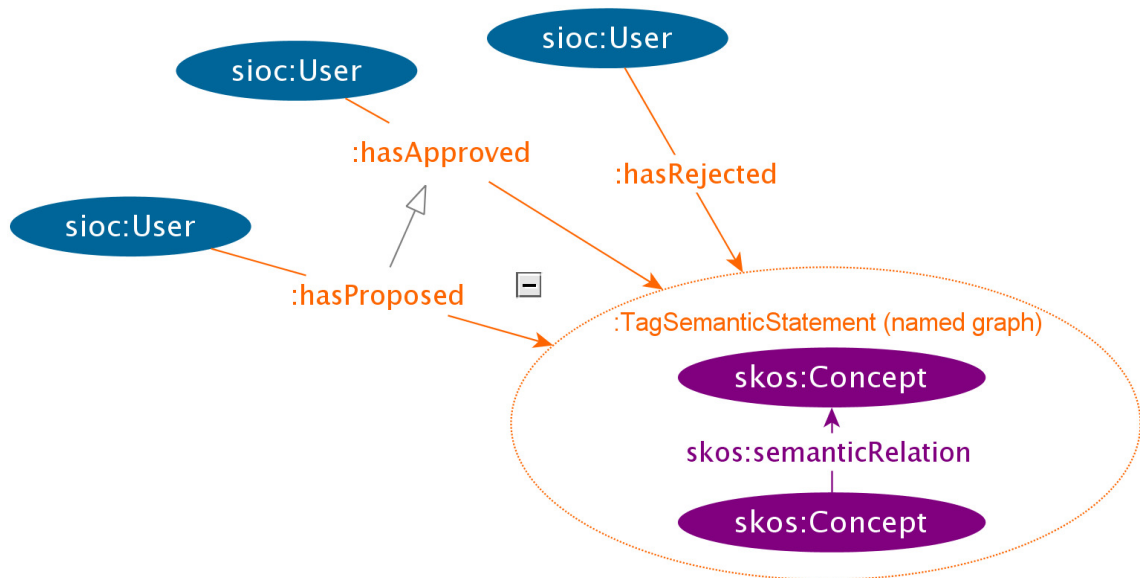


Figure 3. Folksonomy enrichment model

The functionalities of this server can be divided into three categories:

1. Tagging : creating a tag; get tag suggestions based on the input characters; create a tagging, i.e. a link between a resource, a user, and a tag.
2. Computing: an external library (exported as a java jar file) has been developed to perform computations on the tagging data. Two types of computations have been implemented:
 1. Spelling Variant detection.
 2. Related tag detection based on the computation of the similarity between tags [101].
3. Managing Semantic relations between tags: get semantically related tags, reject or propose new semantic relations.

We developed a Firefox extension to help users navigate within a folksonomy and organize semantically the tags. The main idea behind this tool is to combine organization tasks with everyday tasks in the least intrusive way, that is to say, without forcing the user in any way, and by providing a user friendly graphical interface. This tool, developed using the XUL framework ⁶, is supported by the SRTag model and the Semantic Tag Server. Users are provided with search bar for navigating the folksonomy. When available, other tags are suggested and ordered according to their semantic relation with the searched tag (broader, narrower, related, spelling variant). Each suggestion can be either :

- clicked to search content tagged with this tag;
- rejected by clicking a checkbox;
- modified thanks to a drag-and-drop mechanism where a tag can be dropped in another category of semantic relation.

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

⁶<http://developer.mozilla.org/en/XUL>

ZENITH Project-Team

5. Software

5.1. WebSmatch (Web Schema Matching)

Participants: Zohra Bellahsène, Emmanuel Castanier, Rémi Coletta, Duy Hoa Ngo, 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. YAM++ ((not) Yet Another Matcher)

Participants: Zohra Bellahsène [contact], Duy Hoa Ngo, Konstantin Todorov.

URL: <http://www2.lirmm.fr/~dngo/>

YAM++ is a tool for discovering semantic correspondences between ontologies. YAM++ supports several matching strategies: machine learning; generic methods when learning data are not available; discovering alignment of ontologies represented in different languages. Furthermore, since this year YAM++ is able to deal with large scale ontology matching.

5.3. SON (Shared-data Overlay Network)

Participants: Ayoub Ait Lahcen, Fady Draidi, Esther Pacitti, Didier Parigot [contact], Patrick Valduriez, Guillaume Verger.

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.

5.4. P2Prec (P2P recommendation service)

Participants: Fady Draidi, Esther Pacitti [contact], Didier Parigot, Guillaume Verger.

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 the Data-Shared Overlay Network (SON) infrastructure which is the basis for the ANR DataRing project.

5.5. ProbDB (Probabilistic Database)

Participants: Reza Akbarinia [contact], Patrick Valduriez, Guillaume Verger.

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.

5.6. Pl@ntNet-Identify

Participants: Mathias Chouet, Hervé Goëau, Alexis Joly [contact].

URL: <http://identify.plantnet-project.org>

Pl@ntNet-Identify is a web application dedicated to the image-based identification of plants. It has been developed jointly by ZENITH, the AMAP UMR team (CIRAD) and the Inria team IMEDIA. It allows submitting one or several query pictures of a plant and browse the matching species in a large collection of social image data, i.e. plant images collected by the members of a social network. It also allows users to enrich the knowledge of the application by uploading their own pictures in the reference collection. Nowadays, the dataset includes more than 17K images posted by about 100 members of Telabotanica¹ social network. In 2012, about 5000 identification sessions have been recorded. The client side of the application is implemented in Javascript whereas the server side (visual search engine) is mostly implemented in C++.

5.7. Pl@ntNet-DataManager

Participants: Mathias Chouet [contact], Alexis Joly.

Pl@ntNet-DataManager 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.

5.8. SnoopIm

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

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

¹<http://www.tela-botanica.org/>

SnoopIm is a content-based search engine allowing to discover and retrieve small visual patterns or objects in large collections of pictures (such as logos on clothes, road signs in the background, paintings on walls, etc.) and to derive statistics from them (frequency, visual cover, size variations, etc.). Query objects to be searched can be either selected from the indexed collection of photos, or selected from an external picture (by simply providing its URL). The web application allows online search of multiple users and has a cache feature to speed-up the processing of seen queries. It is implemented in Javascript on top of a C++ library developed in collaboration with INA ². The software is used at INA by archivists and sociologists in the context of the Transmedia Observatory project.

²<http://www.ina-sup.com/>