Activity Report 2011

Section Software
1. ABS Project-Team ................................................................. 4
2. AOSTE Project-Team .............................................................. 6
3. APICS Project-Team ............................................................... 9
4. ARIANA Project-Team ............................................................ 13
5. AROBAS Project-Team ........................................................... 14
6. ASCLEPIOS Project-Team ......................................................... 16
7. ATHENA Project-Team ........................................................... 17
8. AXIS Project-Team ................................................................. 18
9. BIOCORE Project-Team .......................................................... 22
10. COPRIN Project-Team .......................................................... 23
11. DEMAR Project-Team ........................................................... 25
12. EDELWEISS Project-Team ....................................................... 27
13. FOCUS Project-Team ............................................................ 31
14. GALAAD Project-Team .......................................................... 32
15. GEOMETRICA Project-Team ................................................ 34
16. GRAPHIK Project-Team ......................................................... 36
17. INDES Project-Team ............................................................ 37
18. MAESTRO Project-Team (section vide) ................................... 40
19. MARELLE Project-Team ........................................................ 41
20. MASCOTTE Project-Team ...................................................... 42
21. MODEMIC Team ................................................................. 47
22. NACHOS Project-Team ......................................................... 48
23. NEUROMATHCOMP Project-Team ......................................... 51
24. OASIS Project-Team ............................................................. 54
25. OPALE Project-Team ............................................................ 56
26. PLANETE Project-Team ......................................................... 59
27. PULSAR Project-Team .......................................................... 65
28. PUMAS Team ..................................................................... 70
29. REVES Project-Team ............................................................ 71
30. SMASH Project-Team (section vide) ....................................... 73
31. TOSCA Project-Team ........................................................... 74
32. TROPICS Project-Team ........................................................ 75
33. VIRTUAL PLANTS Project-Team .............................................. 77
34. ZENITH Team ..................................................................... 81
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 2011 is briefly described as the context is presented in the sections dedicated to new results. On the other hand, the software made available before 2011 is briefly specified in terms of applications targeted.

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

4.1.1. vorpatch and compatch: Modeling and Comparing Protein Binding Patches
Participants: Frédéric Cazals, Noël Malod-Dognin.

Context. Our work on the problem of modeling and comparing atomic resolution protein interfaces has been discussed in sections 5.4.1 and 5.1.1. The programs undertaking these two tasks are respectively named vorpatch and compatch.


4.1.2. voratom: Modeling with Toleranced Models
Participants: Frédéric Cazals, Tom Dreyfus.

Context. Our TOleranced Model framework has been described in sections 5.2.1 and 5.2.2. The corresponding software package 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 aforementioned programs are made available from http://cgal.inria.fr/abs/voratom/.

4.1.3. 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.


4.1.4. intervor: Modeling Macro-molecular Interfaces
Participant: Frédéric Cazals.

In collaboration with S. Loriot, from 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 web site http://cgal.inria.fr/abs/Intervor serves two purposes: on the one hand, calculations can be run from the web site; on the other hand, binaries are distributed for Linux. To the best of our knowledge, this software is the only publicly available one for analyzing Voronoi interfaces in macro-molecular complexes.

4.1.5. vorlume: Computing Molecular Surfaces and Volumes with Certificates

Participant: Frédéric Cazals.

In collaboration with S. Loriot, from the GEOMETRY FACTORY.

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.


4.1.6. ESBTL: the Easy Structural Biology Template Library

Participant: Frédéric Cazals.

In collaboration with S. Loriot (the Geometry Factory), and J. Bernauer, from the EPI AMIB.

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.


4.1.7. 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.

AOSTE Project-Team

5. Software

5.1. TimeSquare

Participants: Charles André, Nicolas Chleq, Julien Deantoni, Benoît Ferrero, 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: Anthony Coadou, 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 represent 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 deadlines and latencies constraints they can minimize the total execution time (makespan) of the application onto the given architecture, as well as the amount of memory. The results of each exploration is visualized as timing diagrams simulating the distributed real-time implementation.

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

5.4. 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.
5. Software

5.1. Tralics

**Participant:** José Grimm [corresponding participant].

Tralics is a LaTeX to XML translator. It is a free software, distributed under the CeCILL license version two, in binary form for Linux, Windows and MacOS X. Sources are also available via its web page [http://www-sop.inria.fr/apics/tralics](http://www-sop.inria.fr/apics/tralics). The development of the LaTeX to XML translator, named Tralics, was continued. Latest release is version 2.14.4, dated 07-11-2011 (see section 6.1).

5.2. RARL2

**Participants:** Jean-Paul Marmorat, Martine Olivi [corresponding participant].

RARL2 (Réalisation interne et Approximation Rationnelle L2) is a software for rational approximation (see section 3.1.4) [http://www-sop.inria.fr/apics/RARL2/rarl2-eng.html](http://www-sop.inria.fr/apics/RARL2/rarl2-eng.html).

This software takes as input a stable transfer function of a discrete time system represented by either of

- its internal realization,
- its first $N$ Fourier coefficients,
- discretized values on the circle.

It computes a local best approximant which is stable, of prescribed McMillan degree, in the $L^2$ norm.

It is akin to the arl2 function of Endymion (see section 5.5) from which it differs mainly in the way systems are represented: a polynomial representation is used in Endymion, while RARL2 uses realizations. It is implemented in Matlab. This software handles multi-variable systems (with several inputs and several outputs), and uses a parametrization with the following advantages:

- it incorporates the stability requirement in a built-in manner,
- it allows the use of differential tools,
- it is well-conditioned, and computationally efficient.

An iterative research strategy on the degree of the local minima, similar in principle to that of arl2, 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 (section 4.2) as well as sources or cracks recovery (section 4.1). It was released to the universities of Delft, Maastricht, Cork and Brussels. The parametrization embodied in RARL2 was recently 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 5.7).

5.3. RGC

**Participants:** Fabien Seyfert [corresponding participant], Jean-Paul Marmorat.
The identification of filters modeled by an electrical circuit that was developed by the team (see section 4.2) 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” \[52\] 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. Moreover, 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 5.6), which is the successor of RGC, solves with guarantees this constraint realization problem.

5.4. PRESTO-HF

Participant: Fabien Seyfert.

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 either on hyperion (see 5.5) (Unix or Linux only) or RARL2 (platform independent), two rational approximation engines 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 and a license agreement has been recently negotiated with Thales airborne systems. 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).

5.5. Endymion

Participant: José Grimm.
The core of the Endymion system (a follow-up to hyperion) is formed by a library that handles numbers (short integers, arbitrary size rational numbers, floating point numbers, quadruple and octuple precision floating point numbers, arbitrary precision real numbers, complex numbers), polynomials, matrices, etc. Specific data structures for the rational approximation algorithm arl2 and the bounded extremal problem bep are also available. One can mention for instance splines, Fourier series, Schur matrices, etc. These data structures are manipulated by dedicated algorithms (matrix inversion, roots of polynomials, a gradient-based algorithm for minimizing $\psi$, Newton method for finding a critical point of $\psi$, etc), and input-output functions that allow one to save data on disk, restore them, plot them, etc. Interactivity is provided through a symbolic interpreter based upon a Lisp interpreter.

The development of Endymion, http://www-sop.inria.fr/apics/endymion/index.html has come to an end. The software is still maintained and sources are available on the ftp server.

5.6. Dedale-HF

Participant: Fabien Seyfert.

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 6.6). 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).

![Flowchart of the design and tuning process of a microwave filter](image)

*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.

As mentioned in 6.6 an extension of this software that handles symmetrical networks is under construction.

5.7. FindSources3D

Participants: Maureen Clerc [EPI Athena], Juliette Leblond [corresponding participant], Jean-PaulMar- morat, Théo Papadopoulo [EPI Athena].

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 section 4.1, it makes use of RARL2 (section 5.2) 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 and those in [30]).

5.8. Sollya

Participants: Sylvain Chevillard, Christoph Lauter [LIP6], Mioara Joldeș [Arénaire team until October; now with Uppsala University].

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/.
ARIANA Project-Team

5. Software

5.1. Software

5.1.1. Deposits

- The software WAIHEKE was deposited with the APP in October 2011. It was developed for classifying 3D-point data generated from airborne lidar systems or multi-view imagery. The input point cloud is labeled into four classes of interest (building, ground, vegetation and clutter).
- The software SIKORA was deposited with the APP in October 2011. It was developed for extracting 3D-segments, planes, cylinders, cones, spheres and tori by region growing from 3D-point clouds.
- The software MOJOPIN was deposited with the APP in October 2011. It was developed for performing planimetric arrangements of urban components, including roof sections and trees, from labeled point clouds.
- The software SCOMBO v1.0 and Hierarchical SCOMBO v1.0 were deposited with the APP in December 2011. It deals with the supervised classification of multiband optical images by using Markov random fields and hierarchical Markov random fields, respectively.
AROBAS Project-Team

5. Software

5.1. Experimental Testbeds

Methodological solutions to the multi-faceted problem of robot autonomy have to be combined with the ever present preoccupation of robustness and real-time implementability. In this respect, validation and testing on physical systems is essential, not only as a means to bring together all aspects of the research done in AROBAS—and thus maintain the coherence and unity of the project-team—but also to understand the core of the problems on which research efforts should focus in priority. The instrumented indoor and outdoor wheeled robots constitute a good compromise in terms of cost, security, maintenance, complexity and usefulness to test much of the research conducted in the project-team and to address real size problems currently under investigation in the scientific community. For the next few years, we foresee on site testbeds dedicated to ground robotic applications (figure 1 Left and Center).

- **HANNIBAL Indoor mobile robot**
  Our cart-like platform, built by Neobotix can operate on flat surfaces, in both indoor and outdoor environments. This platform is equipped with the various sensors needed for SLAM purposes, autonomous navigation and sensor-based control. With its programming further developed to become user-friendly, it has become one of the team’s main testbeds for fast prototyping of perception, control and autonomous navigation algorithms.

- **CyCab Urban electrical car**
  Two instrumented electrical cars of the CyCab family are destined to validate researches in the domain of Intelligent urban vehicle. CyCabs are used as experimental testbeds in several national projects.

- **Hexacopter VTOL vehicle**
  A basic version of this machine was recently acquired from Mikrokopter Inc.(Germany) by our colleagues (T. Hamel, G. Ducard, M.-D. Hua) from the SIS (Signal, Images et Systèmes) research pole at I3S-UNSA-CNRS. It has a diameter of 90cm, weights about 1.5 kg, and can carry a payload up to 1.5 kg (figure 1 Right). The flight time autonomy varies between 6mn and 18mn, depending on the payload, and it can be extended provided that the battery capacity is extended accordingly. The machine’s external envelope has been modified for safety reasons. Initial flight tests have been conducted, and the aircraft is currently being equipped with various sensors (GPS, accelerometers, gyroimeters, camera,...). We are working with our colleagues from I3S to control this vehicle with the aim of providing it with large autonomy capabilities and robust performance. It is also a benchmark to validate various estimation/control issues that we are currently investigating.
Figure 1. Left: The Hannibal platform. – Center: The Cycab vehicle. – Right: Hexacopter
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.

It is mainly developed by the Inria team project 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

4.2. MedINRIA

Participants: Benoît Bleuzé, Olivier Clatz [correspondant], Vincent Garcia, Michael Knopke, Stephan Schmitt, Maxime Sermesant, John Stark, 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.

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), DT-REFInD (Proprietary)
- Programming language: C++
5. Software

5.1. OpenMEEG

Participants: Théodore Papadopoulo, Maureen Clerc, Emmanuel Olivi, Alexandre Gramfort [Parietal project-team].

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, thus providing excellent accuracy. 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 ODYSSÉE 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.

Thanks to the ADT MedInria-NT, this library is in the process to be partly integrated within the interactive medical imaging platform MedINRIA.
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 Télus 5.5.2) which is based on a Service oriented Architecture. The development process has started this year, finding ways to fund human resources. Such a platform aims the community of Living Labs domain.

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 [87], [86], [106], [105], [110], especially those that can handle dissimilarity data (available on Inria’s Gforge server (public access) https://gforge.inria.fr/projects/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 [101], [104], [103], [102] (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 [111]. 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 [84] (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 [85]).

A Web interface developed in C++ and running on our Apache internal Web server is available for the following methods: SCluster, Div, yCdis, CCClust.

Previous versions of the above software have been integrated in the SODAS 2 Software [98] which was the result of the european project ASSO 5 (2001-2004). SODAS 2 softsodaslinkware supports the analysis Stof multidimensional complex data (numerical and non numerical) coming from databases mainly in statistical offices and administration using Symbolic Data Analysis [82]. This software is registrated at APP. 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 [88], [112].

5 ASSO: Analysis System of Symbolic Official data
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 [108] for more details) in collaboration with F. Masseglia and B. Trousse:

- Cluster & Divide [108]
- and Divide & Discover [13], [108].

4.2.3. Mining Data Streams

Participants: Brigitte Trousse [correspondante], Mohamed Gaieb.

In Marascu’s thesis (2009) [95], 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 $O(n \cdot m)$ 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 (cf. 6.2.1) 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 http://iapg.jade-hs.de/personen/brinkhoff/generator/ in the context of MIDAS project (Java version)

This year it has been integrated as a Web service (Java version) in the first version of FocusLab platform in the ELLIOT context (cf. 5.5.2): a demonstration was made on San Rafaele Hospital media use case at the first ELLIOT review at Brussels (cf. 6.3.1.1).

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 Preprocessing and Data Stream extraction) for Web Usage Mining (WUM) is issued from AxISlogminer preprocessing software which implements the multi-site log preprocessing methodology developed by D. Tanasa in his thesis [16] 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. AWLH can extract and structure log files from several Web servers using different input format. The web log files are cleaned as usually before to be used by data mining methods, as they contain many noisy entries (for example, robots bring a lot of noise in the analysis of user behaviour then it is important in this case to identify robot requests). The data are stored within a database whose model has been improved.
Now the current version of our Web log processing (Available on INRIA's gforge website with private access) 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.

ATWUEDA for Web Usage Evolving Data Analysis [ 90 ] was developed by A. Da Silva in her thesis [ 89 ] 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 http://www.r-project.org/ 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.


This year we have demonstrated the efficiency of ATWUEDA [ 51 ] by applying it on another real case study on condition monitoring data streams of an electric power plant provided by EDF (cf. section 5.5.1 ).

ATWUEDA is used by Telecom Paris Tech and EDF [ 51 ].

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 [ 92 ], [ 93 ] is an object-oriented framework [ 94 ], [ 91 ] 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/.

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 an 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 [92], 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. The last release was delivered on November 1st 2010 based on the work of Caffiau and al. [83].

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.
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. 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 two libraries that have been designed in the project and are 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.

These libraries can be freely downloaded.

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.

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. The initial aim of providing the Mathematica users community a transparent access to the functionalities of ALIAS, and of extending the dissemination of our library, has progressively turned into the aim of providing ALIAS advanced users and developers with a high-level modular interface for prototyping, easy testing and quick implementation of new interval analysis algorithms and procedures relying on symbolic computation skills. This includes symbolic preprocessing of expressions, and symbolic specializations of interval analysis algorithms.
5. Software

5.1. Software

5.1.1. FES muscle modeling in opensim framework

Participants: Mitsuhiro Hayashibe, Philippe Fraisse, Emel Demircan, Oussama Khatib (INRIA Equipe Associee, Stanford Univ.).

In FES, movement synthesis and control are still challenging tasks due to the complexity of whole body dynamics computation and the nonlinearity of stimulated muscle dynamics. An efficient movement synthesis means that criteria can be defined and evaluated through an accurate numeric simulation. We perform the implementation of muscle model representing the electrically stimulated muscle into the OpenSim framework which has whole body musculoskeletal geometry. We would like to develop the FES simulator using Stanford Operational Space Whole-Body Controller which allows the real-time motion generation with virtual FES and finally we aim at the development of motion correction controller to find the appropriate FES signals against a disabled motor function.

5.1.2. Further development of gom2n software - a toolchain to simulate and investigate selective stimulation strategies for FES

Participants: Guillaume Jourdain, Pawel Maciejasz, Jeremy Laforet, Christine Azevedo Coste, David Guiraud.

Concurrently with the experiments on selective stimulation of nerve fibres, performed on earthworms (see section 6.1.6), also the gom2n toolchain developed previously by our team was further developed. Main objective of this work was to be able to simulate similar behaviour of nerve fibres, as observed during electrical stimulation of the giant nerve fibres of earthworms, and therefore to be able to compare computational and experimental results. Main improvements which has been implemented in the new version of the gom2n toolchain are:

- improved and more intuitive users interface
- possibility to perform concurrently multiple simulations for various stimulation parameters, as well as various diameters and locations of nerve fibres within the nerve.

Further work is however still needed to adapt electrical properties of simulated fibres, since electrical properties of the earthworm’s giant nerve fibres are different that properties of mammalian nerve fibres."

5.1.3. 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 its embeds among others a dedicated processor (micro-machine with a specific reduced instruction set), a monitoring module and a 3-layer protocol stack. To face the complexity of the unit’s digital part and to ease its prototyping on programmable digital devices (e.g. FPGA), we developed an approach for high level hardware component programming (HILECOP). To support the modularity and the reusability of sub-parts of complex hardware systems, the HILECOP methodology is based on components. An HILECOP component has: a Petri Net (PN) based behavior, a set of functions whose execution is controlled by the PN, and a set of variables and signals. Its interface contains places and transitions from which its PN model can be inter-connected as well as signals it exports or imports. The interconnection of those components, from a behavioral point out view, consists in the interconnection of places and/or transitions according to well-defined mechanisms: interconnection by means of oriented arcs or by means of the "merging" operator (existing for both places and transitions).
The development of an Eclipse-based version of HILECOP has been achieved. This new version of HILECOP has been registered (new deposit) in September 2011, at the French Agence de Protection des Programmes (APP) with the IDDN.FR.001.380008.000.S.P.2011.000.31235. It will be accessible to the academic community at the beginning of 2012.

5.1.4. 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 (Fig. 1; left). This environment allows the manipulation of micro-programs from their edition to their remote control (Fig. 1; right). It also allows the programming of control sequences executed by an external controller in charge of automatically piloting a stimulator.

This new version of SENIS Manager has been registered (updated deposit) in September 2011, at the French Agence de Protection des Programmes (APP), with the IDDN.FR.001.320011.001.S.P.2009.000.31500.
5. Software

5.1. Corese

Participants: Olivier Corby [correspondant], Fabien Gandon.

Corese\(^1\) (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 an 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 45 applications, 21 PhD Thesis and is used for education by 8 institutions. It is used as a Semantic Factory in such projects as Ontorule, Palette, SevenPro and SeaLife European projects, in e-WOK Hub, Neurolog, ISICIL and Kollflow ANR projects, BioMarker and KmP projects, Semantic Web Import Plugin for Gephi visualization and ECCO ontology editor. The work on Corese was published in \[ 57 \], \[ 58 \], \[ 59 \], \[ 56 \], \[ 1 \], \[ 5 \], \[ 3 \], \[ 2 \], \[ 4 \].

This year we released a major new version 3.0 based on the KGRAM SPARQL 1.1 interpreter. KGRAM (see 6.1.1 ) is a generic SPARQL interpreter that can query not only RDF but also labeled graphs.

Web page: http://www.inria.fr/sophia/edelweiss/software/corese

5.2. Semantic Web Import Plugin for Gephi visualization

Participants: Erwan Demairy [correspondant], Fabien Gandon, Olivier Corby.

The SemanticWebImport\(^2\) plugin is intended to allow the import of semantic data into Gephi open graph visualisation 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.


---

1 http://www.inria.fr/sophia/edelweiss/software/corese
2 http://wiki.gephi.org/index.php/SemanticWebImport
5.3. ISICIL

Participants: Sébastien Comos, 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.

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 SIOC3, SCOT, and Newman’s Tag Ontology4 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.

---

3 http://sioc-project.org
4 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 [63].

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[^5], 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/](https://gforge.inria.fr/projects/isicil/)

5. Software

5.1. Jolie

Members of Focus have recently developed Jolie [7] (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 2011 the development of Jolie has continued. The main activities have been:

- A new session message-routing mechanism, based on correlation sets has been implemented. This mechanism makes message routing programmable from inside Jolie code.
- A new primitive for (smart) service aggregation.
- A graphical editor.
- An integrated development environment.

Moreover, this year Jolie has been used for teaching, in a 30-hour master course at IT University of Copenhagen, Denmark.

5.2. Others

Below we list some software that has been developed in Focus in previous years but that during 2011 has not substantially changed. Short descriptions of these items can be found in the Focus activity report for 2010.

- PiDuce (see http://www.cs.unibo.it/PiDuce/) is a prototype for experimenting Web services technologies, based on theories of process calculi and XML documents and schemas [3].
- IntML is a functional programming language guaranteeing sublinear space bounds for all programs [57].
5. Software

5.1. Mathemagix, a free computer algebra environment

**Participants:** Bernard Mourrain, Angelos Mantzaflaris.

http://www.mathemagix.org/

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 are 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:** Angelos Mantzaflaris, Bernard Mourrain, Meriadeg Perrinel.

http://axel.inria.fr.

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 in Exciting 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 Perinnel 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 the 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).

5.3. Maple packages for differential algebra and algebraic invariants

Participant: Evelyne Hubert.

- The Maple package diffalg is a collection of routines to handle systems of polynomial differential equations and inequations. The functionalities include differential elimination, expansion of the solutions into formal power series and analysis of singular solutions. The underlying theory and terminology belongs to differential algebra.
  Collaborators: François Boulier and François Lemaire from University of Lille.

- The Maple AIDA package is a collection of routines to explore algebra of differential invariants: computation of generating sets of invariants, rewritings, syzygies, and their differential analogues. The package builds on the Maple libraries Groebner, Vessiot and diffalg.
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 Lorio, Abdelkrim Mebarki, Naceur Meskini, Andreas Meyer, Sylvain Pion, Marc Pouget, François Rebufat, Laurent Rineau, Laurent Saboret, Stéphane Tayeb, 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 13 editors in the editorial board, with approximately 20 additional developers. The user discussion mailing-list has more than 1000 subscribers with a relatively high traffic of 5-10 mails a day. The announcement mailing-list has more than 3000 subscribers.
5. Software

5.1. Cogui

Participants: Alain Gutierrez, Michel Chein, Michel Leclère, Marie-Laure Mugnier, 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. It relies on CoGITaNT for reasoning tasks.

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

- XML Datatypes are now supported.
- The use of URIs as identifiers and the notion of namespaces have been introduced to facilitate interoperability with RDF/XML.
- A pure java solver has been implemented to preserve reasoning capabilities on all platforms.
- A scripting language has been introduced on top of Cogui to satisfy specific applications requirements and facilitate the writing of prototypes. Scripts can be serialized in Cogui projects and give end-users the ability to manipulate objects of the knowledge base and use reasoning features through the Cogui core API.

5.2. Towards Large Knowledge Bases

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

We have began to study different storage solutions for large databases, first as part of a Master’s thesis, and now with the PhD of Bruno Paiva Lima Da Silva [29]. The goal of this work is to evaluate different storage paradigms and systems (e.g., relational databases MySQL and Sqlite; triple stores Sesame and graph databases Neo4J, DEX, HyperGraphDB and OrientDB) with respect to our particular requirements (mainly ontological conjunctive query answering with large knowledge bases), and to integrate them in a unified way in a software tool (answering our genericity requirement and paving the way for hybrid KBs). We believe this work to be a necessary step for our next generation of software tools.
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/centre/sophia/innovation

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], Thomas Gazagnaire [Inria], Zhengqin Luo [Inria], Cyprien Nicolas [Inria], Tamara Rezk [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. Camloo

Camloo is a caml-light to bigloo compiler, which was developed 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.

5.2.3. 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.mashic.net.
5.5. Old software

5.5.1. 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.

5.5.2. 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.
MAESTRO Project-Team (section vide)
MARELLE Project-Team

5. Software

5.1. 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](http://coq.inria.fr/pylons/pylons/contribs/view/Semantics/v8.3).

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

5.2. Certicrypt

**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/](http://easycrypt.gforge.inria.fr/).
MASCOTTE Project-Team

5. Software

5.1. Grph

Participants: Nathann Cohen, David Coudert, Luc Hogie [correspondant], Aurélien Lacin, Grégory Morel, Issam Tahiri.

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

The GRPH project takes over Dipergrafs which was introduced in the activity report of 2010. A drastic change in the model of Dipergrafs justified the name change.

The objective of GRPH is to provide researchers and engineers a suitable graph library for graph algorithms experimentation and network simulation. GRPH is mainly 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. At every stage, it is designed to be efficient in terms of: computation time (use of parallelism, caching, adequate data structures, native code, etc.); memory requirements (use of Java primitives); and portability (it is written in a Java and C). Its model considers mixed graphs composed of (un)directed simple- and hyper-edges. It can handle large dynamic graphs in the order of millions of nodes. GRPH comes with a collection of base graph algorithms which are regularly augmented.

So far, most known users of the GRPH library are part of INRIA and of the FP7 STREP EULER project. 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 2011, GRPH was augmented over Dipergrafs of a number of features suited to its usage within the MASCOTTE research team. These include: addition of numerous graph manipulation methods; introduction of an incidence-list data structure for the representation of graphs; introduction of an adaptive data structure for the representation of sets (based on hash-tables and bit-vectors); integration of implementations of "maximum clique" and "sub-graph isomorphism" algorithms by Christine Solnon (CNRS, INSA Lyon). These sources, written in C, are compiled on-the-fly; integration of implementation of "graph isomorphism" algorithm by Brendan McKay (Australian National University); iteration of implementation of "number of triangles" algorithm by Matthieu Latapy (LIP6); introduction of a bridge to the Mascot/OpenGVE library; introduction of a bridge to the JUNG library; addition of numerous graph algorithms; introduction of a new layer atop GRPH which allows the representation and manipulation of graph as Java objects, like it is done in other libraries such as Mascot, Jung, etc.; introduction of an efficient mechanism for the definition of graph properties; addition of graph reporting facilities.

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

5.2. DRMSim

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

Around 45,000 lines, developed in Java, collaboration between MASCOTTE and researchers in LaBRI (95% MASCOTTE).
The expansion of the Internet results in a number of issues: BGP (Border Gateway Protocol) starts to show its limits in terms of the number of routing table entries it can manage. More efficient dynamic routing protocols are thus under investigation. However, because deploying under-development routing protocols on the Internet is not practicable at a large-scale, simulation is a necessary step to validate the properties of a newly proposed routing scheme. Unfortunately, the simulation of routing protocols over large networks poses real challenges due to the limited computational capabilities of computers. Existing simulation tools exhibit limitations in terms of the number of nodes they can handle and of the models they propose. This motivated us to conceive and develop DRMSim (Dynamic Routing Model Simulator): a network simulator which addresses the specific problem of large-scale simulations of routing models.

DRMSim relies on a discrete-event simulation engine. 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 [99] and lightweight versions of BGP (Border Gateway Protocol).

Recent developments (in 2011) in the DRMSim simulator include the four following elements:

1. The initial framework was composed of a routing model. It now incorporates a system model and a metric model. In addition, the system model now considers the dynamic evolution of the simulated network. This dynamic behavior includes the maintenance operations on the network infrastructure as well as router failures. This model stores the connectivity of routers and links before their failure is simulated. This information is used for the simulation of the recovery procedure. This model takes as its input parameter the distribution of failure probability for both routers and links.

2. The metric model has been fully rewritten and is now geared towards computational performance and flexibility. Taking measures along a discrete-event simulation can be performed in many ways. DRMSim uses a new approach which consists in a metric model listening to the simulation and system models. The user can define its own metrics. Memory and CPU usages depend on which metrics are defined, to which set of routers/links they are applied, how many measures are taken and their computational complexity. It is possible to restrict the model to a small amount of nodes/links by selectors provided as input parameters. At the cost of memory and CPU usage, metrics measures can be stored as time-ordered sequence of values. To reduce the need of resources, a single global measure for each metric can be computed. Finally, metrics can be computed globally on the set of selected entities (links/routers) but also separately for each entity.

3. DRMSim enables the definition of customized simulation scenarios and stateful simulation campaigns. Commonly, a simulation campaign consists in iterating over the set of combinations of parameter values, calling the simulation function for every combination. These combinations cannot be found randomly nor can they be determined using linear functions. Indeed, most of the time there exist correlations between the parameters involved. Also for performance reasons, the end-user will prefer non-linear (most often logarithmic) evolutions for the values of the parameters. The definition of the set of combinations is strongly linked to the simulated system and the time needed to solve it. DRMSim provides a simulation methodology that describes (programmatically) the way a simulation campaign should be conducted.

The duration of a simulation can be as long as several hours (or days). In the context of a simulation campaign where numerous simulations are executed, it is important that re-starting a simulation campaign that was interrupted does not entail the re-computation of already computed results. In order to do this, 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 G\textsc{RPH} library.

4. Finally, DRMSim manipulates graph abstractions, allowing the user to force the use of a library different from the default one, i.e. G\textsc{RPH}.
5.3. Mascop and openGVE

Participant: Michel Syska [correspondant].

MASCOT [98] (MASCOTTE Optimization) is a Java library distributed under the terms of the LGPL license which is dedicated to graph and network processing. MASCOT includes a collection of Java interfaces and classes that implement fundamental data structures and algorithms. The forthcoming public distribution of MASCOT will appear under the name of the openGVE project, MASCOT being one implementation of the bridge graph interface (see http://opengve.inria.fr/bridge-graph-interface/apidocs/fr/inria/opengve/bridge/interfaces/Graph.html). The objective is to allow easy integration of different implementations. The applications already written will not be affected. They will have different choices of internal implementation which may lead to better performances for specific issues such as large graphs processing.

The main objective of MASCOT project is to ease software development in the field of network optimization. Examples of problems include routing, grooming, survivability, and virtual network design. MASCOT helps implementing a solution to such problems by providing a data model of the network and the demands, classes to handle data and ready to use implementations of existing algorithms or linear programs (e.g. shortest paths or integral multicommodity flow).

A key feature of MASCOT is to provide a generic linear programming object interface which allows users to program the same way whether the target solver is IBM ILOG CPLEX, GLPK (GNU Linear Programming Kit) or CLP/CBC (accessed through JNI).

MASCOT has been intensively used in the past within MASCOTTE industrial cooperation programs for experimentation and validation purposes as for example with Alcatel Space Technologies and Orange Labs. Today, the library is used within the framework of the ANR AGAPE to implement FPT algorithms (work done at LIFO).

See also the web page http://www-sop.inria.fr/mascotte/mascopt/.

5.4. Open Simulation Architecture (OSA)

Participants: Olivier Dalle [correspondant], Van Dan Nguyen, Judicaël Ribault.

Developed in Java (80%) and XML, AspectJ, etc. Represent the work of about 8 man/year during the last 6 years.

Component-based modeling has many well-known good properties. One of these properties is the ability to distribute the modeling effort amongst several experts, each having his/her own area of system expertise. Clearly, the less experts have to care about areas of expertise of others, the more efficient they are in modeling sub-systems in their own area. Furthermore, the process of studying complex systems using discrete-event computer simulations involves several areas of non-system expertise, such as discrete-event techniques or experiment planning.

The Open Simulation Architecture (OSA) [97] is designed to enforce a strong separation of the end-user roles and therefore, ensure a successful cooperation of all the experts involved in the process of simulating complex systems.

The OSA architecture is also intended to meet the expectations of a large part of the discrete-event simulation community: it provides an open platform intended to support researchers in a wide range of their simulation activities, and allows the reuse and sharing of system models in the simulation community by means of a flexible and generic component model (Fractal).
Many discrete-event simulators are developed concurrently, but with identical or similar purpose. Another goal of OSA is to favor the reuse and integration of simulation software components and models. To favor reuse, OSA uses a layered approach to combine the modeling, simulation, and related concerns, such as instrumentation or deployment. This ability is demonstrated by the successful integration and reuse of third-party components, such as ScaFe, the analysis module of Omnet++, or a large number of the James II plugins developed by the University of Rostock. OSA is both a testbed for experimenting new simulation techniques and a tool for real case studies.

OSA is Open Source (LGPL) and is available for download on the INRIA forge server http://osa.gforge.inria.fr/.

See also the web page http://osa.inria.fr/.

5.5. SageMath

Participants: Nathann Cohen [correspondant], David Coudert, Leonardo Sampaio.

Developed in Python, Cython, and C++. N. Cohen wrote more than 180 patches and N. Cohen, D. Coudert and L. Sampaio reviewed more than 120 others for inclusion in Sage.

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). It has currently more than 200 MB of source code and the graph module consists of 40,000 lines. It was initially of interest for Mascotte because of its large library in Combinatorics and Graph Theory. This year, impressive improvements have been made to this library. In particular, N. Cohen contributed a lot into the following: 1) implementation of a generic interface between Sage and existing (Mixed Integer) Linear Program solvers, 2) implementation of exact algorithms for common Polynomial/NP-Complete graph problems, often through the use of Linear Programs, and 3) improving Sage’s documentation by participating to the writing of a french manual on the use of Sage with 10 other french scientists. New patches are in preparation in the group for possible inclusion in Sage.

Sage’s Graph and Linear Programming libraries are currently used by Mascotte members to test algorithms or compare their performances, as well as to prove/disprove theoretical conjectures and for teaching purposes in the Master IFI, stream UBINET.

5.6. Utilities

5.6.1. 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. This module was formerly separated from Java4unix and was referred to as EPR.

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

5.6.2. Jalinopt

Participants: Luc Hogie [correspondant], Grégory Morel.
Developed in Java.

Many mathematical and engineering problems can be expressed as linear programs, and doing so facilitates their resolution. Indeed it is generally more convenient to transform a domain-specific problem into a linear-optimizable one (that can be solved by any solver) rather than writing a complex domain-specific algorithm. In the case of graph theory, problems like flows, minimum vertex cover, maximum stable can be conveniently represented via linear programs.

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. Altought 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.

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

5.6.3. 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/.

5.6.4. 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.

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

5.6.5. P2PVSim

Participant: Remigiusz Modrzejewski [correspondant].

Around 8,000 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.
MODEMIC 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 (INRA Theix), Biomeco (Paris-Grignon), UMR EGC (Paris-Grignon).... and also as a teaching support. See the web page  http://sites.google.com/site/vitelbio/
5. Software

5.1. MAXW-DGTD

Participants: Joseph Charles, Tristan Cabel, 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) or tetrahedral (3D) meshes [14]. 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 message passing programming using the MPI standard. Besides, a peripheral version of the software has been recently developed which is able to exploit the processing capabilities of a hybrid parallel computing system comprising multicore CPU and GPU nodes [20]. Moreover, a recent methodological achievement has been the extension of the implemented DGTD method to deal with a Debye type dispersive propagation medium [35].

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

5.2. MAXW-DGFD

Participants: Mohamed El Bouajaji, 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) or tetrahedral (3D) meshes. Within each element of the mesh, the components of the electromagnetic field are approximated by an 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 message passing programming using the MPI standard. Some recent achievements have been the implementation of non-uniform order DG method in the 2D case [17] and of a new hybridizable discontinuous Galerkin (HDG) formulation also in the 2D case [33].

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

**Participants:** Loula Fezoui, 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 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.

- 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, Tristan Cabel, Thibaud Kloczko [SED 3 team], Régis Duvigneau [OPALE project-team], Thibaud Kloczko [SED team], Stéphane Lanteri, Julien Wintz [SED team].

NUM3SIS [http://num3sis.inria.fr](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].

3Service d’Experimentation et de Développement
Medical Image Extractor http://num3sis.inria.fr/software/apps/numMedicalImageExtractor provides functionalities needed to extract meshes from labeled MR or PET-CT medical images. It puts the emphasis on consistency, 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.

![Figure 2. Medical Image Extractor tool graphical user interface.](image)
4. Software

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

Participants: Bruno Cessac, Hassan Nasser, Pierre Kornprobst [correspondent], Adrien Wohrer [Group for Neural Theory - ENS].

Virtual Retina is a simulation software developed by Adrien Wohrer during his PhD [74], [73] 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 your own sequences. Virtual Retina also offers a web service (v 2.1), so that you may test directly the main software on user’s own data, without any installation. This web service was developed in collaboration with Nicolas Debeissat (engineer, 2002).

Virtual Retina continues it evolution thanks it work done in our team by Bruno Cessac and Hassan Nasser who are interested in the analysis of the collective behavior of ganglion cells responses (see Section 5.1.1).

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 (see, e.g., [68].

- 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. ABFilter: A Simulator Of V1 Simple and Complex Cells

Participants: Pierre Kornprobst [correspondant], Maria-Jose Escobar [Electronics Engineering Department of the Universidad Técnica Federico Santa María, Valparaíso, Chile.], Adrien Wohrer [Group for Neural Theory - ENS].

ABFilter is a C++-library that allows the implementation of spatiotemporal filtering in video sequences using filters proposed by Adelson and Bergen in [62].

Simple cells in V1 are characterized by linear receptive fields where the neuron response is a weighted linear combination of the input stimulus inside its receptive field. By combining two simple cells in a linear manner it is possible to get direction-selective cells. The direction-selectivity refers to the property of a neuron to respond to the direction of the stimulus motion. The way to model this selectivity is by obtaining receptive fields oriented in space and time. Some characteristics of V1 complex cells can be explained using a nonlinear combination of V1 simple cells as it has been proposed by, e.g., Adelson and Bergen in [62]. Implementing these cells properly is a difficult problem and this library offers the possibility to easily implement a V1 layer which can serve as an input to subsequent cortical areas such as MT (see, e.g., the architecture developed in [65]).

The ABFilter library is under a CeCill-C open-source license.

- IDDN.FR.001.280017.000.S.P.2011.000.31235
- Version: v 1.0 (May 2011)
- Download: http://www-sop.inria.fr/neuromathcomp/public/software/abfilter-1.0.tar.gz
4.3. MotionLib: A Neural-Fields Model for Motion Estimation

Participants: Pierre Kornprobst [correspondant], Emilien Tlapale.

MotionLib implements the neural field model of motion estimation described in [52], using the Python programming language.

Motion integration is the core of the model: It implements a two-layer model with feedbacks that selects and diffuses motion signals. The main structure has been implemented here, allowing contributors to enrich this model easily. Several tools are also provided to visualize and analyze the distributed velocity fields obtained by this approach.

- License IDDN.FR.001.210029.000.S.P.2011.000.31235
- Version: v 1.0 (October 2011)
- Download: http://www-sop.inria.fr/neuromathcomp/public/software/motion.zip

4.4. Event neural assembly Simulation

Participants: Frederic Alexandre [INRIA Cortex Nancy], Bruno Cessac [correspondent], Rodrigo Cofre Torres, Jeremy Fix [INRIA Cortex Nancy], Olivier Rochel [INRIA Cortex Nancy], Sélim Kraria, Olivier Marre, Hassan Nasser, Horacio Rostro-Gonzalez, Vivien Robinet, Thierry Viéville [INRIA Cortex Nancy], Juan-Carlos Vasquez.

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 optimaly 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 programing 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.OO1.360008.000.S.P.2009.000.10600.

Its development as a friendly software designed for the neuroscience community is our next purpose (ADT proposal).

Website: http://enas.gforge.inria.fr/
OASIS Project-Team

5. Software

5.1. ProActive


url: 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 (NorduGrid)
- 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] [42] 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, R. Halalai, 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.

- Our main effort this year was based on the development of a quite large case-study, two orders of magnitude larger than our previous experiments. This study was the opportunity to develop new methods for encoding our models using a new combination of CADP formalisms, combining compositional approaches, abstraction techniques, and distributed model-checking [22]. The implementation of these methods in the Vercors tools has started.

We have also been conducting experiments towards the next generation of specification formalism editors for VerCors, using the Papyrus UML-based environment.
5. Software

5.1. NUM3SIS

Participants: Régis Duvigneau [correspondant], Thibaud Kloczko, Nora Aïssiouene.

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 scenarios 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 Duvigneau [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, fault-tolerance and exception-handling capabilities. The platform is used to experiment new resilience algorithms, including monitoring and management of application-level errors. 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. This work is part of Laurentiu Trifan’s PhD thesis. (See Fig. 2.)
Figure 2. Testcase deployment on the Grid5000 infrastructure.
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)
- 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
- OS/Middleware: 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 “pdjkwerl” 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, Anasthesia Fedane.

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, Claudio Freire.

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 2011 we improved the robustness in the experiment control scheme, and we added support for new experimentation environments. We released and registered a second version of the NEPI software (IDDN.FR.001.06003.001.S.A.2010.000.10600). Details on the improvements made can be found in [ 48 ]. See also the web page http://nepihome.org .

- Version: 1.0
- 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
- Programming language: python
5.6. Reference implementation for SFA Federation of experimental testbeds

**Participants:** Thierry Parmentelat [correspondant], Baris Metin, 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](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](https://www.planet-lab.eu/db/doc/PLCAPI.php)
- Codebase: [http://git.onelab.eu](http://git.onelab.eu)

5.7. 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 delivery 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://planete-bcast.inrialpes.fr/](http://planete-bcast.inrialpes.fr/) for more information.

5.8. 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](http://openfec.org) are:
- to share IPR-free, open, AL-FEC codes, to share high performance, ready-to-use, open, free, C-language, software codecs and 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.
5.9. BitHoc  
**Participants:** Chadi Barakat [correspondant], Thierry Turletti, Amir Krifa.

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](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 (http://www.alhem.net/Sockets/, GPL)
- Programming languages: C++, C#
- Documentation: doxygen

5.10. TICP  
**Participant:** Chadi Barakat [correspondant].

TICP 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. TICP 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 TICP is designed with the main objective to be friendly with applications using TCP. Experimental results show that TICP can achieve better performance than using parallel TCP connections for the data collection. The code of TICP is available upon request, it is an open source software under the GPLv3 licence.

See also the web page [http://planete.inria.fr/ticp/](http://planete.inria.fr/ticp/)

- 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
### 5.11. Experimentation Software

**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/](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](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](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/iamine/ns-3-wimax](http://code.nsnam.org/iamine/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](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 in [41] 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 .
PULSAR Project-Team

5. Software

5.1. SUP

SUP is a Scene Understanding Software Platform written in C and C++ (see Figure 2). 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., until scenario recognition. Each module has a precise 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 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, the Orion/Pulsar spin off created in July 2005.

Goals of SUP are twofolds:

1. From a video understanding point of view, to allow the researchers of the Pulsar team can share the implementations of their researches through this platform.
2. From a software engineering point of view, to integrate the results of the dynamic management of the applications when applied to video surveillance.

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 of physical objects of interest, classification of physical objects of interest, 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 add new metrics.

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

1. The video interface defines the way to load the images in the interface. For instance the user can develop her/his plugin based on her/his own video format. The tool is delivered with a plugin to load JPEG image, and ASF video.
2. The object filter selects which objects (e.g. objects far from the camera) are processed to compute 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 (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 is delivered with 4 metrics.

The GUI is composed of different parts (see Figure 3):

- Window 1: the video part 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.
- The object window enables the user to choose the object to be displayed (see Figure 4).
- The multi-view window displays the different points of view of the scene (see Figure 5).

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

The ViSEvAl software was tested and validated into the framework of the Cofriend project through its partners (Akka,...). The tool is also used by IMRA, Nice hospital, Institute for Infocomm Research (Singapore),... The software version 1.0 was delivered to APP (French Program Protection Agency) on August 2010. ViSEvAl is under GNU Affero General Public License AGPL (http://www.gnu.org/licenses/) since July 2011. The tool is available on the web page: http://www-sop.inria.fr/teams/pulsar/EvaluationTool/ViSEvAl_Description.html
Figure 3. GUI of the ViSEvAl software

Figure 4. The object windows enables users to choose the object to display
5.3. Pegase

Since September 1996, the Orion team (and now the Pulsar team) distributes the program supervision engine PEGASE, based on the LAMA platform. The Lisp version has been used at Maryland University and at Genset (Paris). The C++ version (PEGASE+) is now available and is operational at ENSI Tunis (Tunisia) and at CEMAGREF, Lyon (France).

5.4. Clem

The Clem Toolkit [61] (see Figure 6) is a set of tools devoted to design, simulate, verify and generate code for LE [17] [71] programs. This latter is a synchronous language supporting a modular compilation. The language also supports automata possibly designed with a dedicated graphical editor. The Clem toolkit comes with a simulation tool. Hardware description (Vhdl) and software code (C) are generated for LE programs. Moreover, we also generate files to feed the NuSMV model checker [57] in order to perform validation of program behaviors.
Figure 6. The Clem Toolkit
PUMAS 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) is being supported by an ADT action of the D2T. 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. The construction of this platform will integrate the following developments.

1. A (small) database corresponding to axi-symmetrical solutions of the equilibrium plasma equations for realistic geometrical and magnetic configurations (ToreSupra, JET and ITER). The construction of meshes always takes considerable time. Plato will provide meshes and solutions corresponding to equilibrium solutions that will be used as initial data for more complex computations.

2. A set of tool for the handling, manipulation and transformation of meshes and solutions using different discretisations (P1, Q1, P3, etc)

3. Numerical templates allowing the use of 3D discretization schemes using finite element schemes in the poloidal plane and spectral Fourier or structured finite volume representations in the toroidal plane.

4. Several applications (Ideal MHD and drift approximation) used in the framework of the Inria large scale initiative “FUSION”.

This year, after a definition of the PlaTo architecture, the points 1. and 2. have been developed.

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. 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 [21] and in an article currently submitted for publication. 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 team). We then interface software for image matting using the Matting Laplacian, and User-Assissted 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. Imere: Inria Multi-Environment Realistic Simulation Engine

Participants: Adrien David, George Drettakis.

In the context of the ADT Interact3D and the ARC NIEVE, we developed Imere, a middleware to be used as a VR engine, helping in the implementation of realistic simulations for immersive installations. Imere 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 Imere 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 research engineer service at INRIA Sophia-Antipolis Méditerranée. 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 premix them to a representative cluster source, so that all spatialization related processing can be applied only on the representative premixed source [9]. Other audio topics were also considered and developed, like progressive and scalable frequency domain mixing, sound propagation, scalable reverberation, modal sound synthesis and contact sounds generation [1].
In order to maintain all the knowledge in the group and re-use these technologies in the Immersive Space, a previous young engineer, 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 at INRIA Nancy.
SMASH Project-Team (section vide)
5. Software

5.1. CarbonQuant

**Participants:** Mireille Bossy [correspondant], Jacques Morice, El Hadj Aly Dia.

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](http://carbonvalue.gforge.inria.fr).

- Version: 0.1

\(^1\)Technology Development Action
TROPICS Project-Team

5. Software

5.1. AIRONUM

**Participants:** 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 the web page [http://www-sop.inria.fr/tropics/aironum](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 source program into a new source program that computes derivatives of the original program. Automatic Differentiation produces analytical derivatives, that are exact up to machine precision. The reverse mode of Automatic Differentiation is able to compute gradients at a cost which is independent from the number of input variables. TAPENADE accepts source programs written in Fortran 77, Fortran 90, or C. It provides differentiation in the following modes: tangent, multi-directional tangent, and reverse. Documentation is provided on the web site of the research team and as the INRIA technical report RT-0300. TAPENADE runs under Linux or Windows operating systems, and requires installation of Java jdk1.6 or upward.

See also the web page [http://www-sop.inria.fr/tropics/](http://www-sop.inria.fr/tropics/).

- Version: v 3.6, september 2011
- 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.001.040038.000.S.P.2002.000.31235
- 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 is 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. Higher-order derivatives can be obtained through repeated application of tangent AD on tangent and/or reverse 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**: This detects variables whose derivative is either null or useless, to reduce the number of derivative instructions.

- **Adjoint Liveness analysis**: This detects the source statements that are dead code for the computation of derivatives.

- **TBR analysis**: In reverse mode, this 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 robots). The tapenade-users mailing list is over one hundred registered users.
4. Software

4.1. V-Plants

Participants: Frédéric Boudon, Christophe Godin [coordinator], Yann Guédon, Christophe Pradal [software architect], Daniel Barbeau, Thomas Cokelaer, David Da Silva, Jean-Baptiste Durand, Pascal Ferraro, Eric Moscardi.

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 [8]. It is used by many other components to represent the geometry of biological shapes from 3D meristems, plant architectures to plant populations. VPlants.PlantGL is built around a scene-graph data structure and provides efficient algorithms and original geometrical shapes (parametric surfaces, dedicated envelops), that are useful for plant modeling.

- Statistical sequence and tree analysis. Different statistical packages (i.e. VPlants.StatTool, VPlants.SequenceAnalysis, VPlants.TreeMatching and VPlants.TreeAnalysis) are now available in OpenAlea. They provide different models and algorithms for plant architecture analysis and simulation.

- Meristem functioning and development. A first set of components has been created in the last 4-years period to model meristem development in OpenAlea. These tools are currently being integrated thoroughly in the platform so that modelers and biologists can use them, and reuse components easily (for meristem 3D reconstruction, cell tracking, statistical analysis of tissues, creating and manipulating atlases, creating or loading models of growth that can further be run on digitized structures, etc).

- Standard data structure for plants. A new implementation of the MTG formalism for representing and manipulating multiscale plant architecture has been developed. It provides a central data-structure to represent plants in a generic way in OpenAlea. This implementation is available through the packages OpenAlea.MTG. These components make it possible to share plant representations between users and fosters the interoperability of new models.

- Simulation system. The study of plant development requires increasingly powerful modeling tools to help understand and simulate the growth and functioning of plants. In the last decade, the formalism of L-systems has emerged as a major paradigm for modeling plant development. Previous implementations of this formalism were made based on static languages, i.e. languages that require explicit definition of variable types before using them. These languages are often efficient but involve quite a lot of syntactic overhead, thus restricting the flexibility of use for modelers. We developed L-Py [26] 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 has been submitted 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], Daniel Barbeau, Thomas Cokelaer, Christian Fournier, Eric Moscardi.

This research theme is supported by an INRIA ADT Grant and by a RTRA Grant.

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.
**OpenAlea** [9] 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 2011, 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 (link http://vp-continuous.cirad.fr) to test the reliability of all the components after every commit.
   - The OpenAlea project is hosted at the Inria gforge. The web site is visited by more than 300 unique visitor each month, 470000 web pages have been visited and the different available components of OpenAlea have been downloaded more than 500,000 times during the last two years. OpenAlea is the first project at Inria Gforge in term of number of downloads and of page views.

### 4.3. Alinea

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

Other participants: Bruno Andrieu, Michael Chelle, Gaëtan Louarn, Benoît de Solan, Mariem Abichou, Li Qi Han, Elmer Ccopa-Rivera, Frédéric Baret, Rafaële Casa, Youcef Mameri, 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 2011, several research group from INRA and CIRAD have worked together on reconstruction and simulation of plant development for different species of gramineous such as rice, wheat, maize and other species like vine, rose or apple tree.
ZENITH 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. 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.

5.3. P2Prec (P2P recommendation service)

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

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

P2Prec is 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.

5.4. 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.

5.5. SnoopIm
Participants: Julien Champ [contact], Alexis Joly.

SnoopIm is a content-based search engine allowing to 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 collection of photos or 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'sup (http://www.ina-sup.com/).

5.6. SimJoin (Distributed Approximate Similarity Join)
Participant: Alexis Joly [contact].

SimJoin is a distributed software for the efficient computation of the full approximate k-nn graph of large collections of high-dimensional features. It is developed within a MapReduce framework and is therefore easily portable to large cloud computing platform. It is based on recent theoretic contributions related to locality preserving hash functions [34]. Its first main feature is to allow splitting a large collection of high-dimensional features into highly balanced pages that preserve locality according to any given similarity kernel. Its second main feature is to build in $O(n^{1+\gamma})$ operations a candidate set of item pairs that approximate the theoretic knn-graph with high recall. This software is developed in collaboration with INRIA Imedia.