RESEARCH CENTER

FIELD

Activity Report 2014

Section Software

Edition: 2015-03-24

ALGORITHMICS, PROGRAMMING, SOFTWARE AND ARCHITECTURE

ALF Project-Team

5. New Software and Platforms

5.1. Panorama

The ALF team is developing several software prototypes for research purposes: compilers, architectural simulators, programming environments ....

Among the prototypes developed in the project, this section reports only the softwares that had significant revisions in 2014. Among the softwares available from the project website and not reported here, **ATMI** **http://www.irisa.fr/alf/atmi**, a microarchitecture temperature model for processor simulation, **STiMuL** **http://www.irisa.fr/alf/stimul**, a temperature model for steady state studies, **ATC** **http://www.irisa.fr/alf/atc**, an address trace compressor, and **HAVEGE** **http://www.irisa.fr/alf/havege** an unpredictable random number generator.

5.2. TPCalc

Participant: Pierre Michaud.

microarchitecture simulation

TPCalc is a throughput calculator for microarchitecture studies concerned with multi-program workloads consisting of sequential programs. Because microarchitecture simulators are slow, it is difficult to simulate throughput experiments where a multicore executes many jobs that enter and leave the system. The usual practice of measuring instantaneous throughput on independent coschedules chosen more or less randomly is not a rigorous practice because it assumes that all the coschedules are equally important, which is not always true. TPCalc can compute the average throughput of a throughput experiment without actually doing the throughput experiment. The user first defines the workload heterogeneity (number of different job types), the multicore configuration (number of cores and symmetries). TPCalc provides a list of base coschedules. The user then simulates these coschedules, using some benchmarks of his/her choice, and feeds back to TPCalc the measured execution rates (e.g., instructions per cycle or instructions per second).TPCalc eventually outputs the average throughput. Several throughput metrics are available, corresponding to different workload assumptions. These metrics are described in our ACM TACO paper, a collaboration with Ghent University [15].

TPCalc is an open-source software written in C++. It runs on Unix-based systems (Linux, OS X ...). It is available for download at http://www.irisa.fr/alf/downloads/michaud/tpcalc.html.

5.3. Heptane

Participants: Isabelle Puaut, Damien Hardy.

WCET estimation

Status: Registered with APP (Agence de Protection des Programmes). Available under GNU General Public License v3, with number IDDN.FR.001.510039.000.S.P.2003.000.10600.

The aim of Heptane is to produce upper bounds of the execution times of applications. It is targeted at applications with hard real-time requirements (automotive, railway, aerospace domains). Heptane computes WCETs using static analysis at the binary code level. It includes static analyses of microarchitectural elements such as caches and cache hierarchies.

For more information, please contact Damien Hardy or Isabelle Puaut.

5.4. Tiptop

Participant: Erven Rohou.

Performance, hardware counters, analysis tool.

Status: Registered with APP (Agence de Protection des Programmes). Available under GNU General Public License v2, with number IDDN.FR.001.450006.000.S.P.2011.000.10800. Current version is 2.2, released March 2013.

Tiptop has been integrated in major Linux distributions, such as Fedora, Debian, Ubuntu.

Tiptop is a new simple and flexible user-level tool that collects hardware counter data on Linux platforms (version 2.6.31+). The goal is to make the collection of performance and bottleneck data as simple as possible, including simple installation and usage. In particular, we stress the following points.

- Installation is only a matter of compiling the source code. No patching of the Linux kernel is needed, and no special-purpose module needs to be loaded.
- No privilege is required, any user can run *tiptop* — non-privileged users can only watch processes they own, ability to monitor anybody's process opens the door to side-channel attacks.
- The usage is similar to *top*. There is no need for the source code of the applications of interest, making it possible to monitor proprietary applications or libraries. And since there is no probe to insert in the application, understanding of the structure and implementation of complex algorithms and code bases is not required.
- Applications do not need to be restarted, and monitoring can start at any time (obviously, only events that occur after the start of *tiptop* are observed).
- Events can be counted per thread, or per process.
- Any expression can be computed, using the basic arithmetic operators, constants, and counter values.
- A configuration file lets users define their prefered setup, as well as custom expressions.

Tiptop is written in C. It can take advantage of libncurses when available for pseudo-graphic display.

For more information, please contact Erven Rohou or visit http://tiptop.gforge.inria.fr.

5.5. Padrone

Participants: Erven Rohou, Emmanuel Riou.

Performance, profiling, dynamic optimization

Status: Ongoing development, early prototype. Registered with APP (Agence de Protection des Programmes).

Padrone is new platform for dynamic binary analysis and optimization. It provides an API to help clients design and develop analysis and optimization tools for binary executables. Padrone attaches to running applications, only needing the executable binary in memory. No source code or debug information is needed. No application restart is needed either. This is specially interesting for legacy or commercial applications, but also in the context of cloud deployment, where actual hardware is unknown, and other applications competing for hardware resources can vary. The profiling overhead is minimum.

Padrone is instrumental to the PhD developments of Nabil Hallou.

Padrone is written in C.

For more information, please contact Erven Rohou.

5.6. Barra

Participant: Sylvain Collange.

GPU simulator

Other Contributors : David Defour (Université de Perpignan), Alexandre Kouyoumdjian (Inria), Elie Gedeon (ENS Lyon), Fabrice Mouhartem (Inria)

Status : APP registration in progress. Available under the new BSD License

Research on throughput-oriented architectures demands accurate and representative models of GPU architectures in order to be able to evaluate new architectural ideas, explore design spaces and characterize applications. The Barra project [0] is a simulator of the NVIDIA Tesla GPU architecture.

Barra builds upon knowledge acquired through micro-benchmarking, in order to provide a baseline model representative of industry practice. The simulator provides detailed statistics to identify optimization opportunities and is fully customizable to experiment ideas of architectural modifications. Barra incorporates both a functional model and a cycle-level performance model.

Visit http://barra.gforge.inria.fr/ or contact Sylvain Collange.

[0] http://gforge.inria.fr/plugins/mediawiki/wiki/barra/index.php/Main_Page

ANTIQUE Team

5. New Software and Platforms

5.1. The Apron Numerical Abstract Domain Library

Participants: Antoine Miné [correspondent], Bertrand Jeannet [team PopArt, Inria-RA].

The APRON library is dedicated to the static analysis of the numerical variables of a program by abstract interpretation. Its goal is threefold: provide ready-to-use numerical abstractions under a common API for analysis implementers, encourage the research in numerical abstract domains by providing a platform for integration and comparison of domains, and provide a teaching and demonstration tool to disseminate knowledge on abstract interpretation.

The APRON library is not tied to a particular numerical abstraction but instead provides several domains with various precision versus cost trade-offs (including intervals, octagons, linear equalities and polyhedra). A specific C API was designed for domain developers to minimize the effort when incorporating a new abstract domain: only few domain-specific functions need to be implemented while the library provides various generic services and fallback methods (such as scalar and interval operations for most numerical data-types, parametric reduced products, and generic transfer functions for non-linear expressions). For the analysis designer, the APRON library exposes a higher-level API with C, C++, OCaml, and Java bindings. This API is domain-neutral and supports a rich set of semantic operations, including parallel assignments (useful to analyze automata), substitutions (useful for backward analysis), non-linear numerical expressions, and IEEE floating-point arithmetic.

The APRON library is freely available on the web at http://apron.cri.ensmp.fr/library; it is distributed under the LGPL license and is hosted at InriaGForge. Packages exist for the Debian and Fedora Linux distributions. In order to help disseminate the knowledge on abstract interpretation, a simple inter-procedural static analyzer for a toy language is included. An on-line version is deployed at http://pop-art.inrialpes.fr/interproc/interprocweb.cgi.

The APRON library is developed since 2006 and currently consists of 130 000 lines of C, C++, OCaml, and Java.

Current and past external library users include the Constraint team (LINA, Nantes, France), the Proval/Démon team (LRI Orsay, France), the Analysis of Computer Systems Group (New-York University, USA), the Sierum software analysis platform (Kansas State University, USA), NEC Labs (Princeton, USA), EADS CCR (Paris, France), IRIT (Toulouse, France), ONERA (Toulouse, France), CEA LIST (Saclay, France), VERIMAG (Grenoble, France), ENSMP CRI (Fontainebleau, France), the IBM T.J. Watson Research Center (USA), the University of Edinburgh (UK).

Additionally, APRON is used internally by the team to assist the research on numeric domains and static analyses by enabling the development of fast prototypes. Specifically, in 2014, APRON has been used to support the design of piece-wise linear ranking function domains to infer termination and functional liveness properties in the implementation of the FUNCTION prototype analyzer, and to implement and experiment a new numeric domain for octagonal constraints with absolute values. It has also been used in the introductory course on program verification given by members of the team.

5.2. The Astrée Static Analyzer of Synchronous Software

Participants: Patrick Cousot [project scientific leader, correspondent], Radhia Cousot, Jérôme Feret, Laurent Mauborgne, Antoine Miné, Xavier Rival.

ASTRÉE is a static analyzer for sequential programs based on abstract interpretation [40], [35], [41], [36].

The ASTRÉE static analyzer [34], [44][1] www.astree.ens.fr aims at proving the absence of runtime errors in programs written in the C programming language.

ASTRÉE analyzes structured C programs, with complex memory usages, but without dynamic memory allocation nor recursion. This encompasses many embedded programs as found in earth transportation, nuclear energy, medical instrumentation, and aerospace applications, in particular synchronous control/command. The whole analysis process is entirely automatic.

ASTRÉE discovers all runtime errors including:

- undefined behaviors in the terms of the ANSI C99 norm of the C language (such as division by 0 or out of bounds array indexing);
- any violation of the implementation-specific behavior as defined in the relevant Application Binary Interface (such as the size of integers and arithmetic overflows);
- any potentially harmful or incorrect use of C violating optional user-defined programming guidelines (such as no modular arithmetic for integers, even though this might be the hardware choice);
- failure of user-defined assertions.

The analyzer performs an abstract interpretation of the programs being analyzed, using a parametric domain (ASTRÉE is able to choose the right instantiation of the domain for wide families of software). This analysis produces abstract invariants, which over-approximate the reachable states of the program, so that it is possible to derive an *over*-approximation of the dangerous states (defined as states where any runtime error mentioned above may occur) that the program may reach, and produces alarms for each such possible runtime error. Thus the analysis is sound (it correctly discovers *all* runtime errors), yet incomplete, that is it may report false alarms (i.e., alarms that correspond to no real program execution). However, the design of the analyzer ensures a high level of precision on domain-specific families of software, which means that the analyzer produces few or no false alarms on such programs.

In order to achieve this high level of precision, ASTRÉE uses a large number of expressive abstract domains, which allow expressing and inferring complex properties about the programs being analyzed, such as numerical properties (digital filters, floating-point computations), Boolean control properties, and properties based on the history of program executions.

ASTRÉE has achieved the following two unprecedented results:

- **A340–300.** In Nov. 2003, ASTRÉE was able to prove completely automatically the absence of any RTE in the primary flight control software of the Airbus A340 fly-by-wire system, a program of 132,000 lines of C analyzed in 1h20 on a 2.8 GHz 32-bit PC using 300 MB of memory (and 50mn on a 64-bit AMD Athlon 64 using 580 MB of memory).
- **A380.** From Jan. 2004 on, ASTRÉE was extended to analyze the electric flight control codes then in development and test for the A380 series. The operational application by Airbus France at the end of 2004 was just in time before the A380 maiden flight on Wednesday, 27 April, 2005.

These research and development successes have led to consider the inclusion of ASTRÉE in the production of the critical software for the A350. ASTRÉE is currently industrialized by AbsInt Angewandte Informatik GmbH and is commercially available.

5.3. The AstréeA Static Analyzer of Asynchronous Software

Participants: Patrick Cousot [project scientific leader, correspondent], Radhia Cousot, Jérôme Feret, Antoine Miné, Xavier Rival.

ASTRÉEA is a static analyzer prototype for parallel software based on abstract interpretation [42], [43], [37]. It started with support from THÉSÉE ANR project (2006–2010) and is continuing within the ASTRÉEA project (2012–2015).

The ASTRÉEA prototype www.astreea.ens.fr is a fork of the ASTRÉE static analyzer (see 5.2) that adds support for analyzing parallel embedded C software.

ASTRÉEA analyzes C programs composed of a fixed set of threads that communicate through a shared memory and synchronization primitives (mutexes, FIFOs, blackboards, etc.), but without recursion nor dynamic creation of memory, threads nor synchronization objects. ASTRÉEA assumes a real-time scheduler, where thread scheduling strictly obeys the fixed priority of threads. Our model follows the ARINC 653 OS specification used in embedded industrial aeronautic software. Additionally, ASTRÉEA employs a weakly-consistent memory semantics to model memory accesses not protected by a mutex, in order to take into account soundly hardware and compiler-level program transformations (such as optimizations). ASTRÉEA checks for the same run-time errors as ASTRÉE, with the addition of data-races.

Compared to ASTRÉE, ASTRÉEA features: a new iterator to compute thread interactions, a refined memory abstraction that takes into account the effect of interfering threads, and a new scheduler partitioning domain. This last domain allows discovering and exploiting mutual exclusion properties (enforced either explicitly through synchronization primitives, or implicitly by thread priorities) to achieve a precise analysis.

ASTRÉEA is currently being applied to analyze a large industrial avionic software: 1.6 MLines of C and 15 threads, completed with a 2,500-line model of the ARINC 653 OS developed for the analysis. The analysis currently takes a few tens of hours on a 2.9 GHz 64-bit intel server using one core and generates around 1,050 alarms. The low computation time (only a few times larger than the analysis time by ASTRÉE of synchronous programs of a similar size and structure) shows the scalability of the approach (in particular, we avoid the usual combinatorial explosion associated to thread interleavings). Precision-wise, the result, while not as impressive as that of ASTRÉE, is quite encouraging. The development of ASTRÉEA continues within the scope of the ASTRÉEA ANR project.

5.4. ClangML: A binding with the CLANG C-frontend

Participants: François Berenger [Correspondent], Devin Mccoughlin, Pippijn Van Steenhoeven.

CLANGML is an OCaml binding with the CLANG front-end of the LLVM compiler suite. Its goal is to provide an easy to use solution to parse a wide range of C programs, that can be called from static analysis tools implemented in OCaml, which allows to test them on existing programs written in C (or in other idioms derived from C) without having to redesign a front-end from scratch. CLANGML features an interface to a large set of internal AST nodes of CLANG, with an easy to use API. Currently, CLANGML supports all C language AST nodes, as well as a large part of the C nodes related to C++ and Objective-C.

It has been applied to the parsing of the Minix micro-kernel as well as of other C programs.

CLANGML has been implemented in C++, OCaml and Camlp4. It has been released as an open source contribution on GitHUB and as an OPAM package.

5.5. FuncTion: An Abstract Domain Functor for Termination

Participants: Caterina Urban, Antoine Miné [Correspondent].

FUNCTION is a research prototype static analyzer to analyze the termination and functional liveness properties of programs. It accepts programs in a small non-deterministic imperative language. It is also parameterized by a property: either termination, or a recurrence or a guarantee property (according to the classification by Manna and Pnueli of program properties). It then performs a backward static analysis that automatically infers sufficient conditions at the beginning of the program so that all executions satisfying the conditions also satisfy the property. FUNCTION is based on an extension to liveness properties of the framework to analyze termination by abstract interpretation proposed by Patrick Cousot and Radhia Cousot in [39]. FUNCTION infers ranking functions using piecewise-defined abstract domains. Several domains are available to partition the ranking function, including intervals, octagons, and polyhedra. Two domains are also available to represent the value of ranking functions: a domain of affine ranking functions, and a domain of ordinal-valued ranking functions (which allows handling programs with unbounded non-determinism).

The analyzer is written in OCaml and implemented on top of the APRON library. It can be used on-line through a web interface: http://www.di.ens.fr/~urban/FuncTion.html.

FUNCTION participated to SV-COMP 2014 (3rd Competition on Software Verification, demonstration section) and is also selected to participate to SV-COMP 2015 in the termination category [31].

5.6. HOO: Heap Abstraction for Open Objects

Participant: Arlen Cox [Correspondent].

JSAna with HOO is a static analyzer for JavaScript programs. The primary component, HOO, which is designed to be reusable by itself, is an abstract domain for a dynamic language heap. A dynamic language heap consists of open, extensible objects linked together by pointers. Uniquely, HOO abstracts these extensible objects, where attribute/field names of objects may be unknown. Additionally, it contains features to keeping precise track of attribute name/value relationships as well as calling unknown functions through desynchronized separation.

As a library, HOO is useful for any dynamic language static analysis. It is designed to allow abstractions for values to be easily swapped out for different abstractions, allowing it to be used for a wide-range of dynamic languages outside of JavaScript.

5.7. The MemCADstatic analyzer

Participants: Xavier Rival [correspondent], François Berenger, Huisong Li, Antoine Toubhans.

Shape analysis. MEMCADis a static analyzer that focuses on memory abstraction. It takes as input C programs, and computes invariants on the data structures manipulated by the programs. It can also verify memory safety. It comprises several memory abstract domains, including a flat representation, and two graph abstractions with summaries based on inductive definitions of data-structures, such as lists and trees and several combination operators for memory abstract domains (hierarchical abstraction, reduced product). The purpose of this construction is to offer a great flexibility in the memory abstraction, so as to either make very efficient static analyses of relatively simple programs, or still quite efficient static analyses of very involved pieces of code. The implementation consists of over 30 000 lines of ML code, and relies on the CLANGML front-end. The current implementation comes with over 300 small size test cases that are used as regression tests.

5.8. The OpenKappa Modeling Plateform

Participants: Pierre Boutillier [Paris VII], Monte Brown [Harvard Medical School], Vincent Danos [University of Edinburgh], Jérôme Feret [Correspondent], Luca Grieco, Walter Fontana [Harvard Medical School], Russ Harmer [ENS Lyon], Jean Krivine [Paris VII].

Causal traces, Model reduction, Rule-based modeling, Simulation, Static analysis. OPENKAPPA is a collection of tools to build, debug and run models of biological pathways. It contains a compiler for the Kappa Language [50], a static analyzer [49] (for debugging models), a simulator [48], a compression tool for causal traces [47], [45], and a model reduction tool [4], [46], [53].

OPENKAPPA is developed since 2007 and, the OCaml version currently consists of 46 000 lines of OCaml. Software are available in OCaml and in Java. Moreover, an Eclipse pluggin is available.A compiler from CellDesigner into Kappa has been released in 2013.

OPENKAPPA is freely available on the web at http://kappalanguage.org under the LGPL license. Discussion groups are also available on line.

Current external users include the ETH Zürich, the UNAM-Genomics Mexico team. It is used as pedagocical material in graduate lessons at Harvard Medical School, and at the Interdisciplinary Approaches to Life science (AIV) Master Program (Université de Médecine Paris-Descartes).

5.9. QUICr set abstract domain

Participant: Arlen Cox [Correspondent].

QUICr is an OCaml library that implements a parametric abstract domain for sets. It is constructed as a functor that accepts any numeric abstract domain that can be adapted to the interface and produces an abstract domain for sets of numbers combined with numbers. It is relational, flexible, and tunable. It serves as a basis for future exploration of set abstraction.

5.10. Translation Validation

Participant: Xavier Rival [correspondent].

Abstract interpretation, Certified compilation, Static analysis, Translation validation, Verifier. The main goal of this software project is to make it possible to certify automatically the compilation of large safety critical software, by proving that the compiled code is correct with respect to the source code: When the proof succeeds, this guarantees that no compiler bug did cause incorrect code to be generated. Furthermore, this approach should allow to meet some domain specific software qualification criteria (such as those in DO-178 regulations for avionics software), since it allows proving that successive development levels are correct with respect to each other i.e., that they implement the same specification. Last, this technique also justifies the use of source level static analyses, even when an assembly level certification would be required, since it establishes separately that the source and the compiled code are equivalent.

The compilation certification process is performed automatically, thanks to a prover designed specifically. The automatic proof is done at a level of abstraction which has been defined so that the result of the proof of equivalence is strong enough for the goals mentioned above and so that the proof obligations can be solved by efficient algorithms.

The current software features both a C to Power-PC compilation certifier and an interface for an alternate source language frontend, which can be provided by an end-user.

5.11. Zarith

Participants: Antoine Miné [Correspondent], Xavier Leroy [Inria Paris-Rocquencourt], Pascal Cuoq [CEA LIST].

ZARITH is a small (10K lines) OCaml library that implements arithmetic and logical operations over arbitrary-precision integers. It is based on the GNU MP library to efficiently implement arithmetic over big integers. Special care has been taken to ensure the efficiency of the library also for small integers: small integers are represented as Caml unboxed integers and use a specific C code path. Moreover, optimized assembly versions of small integer operations are provided for a few common architectures.

ZARITH is an open-source project hosted at OCamlForge (http://forge.ocamlcore.org/projects/zarith) and distributed under a modified LGPL license.

ZARITH is currently used in the ASTRÉE analyzer to enable the sound analysis of programs featuring 64-bit (or larger) integers. It is also used in the Frama-C analyzer platform developed at CEA LIST and Inria Saclay.

AOSTE Project-Team

5. New Software and Platforms

5.1. TimeSquare

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

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

TimeSquare offers five main functionalities:

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

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

5.2. K-Passa

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

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

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

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

5.3. SynDEx

Participants: Yves Sorel [correspondant], Meriem Zidouni.

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

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

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

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

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

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

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

5.4. Lopht

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

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

Improved efficiency is attained through the use of classical and novel data structures and optimization algorithms pertaining to 3 fields: synchronous language compilation, classical compiler theory, and real-time scheduling. A finer representation of execution conditions allows us to make a better use of double resource reservation and thus improve latency and throughput. The use of software pipelining allows the improvement of computation throughput. The use of post-scheduling optimizations allows a reduction in the number of preemptions. The focus on novel architectures means that architecture descriptions need to define novel communication media such as the networks-on-chips (NoCs), and that real-time characteristics must include those specific to a time-triggered execution model, such as the Major Time Frame (MTF). Attaining efficiency alse requires a fine-grain description of more classical platform resources, such as the multi-bank RAMs, to allow efficient allocation during scheduling.

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

5.5. SAS

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

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

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

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

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

ARIC Project-Team

5. New Software and Platforms

5.1. Overview

AriC software realizations are accessible from the web page http://www.ens-lyon.fr/LIP/AriC/ware. We describe below only those which progressed in 2014.

5.2. GNU MPFR

Participant: Vincent Lefèvre [correspondant].

GNU MPFR is an efficient multiple-precision floating-point library with well-defined semantics (copying the good ideas from the IEEE-754 standard), in particular correct rounding in 5 rounding modes. GNU MPFR provides about 80 mathematical functions, in addition to utility functions (assignments, conversions...). Special data (*Not a Number*, infinities, signed zeros) are handled like in the IEEE-754 standard.

MPFR was one of the main pieces of software developed by the old SPACES team at Loria. Since late 2006, with the departure of Vincent Lefèvre to Lyon, it has become a joint project between the Caramel (formerly SPACES then CACAO) and the AriC (formerly Arénaire) project-teams. MPFR has been a GNU package since 26 January 2009.

An MPFR-MPC developers meeting took place from 20 to 22 January 2014 in Nancy. There was no new release this year, but various developments were done in the trunk.

The main work done in the AriC project-team:

- Changed the behavior of the `mpfr_set_exp` function to avoid undefined behavior in some cases (this change mainly impacted the internal usage).
- Bug fixes and various improvements (portability, efficiency, etc.).
- The `mpfr_sum` function is being rewritten (`new-sum` branch); see Section 6.2.8 .

URL: http://www.mpfr.org/

GNU MPFR is on the Black Duck Open Hub community platform for free and open source software: https://www.openhub.net/p/gnu-mpfr

- ACM: D.2.2 (Software libraries), G.1.0 (Multiple precision arithmetic), G.4 (Mathematical software).
- AMS: 26-04 Real Numbers, Explicit machine computation and programs.
- APP: no longer applicable (copyright transferred to the Free Software Foundation).
- License: LGPL version 3 or later.
- Type of human computer interaction: C library, callable from C or other languages via third-party interfaces.
- OS/Middleware: any OS, as long as a C compiler is available.
- Required library or software: GMP.
- Programming language: C.
- Documentation: API in texinfo format (and other formats via conversion); algorithms are also described in a separate document.

5.3. Exhaustive Tests for the Correct Rounding of Mathematical Functions

Participant: Vincent Lefèvre.

The search for the worst cases for the correct rounding (hardest-to-round cases) of mathematical functions (exp, log, sin, cos, etc.) in a fixed precision (mainly double precision) using Lefèvre's algorithm is implemented by a set of utilities written in Perl, with calls to Maple/intpakX for computations on intervals and with C code generation for fast computations. It also includes a client-server system for the distribution of intervals to be tested and for tracking the status of intervals (fully tested, being tested, aborted).

The Perl scripts have been improved (in particular, for the interaction with Grid Engine).

5.4. FPLLL: A Lattice Reduction Library

Participant: Damien Stehlé [correspondant].

fplll contains several algorithms on lattices that rely on floating-point computations. This includes implementations of the floating-point LLL reduction algorithm, offering different speed/guarantees ratios. It contains a "wrapper" choosing the estimated best sequence of variants in order to provide a guaranteed output as fast as possible. In the case of the wrapper, the succession of variants is oblivious to the user. It also includes a rigorous floating-point implementation of the Kannan-Fincke-Pohst algorithm that finds a shortest non-zero lattice vector, and the BKZ reduction algorithm.

The fplll library is used or has been adapted to be integrated within several mathematical computation systems such as Magma, Sage, and PariGP. It is also used for cryptanalytic purposes, to test the resistance of cryptographic primitives.

This year, several improvements to the BKZ (block Korkine Zolotarev) algorithm have been implemented. Further, the library is now hosted on `github`.

URL: https://github.com/dstehle/fplll

- ACM: D.2.2 (Software libraries), G.4 (Mathematical software)
- APP: Procedure started
- License: LGPL v2.1
- Type of human computer interaction: C++ library callable, from any C++ program.
- OS/Middleware: any, as long as a C++ compiler is available.
- Required library or software: MPFR and GMP.
- Programming language: C++.
- Documentation: available in html format on **URL:**https://github.com/dstehle/fplll

5.5. Sipe

Participant: Vincent Lefèvre.

Sipe is a mini-library in the form of a C header file, to perform radix-2 floating-point computations in very low precisions with correct rounding, either to nearest or toward zero. The goal of such a tool is to do proofs of algorithms/properties or computations of tight error bounds in these precisions by exhaustive tests, in order to try to generalize them to higher precisions. The currently supported operations are addition, subtraction, multiplication (possibly with the error term), fused multiply-add/subtract (FMA/FMS), and miscellaneous comparisons and conversions. Sipe provides two implementations of these operations, with the same API and the same behavior: one based on integer arithmetic, and a new one based on floating-point arithmetic.

New in 2014:

- `sipe_to_mpfr` function;
- support for `__float128` from GCC/libquadmath (implementing the binary128 format);
- some corrections.

URL: https://www.vinc17.net/research/sipe/

- ACM: D.2.2 (Software libraries), G.4 (Mathematical software).
- AMS: 26-04 Real Numbers, Explicit machine computation and programs.
- License: LGPL version 2.1 or later.
- Type of human computer interaction: C header file.
- OS/Middleware: any OS.
- Required library or software: GCC compiler.
- Programming language: C.
- Documentation: comment at the beginning of the code and Research report Inria RR-7832.

5.6. Gfun

Participant: Bruno Salvy.

Gfun is a Maple package for the manipulation of linear recurrence or differential equations. It provides tools for guessing a sequence or a series from its first terms; for manipulating rigorously solutions of linear differential or recurrence equations, using the equation as a data-structure. This year, the implementation effort was focused on speeding up the guessing routines in the case of sequences with symbolic parameters that come up in general hypergeometric identities.

ATEAMS Project-Team

4. New Software and Platforms

4.1. MicroMachinations

Participant: Riemer Van Rozen [correspondent].

Characterization: A-2-up3, SO-4, SM-2-up3, EM-3, SDL-3-up4, OC-DA-3-CD-3-MS-3-TPM-3.

WWW:

Objective: To create an integrated, live environment for modeling and evolving game economies. This will allow game designers to experiment with different strategies to realize game mechanics. The environment integrates with the SPIN model checker to prove properties (reachability, liveness). A runtime system for executing game economies allows MicroMachinations models to be embedded in actual games.

Users: Game designers

Impact: One of the important problems in game software development is the distance between game design and implementation in software. MicroMachinations has the potential to bridge this gap by providing live design tools that directly modify or create the desired software behaviors.

Competition: None.

Engineering: The front-end of MicroMachinations is built using the Rascal language workbench, including visualization, model checking, debugging and standard IDE features. The runtime library is implemented in C++ and will be evaluated in the context of industrial game design.

Publications: [11]

4.1.1. Novelties

- MMLib was finished to allow the execution of game economies directly within games. This supports "Live programming" of the behavior of games. The library has been used in the development of the real-life game "Johnny Jetstream", designed by IC3DMedia.

4.2. Naked Object Algebras

Participant: Tijs Van Der Storm [correspondent].

Characterization: A5, SO-4, SM-4, EM-4, SDL-4-up5, OC-DA-3-CD-3-MS-3-TPM-3.

WWW: https://github.com/cwi-swat/naked-object-algebras

Objective: Supporting modular and extensible language development.

Users: Programmers, language designers.

Impact: Object Algebras promise a new level of modularity and extensibility in the implementation of recursive data types. The NAO framework lifts this to the implementation of software languages, including the declarative declaration of concrete syntax.

Competition: Language prototyping tools.

Engineering: NAO consists of a few hundred lines of Java code. It has no external dependencies, except ANTLR for parsing.

Publications: [27], [33]

4.2.1. Novelties

- NAO has been used to develop an extensible variant of the QL questionnaire language [33].

4.3. Rascal

Participants: Paul Klint, Jurgen Vinju [correspondent], Tijs Van Der Storm, Pablo Inostroza Valdera, Davy Landman, Bert Lisser, Atze Van Der Ploeg, Vadim Zaytsev, Anastasia Izmaylova, Michael Steindorfer, Jouke Stoel, Ali Afroozeh, Ashim Shahi.

Characterization: A5, SO-4, SM-4, EM-4, SDL-4-up5, OC-DA-3-CD-3-MS-3-TPM-3.

WWW: http://www.rascal-mpl.org

Objective: Provide a completely integrated programming language parametric meta programming language for the construction of any kind of meta program for any kind of programming language: analysis, transformation, generation, visualization.

Users: Researchers in model driven engineering, programming languages, software engineering, software analysis, as well as practitioners that need specialized tools.

Impact: Rascal is making the mechanics of meta programming into a non-issue. We can now focus on the interesting details of the particular fact extraction, model, source analysis, domain analysis as opposed to being distracted by the engineering details. Simple things are easy in Rascal and complex things are manageable, due to the integration, the general type system and high-level programming features.

Competition: There is a plethora of meta programming toolboxes and frameworks available, ranging from plain parser generators to fully integrated environments. Rascal is distinguished because it is a programming language rather than a specification formalism and because it completely integrates different technical domains (syntax definition, term rewriting, relational calculus). For simple tools, Rascal competes with scripting languages and for complex tools it competes context-free general parser generators, with query engines based on relational calculus and with term rewriting and strategic programming languages.

Engineering: Rascal is about 100 kLOC of Java code, designed by a core team of three and with a team of around 8 PhD students and post-docs contributing to its design, implementation and maintenance. The goal is to work towards more bootstrapping and less Java code as the project continues.

Publications: [7], [6], [8], [5], [6]

4.3.1. Novelties

- Improvements of the language-parametric model to represent software projects (M3) [9].
- Performance improvements of the Rascal interpreter throughout.
- Further improvements to the compiler for Rascal, based on new language construct guarded coroutines.
- New language feature: keyword parameters. This will further allow simplificiation of the core language, as well as support better extensibility.
- Significant improvements to the Rascal static type checker.
- Further improvements to the new GLL parser (Iguana).
- Design of a new DSL for describing core banking infrastructure was started (ReBEL). Rascal was also used to develop a state machine DSL for use in embedded devices (Machino).

4.4. IDE Meta-tooling Platform

Participants: Jurgen Vinju [correspondent], Michael Steindorfer.

IMP, the IDE meta tooling platform is an Eclipse plugin developed mainly by the team of Robert M. Fuhrer at IBM TJ Watson Research institute. It is both an abstract layer for Eclipse, allowing rapid development of Eclipse based IDEs for programming languages, and a collection of meta programming tools for generating source code analysis and transformation tools.

Characterization: A5, SO-3, SM4-up5, EM-4, SDL-5, DA-2-CD-2-MS-2-TPM-2

WWW: https://github.com/impulse-org/

Objective: The IDE Meta Tooling Platform (IMP) provides a high-level abstraction over the Eclipse API such that programmers can extend Eclipse with new programming languages or domain specific languages in a few simple steps. IMP also provides a number of standard meta tools such as a parser generator and a domain specific language for formal specifications of configuration parameters.

Users: Designers and implementers of IDEs for programming languages and domain specific languages. Also, designers and implementers of meta programming tools.

Impact: IMP is a popular among meta programmers especially for it provides the right level of abstraction.

Competition: IMP competes with other Eclipse plugins for meta programming (such as Model Driven Engineering tools), but its API is more general and more flexible. IMP is a programmers framework rather than a set of generators.

Engineering: IMP is a long-lived project of many contributors, which is managed as an Eclipse incubation project at `eclipse.org`. Currently we are moving the project to Github to explore more different ways of collaboration.

Publications: [2] [29]

4.4.1. Novelties

- Significant performance improvements to the IMP program database. Performance is now better than equivalent data structure libraries in Scala and Clojure.

4.5. Ensō

Participant: Tijs Van Der Storm [correspondent].

Characterization: A5, SO-4, SM-3-up-4, EM-2-up-4, SDL-4, OC-DA-4-CD-4-MS-4-TPM-4

WWW: http://www.enso-lang.org

Objective: Together with Prof. Dr. William R. Cook of the University of Texas at Austin, and Alex Loh, Tijs van der Storm has been designing and implementing a new programming system, called Ensō. Ensō is theoretically sound and practical reformulation of model-based development. It is based on model-interpretation as opposed to model transformation and code generation. Currently, the system already supports models for schemas (data models), web applications, context-free grammars, diagram editors and security.

Users: All programmers.

Impact: Ensō has the potential to revolutionize the activity of programming. By looking at model driven engineering from a completely fresh perspective, with as key ingredients interpreters and partial evaluation, it may make higher level (domain level) program construction and maintenance as effective as normal programming.

Competition: Ensō competes as a programming paradigm with model driven engineering tools and generic programming and languages that provide syntax macros and language extensions.

Engineering: Ensō is a completely self-hosted system in 7000 lines of code.

Publications: [14], [16], [13]

CAIRN Project-Team

5. New Software and Platforms

5.1. Panorama

With the ever raising complexity of embedded applications and platforms, the need for efficient and customizable compilation flows is stronger than ever. This need of flexibility is even stronger when it comes to research compiler infrastructures that are necessary to gather quantitative evidence of the performance/energy or cost benefits obtained through the use of reconfigurable platforms. From a compiler point of view, the challenges exposed by these complex reconfigurable platforms are quite significant, since they require the compiler to extract and to expose an important amount of coarse and/or fine grain parallelism, to take complex resource constraints into consideration while providing efficient memory hierarchy and power management.

Because they are geared toward industrial use, production compiler infrastructures do not offer the level of flexibility and productivity that is required for compiler and CAD tool prototyping. To address this issue, we have designed an extensible source-to-source compiler infrastructure that takes advantage of leading edge model-driven object-oriented software engineering principles and technologies.

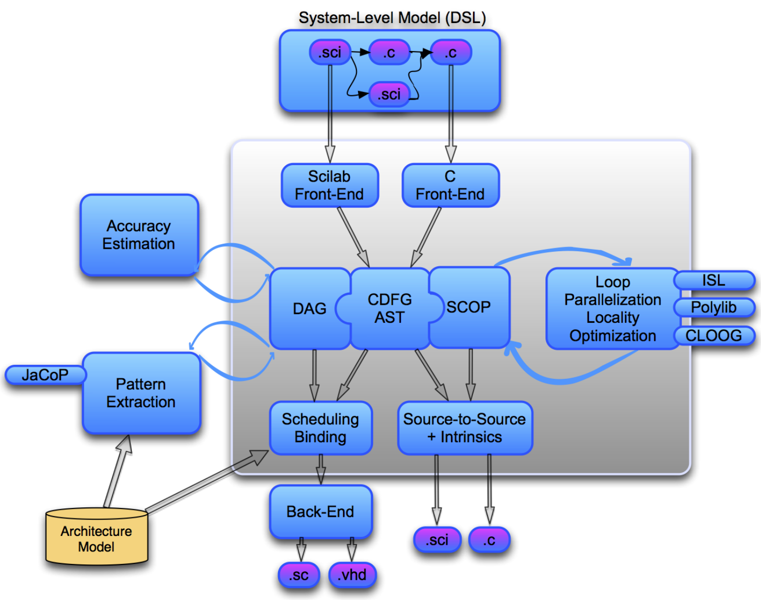


Figure 2. CAIRN's general software development framework.

Figure 2 shows the global framework that is being developed in the group. Our compiler flow mixes several types of intermediate representations. The baseline representation is a simple tree-based model enriched with control flow information. This model is mainly used to support our source-to-source flow, and serves as the backbone for the infrastructure. We use the extensibility of the framework to provide more advanced representations along with their corresponding optimizations and code generation plug-ins. For example, for our pattern selection and accuracy estimation tools, we use a data dependence graph model in all basic blocks instead of the tree model. Similarly, to enable polyhedral based program transformations and analysis, we introduced a specific representation for affine control loops that we use to derive a Polyhedral Reduced Dependence Graph (PRDG). Our current flow assumes that the application is specified as a system level hierarchy of communicating tasks, where each task is expressed using C (or Scilab in the short future), and where the system level representation and the target platform model are defined using Domain Specific Languages (DSL).

Gecos (Generic Compiler Suite) is the main backbone of CAIRN's flow. It is an open source Eclipse-based flexible compiler infrastructure developed for fast prototyping of complex compiler passes. Gecos is a 100% Java based implementation and is based on modern software engineering practices such as Eclipse plugin or model-driven software engineering with EMF (Eclipse Modeling Framework). As of today, our flow offers the following features:

- An automatic floating-point to fixed-point conversion flow (for HLS and embedded processors). **ID.Fix** is an infrastructure for the automatic transformation of software code aiming at the conversion of floating-point data types into a fixed-point representation. http://idfix.gforge.inria.fr.
- A polyhedral-based loop transformation and parallelization engine (mostly targeted at HLS). http://gecos.gforge.inria.fr. It was used for source-to-source transformations in the context of Nano2012 projects in collaboration with STMicroelectronics.
- A custom instruction extraction flow (for ASIP and dynamically reconfigurable architectures). **Durase** and **UPaK** are developed for the compilation and the synthesis targeting reconfigurable platforms and the automatic synthesis of application specific processor extensions. They use advanced technologies, such as graph matching and graph merging together with constraint programming methods.
- Several back-ends to enable the generation of VHDL for specialized or reconfigurable IPs, and SystemC for simulation purposes (e.g., fixed-point simulations).

5.2. Gecos

Participants: Steven Derrien [corresponding author], Nicolas Simon, Antoine Morvan.

Keywords: source-to-source compiler, model-driven software engineering, retargetable compilation.

The Gecos (Generic Compiler Suite) project is a source-to-source compiler infrastructure developed in the Cairn group since 2004. It was designed to enable fast prototyping of program analysis and transformation for hardware synthesis and retargetable compilation domains.

Gecos is 100% Java based and takes advantage of modern model driven software engineering practices. It uses the Eclipse Modeling Framework (EMF) as an underlying infrastructure and takes benefits of its features to make it easily extensible. Gecos is open-source and is hosted on the Inria gforge at http://gecos.gforge.inria.fr.

The Gecos infrastructure is still under very active development, and serves as a backbone infrastructure to projects of the group. Part of the framework is jointly developed with Colorado State University and since 2012 it is used in the context of the ALMA European project. Recent developments in Gecos have focused on polyhedral loop transformations and efficient SIMD code generation for fixed point arithmetic data-types as a part of the ALMA project. Significant efforts were also put to provide a coarse-grain parallelization engine targeting the data-flow actor model in the context of the COMPA ANR project.

5.3. ID.Fix: Infrastructure for the Design of Fixed-point Systems

Participants: Olivier Sentieys [corresponding author], Romuald Rocher, Nicolas Simon.

Keywords: fixed-point arithmetic, source-to-source code transformation, accuracy optimization, dynamic range evaluation

The different techniques proposed by the team for fixed-point conversion are implemented on the ID.Fix infrastructure. The application is described with a C code using floating-point data types and different pragmas, used to specify parameters (dynamic, input/output word-length, delay operations) for the fixed-point conversion. This tool determines and optimizes the fixed-point specification and then, generates a C code using fixed-point data types (ac_fixed) from Mentor Graphics. The infrastructure is made-up of two main modules corresponding to the fixed-point conversion (ID.Fix-Conv) and the accuracy evaluation (ID.Fix-Eval)

The different developments carried out in 2014 allowed to have a complete compatibility with GeCos. The structure of each node in the graph has been changed to simplify the graph modifications. The Octave software has been added instead of Matlab for LTI and recursive systems conversion. A development has started to replace Matlab/Octave tool by a C code algorithm to reduce optimization time. In the context of the ANR DEFIS project, the ID.Fix tool has been reorganized to be integrated in the DEFIS toolflow.

In 2014, ID.Fix has been demonstrated during University Booth at IEEE/ACM DATE.

5.4. UPaK: Abstract Unified Pattern-Based Synthesis Kernel for Hardware and Software Systems

Participants: Christophe Wolinski [corresponding author], François Charot.

Keywords: compilation for reconfigurable systems, pattern extraction, constraint-based programming.

We are developing (with strong collaboration of Lund University, Sweden and Queensland University, Australia) UPaK *Abstract Unified Pattern Based Synthesis Kernel for Hardware and Software Systems* [117]. The preliminary experimental results obtained by the UPak system show that the methods employed in the systems enable a high coverage of application graphs with small quantities of patterns. Moreover, high application execution speed-ups are ensured, both for sequential and parallel application execution with processor extensions implementing the selected patterns. UPaK is one of the basis for our research on compilation and synthesis for reconfigurable platforms. It is based on the HCDG representation of the Polychrony software designed at Inria-Rennes in the project-team Espresso.

5.5. DURASE: Automatic Synthesis of Application-Specific Processor Extensions

Participants: Christophe Wolinski [corresponding author], François Charot.

Keywords: compilation for reconfigurable systems, instruction-set extension, pattern extraction, graph covering, constraint-based programming.

We are developing a framework enabling the automatic synthesis of application specific processor extensions. It uses advanced technologies, such as algorithms for graph matching and graph merging together with constraints programming methods. The framework is organized around several modules.

- CoSaP: Constraint Satisfaction Problem. The goal of CoSaP is to decouple the statement of a constraint satisfaction problem from the solver used to solve it. The CoSaP model is an Eclipse plugin described using EMF to take advantage of the automatic code generation and of various EMF tools.
- HCDG: Hierarchical Conditional Dependency Graph. HCDG is an intermediate representation mixing control and data flow in a single acyclic representation. The control flow is represented as hierarchical guards specifying the execution or the definition conditions of nodes. It can be used in the Gecos compilation framework via a specific pass which translates a CDFG representation into an HCDG.
- Patterns: Flexible tools for identification of computational pattern in a graph and graph covering. These tools model the concept of pattern in a graph and provide generic algorithms for the identification of pattern and the covering of a graph. The following sub-problems are addressed: (sub)-graphs isomorphism, patterns generation under constraints, covering of a graph using a library of patterns. Most of the implemented algorithms use constraints programming and rely on the CoSaP module to solve the optimization problem.

5.6. PowWow: Power Optimized Hardware and Software FrameWork for Wireless Motes (AP-L-10-01)

Participants: Olivier Sentieys [corresponding author], Olivier Berder, Arnaud Carer, Steven Derrien.

Keywords: Wireless Sensor Networks, Low Power, Preamble Sampling MAC Protocol, Hardware and Software Platform

PowWow is an open-source hardware and software platform designed to handle wireless sensor network (WSN) protocols and related applications. Based on an optimized preamble sampling medium access (MAC) protocol, geographical routing and `protothread` library, PowWow requires a lighter hardware system than Zigbee [79] to be processed (memory usage including application is less than 10kb). Therefore, network lifetime is increased and price per node is significantly decreased.

CAIRN's hardware platform (see Figure 3) is composed of:

- The motherboard, designed to reduce power consumption of sensor nodes, embeds an MSP430 microcontroller and all needed components to process PowWow protocol except radio chip. JTAG, RS232, and I2C interfaces are available on this board.
- The radio chip daughter board is currently based on a TI CC2420.
- The coprocessing daughter board includes a low-power FPGA which allows for hardware acceleration for some PowWow features and also includes dynamic voltage scaling features to increase power efficiency. The current version of PowWow integrates an Actel IGLOO AGL250 FPGA and a programmable DC-DC converter. We have shown that gains in energy of up to 700 can be obtained by using FPGA acceleration on functions like CRC-32 or error detection with regards to a software implementation on the MSP430.
- Finally, a last daughter board is dedicated to energy harvesting techniques. Based on the energy management component LTC3108 from Linear Technologies, the board can be configured with several types of stored energy (batteries, micro-batteries, super-capacitors) and several types of energy sources (a small solar panel to recover photovoltaic energy, a piezoelectric sensor for mechanical energy and a Peltier thermal energy sensor).

Figure 3. CAIRN's PowWow motherboard with radio and energy-harvesting boards connected

PowWow distribution also includes a generic software architecture using event-driven programming and organized into protocol layers (PHY, MAC, LINK, NET and APP). The software is based on Contiki [95], and more precisely on the `Protothread` library which provides a sequential control flow without complex state machines or full multi-threading.

To optimize the network regarding a particular application and to define a global strategy to reduce energy, PowWow offers the following extra tools: over-the-air reprogramming (and soon reconfiguration), analytical power estimation based on software profiling and power measurements, a dedicated network analyzer to probe and fix transmissions errors in the network. More information can be found at http://powwow.gforge.inria.fr.

5.7. Ziggie: a Platform for Wireless Body Sensor Networks

Participants: Olivier Sentieys, Olivier Berder, Arnaud Carer, Antoine Courtay [corresponding author], Robin Bonamy.

Keywords: Wireless Body Sensor Networks, Low Power, Gesture Recognition, Localization, Hardware and Software Platform

The Zyggie sensor node has been developed in the team to create an autonomous Wireless Body Sensor Network (WBSN) with the capabilities of monitoring body movements. The Zyggie platform is part of the BoWI project funded by CominLabs. Zyggie is composed of: an ATMEGA128RFA1 microcontroller, an MPU9150 Inertial Measurement Unit (IMU), an RF AS193 switch with two antennas, an LSP331AP barometer, a DC/DC voltage regulator with a battery charge controller, a wireless inductive battery charge controller, and some switches and control LEDs.

Figure 4. CAIRN's Ziggie platform for WBSN

The IMU is composed of a 3-axis accelerometer, a 3-axis gyrometer and a 3-axis magnetometer. The IMU is communicating its data to the embedded microcontroller via an I2C protocol. We also developed our own MAC protocol for synchronization and data exchanges between nodes. The Zyggie platform is used in many PhD works for evaluating data fusion algorithms (RSSI + IMU data) (Zhongwei Zheng, UR1 and Alexis Aulery, UBS/UR1), low power computing algorithms (Alexis Aulery, UBS/UR1), wireless protocols (Viet Hoa Nguyen, UR1) and body channel characterization (Rizwan Masood, TB).

CAMUS Team

5. New Software and Platforms

5.1. PolyLib

Participant: Vincent Loechner.

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

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

5.2. APOLLO software and LLVM

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

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

5.3. IBB source-to-source xfor compiler

Participants: Imen Fassi, Philippe Clauss, Cédric Bastoul.

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

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

5.4. CLooG

Participant: Cédric Bastoul.

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

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

5.5. OpenScop

Participant: Cédric Bastoul.

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

5.6. Clan

Participants: Cédric Bastoul, Imen Fassi.

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

5.7. Clay

Participant: Cédric Bastoul.

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

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

CARAMEL Project-Team

5. New Software and Platforms

5.1. Introduction

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

5.2. GNU MPFR

Participant: Paul Zimmermann [contact].

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

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

5.3. GNU MPC

Participant: Paul Zimmermann [contact].

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

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

5.4. Finite Fields

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

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

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

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

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

5.5. gf2x

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

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

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

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

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

5.6. CADO-NFS

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

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

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

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

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

5.7. Belenios

Participants: Pierrick Gaudry, Stéphane Glondu [contact].

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

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

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

5.8. CMH

Participant: Emmanuel Thomé [contact].

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

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

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

5.9. Platforms

5.9.1. CATREL cluster

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

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

CARTE Project-Team

5. New Software and Platforms

5.1. Morphus/MMDEX

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

5.2. DynamicTracer

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

5.3. CoDisasm

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

CASCADE Project-Team (section vide)

CASSIS Project-Team

5. New Software and Platforms

5.1. Protocol Verification Tools

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

5.1.1. CL-AtSe

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

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

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

5.1.2. Akiss

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

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

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

5.1.3. Belenios

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

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

5.1.4. SAPIC

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

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

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

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

5.2. Testing Tools

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

5.2.1. Hydra

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

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

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

5.2.2. jMuHLPSL

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

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

5.2.3. Praspel

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

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

5.3. Other Tools

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

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

CELTIQUE Project-Team

4. New Software and Platforms

4.1. Javalib

Participants: Frédéric Besson [correspondant], David Pichardie, Pierre Vittet, Laurent Guillo.

Javalib is an efficient library to parse Java .class files into OCaml data structures, thus enabling the OCaml programmer to extract information from class files, to manipulate and to generate valid .class files.
See also the web page http://sawja.inria.fr/.

- Version: 2.3
- Programming language: Ocaml

4.2. SAWJA

Participants: Frédéric Besson [correspondant], David Pichardie, Pierre Vittet, Laurent Guillo.

Sawja is a library written in OCaml, relying on Javalib to provide a high level representation of Java bytecode programs. It name comes from Static Analysis Workshop for JAva. Whereas Javalib is dedicated to isolated classes, Sawja handles bytecode programs with their class hierarchy and with control flow algorithms.

Moreover, Sawja provides some stackless intermediate representations of code, called JBir and A3Bir. The transformation algorithm, common to these representations, has been formalized and proved to be semantics-preserving.
See also the web page http://sawja.inria.fr/.

- Version: 1.5
- Programming language: Ocaml

4.3. Jacal

Participants: Frédéric Besson [correspondant], Thomas Jensen, David Pichardie, Delphine Demange.

Static program analysis, Javacard, Certification, AFSCM

`Jacal` is a JAvaCard AnaLyseur developed on top of the SAWJA (see Section 4.2) platform. This proprietary software verifies automatically that Javacard programs conform with the security guidelines issued by the AFSCM (Association Française du Sans Contact Mobile). `Jacal` is based on the theory of abstract interpretation and combines several object-oriented and numeric analyses to automatically infer sophisticated invariants about the program behaviour. The result of the analysis is thereafter harvest to check that it is sufficient to ensure the desired security properties.

4.4. Timbuk

Participant: Thomas Genet [correspondant].

Timbuk is a library of OCAML functions for manipulating tree automata. More precisely, Timbuk deals with finite bottom-up tree automata (deterministic or not). This library provides the classical operations over tree automata (intersection, union, complement, emptiness decision) as well as exact or approximated sets of terms reachable by a given term rewriting system. This last operation can be certified using a checker extracted from a Coq specification. The checker is now part of the Timbuk distribution. Timbuk distribution now also provides a CounterExample Guided Abstraction Refinement (CEGAR) tool for tree automata completion. The CEGAR part is based on the Buddy BDD library. Timbuk also provides an implementation of Lattice Tree Automata to (efficiently) represent built-in values such as integers, strings, etc. in recognized tree languages. See also the web page http://www.irisa.fr/celtique/genet/timbuk/.

- Version: 3.1
- Programming language: Ocaml

4.5. JSCert

Participants: Alan Schmitt [correspondant], Martin Bodin.

The JSCert project aims to really understand JavaScript. JSCert itself is a mechanised specification of JavaScript, written in the Coq proof assistant, which closely follows the ECMAScript 5 English standard. JSRef is a reference interpreter for JavaScript in OCaml, which has been proved correct with respect to JSCert and tested with the Test 262 test suite.

We plan to build other verification and analysis projects on top of JSCert and JSRef, in particular the certification of derivations in program logics or static analyses.

This project is an ongoing collaboration between Inria and Imperial College. More information, including the source code, is available at http://jscert.org/.

COMETE Project-Team

5. New Software and Platforms

5.1. Location Guard

Participants: Konstantinos Chatzikokolakis [correspondant], Marco Stronati.

https://github.com/chatziko/location-guard

The purpose of *Location Guard* is to implement obfuscation techniques for achieving location privacy, in a an easy and intuitive way that makes them available to the general public. Various modern applications, running either on smartphones or on the web, allow third parties to obtain the user's location. A smartphone application can obtain this information from the operating system using a system call, while web application obtain it from the browser using a JavaScript call.

Although both mobile operating systems and browsers require the user's permission to disclose location information, the user faces an "all-or-nothing" choice: either disclose his exact location and give up his privacy, or stop using the application. This forces many users to disclose their location, although ideally they would like to enjoy some privacy.

The API level of a browser or an operating system is an ideal place for integrating a location obfuscation technique, in a way that is easy to understand for the average user, and readily available to all applications. When an application asks for the user's location, the browser or operating system can ask the user's permission, but including the option to provide an obfuscated location instead of the real one! Different levels of obfuscation can be also offered, so that the user can chose to provide more accurate location to applications that really need it, and more noisy location to those that don't.

Location Guard was created as a prototype for Google Chrome at the end for 2013. In 2014, Location Guard matured from a prototype to a high quality software, supporting both desktop and mobile browsers:

- Google Chrome / Chromium
- Mozilla Firefox and Firefox for Android
- Opera

After only a short period online, the extension has more than 8500 daily users, and it was presented in an article by the popular technology news site Ghacks. Our experience so far shows that end users do care about location privacy, and geo-indistinguishability is a practical approach for providing it.

In the future we plan to make Location Guard more widely available on smartphones, supporting more mobile browsers as well as providing direct integration into the operating system, primarily on Android.

5.2. libqif - A Quantitative Information Flow C++ Toolkit Library

Participants: Konstantinos Chatzikokolakis [correspondant], Marco Stronati.

https://github.com/chatziko/libqif

The goal of libqif is to provide an efficient C++ toolkit implementing a variety of techniques and algorithms from the area of quantitative information flow and differential privacy. We plan to implement all techniques produced by Comète in recent years, as well as several ones produced outside the group, giving the ability to privacy researchers to reproduce our results and compare different techniques in a uniform and efficient framework.

Some of these techniques were previously implemented in an ad-hoc fashion, in small, incompatible with each-other, non-maintained and usually inefficient tools, used only for the purposes of a single paper and then abandoned. We aim at reimplementing those – as well as adding several new ones not previously implemented – in a structured, efficient and maintainable manner, providing a tool of great value for future research. Of particular interest is the ability to easily re-run evaluations, experiments and case-studies from all our papers, which will be of great value for comparing new research results in the future.

The library is still in under heavy development but substantial progress has been made in 2014. Some of the techniques already implemented are:

- Standard leakage measures: Shannon, min-entropy, guessing entropy
- Measures from the g-leakage framework [32]
- Channel factorization
- Standard differential privacy mechanisms from the literature
- The planar Laplace mechanism of [33]
- The standard Kantorovich metric as well as the multiplicative variant from [19]
- All operations are supported for both doubles (for precision) and floats (for memory efficiency)
- All operations involving only rational quantities are supported using arbitrary precision rational arithmetic, allowing to obtain exact results
- Native linear programing for rationals

Many more are scheduled to be added in the near future.

5.3. LeakWatch: Estimating Information Leakage from Java Programs

Participant: Yusuke Kawamoto.

http://www.cs.bham.ac.uk/research/projects/infotools/leakwatch/

Comète contributed to the development of LeakWatch, a quantitative information leakage analysis tool for the Java programming language, created by several people at the University of Birmingham.

LeakWatch is based on a flexible "point-to-point" information leakage model, where secret and publicly-observable data may occur at any time during a program's execution. LeakWatch repeatedly executes a Java program containing both secret and publicly-observable data and uses robust statistical techniques to provide estimates, with confidence intervals, for min-entropy leakage (using a new theoretical result from [23]) and mutual information.

COMPSYS Project-Team

5. New Software and Platforms

5.1. Introduction

This section lists and briefly describes the software developments conducted within Compsys. Most are tools that we extend and maintain over the years. They mainly concern three activities: a) the development of research tools, in general available on demand, linked to polyhedra and loop/array transformations, b) the development of tools linked to the start-up XTREMLOGIC (mostly done outside Compsys but partly inspired by work from Compsys), and c) the development of algorithms in the context of our collaborations with STMicroelectronics. These last developments have been stopped right now, since the end of the Mediacom project in 2012. They are described in previous Compsys activity reports: they concerned register allocation, SSA deconstruction, liveness analysis, intermediate representations, etc. They were done within the compilers of STMicroelectronics, not as stand-alone tools, so they are not available as the other tools.

Concerning tools based on a polyhedral representation of nested loops, many of them are now available. They have been developed and maintained over the years by different teams, after the introduction of Paul Feautrier's Pip, a tool for parametric integer linear programming. This "polyhedral model" view of codes is now widely accepted: it was or is used by Cairn, Parkas, and Camus Inria project-teams, PIPS at École des Mines de Paris, Suif from Stanford University, Compaan at Berkeley and Leiden, PiCo from the HP-Labs (continued as PicoExpress by Synfora and now Synopsis), the DTSE methodology at Imec, Sadayappan's group at Ohio State University, Rajopadhye's group at Colorado State's University, etc. In the last 10 years, several compiler groups have shown their interest in polyhedral methods, e.g., the Gcc group, IBM, and Reservoir Labs, a company that develops a compiler fully based on the polyhedral model and on the techniques that we (the french community) introduced for loop and array transformations. Polyhedra are also used in test and certification projects (Verimag, Lande, Vertecs). Now that these techniques are well-established and disseminated in and by other groups, we prefer to focus on the development of new techniques and tools, which are described here. Some of these tools can be used through a web interface on the Compsys tool demonstrator web page http://compsys-tools.ens-lyon.fr/.

Recent activity concerns the development, by Christophe Alias, of HLS compiler parts to be transferred to the XTREMLOGIC start-up (Zettice project) (see Section 7.2). An important effort of applied research and software development [12] has been achieved, resulting in the Dcc (DPN C Compiler) tool, outlined in Section 5.5 . Also, optimization developments (scalability, memory leaks, parallelization, etc) were performed on the PoCo compiler framework (see Section 5.6) and the Bee tool (see Section 5.7).

Also, several successive developments have been made related to termination tools. Our first implementation, RanK (see Section 5.9), was extended by other tools such as SToP (see Section 5.12) and, more recently Termite, (see Section 5.13).

5.2. Pip

Participants: Cédric Bastoul [professor, Strasbourg University and Inria/CAMUS], Paul Feautrier.

Paul Feautrier is the main developer of Pip (Parametric Integer Programming) since its inception in 1988. Basically, Pip is an "all integer" implementation of the Simplex, augmented for solving integer programming problems (the Gomory cuts method), which also accepts parameters in the non-homogeneous term. Pip is freely available under the GPL at http://www.piplib.org. It is widely used in the automatic parallelization community for testing dependences, scheduling, several kind of optimizations, code generation, and others. Beside being used in several parallelizing compilers, Pip has found applications in some unconnected domains, as for instance in the search for optimal polynomial approximations of elementary functions (see the Inria project Aric, previously known as Arénaire).

5.3. Cl@k

Participants: Fabrice Baray [Mentor Graphics, Former post-doc in Compsys], Alain Darte.

Cl@k (Critical LAttice Kernel) is a stand-alone optimization tool which computes or approximates the critical lattice for a given 0-symmetric polytope. (An admissible lattice is a lattice whose intersection with the polytope is reduced to 0; a critical lattice is an admissible lattice with minimal determinant). This tool is useful for the automatic derivation of array mappings that enable memory reuse, based on the notions of admissible lattice and of modular allocation (linear mapping plus modulo operations). It has been developed in 2005-2006 by Fabrice Baray, former post-doc Inria under Alain Darte's supervision.

Its application to array contraction has been implemented by Christophe Alias in a tool called Bee (see Section 5.7). More information is available at http://compsys-tools.ens-lyon.fr/clak/. The Cl@k tool is unfortunately outdated today (it is hard, if not impossible, to recompile it) and would need to be re-implemented. An extension of its underlying theory is also in progress.

5.4. Syntol

Participant: Paul Feautrier.

Syntol is a modular process network scheduler. The source language is C augmented with specific constructs for representing communicating regular process (CRP) systems. The present version features a syntax analyzer, a semantic analyzer to identify DO loops in C code, a dependence computer, a modular scheduler, and interfaces for CLooG (loop generator developed by C. Bastoul) and Cl@k (see Sections 5.3 and 5.7). The dependence computer now handles casts, records (`structures`), and the modulo operator in subscripts and conditional expressions. The latest developments are, firstly, a new code generator, and secondly, several experimental tools for the construction of bounded parallelism programs.

- The new code generator, based on the ideas of Boulet and Feautrier [20], generates a counter automaton that can be presented as a C program, as a rudimentary VHDL program at the RTL level, as an automaton in the Aspic input format, or as a drawing specification for the DOT tool.
- Hardware synthesis can only be applied to bounded parallelism programs. Our present aim is to construct threads with the objective of minimizing communications and simplifying synchronization. The distribution of operations among threads is specified using a placement function, which is found using techniques of linear algebra and combinatorial optimization.

5.5. Dcc

Participants: Christophe Alias, Alexandru Plesco [XtremLogic].

Dcc (DPN C Compiler) compiles a C program annotated with pragmas to a data-aware process network (DPN), a regular process network close to a circuit description that makes explicit the I/O transfers and the synchronizations. Dcc features throughput optimization, communication vectorization, and automatic parallelization.

Dcc is registered at the APP ("agence de protection des programmes") and has been transferred to the XTREMLOGIC start-up under an Inria licence. It uses a patented technology [12] and serves as a *front-end* for the HLS suite of the XTREMLOGIC start-up. Dcc has been implemented by Christophe Alias, using the PoCo compiler infrastructure (Section 5.6) and the Bee tool (Section 5.7). It represents more than 3000 lines of C++ code.

5.6. PoCo

Participant: Christophe Alias.

PoCo is a polyhedral compilation framework providing many features to prototype program analysis and optimizations in the polyhedral model:

- C parsing and extraction of the polyhedral representation.
- Symbolic layer on the state-of-art polyhedral libraries Polylib (set operations on polyhedra) and Piplib (parameterized ILP, see Section 5.2).
- Dependence analysis (PRDG, array dataflow analysis), array region analysis, array liveness analysis.
- C and VHDL code generation.

PoCo is registered at the APP ("agence de protection des programmes") and has been transferred to the XTREMLOGIC start-up under an Inria licence. The Bee, Chuba, and RanK tools presented thereafter make an extensive use of PoCo abstractions. PoCo has been developed by Christophe Alias. It represents more than 19000 lines of C++ code.

5.7. Bee

Participants: Christophe Alias, Alain Darte.

Bee is a source-to-source optimizer that resizes and reallocates optimally the arrays used by a program under scheduling constraints. Bee applies a fine-grain lifetime analysis for arrays. Then, the mathematical optimization of the Cl@k tool (Section 5.3) finds the array allocation (expressed as an affine lattice). Finally, Bee derives the actual array allocation and generates the C code accordingly. Bee was – to our knowledge – the first complete (i.e., with an element-wise lifetime analyzer integrated with an allocator) array contraction tool. Bee allows to allocate and to size the channels in process networks, providing a global affine schedule. This feature is fundamental in HLS (see Section 3.1.2) and more generally in automatic parallelization where communication buffers must be allocated and sized. An online demonstrator is available at http://compsys-tools.ens-lyon.fr/bee/index.html.

Bee is registered at the APP ("agence de protection des programmes") and has been transferred to the XTREMLOGIC start-up under an Inria licence. It is also used as an external tool by the compiler Gecos developed in the Cairn team at Irisa. Bee has been implemented by Christophe Alias, using the PoCo compiler infrastructure (see Section 5.6). It represents more than 2400 lines of C++ code.

5.8. Chuba

Participants: Christophe Alias, Alain Darte, Alexandru Plesco [Compsys/Zettice].

Chuba is a source-level optimizer that improves a C program in the context of the high-level synthesis (HLS) of hardware. Chuba is an implementation of the work described in the PhD thesis of Alexandru Plesco. The optimized program specifies a system of multiple communicating accelerators, which optimizes the data transfers with the external DDR memory. The program is divided into blocks of computations obtained thanks to tiling techniques, and, in each block, data are fetched by block to reduce the penalty due to line changes in the DDR accesses. Four accelerators achieve data transfers in a macro-pipeline fashion so that data transfers and computations (performed by a fifth accelerator) are overlapped.

So far, the back-end of Chuba is specific to the HLS tool C2H but the analysis is quite general and adapting Chuba to other HLS tools should be possible. Besides, it is interesting to mention that the program analysis and optimizations implemented in Chuba address a problem that is also very relevant in the context of GPGPUs. The underlying theory and corresponding experiments are described in [17].

Chuba has been implemented by Christophe Alias, using the PoCo compiler infrastructure (see Section 5.6). It represents more than 900 lines of C++ code.

5.9. RanK

Participants: Christophe Alias, Alain Darte, Paul Feautrier, Laure Gonnord [Compsys].

RanK is a software tool that can prove the termination of a program (in some cases) by computing a *ranking function*, i.e., a mapping from the operations of the program to a well-founded set that *decreases* as the computation advances. In case of success, RanK can also provide an upper bound of the worst-case time complexity of the program as a symbolic affine expression involving the input variables of the program (parameters), when it exists. In case of failure, RanK tries to prove the non-termination of the program and then to exhibit a counter-example input. This last feature is of great help for program understanding and debugging. The theory underlying RanK was presented at SAS'10 [15].

The input of RanK is an integer automaton, computed by C2fsm (see Section 5.11), representing the control structure of the program to be analyzed. RanK uses the Aspic tool (see Section 5.10), developed by Laure Gonnord during her PhD thesis, to compute automaton invariants. RanK has been used to discover successfully the worst-case time complexity of many benchmarks kernels of the community (see the WTC benchmark suite at http://compsys-tools.ens-lyon.fr/wtc/index.html). It uses the libraries Piplib (Section 5.2) and Polylib.

RanK has been implemented by Christophe Alias, using the compiler infrastructure PoCo (Section 5.6). It represents more than 3000 lines of C++. The tool has been presented at the CSTVA'13 workshop [16]. An online demonstrator is available at http://compsys-tools.ens-lyon.fr/rank.

5.10. Aspic

Participant: Laure Gonnord.

Aspic is an invariant generator for general counter automata. Used with C2fsm (see Section 5.11), it can be used to derivate invariants for numerical C programs, and also to prove safety. It is also part of the WTC toolsuite (see http://compsys-tools.ens-lyon.fr/wtc/index.html), a set of examples to demonstrate the capability of the RanK tool (see Section 5.9) for evaluating worse-case time complexity (number of transitions when executing an automaton).

Aspic implements the theoretical results of Laure Gonnord's PhD thesis on acceleration techniques and has been maintained since 2007.

5.11. C2fsm

Participant: Paul Feautrier.

C2fsm is a general tool that converts an arbitrary C program into a counter automaton. This tool reuses the parser and pre-processor of Syntol (see Section 5.4), which has been extended to handle `while` and `do while` loops, `goto`, `break`, and `continue` statements. C2fsm reuses also part of the code generator of Syntol and has several output formats, including FAST (the input format of Aspic, see Section 5.10), a rudimentary VHDL generator, and a DOT generator which draws the output automaton. In contrast to the FAST format, an ad hoc format, FLOW, uses a relational representation and retains non-affine constructs. C2fsm is also able to do elementary transformations on the automaton, such as eliminating useless states, transitions and variables, simplifying guards, or selecting cut-points, i.e., program points on loops that can be used by RanK (see Section 5.9) to prove program termination.

5.12. SToP

Participants: Christophe Alias, Guillaume Andrieu [University of Lille], Laure Gonnord [Compsys].

SToP (Scalable Termination of Programs) is the implementation of the modular termination technique presented at the TAPAS'12 workshop [18]. It takes as input a large irregular C program and conservatively checks its termination. To do so, SToP generates a set of small programs whose termination implies the termination of the whole input program. Then, the termination of each small program is checked thanks to RanK (see Section 5.9). In case of success, SToP infers a ranking (schedule) for the whole program. This schedule can be used in a subsequent analysis to optimize the program.

SToP represents more than 2000 lines of C++. The first results are available at http://compsys-tools.ens-lyon.fr/stop.

5.13. Termite

Participants: Laure Gonnord, Gabriel Radanne [ENS Rennes], David Monniaux [CNRS/VERIMAG].

Termite (Termination of C programs) is the implementation of our new algorithm "Counter-example based generation of ranking functions" (see Section 6.4). Based on LLVM and Pagai (a tool that generates invariants), the tool automatically generates a ranking function for each *head of loop*. Its implementation is under consolidation, it will be publicly available soon.

Termite represents 3000 lignes of OCaml.

5.14. Simplifiers

Participant: Paul Feautrier.

The aim of the `simple` library is to simplify Boolean formulas on affine inequalities. It works by detecting redundant inequalities in the representation of the subject formula as an ordered binary decision diagram (OBDD), see details in [23]. It uses PIP (see Section 5.2) for testing the feasibility – or unfeasibility – of a conjunction of affine inequalities.

The library is written in Java and is presented as a collection of class files. For experimentation, several front-ends have been written. They differ mainly in their input syntax, among which are a C like syntax, the Mathematica and SMTLib syntaxes, and an ad hoc Quast (quasi-affine syntax tree) syntax. See the first results at http://compsys-tools.ens-lyon.fr/stop.

CONVECS Project-Team

5. New Software and Platforms

5.1. The CADP Toolbox

Participants: Hubert Garavel [correspondent], Frédéric Lang, Radu Mateescu, Wendelin Serwe.

We maintain and enhance CADP (*Construction and Analysis of Distributed Processes* – formerly known as *CAESAR/ALDEBARAN Development Package*) [1], a toolbox for protocols and distributed systems engineering [0]. In this toolbox, we develop and maintain the following tools:

- CAESAR.ADT [41] is a compiler that translates LOTOS abstract data types into C types and C functions. The translation involves pattern-matching compiling techniques and automatic recognition of usual types (integers, enumerations, tuples, etc.), which are implemented optimally.
- CAESAR [47], [46] is a compiler that translates LOTOS processes into either C code (for rapid prototyping and testing purposes) or finite graphs (for verification purposes). The translation is done using several intermediate steps, among which the construction of a Petri net extended with typed variables, data handling features, and atomic transitions.
- OPEN/CAESAR [42] is a generic software environment for developing tools that explore graphs on the fly (for instance, simulation, verification, and test generation tools). Such tools can be developed independently of any particular high level language. In this respect, OPEN/CAESAR plays a central role in CADP by connecting language-oriented tools with model-oriented tools. OPEN/CAESAR consists of a set of 16 code libraries with their programming interfaces, such as:
  - CAESAR_GRAPH, which provides the programming interface for graph exploration,
  - CAESAR_HASH, which contains several hash functions,
  - CAESAR_SOLVE, which resolves Boolean equation systems on the fly,
  - CAESAR_STACK, which implements stacks for depth-first search exploration, and
  - CAESAR_TABLE, which handles tables of states, transitions, labels, etc.

  A number of on-the-fly analysis tools have been developed within the OPEN/CAESAR environment, among which:
  - BISIMULATOR, which checks bisimulation equivalences and preorders,
  - CUNCTATOR, which performs steady-state simulation of continuous-time Markov chains,
  - DETERMINATOR, which eliminates stochastic nondeterminism in normal, probabilistic, or stochastic systems,
  - DISTRIBUTOR, which generates the graph of reachable states using several machines,
  - EVALUATOR, which evaluates MCL formulas,
  - EXECUTOR, which performs random execution,
  - EXHIBITOR, which searches for execution sequences matching a given regular expression,
  - GENERATOR, which constructs the graph of reachable states,
  - PROJECTOR, which computes abstractions of communicating systems,
  - REDUCTOR, which constructs and minimizes the graph of reachable states modulo various equivalence relations,

[0] http://cadp.inria.fr

  - SIMULATOR, XSIMULATOR, and OCIS, which enable interactive simulation, and
  - TERMINATOR, which searches for deadlock states.
- BCG (*Binary Coded Graphs*) is both a file format for storing very large graphs on disk (using efficient compression techniques) and a software environment for handling this format. BCG also plays a key role in CADP as many tools rely on this format for their inputs/outputs. The BCG environment consists of various libraries with their programming interfaces, and of several tools, such as:
  - BCG_CMP, which compares two graphs,
  - BCG_DRAW, which builds a two-dimensional view of a graph,
  - BCG_EDIT, which allows the graph layout produced by BCG_DRAW to be modified interactively,
  - BCG_GRAPH, which generates various forms of practically useful graphs,
  - BCG_INFO, which displays various statistical information about a graph,
  - BCG_IO, which performs conversions between BCG and many other graph formats,
  - BCG_LABELS, which hides and/or renames (using regular expressions) the transition labels of a graph,
  - BCG_MIN, which minimizes a graph modulo strong or branching equivalences (and can also deal with probabilistic and stochastic systems),
  - BCG_STEADY, which performs steady-state numerical analysis of (extended) continuous-time Markov chains,
  - BCG_TRANSIENT, which performs transient numerical analysis of (extended) continuous-time Markov chains, and
  - XTL (*eXecutable Temporal Language*), which is a high level, functional language for programming exploration algorithms on BCG graphs. XTL provides primitives to handle states, transitions, labels, *successor* and *predecessor* functions, etc.

    For instance, one can define recursive functions on sets of states, which allow evaluation and diagnostic generation fixed point algorithms for usual temporal logics (such as HML [51], CTL [37], ACTL [39], etc.) to be defined in XTL.
- PBG (*Partitioned BCG Graph*) is a file format implementing the theoretical concept of *Partitioned LTS* [45] and providing a unified access to a graph partitioned in fragments distributed over a set of remote machines, possibly located in different countries. The PBG format is supported by several tools, such as:
  - PBG_CP, PBG_MV, and PBG_RM, which facilitate standard operations (copying, moving, and removing) on PBG files, maintaining consistency during these operations,
  - PBG_MERGE (formerly known as BCG_MERGE), which transforms a distributed graph into a monolithic one represented in BCG format,
  - PBG_INFO, which displays various statistical information about a distributed graph.
- The connection between explicit models (such as BCG graphs) and implicit models (explored on the fly) is ensured by OPEN/CAESAR-compliant compilers, e.g.:
  - BCG_OPEN, for models represented as BCG graphs,
  - CAESAR.OPEN, for models expressed as LOTOS descriptions,
  - EXP.OPEN, for models expressed as communicating automata,
  - FSP.OPEN, for models expressed as FSP [55] descriptions,
  - LNT.OPEN, for models expressed as LNT descriptions, and
  - SEQ.OPEN, for models represented as sets of execution traces.

The CADP toolbox also includes TGV (*Test Generation based on Verification*), which has been developed by the VERIMAG laboratory (Grenoble) and the VERTECS project-team at Inria Rennes – Bretagne-Atlantique.

The CADP tools are well-integrated and can be accessed easily using either the EUCALYPTUS graphical interface or the SVL [43] scripting language. Both EUCALYPTUS and SVL provide users with an easy and uniform access to the CADP tools by performing file format conversions automatically whenever needed and by supplying appropriate command-line options as the tools are invoked.

5.2. The TRAIAN Compiler

Participants: Hubert Garavel [correspondent], Frédéric Lang, Wendelin Serwe.

We develop a compiler named TRAIAN for translating LOTOS NT descriptions into C programs, which will be used for simulation, rapid prototyping, verification, and testing.

The current version of TRAIAN, which handles LOTOS NT types and functions only, has useful applications in compiler construction [44], being used in all recent compilers developed by CONVECS.

The TRAIAN compiler can be freely downloaded from the CONVECS Web site [0].

5.3. The PIC2LNT Translator

Participants: Radu Mateescu, Gwen Salaün [correspondent].

We develop a translator named PIC2LNT from an applied π-calculus (see § 6.1) to LNT, which enables the analysis of concurrent value-passing mobile systems using CADP.

PIC2LNT is developed by using the SYNTAX tool (developed at Inria Paris-Rocquencourt) for lexical and syntactic analysis together with LOTOS NT for semantical aspects, in particular the definition, construction, and traversal of abstract trees.

The PIC2LNT translator can be freely downloaded from the CONVECS Web site [0].

5.4. The PMC Partial Model Checker

Participants: Radu Mateescu, Frédéric Lang.

We develop a tool named PMC (*Partial Model Checker*, see § 6.4), which performs the compositional model checking of dataless MCL formulas on networks of communicating automata described in the EXP language.

PMC can be freely downloaded from the CONVECS Web site [0].

[0] http://convecs.inria.fr/software/traian
[0] http://convecs.inria.fr/software/pic2lnt
[0] http://convecs.inria.fr/software/pmc

CRYPT Team (section vide)

DEDUCTEAM Exploratory Action

5. New Software and Platforms

5.1. Dedukti

Dedukti is a proof-checker for the $\lambda\Pi$-calculus modulo. As it can be parametrized by an arbitrary set of rewrite rules, defining an equivalence relation, this calculus can express many different theories. Dedukti has been created for this purpose: to allow the interoperability of different theories.

Dedukti's core is based on the standard algorithm [29] for type-checking semi-full pure type systems and implements a state-of-the-art reduction machine inspired from Matita's [28] and modified to deal with rewrite rules.

Dedukti's input language features term declarations and definitions (opaque or not) and rewrite rule definitions. A basic module system allows the user to organize its project in different files and compile them separately.

Dedukti now features matching modulo beta for a large class of patterns called Miller's patterns, allowing for more rewriting rules to be implemented in Dedukti.

Dedukti has been developed by Mathieu Boespflug, Olivier Hermant, Quentin Carbonneaux and Ronan Saillard. It is composed of about 2500 lines of OCaml.

5.2. Coqine, Holide, Focalide and Sigmaid

Dedukti comes with four companion tools: Holide, an embedding of HOL proofs through the OpenTheory format [41], Coqine, an embedding of Coq proofs, Focalide, an embedding of FoCaLiZe certified programs, and Sigmaid, a type-checker for the simply-typed ς-calculus with subtyping and a translator to Dedukti. All of the OpenTheory standard library and a part of Coq's and FoCaLiZe's libraries are checked by Dedukti.

A preliminary version of Coqine supports the following features of Coq: the raw Calculus of Constructions, inductive types, and fixpoint definitions. Coqine is currently being rewritten to support universes. Coqine has been developed by Mathieu Boespflug, Guillaume Burel, and Ali Assaf.

Holide supports all the features of HOL, including ML-polymorphism, constant definitions, and type definitions. It is able to translate all of the OpenTheory standard theory library. Holide has been developed by Ali Assaf.

Focalide has been improved to support FoCaLiZe proofs found by Zenon using the Dedukti backend for Zenon developped by Frédéric Gilbert. This backend has been improved by a simple typing mechanism in order to work with Focalide. Focalide has also been updated again to work with the latest version of FoCaLiZe.

Sigmaid implements a shallow embedding of the simply-typed ς-calculus of Abadi and Cardelli with subtyping in the $\lambda\Pi$-calculus modulo. This translation has been proved[21] to preserve the typing judgments and the semantics of the simply-typed ς-calculus and tested on the examples of Abadi and Cardelli.

Focalide and Sigmaid have been developed by Raphaël Cauderlier.

Translators from Version 2.0 of the SMT-LIB standard and from the SMT-solver veriT have been initiated. They are currently developed by Frédéric Gilbert.

5.3. iProver Modulo

iProver Modulo is an extension of the automated theorem prover iProver originally developed by Konstantin Korovin at the University of Manchester. It implements Ordered polarized resolution modulo, a refinement of the Resolution method based on Deduction modulo. It takes as input a proposition in predicate logic and a clausal rewriting system defining the theory in which the formula has to be proved. Normalization with respect to the term rewriting rules is performed very efficiently through translation into OCaml code, compilation and dynamic linking. Experiments have shown that Ordered polarized resolution modulo dramatically improves proof search compared to using raw axioms. iProver modulo is also able to produce proofs that can be checked by Dedukti, therefore improving confidence. iProver modulo is written in OCaml, it consists of 1,200 lines of code added to the original iProver.

A tool that transforms axiomatic theories into polarized rewriting systems, thus making them usable in iProver Modulo, has also been developed. Autotheo supports several strategies to orient the axioms, some of them being proved to be complete, in the sense that Ordered polarized resolution modulo the resulting systems is refutationally complete, some others being merely heuristics. In practice, autotheo takes a TPTP input file and transforms the axioms into rewriting rules, and produces an input file for iProver Modulo.

iProver Modulo and autotheo have been developed by Guillaume Burel. iProver Modulo is released under a GPL license.

5.4. Super Zenon and Zenon Modulo

Several extensions of the *Zenon* automated theorem prover (developed by Damien Doligez at *Inria* in the *Gallium* team) to Deduction modulo have been studied. These extensions intend to be applied in the context of the automatic verification of proof rules and obligations coming from industrial applications formalized using the *B* method.

The first extension, developed by Mélanie Jacquel and David Delahaye, is called *Super Zenon* and is an extension of *Zenon* to superdeduction, which can be seen as a variant of Deduction modulo. This extension is a generalization of previous experiments [42] together with Catherine Dubois and Karim Berkani (*Siemens*), where *Zenon* has been used and extended to superdeduction to deal with the *B* set theory and automatically prove proof rules of *Atelier B*. This generalization consists in allowing us to apply the extension of *Zenon* to superdeduction to any first order theory by means of a heuristic that automatically transforms axioms of the theory into rewrite rules. This work is described in [13] [35], which also proposes a study of the possibility of recovering intuition from automated proofs using superdeduction.

The second extension, developed by Pierre Halmagrand, David Delahaye, Damien Doligez, and Olivier Hermant, is called *Zenon Modulo* and is an extension of *Zenon* to Deduction modulo. Compared to *Super Zenon*, this extension allows us to deal with rewrite rules both over propositions and terms. Like *Super Zenon*, *Zenon Modulo* is able to deal with any first order theory by means of a similar heuristic. To assess the approach of *Zenon Modulo*, we have applied this extension to the first order problems coming from the TPTP library. An increase of the number of proved problems has been observed, with in particular a significant increase in the category of set theory. Over these problems of the TPTP library, we have also observed a significant proof size reduction, which confirms this aspect of Deduction modulo. These results are gathered into two publications [33], [34].

The third extension, developed by Guillaume Bury and David Delahaye, is an extension of *Zenon* to (rational and integer) linear arithmetic (using the simplex algorithm), that has been integrated to *Zenon Modulo* by Guillaume Bury and Pierre Halmagrand, in order to be applied in the framework of the *B* set theory to the verification of proof obligations of *Atelier B* [17]. Experiments have been conducted over the benchmarks of the *BWare* project, and it turns out that more than 95% of the proof obligations are proved thanks to this extension.

5.5. Zipperposition (and extensions) and Logtk

Zipperposition is an implementation of the superposition method; it relies on the library Logtk for basic logic data structures and algorithms. Zipperposition is designed as a testbed for extensions to superposition, and can currently deal with polymorphic typed logic, integer arithmetic, and total orderings; an extension to handle structural induction is being worked on by Simon Cruanes.

Those pieces of software also depend on many smaller tools and libraries developped by Simon Cruanes in OCaml. In particular, efficient iterators were key to implementing arithmetic rules successfully, and a lightweight extension to the standard library has been developped steadily and released regularly.

5.6. CoLoR

CoLoR is a Coq library on rewriting theory and termination of more than 83,000 lines of code [2]. It provides definitions and theorems for:

- Mathematical structures: relations, (ordered) semi-rings.
- Data structures: lists, vectors, polynomials with multiple variables, finite multisets, matrices, finite graphs.
- Term structures: strings, algebraic terms with symbols of fixed arity, algebraic terms with varyadic symbols, pure and simply typed λ-terms.
- Transformation techniques: conversion from strings to algebraic terms, conversion from algebraic to varyadic terms, arguments filtering, rule elimination, dependency pairs, dependency graph decomposition, semantic labelling.
- Termination criteria: polynomial interpretations, multiset ordering, lexicographic ordering, first and higher order recursive path ordering, matrix interpretations.

CoLoR is distributed under the CeCILL license. It is currently developed by Frédéric Blanqui and Kim-Quyen Ly, but various people participated to its development since 2006 (see the website for more information).

5.7. HOT

HOT is an automated termination prover for higher-order rewrite systems based on the notion of computability closure and size annotation [31]. It won the 2012 competition in the category “higher-order rewriting union beta”. The sources (5000 lines of OCaml) are not public. It is developed by Frédéric Blanqui.

5.8. Moca

Moca is a construction functions generator for OCaml data types with invariants.

It allows the high-level definition and automatic management of complex invariants for data types. In addition, it provides the automatic generation of maximally shared values, independently or in conjunction with the declared invariants.

A relational data type is a concrete data type that declares invariants or relations that are verified by its constructors. For each relational data type definition, Moca compiles a set of construction functions that implements the declared relations.

Moca supports two kinds of relations:

- predefined algebraic relations (such as associativity or commutativity of a binary constructor),
- user-defined rewrite rules that map some pattern of constructors and variables to some arbitrary user’s define expression.

The properties that user-defined rules should satisfy (completeness, termination, and confluence of the resulting term rewriting system) must be verified by a programmer’s proof before compilation. For the predefined relations, Moca generates construction functions that allow each equivalence class to be uniquely represented by their canonical value.

Moca is distributed under QPL. It is written in OCaml (14,000 lines) It is developed by Frédéric Blanqui, Pierre Weis (EPI Pomdapi) and Richard Bonichon (CEA).

5.9. Rainbow

Rainbow is a set of tools for automatically verifying the correctness of termination certificates expressed in the CPF format used in the termination competition. It contains:

- a tool `xsd2coq` for generating Coq data types for representing XML files valid wrt some XML Schema,
- a tool `xsd2ml` for generating OCaml data types and functions for parsing XML files valid wrt some XML Schema,
- a tool for translating a CPF file into a Coq script,
- a standalone Coq certified tool for verifying the correctness of a CPF file.

Rainbow is distributed under the CeCILL license. It is developed in OCaml (10,000 lines) and Coq (19,000 lines). It is currently developed by Frédéric Blanqui and Kim-Quyen Ly. See the website for more information.

5.10. mSAT

mSAT is a modular, proof-producing, SAT and SMT core based on Alt-Ergo Zero, written in OCaml. The solver accepts user-defined terms, formulas and theory, making it a good tool for experimenting. This tool produces resolution proofs as trees in which the leaves are user-defined proof of lemmas.

An encoding of tableaux rules as a theory for SMT solvers has been implemented and tested in mSAT. mSat has also been extended to implement model constructing satisfiability calculus, a variant of SMT solvers in which assignment of variables to values are propagated along with the usual boolean assignment of litterals.

DICE Team

4. New Software and Platforms

4.1. GPeer: a peer-to-peer javascript communication library

Our software development has been oriented towards systems working in browsers, with the support of an **Inria ADT project** in cooperation with the ASAP team. To answer our technological objectives, we are working on decentralized architectures, browser to browser, developed in javascript/HTML5. We rely on the WebRTC JavaScript protocol proposed by Google to develop a communication layer between peers. Many peer-to-peer protocols share common elements, that we group in a generic library for developing peer-to-peer systems. The joint library developed with the ASAP team handles any gossip based communication overlay. We design peer messages, tracker management and resilient behavior. The library is a standard bridge between complex browser to browser applications and low level networking layers such as WebRTC. With the use of our library, we can reproduce systems such as BitTorrent, but also provide new applications without the need of either native applications or identified servers.

4.2. Fluxion: a software plugin for flows in AngularJS

The **joint project with Worldline** aims at managing mobile code in complex Web architectures. Load variation in data-centers is currently poorly resolved. Most of the time, systems overestimate resource consumption in order to absorb burst usage. These consumption overestimation has a cost both in terms of the SLA negotiated with the client and the non-availability of reserved resources. With Wordline we focus on code mobility for high performance Web architectures and design a fast and reactive framework, transparently moving functions between running systems. The Fluxion model is our approach to design mobile application modules that are a mix of functional programming and flow based reactive systems.

4.3. BitBallot: a decentralized voting protocol

The BitBallot voting protocol is designed to target large scale communities. The protocol allows users to share only restricted amounts of their data and computation with central platforms as well as other peers. Convinced by the need of new election mechanisms, to support emerging forms of more continuous democracy, we are developing BitBallot, to allow elections to be organized independently of any central authority. The protocol guarantees the following properties, anonymity of the data sources, non interruptible run-time, global access to results, and non predictability of results through partial communication spying.

4.4. Odin: an intermediation platform

Odin is a middleware framework for building intermediation platforms. It is build over a kernel that stores users data and activities on a noSQL database and a full client/server JavaScript communication stack. The kernel is used to build intermediation platforms for any kind of project management systems, and where projects peculiarities are handled through a plugin architecture. Plugins are used to define dedicated crawlers over the Web that gather information and push recommendation toward users. The framework maintains an internal currency used to trigger a subset of agents used for recommendation. These recommendations must be mapped to the project keywords and user profile. Each user project is associated to a specific amount of money in our currency, and project users may use this currency to drive their virtual agents. If agents are correctly driven, projects may gain more money used to obtain better recommendations or used on other projects. Our goal is to gather a huge amount of users in order to study system scalability in a real life application. We use odin to test and validate search engines, recommendation engines, external resource crawling, and social network user experiences.

4.5. C3PO: Collaborative Creation of Contents and Publishing using Opportunistic Networks

Social networks put together individuals with common interests and/or existing real-life relationships so that they can produce and share information. There is a strong interest of individuals towards those networks. They rely on a stable, centralized network infrastructure and a user will always be provided with the same services no matter what their current context is. By contrast, the C3PO project aims at promoting “spontaneous and ephemeral social networks” (SESN), built on top of a peer-to-peer distributed architecture leveraging ad-hoc mobile networks and the resources and services offered by mobile devices. As with traditional social networks, SESN can put together nomad individuals based on their affinities and common interests so that they can collaboratively work on tasks as part of a SESN. In C3PO, we strive for incitation in collaborating through a SESN. Several application domains have been anticipated for SESN, especially those involving gathering information and producing content as part of cultural or sport events. In such types of SESN, photo sharing, collaborative document edition and sport results spreading services can be used for building structured digital contents that relate the events of sports gatherings. Generated contents can be consulted through the multiple production sources. They can then be replicated on dedicated servers or published to traditional, centralized social networks and made available to Internet users beyond the lifespan of the SESN where they were initially produced. The C3PO project aims at investigating the problems posed by SESN, and especially those induced by the dynamic and unreliable nature of the ad-hoc mobile networks. It will offer innovative scientific and software solutions for services provision with intermittent connectivity, the definition of an infrastructure for the collaborative management of services in the context of SESN, and an analysis of the value adapted to this context. C3PO is a 3 years ANR industrial research project involving 4 academic research groups and an industrial partner. The proposed contributions will be validated by experimentation in real-world conditions.

DREAMPAL Team

4. New Software and Platforms

4.1. New Software and Platforms

Download page : https://gforge.inria.fr/frs/?group_id=3646

4.1.1. HoMade

HoMade is a softcore processor that we have started developing in 2012. The current version is reflective (i.e., the program it executes is self-modifiable), and statically configurable; dynamically reconfigurable multi-processors are the next steps. Users have to add to it the functionality they need in their applications via IPs. We have also being developing a library of IPs for the most common processor functions (ALU, registers, ...). All the design is in VHDL except for some schematic specifications.

The V5 version of HoMade has been developed in the Spring 2014. It has been used by ∼140 4th-year computer science students at Univ. Lille enrolled in the hardware architecture course (https://sites.google.com/site/tpm1aev/home). The new features of V5 are listed in Section 5.2 .

4.1.2. JHomade

JHomade is a software suite written in JAVA, including compilers and tools for the HoMade processor. It allows us to compile HiHope programs to Homade machine code and load the resulting binaries on FPGA boards. It was first released in 2013. The second version in 2014 includes several new features, like a C-frontend, a binary decoder and a code-generator for VHDL simulation. New features of the HiHope language are described in more detail in Section 5.3 .

4.1.3. Kcheck

Kcheck is a tool for the symbolic execution of programs in arbitrary languages defined in the $\mathbb{K}$ framework (http://k-framework.org), such as C and Java as well as the languages HiHope and Homade machine-code languages developed in out team. It also allow users to formally verify programs against specifications written in Reachability Logic, a specification formalism that can be seen as a language-independent Hoare logic. More information about the theory underlying Kcheck is given in Section 5.5 .

In 2014 we have developed a new and improved version of our tool, in order to keep up with the new modular infrastructure of the $\mathbb{K}$ framework. An online interface has been developed and is available at https://fmse.info.uaic.ro/tools/kcheck/. We have also started (since Nov. 2014) a development in the Coq proof assistant in order to obtained certificates for the program verifications performed by our tool.

ESTASYS Exploratory Action

5. New Software and Platforms

5.1. The Plasma Statistical Model Checker

Participants: Axel Legay [Coordinator], Sean Sedwards, Benoît Boyer, Louis-Marie Traonouez, Kevin Corre.

5.1.1. PLASMA

Statistical model checking (SMC) is a fast emerging technology for industrial scale verification and optimisation problems. In recognition of this, our group is developing a *P*latform for *L*earning and *A*dvanced *S*tatistical *M*odel checking *A*lgorithms: PLASMA.

PLASMA (see https://project.inria.fr/plasma-lab/) was conceived to have high performance and be extensible, using a proprietary virtual machine [48]. Since SMC requires only an executable semantics and is not constrained by decidability, we can easily implement different modelling languages and logics. Our involvement in the DANSE [0] and DALi [0] European projects has also made us aware of the need to provide efficient verification for externally implemented simulators. We thus devised PLASMA-lab, a modular SMC library that allows external users to tightly integrate their own code with our efficient SMC algorithms and integrated development environment [47]. PLASMA-lab has now been successfully integrated with DESYRE [0], Scilab [0] and MATLAB [0].

The PLASMA-lab architecture is now the basis of our free-standing tool, [0] which includes all the modelling languages, logics and algorithms developed by our group. In particular, we have recently developed cutting edge algorithms for rare events [50], [49], [26], nondeterminism [28], [34], [37] and learning [14], [41].

5.2. Quail

Participants: Axel Legay [Coordinator], Fabrizio Biondi [Coordinator], Jean Quilbeuf.

Privacy is a central for Systems of Systems and interconnected objects. We propose QUAIL, a tool that can be used to quantify privacy of components. QUAIL is the only tool able to perform an arbitrary-precision quantitative analysis of the security of a system depending on private information. Thanks to its Markovian semantics model, QUAIL computes the correlation between the system's observable output and the private information, obtaining the amount of bits of the secret that the attacker will infer by observing the output. QUAIL is open source and can be downloaded at https://project.inria.fr/quail/.

QUAIL is able to evaluate the safety of randomized protocols depending on secret data, allowing to verify a security protocol's effectiveness. QUAIL can also be used to find previously unknown security vulnerabilities in software systems and security protocols. The tool can verify whether a protocol is protecting its secret in a perfect way, and quantify how much the secret is exposed to being revealed otherwise.

QUAIL has been used to quantify whether voting protocols respect the anonymity of the voters, proving that preference ranking voting schemes are more secure than single preference ones. It has also been applied to the security of smart grids and a number of classic examples like dining cryptographers, authentication protocols and grades protocol.

[0] http://www.danse-ip.eu
[0] http://www.ict-dali.eu
[0] http://www.ales.eu.com
[0] http://www.scilab.org
[0] http://www.mathworks.com
[0] https://project.inria.fr/plasma-lab

Since its initial release in 2013, QUAIL's algorithm has been improved employing abstract trace exploration and statistical estimation techniques, making it thousands of times faster than the initial version and outperforming other comparable analysis tools on most use cases.

5.3. PyEcdar

Participants: Axel Legay [Coordinator], Louis-Marie Traonouez [Coordinator].

One of the main difficulties with Systems of Systems is to describe the connection and interactions between the components. We propose PyEcdar as a solution to this problem. PyEcdar (https://project.inria.fr/pyecdar/) is a free software that analyses timed games and timed specifications. The goal of the tool is to allow a fast prototyping of new analysis techniques. It currently allows to solve timed games based on timed automata models. These can be extended with adaptive features to represent dynamicity and to model software product lines.

The tool has been originally developed to analyze the robustness of timed specifications, in extension of the tool Ecdar (http://people.cs.aau.dk/~adavid/ecdar/). As Ecdar, it allows to compose components specifications based on Timed I/O Automata (TIOA), and it implements timed game algorithms for checking consistency and compatibility. Additionally, it features original methods for checking the robustness of these specifications.

The tool has been later extended to analyse adaptive systems. It therefore implements original algorithms for checking featured timed games against requirements expressed in the timed AdaCTL logic.

The tool is written in Python with around 3'000 lines of code. It uses a Python console as user interface, from which it can load TIOA components from XML files written in the UPPAAL format (http://www.uppaal.org/), and design complex system by combining the components using a simple algebra. Then, it can analyze these systems, transform them and save them in a new XML file.

GALAAD2 Team

5. New Software and Platforms

5.1. Mathemagix, a free computer algebra environment

Participant: Bernard Mourrain.

http://www.mathemagix.org/

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

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

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

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

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

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

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

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

5.2. Axel, a geometric modeler for algebraic objects

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

http://axel.inria.fr.

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

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

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

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

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

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

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

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

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

GALLIUM Project-Team

5. New Software and Platforms

5.1. OCaml

Participants: Damien Doligez [correspondant], Alain Frisch [LexiFi], Jacques Garrigue [Nagoya University], Fabrice Le Fessant, Xavier Leroy, Luc Maranget, Gabriel Scherer, Mark Shinwell [Jane Street], Leo White [OCaml Labs, Cambridge University], Jeremy Yallop [OCaml Labs, Cambridge University].

OCaml, formerly known as Objective Caml, is our flagship implementation of the Caml language. From a language standpoint, it extends the core Caml language with a fully-fledged object and class layer, as well as a powerful module system, all joined together by a sound, polymorphic type system featuring type inference. The OCaml system is an industrial-strength implementation of this language, featuring a high-performance native-code compiler for several processor architectures (IA32, AMD64, PowerPC, ARM, ARM64) as well as a bytecode compiler and interactive loop for quick development and portability. The OCaml distribution includes a standard library and a number of programming tools: replay debugger, lexer and parser generators, documentation generator, and compilation manager.

Web site: http://caml.inria.fr/

5.2. CompCert C

Participants: Xavier Leroy [correspondant], Sandrine Blazy [EPI Celtique], Jacques-Henri Jourdan.

The CompCert C verified compiler is a compiler for a large subset of the C programming language that generates code for the PowerPC, ARM and x86 processors. The distinguishing feature of Compcert is that it has been formally verified using the Coq proof assistant: the generated assembly code is formally guaranteed to behave as prescribed by the semantics of the source C code. The subset of C supported is quite large, including all C types except `long double`, all C operators, almost all control structures (the only exception is unstructured `switch`), and the full power of functions (including function pointers and recursive functions but not variadic functions). The generated PowerPC code runs 2–3 times faster than that generated by GCC without optimizations, and only 7% (resp. 12%) slower than GCC at optimization level 1 (resp. 2).

Web site: http://compcert.inria.fr/

5.3. The diy tool suite

Participants: Luc Maranget [correspondant], Jade Alglave [Microsoft Research, Cambridge], Jacques-Pascal Deplaix, Susmit Sarkar [University of St Andrews], Peter Sewell [University of Cambridge].

The **diy** suite provides a set of tools for testing shared memory models: the **litmus** tool for running tests on hardware, various generators for producing tests from concise specifications, and **herd**, a memory model simulator. Tests are small programs written in x86, Power or ARM assembler that can thus be generated from concise specification, run on hardware, or simulated on top of memory models. Test results can be handled and compared using additional tools.

Web site: http://diy.inria.fr/

5.4. Zenon

Participant: Damien Doligez.

Zenon is an automatic theorem prover based on the tableaux method. Given a first-order statement as input, it outputs a fully formal proof in the form of a Coq proof script. It has special rules for efficient handling of equality and arbitrary transitive relations. Although still in the prototype stage, it already gives satisfying results on standard automatic-proving benchmarks.

Zenon is designed to be easy to interface with front-end tools (for example integration in an interactive proof assistant), and also to be easily retargeted to output scripts for different frameworks (for example, Isabelle and Dedukti).

Web site: http://zenon-prover.org/

GCG Team

5. New Software and Platforms

5.1. Givy

Givy is a runtime currently developed as part of the phd thesis of François Gindraud. It is designed for architectures with distributed memories, with the Kalray MPPA as the main target. It will execute dynamic data-flow task graphs, annotated with memory dependencies. It will automatically handle scheduling and placement of tasks (using the memory dependency hints), and generate memory transfers between distributed memory nodes when needed by using a software cache coherence protocol. Most of the work this year was done on implementing and testing a memory allocator with specific properties that is a building block of the whole runtime. This memory allocator is also tuned to work on the MPPA and its constraints, turning with very little memory and being efficient in the context of multithreaded calls.

5.2. Tirex

The Tirex Intermediate Representation has previously been generated from within both the Path64 and GCC compilers. In order to increase the usability of Tirex and to decrease the amount of required code maintenance that is induced by compiler evolutions a Tirex-generator has been written that is capable of creating the Tirex representation of a program based on its corresponding assembler code.

5.3. LLVM plugins

Work has been started on multiple plugins for the LLVM compiler framework that implement the code optimisations that have been elaborated by the team. While being work in progress this already provides us with crucial information for program analysis such as data-dependencies.

GEOMETRICA Project-Team

5. New Software and Platforms

5.1. CGAL, the Computational Geometry Algorithms Library

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

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

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

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

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

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

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

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

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

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

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

5.1.1. High-dimensional kernel Epick_d

Participant: Marc Glisse.

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

5.1.2. Number type Mpzf

Participant: Marc Glisse.

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

5.1.3. CGALmesh: a Generic Framework for Delaunay Mesh Generation

Participants: Jean-Daniel Boissonnat, Mariette Yvinec.

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

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

5.1.4. Periodic Meshes

Participants: Aymeric Pellé, Monique Teillaud.

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

5.2. Gudhi library

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

With the collaboration of David Salinas *https://project.inria.fr/gudhi/software/*

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

GRACE Project-Team

5. New Software and Platforms

5.1. CADO-NFS-DLOG

F. Morain is one of the developers of CADO-NFS (available at http://cado-nfs.gforge.inria.fr/), which now includes new algorithms for discrete logarithm computations over finite fields.

5.2. Fast Compact Diffie–Hellman software

Working with C. Costello (Microsoft Research) and H. Hisil (Yasar), B. Smith contributed to the development of a competitive, high-speed, open implementation of the Diffie–Hellman protocol (described in [21]), targeting the 128-bit security level on Intel platforms. The source code is freely available at http://research.microsoft.com/en-us/downloads/ef32422a-af38-4c83-a033-a7aafbc1db55/ and http://hhisil.yasar.edu.tr/files/hisil20140318compact.tar.gz.

5.3. Platforms

5.3.1. ACTIS: Contribution to Sage

In the beginning of 2014,D. Augot and C. Pernet submitted an IJD proposal (ingénieur jeune diplomé) to Inria, called Projet Actis (Algorithmic Coding Theory In Sage). The aim of this project is to vastly improve the state of the error correcting library in Sage. The existing library does not present a good and usable API, and the provided algorithms are very basic, irrelevant, and outdated. We thus have two directions for improvement: renewing the APIs to make them actually usable by researchers, and incorporating efficient programs for decoding, like J. Nielsen's CodingLib, which contains many new algorithms.

We hired D. Lucas on October 1st; he has started implementing various basic things, in a standalone manner. We plan to publish these snippets of code to the Sage community in January 2015. Our plan is to interact a lot with the Sage community, to ensure that our new APIs will cover most of the needs of various communities.

HYCOMES Team

5. New Software and Platforms

5.1. Mica: A Modal Interface Compositional Analysis Toolbox

Participant: Benoît Caillaud.

http://www.irisa.fr/s4/tools/mica/

Mica is an Ocaml library developed by Benoît Caillaud implementing the Modal Interface algebra published in [5], [4]. The purpose of Modal Interfaces is to provide a formal support to contract based design methods in the field of system engineering. Modal Interfaces enable compositional reasoning methods on I/O reactive systems.

In Mica, systems and interfaces are represented by extension. However, a careful design of the state and event heap enables the definition, composition and analysis of reasonably large systems and interfaces. The heap stores states and events in a hash table and ensures structural equality (there is no duplication). Therefore complex data-structures for states and events induce a very low overhead, as checking equality is done in constant time.

Thanks to the Inter module and the mica interactive environment, users can define complex systems and interfaces using Ocaml syntax. It is even possible to define parameterized components as Ocaml functions.

Mica is available as an open-source distribution, under the CeCILL-C Free Software License Agreement (http://www.cecill.info/licences/Licence_CeCILL-C_V1-en.html).

5.2. Flipflop and TnF-C++: Test and Flip Net Synthesis Tools for the Automated Synthesis of Surgical Procedure Models

Participant: Benoît Caillaud.

http://tinyurl.com/oql6f3y

Flipflop is a Test and Flip net synthesis tool implementing a linear algebraic polynomial time algorithm. Computations are done in the $\mathbb{Z}/2\mathbb{Z}$ ring. Test and Flip nets extend Elementary Net Systems by allowing test to zero, test to one and flip arcs. The effect of flip arcs is to complement the marking of the place. While the net synthesis problem has been proved to be NP hard for Elementary Net Systems, thanks to flip arcs, the synthesis of Test and Flip nets can be done in polynomial time. Test and flip nets have the required expressivity to give concise and accurate representations of surgical processes (models of types of surgical operations). Test and Flip nets can express causality and conflict relations. The tool takes as input either standard XES log files (a standard XML file format for process mining tools) or a specific XML file format for surgical applications. The output is a Test and Flip net, solution of the following synthesis problem: Given a finite input language (log file), compute a net, which language is the least language in the class of Test and Flip net languages, containing the input language.

TnF-C++ is a robust and portable re-implementation of Flipflop, developed in 2014 and integrated in the S3PM toolchain. Both software have been designed in the context of the S3PM project on surgical procedure modeling and simulation (see section 7.1).

LFANT Project-Team

4. New Software and Platforms

4.1. Pari/Gp

Participants: Karim Belabas [correspondent], Bill Allombert, Henri Cohen, Andreas Enge, Hamish Ivey-Law.

http://pari.math.u-bordeaux.fr/

PARI/GP is a widely used computer algebra system designed for fast computations in number theory (factorisation, algebraic number theory, elliptic curves, ...), but it also contains a large number of other useful functions to compute with mathematical entities such as matrices, polynomials, power series, algebraic numbers, etc., and many transcendental functions.

- PARI is a C library, allowing fast computations.
- GP is an easy-to-use interactive shell giving access to the PARI functions.
- gp2c, the GP-to-C compiler, combines the best of both worlds by compiling GP scripts to the C language and transparently loading the resulting functions into GP; scripts compiled by gp2c will typically run three to four times faster.
- Version of PARI/GP: 2.7.2
- Version of gp2c: 0.0.9
- License: GPL v2+
- Programming language: C

4.2. GNU MPC

Participants: Andreas Enge [correspondent], Mickaël Gastineau [CNRS], Philippe Théveny [INRIA project-team ARIC], Paul Zimmermann [INRIA project-team CARAMEL].

http://mpc.multiprecision.org/.

GNUMPC is a C library for the arithmetic of complex numbers with arbitrarily high precision and correct rounding of the result. It is built upon and follows the same principles as GNUMPFR.

It is a prerequisite for the GNU compiler collection GCC since version 4.5, where it is used in the C and Fortran front ends for constant folding, the evaluation of constant mathematical expressions during the compilation of a program. Since 2011, it is an official GNU project.

2012 has seen the first release of the major version 1.0.

- Version: 1.0.2 *Fagus silvatica*
- License: LGPL v3+
- ACM: G.1.0 (Multiple precision arithmetic)
- AMS: 30.04 Explicit machine computation and programs
- APP: Dépôt APP le 2003-02-05 sous le numéro IDDN FR 001 060029 000 R P 2003 000 10000
- Programming language: C

4.3. MPFRCX

Participant: Andreas Enge.

http://mpfrcx.multiprecision.org/

MPFRCX is a library for the arithmetic of univariate polynomials over arbitrary precision real (MPFR) or complex (MPC) numbers, without control on the rounding. For the time being, only the few functions needed to implement the floating point approach to complex multiplication are implemented. On the other hand, these comprise asymptotically fast multiplication routines such as Toom-Cook and the FFT.

- Version: 0.4.2 *Cassava*
- License: LGPL v2.1+
- Programming language: C

4.4. CM

Participant: Andreas Enge.

http://cm.multiprecision.org/

The CM software implements the construction of ring class fields of imaginary quadratic number fields and of elliptic curves with complex multiplication via floating point approximations. It consists of libraries that can be called from within a C program and of executable command line applications. For the implemented algorithms, see [8].

- Version: 0.2 *Blindhühnchen*
- License: GPL v2+
- Programming language: C

4.5. AVIsogenies

Participants: Damien Robert [correspondent], Gaëtan Bisson, Romain Cosset [INRIA project-team CARAMEL].

http://avisogenies.gforge.inria.fr/.

AVISOGENIES (Abelian Varieties and Isogenies) is a MAGMA package for working with abelian varieties, with a particular emphasis on explicit isogeny computation.

Its prominent feature is the computation of (ℓ, ℓ)-isogenies between Jacobian varieties of genus-two hyperelliptic curves over finite fields of characteristic coprime to ℓ; practical runs have used values of ℓ in the hundreds.

It can also be used to compute endomorphism rings of abelian surfaces, and find complete addition laws on them.

- Version: 0.6
- License: LGPL v2.1+
- Programming language: Magma

4.6. APIP

Participant: Jérôme Milan.

http://www.lix.polytechnique.fr/~milanj/apip/apip.xhtml

APIP, Another Pairing Implementation in PARI, is a library for computing standard and optimised variants of most cryptographic pairings.

The following pairings are available: Weil, Tate, ate and twisted ate, optimised versions (à la Vercauteren–Hess) of ate and twisted ate for selected curve families.

The following methods to compute the Miller part are implemented: standard Miller double-and-add method, standard Miller using a non-adjacent form, Boxall et al. version, Boxall et al. version using a non-adjacent form.

The final exponentiation part can be computed using one of the following variants: naive exponentiation, interleaved method, Avanzi–Mihailescu's method, Kato et al.'s method, Scott et al.'s method.

Part of the library has been included into PARI/GP proper.

- Version: 2012-10-17
- License: GPL v2+
- Programming language: C with libpari

4.7. CMH

Participants: Andreas Enge, Emmanuel Thomé [INRIA project-team CARAMEL].

http://cmh.gforge.inria.fr/

CMH computes Igusa class polynomials, parameterising two-dimensional abelian varieties (or, equivalently, Jacobians of hyperelliptic curves of genus 2) with given complex multiplication.

- Version: 1.0
- License: GPL v3+
- Programming language: C

4.8. Cubic

Participant: Karim Belabas.

http://www.math.u-bordeaux1.fr/~belabas/research/software/cubic-1.2.tgz

CUBIC is a stand-alone program that prints out generating equations for cubic fields of either signature and bounded discriminant. It depends on the PARI library. The algorithm has quasi-linear time complexity in the size of the output.

- Version: 1.2
- License: GPL v2+
- Programming language: C

4.9. Euclid

Participant: Pierre Lezowski.

http://www.math.u-bordeaux1.fr/~plezowsk/euclid/index.php.

Euclid is a program to compute the Euclidean minimum of a number field. It is the practical implementation of the algorithm described in [38]. Some corresponding tables built with the algorithm are also available. Euclid is a stand-alone program depending on the PARI library.

- Version: 1.2
- License: LGPL v2+
- Programming language: C

4.10. KleinianGroups

Participant: Aurel Page.

http://www.normalesup.org/~page/Recherche/Logiciels/logiciels.html

KLEINIANGROUPS is a Magma package that computes fundamental domains of arithmetic Kleinian groups.

- Version: 1.0
- License: GPL v3+
- Programming language: Magma

MARELLE Project-Team

5. New Software and Platforms

5.1. Coq

Participants: Enrico Tassi, Benjamin Grégoire.

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

5.2. Easycrypt

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

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

5.3. zoocrypt

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

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

5.4. CoqApprox

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

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

5.5. Ssreflect and Mathematical Components

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

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

MEXICO Project-Team

5. New Software and Platforms

5.1. Software

5.1.1. Software

5.1.1.1. Mole/Cunf: unfolders for Petri Nets

Participant: Stefan Schwoon [correspondant].

Mole computes, given a safe Petri net, a finite prefix of its unfolding. It is designed to be compatible with other tools, such as PEP and the Model-Checking Kit, which are using the resulting unfolding for reachability checking and other analyses. The tool Mole arose out of earlier work on Petri nets. Details on Mole can be found at http://www.lsv.ens-cachan.fr/~schwoon/tools/mole/. Mole served as an experimentation platform for several of our papers in recent years, most recently [33].

In the context of MExICo, we have created a new tool called Cunf [82], which is able to handle contextual nets, i.e. Petri nets with read arcs [80].While in principle every contextual net can be transformed into an equivalent Petri net and then unfolded using Mole, Cunf can take advantage of their special features to do the job faster and produce a smaller unfolding. Cunf has recently been extended with a verification component that takes advantage of these features; More details can be found at http://www.lsv.ens-cachan.fr/~rodrigue/tools/cunf/. Moreover, Cunf has been integrated into the CosyVerif environment (see section 5.2.1). Cunf has also participated in the Model Checking Contest held at the Petri Nets conference in 2013 and 2014.

5.1.1.2. TOURS: Testing On Unfolded Reactive Systems

Participant: Hernán Ponce de León [correspondant].

The MOLE - based testing tool TOURS [42] has been developed in 2014 with the help of intern Konstantinos Athanasiou, jointly supervised by Hernán Ponce de León and Stefan Schwoon of the MExICo team at LSV); it has served successfully to experiment the partial-order based testing methodology on a scalable benchmark example (elevator control).

5.1.1.3. COSMOS : a Statistical Model Checker for the Hybrid Automata Stochastic Logic

Participant: Benoît Barbot [correspondant].

COSMOS is a statistical model checker for the Hybrid Automata Stochastic Logic (HASL). HASL employs Linear Hybrid Automata (LHA), a generalization of Deterministic Timed Automata (DTA), to describe accepting execution paths of a Discrete Event Stochastic Process (DESP), a class of stochastic models which includes, but is not limited to, Markov chains. As a result HASL verification turns out to be a unifying framework where sophisticated temporal reasoning is naturally blended with elaborate reward-based analysis. COSMOS takes as input a DESP (described in terms of a Generalized Stochastic Petri Net), an LHA and an expression Z representing the quantity to be estimated. It returns a confidence interval estimation of Z; recently, it has been equipped with functionalities for rare event analysis. COSMOS is written in C++ and is freely available to the research community.

Details on COSMOS can be found at http://www.lsv.ens-cachan.fr/~barbot/cosmos/

5.2. Platforms

5.2.1. Platform CosyVerif

CosyVerif (http://www.cosyverif.org/) is a platform dedicated to the formal specification and verification of dynamic systems. It allows to specify systems using several formalisms (such as automata and Petri nets), and to run verification tools on these models. CosyVerif integrates several tools, that are mainly developed by researchers of the MeFoSyLoMa group (a Parisian verification group, http://www.mefosyloma.fr/).

The platform is client/server based. The modeler creates models on the client side, either programmatically, or in a dedicated graphical editor. Tools are then executed on the server side.

CosyVerif is available as installable bundles, that embed the client, the server, and also the tools. It is also usable through a public server hosted within the laboratory.

The platform offers a common language for the description of the models, in order to create interoperability between clients and tools. It also provides a way to define easily new formalisms within the platform, and to manipulate models that are instances of these formalisms. To the best of our knowledge, no other verification framework presents such a feature.

CosyVerif targets three different kinds of users:

- Students use this platform in two M2 courses in modeling and verification courses. *Citer les deux cours*
- Tool developers, that are usually researchers, use the platform to distribute their tools, and have a demonstration version easily available. They also use CosyVerif for tutorials in conferences or workshops *Citer Petri nets 2014*.
- Industrial case studies have used the platform since its creation to prove properties on systems in various fields, such as: transportation systems, scheduling, hardware, robotics, databases, banking systems, home automation...

The platform is managed by a steering committee consisting of researchers and engineers of three laboratories: LIP6, LIPN, LSV. This committee decides strategic orientations as well as technical choices.

This year, we have fully redesigned the platform, with two goals in mind: first, to use technologies that target better our users; and second, to provide more functionalities.

- We switched to lightweight web technologies, in order to ease the deployment and use of CosyVerif. For the users, it means that they can access a graphical editor within their web browser. They can also access the platform through an API, usable with any HTTP client.
- We improved the language for formalisms and models in order to allow the modular definition of new formalisms. We switched from a class/instance paradigm to a prototype one, that allows to represent complex models in a both efficient and usable way.
- We extended the server to handle not only executions. It is now primarily a repository of formalismes, models, services and executions, that belong to users or project. It also handles the tools executions, and the collaborative edition of models.
- We started working on a system to help building packages for the various components of the platform (client, server, tools, ...), to ease its installation. It is used to create the bundles of CosyVerif, that are available to download. Another team (Secsi) of the LSV laboratory is interested in this system, and will support its development in 2015.

All the developed software are open source and free software tools.

Two engineers have worked this year on CosyVerif:

- Francis Hulin-Hubard, part-time (CNRS engineer);
- Alban Linard, full-time (Inria engineer).

CosyVerif has been used for teaching in two master programs (Universities Paris 6 and Paris 13/Villetaneuse) It has been used in a tutorial in the Petri Nets 2014 conference.

We are currently in the process of giving a better visibility to the project, by transforming it into a consortium. Our goal is to identify industrial fields where the tools of the platform can be applied successfully, by proposing services to the industry. The strength of the platform relies on the variety of techniques offered by the tools, that adapt to a wide range of problems. In order to increase the number of techniques, we have been joined by another partner from Geneva.

MUTANT Project-Team

5. New Software and Platforms

5.1. Antescofo

Participants: Arshia Cont, Jean-Louis Giavitto, Florent Jacquemard, José Echeveste.

Antescofo is a modular polyphonic Score Following system as well as a Synchronous Programming language for musical composition. The module allows for automatic recognition of music score position and tempo from a realtime audio Stream coming from performer(s), making it possible to synchronize an instrumental performance with computer realized elements. The synchronous language within Antescofo allows flexible writing of time and interaction in computer music.

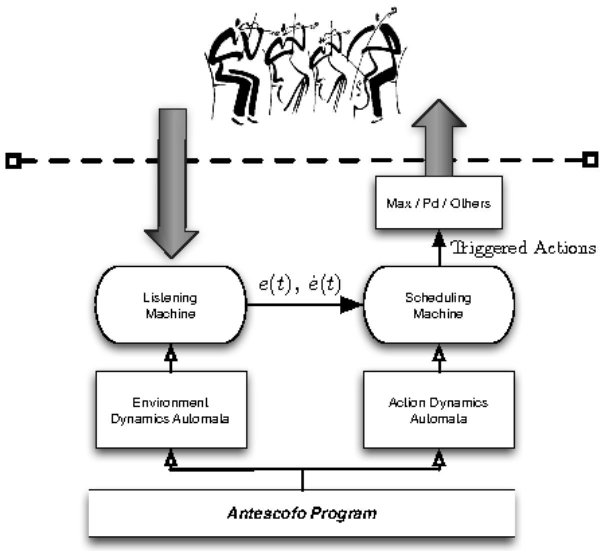


Figure 4. General scheme of Antescofo virtual machine

A complete new version of Antescofo has been released in November 2014 on Ircam Forumnet. This version includes major improvements over the previous version as a result of MuTant's research and development. Its development has benefited from of intensive interactions with composers, especially Julia Blondeau, José-Miguel Fernandez, and Marco Stroppa.

One major and sensible improvement is a total review of *Antescofo*'s realtime machine listening as a result of [12], which allows robust recognition in highly polyphonic and noisy environments with the help of novel notions of *Probabilistic Time Coherency*. This improvement allowed team members to participate in collaborative work with Paris Orchestra among others.

The 2014 release of *Antescofo* also includes new anticipative synchronization strategies. In the context of the PhD of José Echeveste, they are systematically studied with the help of the Orchestre de Paris and the composer Marco Stroppa. This work won the best presentation award at ICMC 2014.

The new internal architecture unifies the handling of external (musical) events and the handling of internal (logical) events in a framework able to manage multiple time frames (relative, absolute or computed). The notion of synchronization has been extended to be able to synchronize on the update of a variable in addition to the update of the listening machine. This mechanism offers new opportunities, especially in the context of open scores and improvised music [16].

The 2014 version targets the Max and PureData (Pd) environments on Mac, but also on Linux on Windows (Pd version). A standalone version is used to simulate a performance in Ascograph and to offer batch processing capabilities as well as a testing framework.

5.2. Ascograph: Antescofo Visual Editor

Participants: Thomas Coffy [ADT], Arshia Cont.

The Antescofo programming language can be extended to visual programing to better integrate existing scores and to allow users to construct complex and embedded temporal structures that are not easily integrated into text. This project is held since October 2012 thanks to Inria ADT Support.

AscoGraph, the Antescofo graphical score editor released in 2013, provides a autonomous Integrated Development Environment (IDE) for the authoring of Antescofo scores. Antescofo listening machine, when going forward in the score during recognition, uses the message passing paradigm to perform tasks such as automatic accompaniment, spatialization, etc. The Antescofo score is a text file containing notes (chord, notes, trills, ...) to follow, synchronization strategies on how to trigger actions, and electronic actions (the reactive language). This editor shares the same score parsing routines with Antescofo core, so the validity of the score is checked on saving while editing in AscoGraph, with proper parsing errors handling. Graphically, the application is divided in two parts (see Figure 2). On the left side, a graphical representation of the score, using a timeline with tracks view. On the right side, a text editor with syntax coloring of the score is displayed. Both views can be edited and are synchronized on saving. Special objects such as "curves", are graphically editable: they are used to provide high-level variable automation facilities like breakpoints functions (BPF) with more than 30 interpolations possible types between points, graphically editable.

An important feature of AscoGraph is the score import from MusicXML or MIDI files, which make the complete workflow of the composition of a musical piece much easier than before.

AscoGraph is strongly connected with Antescofo core object (using OSC over UDP): when a score is edited and modified it is automatically reloaded in Antescofo, and on the other hand, when Antescofo follows a score (during a concert or rehearsal) both graphical and textual view of the score will scroll and show the current position of Antescofo.

AscoGraph is released under Open-Source MIT license and has been released publicly along with new Antescofo architecture during IRCAM Forum 2013. Recent development was published in [11].

5.3. Antescofo Timed Test Platform

Participants: Florent Jacquemard, Clément Poncelet.

The frequent use of Antescofo in live and public performances with human musicians implies strong requirements of temporal reliability and robustness to unforeseen errors in input. To address these requirements and help the development of the system and authoring of pieces by users, we are developing a platform for the automation of testing the behavior of Antescofo on a given score, with of focus on timed behavior. It makes possible to automate the following main tasks:

1. (1) generation of relevant input data for testing, with the sake of exhaustiveness,
2. (2) computation of the corresponding expected output, according to a formal specification of the expected behavior of the system on a given mixed score,
3. (3) black-box execution of the input test data,
4. (4) comparison of expected and real output and production of a test verdict.

The input and output data are timed traces (sequences of events together with inter-event durations).

Our platform uses state of the art techniques and tools for *model-based testing* of embedded systems [31]. Some models of The environment of the system (the musicians) and the expected behavior of the system are both represented by formal models. We have developed a compiler for producing automatically such models, in an intermediate representation language (IR), from mixed scores. The IR are in turn converted into Timed Automata and passed, to the model-checker Uppaal.

Uppaal is used, with its extension Cover, for the above generation Task (1). Following some *coverage criteria*, this tools makes a systematic exploration of the state space of the model. We propose also an alternative approach for the generation of input traces by *fuzzing* of an ideal trace obtained from the score (a trace represented a perfectly timed performance of the score).

Task (2) is also performed by Uppaal, by simulation, using the model of the system and the generated test input.

Moreover, we have implemented several tools for Tasks (3) and (4), corresponding to different boundaries for the implementation under test (black box): e.g. the interpreter of Antescofo's synchronous language alone, or with tempo detection, or the whole system.

PAREO Project-Team

5. New Software and Platforms

5.1. ATerm

Participant: Pierre-Etienne Moreau [correspondant].

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

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

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

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

5.2. Tom

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

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

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

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

PARKAS Project-Team

5. New Software and Platforms

5.1. Lucid Synchrone

Participant: Marc Pouzet [contact].

Synchronous languages, type and clock inference, causality analysis, compilation

Lucid Synchrone is a language for the implementation of reactive systems. It is based on the synchronous model of time as provided by Lustre combined with features from ML languages. It provides powerful extensions such as type and clock inference, type-based causality and initialization analysis and allows to arbitrarily mix data-flow systems and hierarchical automata or flows and valued signals.

It is distributed under binary form, at URL http://www.di.ens.fr/~pouzet/lucid-synchrone/.

The language was used, from 1996 to 2006 as a laboratory to experiment various extensions of the language Lustre. Several programming constructs (e.g. merge, last, mix of data-flow and control-structures like automata), type-based program analysis (e.g., typing, clock calculus) and compilation methods, originaly introduced in Lucid Synchrone are now integrated in the new SCADE 6 compiler developped at Esterel-Technologies and commercialized since 2008.

Three major release of the language has been done and the current version is V3 (dev. in 2006). As of 2014, the language is still used for teaching and in our research but we do not develop it anymore. Nonetheless, we have integrated several features from Lucid Synchrone in new research prototypes described below. The Heptagon language and compiler are a direct descendent of it. The new language Zélus for hybrid systems modeling borrows many features originaly introduced in Lucid Synchrone.

5.2. ReactiveML

Participant: Guillaume Baudart [contact].

Programming language, synchronous reactive programming, concurrent systems, dedicated type-systems.

With Louis Mandel (IBM Watson, USA) and Cédric Pasteur.

ReactiveML is a programming language dedicated to the implementation of interactive systems as found in graphical user interfaces, video games or simulation problems. ReactiveML is based on the synchronous reactive model due to Boussinot, embedded in an ML language (OCaml).

The Synchronous reactive model provides synchronous parallel composition and dynamic features like the dynamic creation of processes. In ReactiveML, the reactive model is integrated at the language level (not as a library) which leads to a safer and a more natural programming paradigm.

ReactiveML is distributed at URL http://reactiveml.org. The compiler is distributed under the terms of the Q Public License and the library is distributed under the terms of the GNU Library General Public License. The development of ReactiveML started at the University Paris 6 (from 2002 to 2006).

The language was mainly used for the simulation of mobile ad hoc networks at the Pierre and Marie Curie University and for the simulation of sensor networks at France Telecom and Verimag (CNRS, Grenoble). A new application to mixed music programming has been developed.

5.3. Heptagon

Participants: Adrien Guatto, Marc Pouzet [contact].

Synchronous languages, compilation, optimizing compilation, parallel code generation, behavioral synthesis.

With Cédric Pasteur, Léonard Gérard, and Brice Gelineau.

Heptagon is an experimental language for the implementation of embedded real-time reactive systems. It is developed inside the Synchronics large-scale initiative, in collaboration with Inria Rhones-Alpes. It is essentially a subset of Lucid Synchrone, without type inference, type polymorphism and higher-order. It is thus a Lustre-like language extended with hierchical automata in a form very close to SCADE 6. The intention for making this new language and compiler is to develop new aggressive optimization techniques for sequential C code and compilation methods for generating parallel code for different platforms. This explains much of the simplifications we have made in order to ease the development of compilation techniques.

Some extensions have already been made, most notably automata, a parallel code generator with Futures, support for correct and efficient in-place array computations. It's currently used to experiment with linear typing for arrays and also to introduce a concept of asynchronous parallel computations. The compiler developed in our team generates C, C++, java and VHDL code.

Transfer activities based on our experience in Heptagon are taking place through the "Fiabilité and Sûreté de Fonctionnement" project at IRT SystemX, led by Alstom Transport, since 2013.

Heptagon is jointly developed with Gwenael Delaval and Alain Girault from the Inria POP ART team (Grenoble). Gwenael Delaval is developing the controller synthesis tool BZR (http://bzr.inria.fr/) above Heptagon. Both software are distributed under a GPL licence.

5.4. Lucy-n: an n-synchronous data-flow programming language

Participants: Albert Cohen, Adrien Guatto, Marc Pouzet.

With Louis Mandel (IBM Watson, USA).

Lucy-n is a language to program in the n-synchronous model. The language is similar to Lustre with a buffer construct. The Lucy-n compiler ensures that programs can be executed in bounded memory and automatically computes buffer sizes. Hence this language allows to program Kahn networks, the compiler being able to statically compute bounds for all FIFOs in the program.

The language compiler and associated tools are available in a binary form at http://www.lri.fr/~mandel/lucy-n.

In 2013, a complete re-implementation has been started. This new version will take into account the new features developed during the PhD of Adrien Guatto. Parallel code generation for this new version also involves compilation and runtime system research in collaboration with Nhat Minh Lê and Robin Morisset.

5.5. ML-Sundials

Participants: Timothy Bourke, Jun Inoue, Marc Pouzet [contact].

Sundials/ML is a comprehensive OCaml interface to the Sundials suite of numerical solvers (CVODE, CVODES, IDA, IDAS, KINSOL). Its structure mostly follows that of the Sundials library, both for ease of reading the existing documentation and for adapting existing source code, but several changes have been made for programming convenience and to increase safety, namely:

- solver sessions are mostly configured via algebraic data types rather than multiple function calls;
- errors are signalled by exceptions not return codes (also from user-supplied callback routines);
- user data is shared between callback routines via closures (partial applications of functions);
- vectors are checked for compatibility (using a combination of static and dynamic checks); and
- explicit free commands are not necessary since OCaml is a garbage-collected language.

OCaml versions of the standard examples usually have an overhead of about 50% compared to the original C versions, and almost never more than 100%.

The current version of Sundials/ML comprises about 30,000 lines of OCaml (plus 10,000 lines of api documentation) and 12,000 lines of C (plus 1000 lines of commentary). In comparison to our previous development (called ML-Sundials), the current version includes a major rewrite of the 'nvector' interface to allow easier generalisation to parallel and custom vectors (both of which have now been implemented), a rewrite of the linear solver interfaces, a redesign of the linear solver interface (now including the ability to specify linear solvers in OCaml), and the inclusion of the CVODES and IDAS solvers.

Sundials/ML allows the use of the state-of-the-art Sundials numerical simulation library from OCaml programs. We use it within PARKAS for the Zélus compiler (documented elsewhere) and our ongoing experiments with Modelica. The binding is, however, complete and general purpose. It can potentially replace the less complete libraries underlying three or four open source projects.

The Sundials/ML source code has now been released under a BSD-3 license. It is available on github and through opam.

5.6. Zélus

Participants: Timothy Bourke, Marc Pouzet [contact].

Zélus is a new programming language for hybrid system modeling. It is based on a synchronous language but extends it with Ordinary Differential Equations (ODEs) to model continuous-time behaviors. It allows for combining arbitrarily data-flow equations, hierarchical automata and ODEs. The language keeps all the fundamental features of synchronous languages: the compiler statically ensure the absence of deadlocks and critical races; it is able to generate statically scheduled code running in bounded time and space and a type-system is used to distinguish discrete and logical-time signals from continuous-time ones. The ability to combines those features with ODEs made the language usable both for programming discrete controllers and their physical environment.

The Zélus implementation has two main parts: a compiler that transforms Zélus programs into OCaml programs and a runtime library that orchestrates compiled programs and numeric solvers. The runtime can use the Sundials numeric solver, or custom implementations of well-known algorithms for numerically approximating continuous dynamics.

This year we reimplemented several basic numeric solver algorithms after a careful analysis of the Simulink versions together with the binding to SUNDIALS CVODE. This was necessary to enable detailed comparsions between our tool and Simulink (the de facto industrial standard in this domain). We also improved the algorithm for zero-crossing detection, simplified and streamlined the back-end interface.

We developed several new examples to aid in the development, debugging, and dissemination of our work together with various talks and demonstrations. These included a simple backhoe model (which served as a introducing example in the HSCC paper), an adaptive control example from Astrom and Wittenmark's text, and a model of Zeno behaviour based on a zig-zagging object (presented at Synchron).

Zélus was been released officially in 2013 with several complete documented examples on http://zelus.di.ens.fr. Work continued in 2014 with many refinements to the compilation passes. The runtime has also been improved and simplified.

5.7. GCC

Participants: Albert Cohen [contact], Tobias Grosser, Feng Li, Riyadh Baghdadi, Nhat Minh Lê.

Compilation, optimizing compilation, parallel data-flow programming automatic parallelization, polyhedral compilation. http://gcc.gnu.org

Licence: GPLv3+ and LGPLv3+

The GNU Compiler Collection includes front ends for C, C++, Objective-C, Fortran, Java, Ada, and Go, as well as libraries for these languages (libstdc++, libgcj,...). GCC was originally written as the compiler for the GNU operating system. The GNU system was developed to be 100% free software, free in the sense that it respects the user's freedom.

PARKAS contributes to the polyhedral compilation framework, also known as Graphite. We also distribute an experimental branch for a stream-programming extension of OpenMP called OpenStream (used in numerous research activites and grants). This effort borrows key design elements to synchronous data-flow languages.

Tobias Grosser is one of main contributors of the Graphite optimization pass of GCC.

5.8. isl

Participants: Sven Verdoolaege [contact], Tobias Grosser, Albert Cohen.

Presburger arithmetic, integer linear programming, polyhedral library, automatic parallelization, polyhedral compilation. http://freshmeat.net/projects/isl

Licence: MIT

isl is a library for manipulating sets and relations of integer points bounded by linear constraints. Supported operations on sets include intersection, union, set difference, emptiness check, convex hull, (integer) affine hull, integer projection, transitive closure (and over-approximation), computing the lexicographic minimum using parametric integer programming. It includes an ILP solver based on generalized basis reduction, and a new polyhedral code generator. isl also supports affine transformations for polyhedral compilation, and increasingly abstract representations to model source and intermediate code in a polyhedral framework.

isl has become the de-facto standard for every recent polyhedral compilation project. Thanks to a license change from LGPL to MIT, its adoption is also picking up in industry.

5.9. ppcg

Participants: Sven Verdoolaege [contact], Tobias Grosser, Riyadh Baghdadi, Albert Cohen.

Presburger arithmetic, integer linear programming, polyhedral library, automatic parallelization, polyhedral compilation. http://freshmeat.net/projects/ppcg

Licence: MIT

More tools are being developed, based on isl. PPCG is our source-to-source research tool for automatic parallelization in the polyhedral model. It serves as a test bed for many compilation algorithms and heuristics published by our group, and is currently the best automatic parallelizer for CUDA and OpenCL (on the Polybench suite).

5.10. Tool support for the working semanticist

Participants: Basile Clément, Francesco Zappa Nardelli [contact].

Languages, semantics, tool support, theorem prouvers.

We are working on tools to support large scale semantic definitions, for programming languages and architecture specifications. For that we develop two complementary tools, Ott and Lem.

Ott is a tool for writing definitions of programming languages and calculi. It takes as input a definition of a language syntax and semantics, in a concise and readable ASCII notation that is close to what one would write in informal mathematics. It generates output:

1. a LaTeX source file that defines commands to build a typeset version of the definition;
2. a Coq version of the definition;
3. an Isabelle version of the definition; and
4. a HOL version of the definition.

Additionally, it can be run as a filter, taking a LaTeX/Coq/Isabelle/HOL source file with embedded (symbolic) terms of the defined language, parsing them and replacing them by typeset terms.

The main goal of the Ott tool is to support work on large programming language definitions, where the scale makes it hard to keep a definition internally consistent, and to keep a tight correspondence between a definition and implementations. We also wish to ease rapid prototyping work with smaller calculi, and to make it easier to exchange definitions and definition fragments between groups. The theorem-prover backends should enable a smooth transition between use of informal and formal mathematics.

Lem is a lightweight tool for writing, managing, and publishing large scale semantic definitions. It is also intended as an intermediate language for generating definitions from domain-specific tools, and for porting definitions between interactive theorem proving systems (such as Coq, HOL4, and Isabelle). As such it is a complementary tool to Ott. Lem resembles a pure subset of Objective Caml, supporting typical functional programming constructs, including top-level parametric polymorphism, datatypes, records, higher-order functions, and pattern matching. It also supports common logical mechanisms including list and set comprehensions, universal and existential quantifiers, and inductively defined relations. From this, Lem generates OCaml, HOL4, Coq, and Isabelle code.

In collaboration with Peter Sewell (Cambridge University) and Scott Owens (University of Kent).

The current version of Ott is about 30000 lines of OCaml. The tool is available from http://moscova.inria.fr/~zappa/software/ott (BSD licence). It is widely used in the scientific community.

The development version of Lem is available from http://www.cs.kent.ac.uk/people/staff/sao/lem/.

In addition to the usual bug-fixes, in 2014 we have investigated several approaches to interactively explore a semantics definition, with the aim of building a toolbox to debug operational semantics and to attempt to falsify expected properties. This code is not yet released.

5.11. Cmmtest: a tool for hunting concurrency compiler bugs

Participants: Francesco Zappa Nardelli [contact], Robin Morisset, Pejman Attar.

Languages, concurrency, memory models, C11/C++11, compiler, bugs.

The Cmmtest tool performs random testing of C and C++ compilers against the C11/C++11 memory model. A test case is any well-defined, sequential C program; for each test case, cmmtest:

1. compiles the program using the compiler and compiler optimisations that are being tested;
2. runs the compiled program in an instrumented execution environment that logs all memory accesses to global variables and synchronisations;
3. compares the recorded trace with a reference trace for the same program, checking if the recorded trace can be obtained from the reference trace by valid eliminations, reorderings and introductions.

Cmmtest identified several mistaken write introductions and other unexpected behaviours in the latest release of the gcc compiler. These have been promptly fixed by the gcc developers.

Cmmtest is available from http://www.di.ens.fr/~zappa/projects/cmmtest/ and a list of bugs reported thanks to cmmtest is available from http://www.di.ens.fr/~zappa/projects/cmmtest/gcc-bugs.html.

In 2014 Cmmtest has been used by the ThreadSanitizer team at Google to debug some subtle false positive race reports, due to the compiler introducing memory accesses.

PARSIFAL Project-Team

5. New Software and Platforms

5.1. Abella

Participants: Kaustuv Chaudhuri [correspondant], Matteo Cimini [Indiana University], Dale Miller, Olivier Savary-Bélanger [Princeton University], Mary Southern [University of Minnesota], Yuting Wang [University of Minnesota].

Main web-site: http://abella-prover.org.

Abella is an interactive theorem prover for reasoning about data structures with binding constructs using the λ-tree approach to syntax. It consists of a sophisticated reasoning logic that supports induction, co-induction, and generic reasoning. Abella also supports the *two-level logic approach* by means of a specification logic based on the logic programming language λProlog.

In 2014, the following additions were made to the system.

- A new translation layer was added to Abella's specification layer, which was used to build an interface to the LF dependent type theory [61]. This extension was documented in the following paper: [27]. A number of examples of the use of this new specification language are available at the following URL: http://abella-prover.org/lf
- Two minor releases were made, versions 2.0.2 and 2.0.3, that fixed a number of bugs and added several convenience features. Consult the change log for more details.

Accompanying these additions were the following publications.

- A new comprehensive tutorial for the Abella system has been accept to appear in the *Journal of Formalized Reasoning* [31].
- The new tactical plugin architecture and the dynamic contexts plugin of Abella in the following paper: [26].
- The use of co-induction and higher-order relations to formalize the meta-theory of various bisimulation-up-to techniques for common process calculi: [19].

5.2. Bedwyr

Participants: Quentin Heath, Dale Miller [correspondant].

Main web-site: http://slimmer.gforge.inria.fr/bedwyr/.

Quentin Heath has continued to maintain and enhance this model checking system. In particular, the tabling mechanism has been extended and formalized to a greater extent. The tabling mechanism is now able to use Horn clause lemmas in order to increase the power of the table. For example, given this enhancement it is possible to tell Bedwyr that if a given board position (in some game) has a winning strategy then symmetric versions of that board also have winning strategies. Thus, when a given board position is recognized as winning, then table will understand that all symmetric versions of that board are winning.

Significant energies have also gone into trying to understand how cyclic proofs (recognized using the tabling mechanism) can be turned into certifiable proof evidence. Good results are currently developed for treating bisimulation and non-reachability: in these cases, cyclic proofs are used to supply invariants for induction and co-induction.

5.3. Psyche

Participants: Stéphane Graham-Lengrand [correspondant], Assia Mahboubi, Jean-Marc Notin.

Psyche (*Proof-Search factorY for Collaborative HEuristics*) is a modular proof-search engine whose first version, 1.0, was released in 2012:
http://www.lix.polytechnique.fr/~lengrand/Psyche/

The engine implements the ideas developed in the section "Trustworthy implementations of theorem proving techniques" above, and was the object of the system description [56].

Psyche's proof-search mechanism is simply the incremental construction of proof-trees in the polarized and focused sequent calculus. Its architecture organizes an interaction between a trusted universal kernel and smart plugins that are meant be efficient at solving certain kinds of problems:

The kernel contains the mechanisms for exploring the proof-search space in a sound and complete way, taking into account branching and backtracking. The output of Psyche comes from the (trusted) kernel and is therefore correct by construction. The plugins then drive the kernel by specifying how the branches of the search space should be explored, depending on the kind of problem that is being treated. The quality of the plugin is how fast it drives the kernel towards the final answer.

In 2014, major developments were achieved in Psyche, whose version 2.0 was released on 20th September 2014. It is now equipped with the machinery to handle quantifiers and quantifier-handling techniques. Concretely, it uses meta-variables to delay the instantiation of existential variables, and constraints on meta-variables are propagated through the various branches of the search-space, in a way that allows local backtracking. The kernel, of about 800 l.o.c., is purely functional.

PI.R2 Project-Team

4. New Software and Platforms

4.1. COQ (http://coq.inria.fr)

Participants: Bruno Barras [Inria Saclay], Yves Bertot [Marelle team, Sophia], Pierre Boutillier, Xavier Clerc [SED team], Pierre Courtieu [CNAM], Maxime Dénès [Gallium team, Rocquencourt], Julien Forest [CNAM], Stéphane Glondu [CARAMEL team, Nancy Grand Est], Benjamin Grégoire [Marelle team, Sophia], Vincent Gross [Consultant at NBS Systems], Hugo Herbelin [correspondant], Pierre Letouzey, Assia Mahboubi [SpecFun team, Saclay], Julien Narboux [University of Strasbourg], Jean-Marc Notin [Ecole Polytechnique], Christine Paulin [Toccata team, Saclay], Pierre-Marie Pédrot, Loïc Pottier [Marelle team, Sophia], Matthias Puech, Yann Régis-Gianas, François Ripault, Matthieu Sozeau, Arnaud Spiwack [Mines Paritech], Pierre-Yves Strub [IMDEA, Madrid], Enrico Tassi [Marelle team, Sophia], Benjamin Werner [Ecole Polytechnique].

4.1.1. Version 8.5

Version 8.5 was expected to be released after the summer of 2014, but this got delayed until the Coq Programming Language workshop mid-January 2015.

Coq 8.5 is a major release of the Coq proof assistant, including 5 major new features:

- Parallel development and compilation, inside files and across files, by Enrico Tassi (Inria SpecFun, then Marelle), a result of the Paral-ITP ANR project.
- Availability of all the features of Arnaud Spiwack's new proof engine, with more expressive, clearer semantics, multigoal tactics, deep backtracking,
- A compilation scheme from Coq to OCaml to native code by Maxime Dénès and Benjamin Grégoire (Inria Marelle, then University of Pennsylvania, then Inria Gallium), considerably improving on the previous virtual machine implementation by B. Grégoire.
- A Universe Polymorphic extension by Matthieu Sozeau that allows universe-generic developments, as required by the Homotopy Type Theory library for example,
- Primitive projections for records by Matthieu Sozeau, with significant efficiency improvements.

Coq 8.5 also includes many improvements at different levels: the primitive tactics, the tactic language, the specification language, the tools associated to Coq, etc. For a full list of changes, the reader is invited to look at http://coq.inria.fr or at the files CHANGES of the Coq archive.

4.1.2. Evaluation algorithms

The new unfolding algorithm for global constants that was proposed by Pierre Boutillier is ready for use in Coq 8.5.

4.1.3. Internal representation of projections

A new internal representation of record projections has been implemented in the 8.5 release by Matthieu Sozeau. During the stabilisation of this feature, we added a backwards compatibility layer that allows users to switch seamlessly to the new representation, keeping the same user-level interface for primitive and non-primitive projections (the record types and values being unchanged). This new representation adds eta-conversion of records defined with primitive projections to the definitional equality of Coq, enlarging the set of conversion problems that can be automatically handled by the system.

4.1.4. Universes

The new universe polymorphism system by Matthieu Sozeau is part of the 8.5 release. The implementation has been stabilised, benchmarked and tested heavily in the last year, with much input from the Homotopy Type Theory development team. In [27], Matthieu Sozeau and Nicolas Tabareau presented the system formally. It has since been extended with user-friendly features like named universes and commands to display the status of universe constraints. With the help from Maxime Dénès (Gallium Team), the native compilation system has also been extended to fully support universe polymorphism.

4.1.5. Internal architecture of the Coq software

Pierre Letouzey, Pierre-Marie Pédrot and Xavier Clerc have continued to work at improving the quality of the OCaml code which composes Coq :

- Many modules have been revised, in particular with cleaner naming conventions.
- Almost all uses of the generic OCaml comparison have been chased and transformed into specific code, thereby avoiding many potential bugs with advanced structures, while improving performances at the same time.
- The codes handling OCaml exceptions have been reworked to avoid undue interceptions of critical exceptions.
- Issues involving exceptions are now quite simpler to debug, thanks to easy-to-obtain backtraces.

4.1.6. Efficiency

Pierre-Marie Pédrot has been working on the overall optimisation of Coq, by tracking hotspots in the code. Coq trunk is currently much more efficient than its v8.4 counterpart, and is about as quick as v8.3, while having been expanded with a lot of additional features.

4.1.7. Documentation generation

Yann Régis-Gianas continued the development of a new version of coqdoc, the documentation generator of Coq. This new implementation is based on the interaction protocol with the Coq system and should be more robust with respect to the evolution of Coq.

4.1.8. Maintenance and coordination

The maintenance and coordination of Coq has been jointly done by Hugo Herbelin, Pierre Boutillier, Pierre Letouzey, Matthieu Sozeau, Pierre-Marie Pédrot, in relation with the other participants to the development.

A Coq working group is organised every two months (5 times a year). From the end of October, a Coq lunch holds weekly welcoming any person interested in the development of Coq in general. Discussions about the development happen, in particular, on `coq-dev@inria.fr` and http://coq.inria.fr/bugs.

4.1.9. The Coq extraction

In 2014, Pierre Letouzey built an extension of the Coq extraction that targets directly one of the internal layers of the OCaml compiler. This way, it is possible to avoid the generation of OCaml concrete syntax by the extraction, followed by a parsing phase when the OCaml compiler is launched on the extracted code. Our extension is able to shortcut these two phases. The interest is twofold. First, it seriously reduces the amount of code that should be considered as critical during a program development via extraction. Secondly, with this approach we are able to directly compile and run certain extracted examples, and internalise the result back into Coq, leading to a new promising command Extraction Compute. This extension is currently quite experimental.

4.1.10. Parametricity for the Coq proof assistant

During his stay in the πr^2 team, Marc Lasson developed a plugin for parametricity theory in the Coq proof assistant.

Parametricity theory was originally introduced by John Reynolds in his seminal paper about polymorphic λ-calculus (also known as System F). It is used to formalise the opacity of abstract datatypes in programming languages that provide idioms to handle types generically. Polymorphic functions cannot inspect their arguments with an abstract type, and have to use them uniformly. The main tool of parametricity theory is that of logical relations, which are relations between programs of the same type that are defined by induction on the structure of types.

Marc Lasson's work consisted in developing a parametricity theory for the terms of Coq. The result of this work is a new plugin for the proof assistant that computes logical relations as well as the proof witnesses that programs satisfy these logical relations. It is available on github http://github.com/mlasson/paramcoq.

The purpose of this plugin is to allow to use parametric arguments in Coq proofs, the main direct application is the certification of parametric programs. Thanks to powerful expressiveness of the proof assistant, this plugin will allow future users to use parametric arguments to a larger scale. Although parametricity theory was originally developed for studying programs, the fact that we can use it in a proof assistant enables new uses in other contexts, such as the formalisation of mathematics and the meta-theory of proof assistants).

In [24], Marc Lasson showed that parametricity may also be useful to derive properties about the groupoidal interpretation of Type Theory. It was known that the equality types (also known as identity types) of type theory carry the algebraic structure of ω-groupoids (which is a higher-dimensional version of groups). Parametricity theory allows us to prove that the terms witnessing these algebraic laws are canonical, in the sense that there is only one way to implement them (up to higher-order equalities).

4.1.11. Formalisation in Coq

Hugo Herbelin's type-theoretic construction of semi-simplicial sets [9] has been formalised in Coq.

Matthieu Sozeau and Nicolas Tabareau formalised a setoid model of type theory in Coq http://github.com/mattam82/groupoid. They are working on extending this work to the groupoid model using the latest tools available in Coq 8.5.

Frédéric Loulergue collaborates with Frédéric Dabrowski and Thomas Pinsard (Univ. Orléans) to verify in Coq the compilation pass [21] for a language with nested atomic sections and thread escape to a language with only threads and locks, building on [45].

4.1.12. Systematic development of programs for parallel and cloud computing

During his stay in the πr^2 team, Frédéric Loulergue continues to collaborate with Kento Emoto (Kyushu Institute of Technology), Zhenjiang Hu (National Institute for Informatics, Japan), Julien Tesson (Univ. Paris-Est Créteil), Wadoud Bousdira (Univ. Orléans), Kiminori Matsuzaki (Kochi University of Technology) and Vitor Rodrigues (Rochester Institute of Technology) to develop the SyDPaCC framework (http://traclifo.univ-orleans.fr/SyDPaCC).

The goal of this framework is to ease the systematic development of correct parallel programs, in particular large scale data-intensive applications. In Coq, users write inefficient (sequential) functional programs and through (partly automated) program transformations based on the theory of list homomorphisms [32], bulk synchronous parallel homomorphisms [59] and semi-ring homomorphisms [48], an efficient sequential version is obtained. This version can then be automatically parallelised thanks to type class instance resolution and instances relating specific functions to their parallel counterparts. The parallel versions of the programs are written with a Coq axiomatisation of Bulk Synchronous Parallel ML (BSML) primitives. To obtain the final code, these Coq programs are extracted towards OCaml with calls to a parallel implementation of the BSML library.

As the SyDPaCC framework currently mixes certified code extracted from Coq and unverified code, Frédéric Loulergue and Pierre Letouzey are working on an extended extraction that generates, when possible, OCaml asserts for preconditions on function arguments. The next version of the generate-test-aggregate library of SyDPaCC will use Marc Lasson's plugin for parametricity to prove a "theorem for free": currently only instantiations of this theorem for each provided generator are proved.

4.1.13. Proofs of algorithms on graphs

Jean-Jacques Lévy's current research is to review basic algorithms and make their formal proofs of correctness in Why3 + Coq. Filliâtre and Pottier already started this research, but we plan to focus on graph algorithms, with concerns on the feasability of these formal proofs and on the design of good libraries on top of Coq or Ssreflect. The goal is not to disprove these algorithms which are most probably correct, but to develop a theory of tools for proving algorithms with proof assistants and provers. Standard techniques use assertions in Hoare logic or TLA or any other logic, which are written on paper. With the recent development of good computer proof-assistants and the fantastic progress of SMT provers, the goal of providing algorithms with their correctness proofs checked by computer seems possible. The plan of this research is to use Why3, Coq, Ssreflect on standard computing systems, and also to motivate a few students to work on this project. The challenge would be to compete with Filliâtre, Pottier and Monate's group at CEA (France), or Fournet, Swamy and Pierce at Microsoft Research or Univ. of Pennsylvania. We want to demonstrate that the use of SMT provers can be well coupled with the one of interactive provers as already done in Why3 and in F* with refined types in probable future. The expected outcome would be to extend to larger programs and real software. But this seems quite ambitious at present time, since large scale needs more technology as showed by Gonthier for his long proofs of mathematical theorems, and since the world of programming is much less structured than the world of mathematics.

We completed proofs of the following major algorithms as exposed in Sedgewick's book: sorting, searching, depth-first search in graphs. This work is performed in collaboration with Chen Ran at Iscas (Institute of Software, Chinese Academy of Sciences). Proofs can be found at http://jeanjacqueslevy.net/why3 (see also [10]).

4.2. Other software developments

In collaboration with François Pottier (Inria Gallium), Yann Régis-Gianas maintained Menhir, an LR parser generator for OCaml.

Yann Régis-Gianas has been developing the "Hacking Dojo", with the help of Alexandre Ly (master student of Paris Diderot). a web platform to automatically grade programming exercises. The platform is now used in several courses of the University Paris Diderot.

In collaboration with Grégoire Duchêne (master student at Paris Diderot), Yann Régis-Gianas developed Tamasheq, a fully-customisable interpreter for the OCaml programming language. Users of this interpreter can write plugins to instrument the interpretation of an OCaml program with visualisation, interactive debugging or logging. A paper is in preparation.

Yves Guiraud has updated the Catex tool for Latex, whose purpose is to automate the production of string diagrams from algebraic expressions (http://www.pps.univ-paris-diderot.fr/~guiraud/catex/catex.zip).

Yves Guiraud has developed the Python library Cox for the computation of coherent presentations of Artin monoids, after [18] (http://www.pps.univ-paris-diderot.fr/~guiraud/cox/cox.zip).

Yves Guiraud collaborates with Samuel Mimram (LIX) to develop the prototype Rewr that implements the homotopical completion-reduction procedure of [6] (http://www.pps.univ-paris-diderot.fr/~smimram/rewr).

Eric Finster has developed a new proof assistant, called Orchard, which aims to pursue the emerging connections between type theory and higher category theory by providing an environment in which to explicitly manipulate higher categorical diagrams using a notation based on a collection of shapes called opetopes. Opetopes have strong connections to concepts from computer science: they have a natural interpretation as a series of canonical indexed inductive types, and thus can be implemented and reasoned about using standard techniques from functional programming. The goal of the Orchard project is to forge links between the homotopical ideas of homotopy type theory, and the higher categorical ideas coming from higher-dimensional rewriting theory by providing a common language in which to reason about both. A preliminary implementation is available at https://github.com/ericfinster/orchard.

POLSYS Project-Team

5. New Software and Platforms

5.1. FGb

Participant: Jean-Charles Faugère [contact].

FGb is a powerful software for computing Gröbner bases.It includes the new generation of algorihms for computing Gröbner bases polynomial systems (mainly the F4, F5 and FGLM algorithms). It is implemented in C/C++ (approximately 250000 lines), standalone servers are available on demand. Since 2006, FGb is dynamically linked with Maple software (version 11 and higher) and is part of the official distribution of this software.
See also the web page http://www-polsys.lip6.fr/~jcf/Software/FGb/index.html.

5.2. GBLA

Participants: Jean-Charles Faugère [contact], Brice Boyer.

- ACM: I.1.2 Algebraic algorithms
- Programming language: C/C++

GBLA a new open source C library for linear algebra dedicated to Gröbner bases computations (see http://www-polsys.lip6.fr/~jcf/Software/index.html).

5.3. RAGlib

Participant: Mohab Safey El Din [contact].

RAGLib is a Maple library for solving over the reals polynomial systems and computing sample points in semi-algebraic sets.

5.4. Epsilon

Participant: Dongming Wang [contact].

Epsilon is a library of functions implemented in Maple and Java for polynomial elimination and decomposition with (geometric) applications.

5.5. SLV

Participant: Elias Tsigaridas [contact].

SLV is a software package in C that provides routines for isolating (and subsequently refine) the real roots of univariate polynomials with integer or rational coefficients based on subdivision algorithms and on the continued fraction expansion of real numbers. Special attention is given so that the package can handle polynomials that have degree several thousands and size of coefficients hundrends of Megabytes. Currently the code consists of $\sim 5\,000$ lines.

- ACM: I.1.2 Algebraic algorithms
- Programming language: C/C++

POSTALE Team

4. New Software and Platforms

4.1. New Software

4.1.1. MyNRC: image-oriented library for allocation and manipulation of SIMD 1D, 2D and 3D structures

Participant: Lionel Lacassagne.

MyNRC is multi-plateform library that can handle SSE, AVX, Neon and ST VECx registers.

4.1.2. CovTrack: agile realtime multi-target tracking algorithm

Participants: Michèle Gouiffès, Lionel Lacassagne, Florence Laguzet, Andrés Romero.

4.1.3. tmLQCD for Blue Gene/Q

Participant: Michael Kruse [correspondant].

tmLQCD is a program suite for lattice quantum chromodynamics (Lattice QCD) using the chirally twisted Wilson quarks to reduce discretization artifacts. This software is in productive use by the European Twisted Mass Collaboration (ETMC).

As to not waste precious computation time on the supercomputers it is running on, it is important to optimize the code in order to run as fast as possible. tmLQCD has already been customized for Intel Xeon processors, the Blue Gene/L and Blue Gene/P from IBM. For the latter's successor, the Blue Gene/Q, more profound changes to the program are necessary. With these changes, tmLQCD reaches a peak performance of up to 55% of the machines theoretical floating point performance.

The Blue Gene/Q optimized tmLQCD is available at: http://github.com/Meinersbur/tmLQCD

4.1.4. Molly

Participant: Michael Kruse [correspondant].

Using Polly extension, the LLVM compiler framework is able to automatically parallelize general programs for shared memory threading for by exploiting the powerful analysis and transformations of the polyhedral model.

Molly adds the ability to manage distributed memory using the polyhedral model and is therefore able to automatically parallelize even for the largest of today's supercomputer. Once the distribution of data between the computer's nodes is known, Molly determines the values that are required to be transferred between the nodes and chunks them into as few messages as possible. It also keeps tracks of the buffers required by the MPI interface. Transfers are asynchronous such that further computations take place while the data is being transferred.

Molly has not yet been officially released.

4.1.5. Dohko (http://dohko.io/)

Participant: Alessandro Ferreira Leite [correspondant].

Automating multi-cloud configuration is a difficult task. The difficulties are mostly due to clouds' heterogeneity and the lack of tools to coordinate cloud computing configurations automatically. As a result, virtual machine image (VMI) is the common approach to configure cloud environment. Although VMI can handle functional properties like minimum disk size, operating system, and software packages, it leads to a high number of configuration options, increasing the difficulty to select one that matches users' requirements. Moreover, the usage of VMI usually results in vendor lock-in. Furthermore, VMI leaves for the users the work of selecting a resource to deploy the image and for orchestrating them accordingly, i.e., the work of selecting and instantiating the VMI in each cloud. In addition, VMI migration across multiple clouds normally has a high cost due to network traffic. Finally, in case of cloud's failure, it may be difficult for users to re-create the failed environment in another cloud, since the image will be inaccessible.

Therefore, to overcome these issues, we developed a configuration management tool for cloud computing. Our tool, called Dohko, allows the users to configure a multi-cloud computing environment, following a declarative strategy. In this case, the users describe their applications and requirements and use our tool to select the resources and to set up the whole computing environment automatically, taking into account temporal and functional dependencies between the resources. Moreover, following a software product line (SPL) engineering method, Dohko captures the knowledge of configuring cloud environments in form of reusable assets. In this case, a product is a cloud computing environment that meets the user requirements, where the requirements can be either based on high or low-level descriptions. Examples of low-level descriptions include: virtualization type, disk technology, sustainable performance, among others, whereas high-level descriptions include the number of CPU cores, the RAM memory size, and the maximum monetary cost per hour. In this context, a cloud computing environment also matches cloud's configuration constraints. Besides that, thanks to the usage of an extended feature model (EFM) with attributes, our approach enables the description of the whole computing environment (i.e., hardware and software) independent of cloud provider and without requiring the usage of virtual machine image. In this case, it relies on an off-the-shelf constraint satisfaction problem (CSP) solver to implement the feature model and to select the resources.

By employing a declarative strategy, Dohko could execute a biological sequence comparison application in two distinct cloud providers (i.e., Amazon EC2 and Google Compute Engine) considering a single and a multi-cloud scenario, demanding minimal users' intervention to instantiate the whole cloud environment, as well as to execute the application. In particular, our solution tackles the lack of middleware prototypes that can support different scenarios when using services from many clouds. Moreover, it meets the functional requirements identified for multiple cloud-unaware systems [136] such as: (a) it provides a way to describe functional and non-functional requirements through the usage of an SPL engineering method; (b) it can aggregate services from distinct clouds; (c) it provides a homogeneous interface to access services of multiple clouds; (d) it allows the service selection of the clouds; (e) it can deploy its components across many clouds; (f) it provides automatic procedures for deployments; (g) it utilizes an overlay network to connect and to organize the resources; (h) it does not impose any constraint for the connected clouds.

4.2. Platforms

4.2.1. Fast linear system solvers in public domain libraries (http://icl.cs.utk.edu/magma/)

Participant: Marc Baboulin [correspondant].

Hybrid multicore+GPU architectures are becoming commonly used systems in high performance computing simulations. In this research, we develop linear algebra solvers where we split the computation over multicore and graphics processors, and use particular techniques to reduce the amount of pivoting and communication between the hybrid components. This results in efficient algorithms that take advantage of each computational unit [12]. Our research in randomized algorithms yields to several contributions to propose public domain libraries PLASMA and MAGMA in the area of fast linear system solvers for general and symmetric indefinite systems. These solvers minimize communication by removing the overhead due to pivoting in LU and $LDLT$ factorization. Different approaches to reduce communication are compared in [2].

See also the web page http://icl.cs.utk.edu/magma/.

4.2.2. cTuning Framework (http://cTuning.org): Repository and Tools for Collective Characterization and Optimization of Computing Systems

Participant: Grigori Fursin [correspondant].

Designing, porting and optimizing applications for rapidly evolving computing systems is often complex, ad-hoc, repetitive, costly and error prone process due to an enormous number of available design and optimization choices combined with the complex interactions between all components. We attempt to solve this fundamental problem based on collective participation of users combined with empirical tuning and machine learning.

We developed cTuning framework that allows to continuously collect various knowledge about application characterization and optimization in the public repository at cTuning.org. With continuously increasing and systematized knowledge about behavior of computer systems, users should be able to obtain scientifically motivated advices about anomalies in the behavior of their applications and possible solutions to effectively balance performance and power consumption or other important characteristics.

Currently, we use cTuning repository to analyze and learn profitable optimizations for various programs, datasets and architectures using machine learning enabled compiler (MILEPOST GCC). Using collected knowledge, we can quickly suggest better optimizations for a previously unseen programs based on their semantic or dynamic features [10].

We believe that such approach will be vital for developing efficient Exascale computing systems. We are currently developing the new extensible cTuning2 framework for automatic performance and power tuning of HPC applications.

For more information, see the web page http://cTuning.org.

4.2.3. NT2 (http://www.github.com/MetaScale/nt2)

Participants: Pierre Esterie, Joël Falcou, Mathias Gaunard, Ian Masliah, Antoine Tran Tan.

NT2 is a C++ high level framework for scientific computing.[18]

4.2.4. Boost.SIMD (http://www.github.com/MetaScale/nt2)

Participants: Pierre Esterie, Joël Falcou, Mathias Gaunard.

Boost.SIMD provides a portable way to vectorize computation on Altivec, SSE or AVX while providing a generic way to extend the set of supported functions and hardwares.

PRIVATICS Project-Team

4. New Software and Platforms

4.1. Mobilitics

Mobilitics is a joint project, started in 2012 between Inria and CNIL, which targets privacy issues on smartphones. The goal is to analyze the behavior of smartphones applications and their operating system regarding users private data, that is, the time they are accessed or sent to third party companies usually neither with user's awareness nor consent.

In the presence of a wide range of different smartphones available in terms of operating systems and hardware architecture, Mobilitics project focuses actually its study on the two mostly used mobile platforms, IOS (Iphone) and Android. Both versions of the Mobilitics software: (1) capture any access to private data, any modification (e.g., ciphering or hashing of private data), or transmission of data to remote locations on the Internet; (2) store these events in a local database on the phone for offline analysis; and (3) provide the ability to perform an in depth database analysis in order to identify personnal information leakage.

A Mobilitics prototype for iOS has been developed since early 2012. A Mobilitics prototype for Android has been developped since mid-2013, running on Galaxy Nexus smartphones. In parallel an analysis tool has been developped, capable of analyzing the databases containing the raw data of both Mobile Operating Systems.

A first live experiment has been conducted by CNIL with the Mobilitics sofwtare for IOS with the help of volunteers equipped with iphones in September 2012-January 2013. As a result, some visualization tools have been developed for the data collected in order to showcase private data leakage by the apps which the participants of the experiment have used. A press conference has been held by CNIL and Inria in Paris in April 2013 and several Mobilitics results have been published in French newspapers (see Section 8.3).

A second live experiment has been conducted by CNIL with the Mobilitics software for Android, with the help of volunteers equipped with Galaxy Nexus smartphones, in June-September 2014. A press conference has been held by CNIL and Inria in December 2014, and several results have been published in French newspapers (see Section 8.3).

4.2. Omen+

Omen+ is a password cracker following our previous work. It is used to guess possible passwords based on specific information about the target. It can also be used to check the strength of user password by effectively looking at the similarity of that password with both usual structures and information relative to the user, such as his name, birth date...

It is based on a Markov analysis of known passwords to build guesses. The previous work Omen needs to be cleaned in order to be scaled to real problems and to be distributed or transfered to the security community (maintainability): eventually it will become an open source software. The main challenge of Omen+ is to optimize the memory consumption.

The actual efficiency of that implementation in the cracking of passwords will be tested in the coming days. The processing of the personal information will be implemented before the end of January. The hardest part of that side of Omen+ will be the collection and classification of the information for a particular target.

4.3. OpenFEC

OpenFEC (http://openfec.org) is an open-source C-language implementation of several Application-Level Forward Erasure Correction (AL-FEC) codecs, namely: 2D-parity, Reed-Solomon (RFC 5510, http://tools.ietf.org/html/rfc5510) and LDPC-Staircase (RFC 5170, http://tools.ietf.org/html/rfc5170) codes. The OpenFEC project also provides a complete performance evaluation tool-set, capable of automatically assessing the performance of various codecs, both in terms of erasure recovery and encoding/decoding speed or memory consumption.

A commercial, highly optimized version of OpenFEC is available, along with an implementation of the FLUTE (RFC 6726, http://tools.ietf.org/html/rfc6726) large scale content delivery protocol, and both softwares are currently commercialized by the Expway (http://expway.com) French SME. These softwares have been deployed in many places throughout the world (for instance there were more than 1.5 millions of terminals in Japan implementing the ISDB-Tmm standard, powered by our FLUTE/LDPC-Staircase softwares, in Q3-2013).

Thanks to the success of the industrial transfer of the OpenFEC and FLUTE softwares to Expway, Vincent Roca has been awarded the third FIEEC (Federation des Industries Electriques, Electroniques et Communications) applied research prize in October 2014.

PROSECCO Project-Team

5. New Software and Platforms

5.1. ProVerif

Participants: Bruno Blanchet [correspondant], Xavier Allamigeon [April–July 2004], Vincent Cheval [Sept. 2011–], Benjamin Smyth [Sept. 2009–Feb. 2010].

PROVERIF (proverif.inria.fr) is an automatic security protocol verifier in the symbolic model (so called Dolev-Yao model). In this model, cryptographic primitives are considered as black boxes. This protocol verifier is based on an abstract representation of the protocol by Horn clauses. Its main features are:

- It can handle many different cryptographic primitives, specified as rewrite rules or as equations.
- It can handle an unbounded number of sessions of the protocol (even in parallel) and an unbounded message space.

The PROVERIF verifier can prove the following properties:

- secrecy (the adversary cannot obtain the secret);
- authentication and more generally correspondence properties, of the form "if an event has been executed, then other events have been executed as well";
- strong secrecy (the adversary does not see the difference when the value of the secret changes);
- equivalences between processes that differ only by terms.

PROVERIF is widely used by the research community on the verification of security protocols (see http://proverif.inria.fr/proverif-users.html for references).

PROVERIF is freely available on the web, at proverif.inria.fr, under the GPL license.

5.2. CryptoVerif

Participants: Bruno Blanchet [correspondant], David Cadé [Sept. 2009–].

CRYPTOVERIF(cryptoverif.inria.fr) is an automatic protocol prover sound in the computational model. In this model, messages are bitstrings and the adversary is a polynomial-time probabilistic Turing machine. CRYPTOVERIF can prove secrecy and correspondences, which include in particular authentication. It provides a generic mechanism for specifying the security assumptions on cryptographic primitives, which can handle in particular symmetric encryption, message authentication codes, public-key encryption, signatures, hash functions, and Diffie-Hellman key agreements.

The generated proofs are proofs by sequences of games, as used by cryptographers. These proofs are valid for a number of sessions polynomial in the security parameter, in the presence of an active adversary. CRYPTOVERIF can also evaluate the probability of success of an attack against the protocol as a function of the probability of breaking each cryptographic primitive and of the number of sessions (exact security).

CRYPTOVERIF has been used in particular for a study of Kerberos in the computational model, and as a back-end for verifying implementations of protocols in F# and C.

CRYPTOVERIF is freely available on the web, at cryptoverif.inria.fr, under the CeCILL license.

5.3. Cryptosense Analyzer

Participants: Graham Steel [correspondant], Romain Bardou.

See also the web page http://cryptosense.com.

Cryptosense Analyzer (formerly known as Tookan) is a security analysis tool for cryptographic devices such as smartcards, security tokens and Hardware Security Modules that support the most widely-used industry standard interface, RSA PKCS#11. Each device implements PKCS#11 in a slightly different way since the standard is quite open, but finding a subset of the standard that results in a secure device, i.e. one where cryptographic keys cannot be revealed in clear, is actually rather tricky. Cryptosense Analyzer analyses a device by first reverse engineering the exact implementation of PKCS#11 in use, then building a logical model of this implementation for a model checker, calling a model checker to search for attacks, and in the case where an attack is found, executing it directly on the device. It has been used to find at least a dozen previously unknown flaws in commercially available devices.

In June 2013 we submitted a patent application (13 55374) on the reverse engineering process. We also concluded a license agreement between Inria PROSECCO and the nascent spin-off company Cryptosense to commercialize the tool.

5.4. miTLS

Participants: Karthikeyan Bhargavan [correspondant], Antoine Delignat-Lavaud, Cedric Fournet [Microsoft Research], Markulf Kohlweiss [Microsoft Research], Alfredo Pironti, Pierre-Yves Strub [IMDEA], Santiago Zanella-Béguelin [Microsoft Research], Jean Karim Zinzindohoue.

miTLS is a verified reference implementation of the TLS security protocol in F#, a dialect of OCaml for the .NET platform. It supports SSL version 3.0 and TLS versions 1.0-1.2 and interoperates with mainstream web browsers and servers. miTLS has been verified for functional correctness and cryptographic security using the refinement typechecker F7.

A paper describing the miTLS library was published at IEEE S&P 2013, and two updates to the software were released in 2013. The software and associated research materials are available from http://mitls.rocq.inria.fr.

5.5. WebSpi

Participants: Karthikeyan Bhargavan [correspondant], Chetan Bansal [Microsoft], Antoine Delignat-Lavaud, Sergio Maffeis [Imperial College London].

WebSpi is a library that aims to make it easy to develop models of web security mechanisms and protocols and verify them using ProVerif. It captures common modeling idioms (such as principals and dynamic compromise) and defines a customizable attacker model using a set of flags. It defines an attacker API that is designed to make it easy to extract concrete attacks from ProVerif counterexamples.

WebSpi has been used to analyze social sign-on and social sharing services offered by prominent social networks, such as Facebook, Twitter, and Google, on the basis of new open standards such as the OAuth 2.0 authorization protocol.

WebSpi has also been used to investigate the security of a number of cryptographi web applications, including password managers, cloud storage providers, an e-voting website and a conference management system.

WebSpi is under development and released as an open source library at http://prosecco.inria.fr/webspi/

5.6. Defensive JavaScript

Participants: Antoine Delignat-Lavaud [correspondant], Karthikeyan Bhargavan, Sergio Maffeis [Imperial College London].

Defensive JavaScript (DJS) is a subset of the JavaScript language that guarantees the behaviour of trusted scripts when loaded in an untrusted web page. Code in this subset runs independently of the rest of the JavaScript environment. When propertly wrapped, DJS code can run safely on untrusted pages and keep secrets such as decryption keys. DJS is especially useful to write security APIs that can be loaded in untrusted pages, for instance an OAuth library such as the one used by "Login with Facebook". It is also useful to write secure host-proof web applications, and more generally for cryptography that happens on the browser.

The DJS type checker and various libraries written in DJS are available from http://www.defensivejs.com.

5.7. F*

Participants: Nikhil Swamy [Microsoft Research], Karthikeyan Bhargavan, Antoine Delignat-Lavaud, Cedric Fournet [Microsoft Research], Catalin Hritcu, Chantal Keller, Aseem Rastogi, Pierre-Yves Strub.

F* is a new higher order, effectful programming language (like ML) designed with program verification in mind. Its type system is based on a core that resembles System Fω (hence the name), but is extended with dependent types, refined monadic effects, refinement types, and higher kinds. Together, these features allow expressing precise and compact specifications for programs, including functional correctness properties. The F* type-checker aims to prove that programs meet their specifications using an automated theorem prover (usually Z3) behind the scenes to discharge proof obligations. Programs written in F* can be translated to OCaml, F#, or JavaScript for execution.

A detailed description of F* (circa 2011) appeared in the Journal of Functional Programming [88]. F* has evolved substantially since then. The latest version of F* is written entirely in F*, and bootstraps in OCaml and F#. It is under active development at GitHub: https://github.com/FStarLang and the official webpage is at http://fstar-lang.org.

SECRET Project-Team

5. New Software and Platforms

5.1. New Software

5.1.1. CFS Implementation

Participants: Grégory Landais, Nicolas Sendrier.

https://gforge.inria.fr/projects/cfs-signature/

Reference implementation of parallel CFS (reinforced version of the digital signature scheme CFS [93] due to Matthieu Finiasz [95]). Two variants are proposed, one with a « bit-packing » finite field arithmetic and an evolution with a « bit-slicing » finite-field arithmetic (collaboration with Peter Schwabe). For 80 bits of security the running time for producing one signature with the « bit-packing » variant is slightly above one second. This is high but was still the fastest so far. The evolution with the « bit-slicing » arithmetic produces the same signature in about 100 milliseconds.

5.1.2. Collision Decoding

Participants: Grégory Landais, Nicolas Sendrier.

https://gforge.inria.fr/projects/collision-dec/

Implementation of two variants of information set decoding, Stern-Dumer [97], [94] and MMT [96]. To our knowledge it is the best full-fledged open-source implementation of generic decoding of binary linear codes. It is the best generic attack against code-based cryptography. This software has the best score for breaking existing publicly available challenges (see http://pqcrypto.org/wild-challenges.html).

SPADES Team

5. New Software and Platforms

5.1. Prototypes

5.1.1. Logical Causality

Participant: Gregor Goessler.

We are developing LoCA, a prototype tool written in Scala that implements the analysis of logical causality described in 6.3.3 . LoCA currently supports causality analysis in BIP and networks of timed automata. The core analysis engine is implemented as an abstract class, such that support for other models of computation (MoC) can be added by instantiating the class with the basic operations of the MoC.

5.1.2. Cosyma

Participant: Gregor Goessler.

We have developed COSYMA, a tool for automatic controller synthesis for incrementally stable switched systems based on multi-scale discrete abstractions. The tool accepts a description of a switched system represented by a set of differential equations and the sampling parameters used to define an approximation of the state-space on which discrete abstractions are computed. The tool generates a controller — if it exists — for the system that enforces a given safety or time-bounded reachability specification.

5.1.3. The SIAAM virtual machine

Participant: Jean-Bernard Stefani.

The SIAAM abstract machine is an object-based realization of the Actor model of concurrent computation. Actors can exchange arbitrary object graphs in messages while still enjoying a strong isolation property. It guarantees that each actor can only directly access objects in its own local heap, and that information between actors can only flow via message exchange. The SIAAM machine has been implemented for Java as a modified Jikes virtual machine. The resulting SIAAM software comprises:

- A modified Jikes RVM that implements actors and actor isolation as specified by the SIAAM machine.
- A set of static analyses build using the Soot Java optimization framework for optimizing the execution of the SIAAM/Jikes virtual machine, and for helping programmers diagnose potential performance issues.
- A formal proof using the Coq proof assistant of the SIAAM isolation property.

The SIAAM machine is the subject of Quentin Sabah's PhD thesis [67].

5.1.4. pyCPA_TCA

Participant: Sophie Quinton.

We are developing PYCPA_TCA, a PYCPA plugin for Typical Worst-Case Analysis as described in Section 6.2.2 . PYCPA is an open-source Python implementation of Compositional Performance Analysis developed at TU Braunschweig, which allows in particular response-time analysis. PYCPA_TCA is an extension of this tool that is co-developed by Sophie Quinton and Zain Hammadeh at TU Braunschweig. It allows in particular the computation of weakly-hard guarantees for real-time tasks, i.e. number of deadline misses out of a sequence of executions. So far, PYCPA_TCA is restricted to uniprocessor systems of independent tasks, scheduled according to static priority scheduling.

SPECFUN Project-Team

5. New Software and Platforms

5.1. SSReflect

SSReflect is a language extension of the Coq system and was originally written by G. Gonthier for his formal proof of the Four-Color Theorem [0]. In the team, A. Mahboubi and E. Tassi participate to its development, maintenance, distribution, documentation, and user support. A new version (1.5) was released in March 2014. The proof language now offers fine-grained control on type-classes inference and offers new proof commands to ease forward reasoning. In particular the 'have' tactic now supports new modifiers to ease stating generalized formulas as well as hoisting out deeply nested forward steps.

5.2. The Mathematical Components library

The Mathematical Components library is a set of Coq libraries that cover the mechanization of the proof of the Odd Order Theorem, with large contributions by A. Mahboubi and E. Tassi. After the formal proof was completed in September 2012, stable libraries had been distributed [0] with the SSReflect extension, while remaining parts of the libraries had remained under continued improvements in view of potential reuse. In March 2014, version 1.5 of library was released. With it, the library includes 16 more theory files, covering in particular field and Galois theory, advanced character theory, and a construction of algebraic numbers.

5.3. Coq

The way Coq processes theory files has been improved. When used as a batch compiler, Coq is now able to decouple the checking of statements and definitions from the checking of proofs. All proofs can be checked independently taking advantage of modern parallel hardware. When used interactively in conjunction with PIDE-based interfaces, Coq is now able to process the document asynchronously by delegating most of the task to external workers.

The Coq build process was also improved to better support the Windows platform and to enable third parties to provide pre-compiled plugins for such platform.

5.4. Coq/jEdit

Building on top of the asynchronous processing of Coq proofs, we have implemented a plugin that connects the jEdit generic text editor to Coq. This plugin is an adaptation of a similar plugin, written by M. Wenzel, for the Isabelle proof assistant. The interaction using this plugin is a significant change from existing user interfaces, making full use of Coq's asynchronous processing capabilities to provide richer feedback about the proof a user is editing.

The plugin was released as a beta in November 2014 and is available at http://pages.saclay.inria.fr/carst.tankink/jedit.html.

5.5. Other maintained software

We still actively maintain the following other software, which have not had a new release this year.

[0] http://www.msr-inria.fr/projects/mathematical-components/
[0] http://www.msr-inria.fr/projects/mathematical-components/

5.5.1. DDMF

(2007–): Web site consisting of interactive tables of mathematical formulas on elementary and special functions. The formulas are automatically generated by OCaml and computer-algebra routines. Users can ask for more terms of the expansions, more digits of the numerical values, proofs of some of the formulas, etc. See http://ddmf.msr-inria.inria.fr/1.9.1/ddmf. We count hundreds of user sessions per month. Source code distributed under CeCILL-B. A next release is under preparation: it will base on a different, more user-friendly rendering tool (MathJax) and will display more contents.

5.5.2. DynaMoW

(2007–): Programming tool for controlling the generation of mathematical websites that embed dynamical mathematical contents generated by computer-algebra calculations. Implemented in OCaml. See http://ddmf.msr-inria.inria.fr/DynaMoW/. Source code distributed under CeCILL-B.

5.5.3. Ring

(2004–): Coq normalization tool and decision procedure for expressions in commutative ring theories. Implemented in Coq and OCaml. Integrated in the standard distribution of the Coq proof assistant since 2005.

5.5.4. Mgfun

(1994–): Maple package for symbolic summation, integration, and other closure properties of multivariate special functions. Now distributed as part of Algolib, a collection of packages for combinatorics and manipulations of special functions, available at http://algo.inria.fr/libraries/. This software has been used this year for our formal proof of irrationality of $\zeta(3)$.

SUMO Project-Team

5. New Software and Platforms

5.1. Sigali

Participants: Hervé Marchand, Nicolas Berthier.

Sigali is a model-checking tool that operates on ILTS (Implicit Labeled Transition Systems, an equational representation of an automaton), an intermediate model for discrete event systems. It offers functionalities for verification of reactive systems and discrete controller synthesis. It is developed jointly by the TEA and SUMO teams. The techniques used consist in manipulating the system of equations instead of the set of solutions, which avoids the enumeration of the state space. Each set of states is uniquely characterized by a predicate and the operations on sets can be equivalently performed on the associated predicates. Therefore, a wide spectrum of properties, such as liveness, invariance, reachability and attractivity, can be checked. Algorithms for the computation of predicates on states are also available. Sigali is connected with the Polychrony environment (Tea project-team) as well as the Matou environment (VERIMAG), thus allowing the modeling of reactive systems by means of Signal Specification or Mode Automata and the visualization of the synthesized controller by an interactive simulation of the controlled system. Sigali is registered at APP under the identification number IDDN.FR.001.370006.S.P.1999.000.10600.

Sigali is also integrated as part of the compiler of the language BZR (web site).

We are currently developing a new version of Sigali that will be able to handle numerical variables.

5.2. Tipex

Participants: Thierry Jéron, Hervé Marchand, Srinivas Pinisetty.

We are implementing a prototype tool named Tipex (TImed Properties Enforcement during eXecution) for the enforcement of timed properties, in collaboration with Ylies Falcone (LIG, Grenoble). Tipex is based on the theory and algorithms that we develop for the synthesis of enforcement monitors for properties specified by timed automata (TA). The prototype is developped in python, and uses the PyUPPAAL and DBMpyuppaal libraries of the UPPAAL tool. It is currently restricted to safety and co-safety timed property. The property provided as input to the tool is a TA that can be specified using the UPPAAL tool, and is stored in XML format. The tool synthesizes an enforcement monitor from this TA, which can then be used to enforce a sequence of timed events to satisfy the property. Experiments have been conducted on a set of case studies. This allowed to validate the architecture and feasibility of enforcement monitoring in a timed setting and to have a first assessment of performance (and to what extent the overhead induced by monitoring is negligible).

5.3. DAXML

Participant: Loïc Hélouët.

DAXML is an implementation of Distributed Active Documents, a formalism for data centric design of Web Services proposed by Serge Abiteboul. This implementation is based on a REST framework, and can run on a network of machines connected to internet and equipped with JAVA. This implementation was realized during the post doc of Benoit Masson in 2011. A demo of the software is available at this web page. This year, the source code of DAXML has been submitted at the APP, and a distribution with free ad-hoc licence will follow in 2015.

TASC Project-Team

5. New Software and Platforms

5.1. Platforms

5.1.1. CHOCO

Participants: Nicolas Beldiceanu, Jean-Guillaume Fages, Xavier Lorca [correspondant], Thierry Petit, Charles Prud'Homme [main developer], Rémi Douence.

CHOCO is a Java discrete constraints library integrating within a same system *explanations*, *soft constraints* and *global constraints* (90000 lines of source code). In 2014 developments were focusing on the following aspects:

- For second consecutive year, CHOCO has participated at the MiniZinc Challenge, an annual competition of constraint programming solvers. In competition with 16 other solvers, CHOCO has won three bronze medals in three out of four categories (Free search, Parallel search and Open class).
- Five versions have been released all year long, the last one (v3.3.0, Dec. 17th) has the particularity to be promoted on Maven Central Repository. The major modifications were related to a simplification of the API but also improvement of the overall solver.
- A User Guide is now available: 164 pages describing how to use CHOCO, together with a new website.
- Finally, Charles Prud'homme and Jean-Guillaume Fages, the main contributors of CHOCO, have defended their Phd, publishing at the same time their work in the source code. In particular, an extension of CHOCO now provides support for constraints involving graph variables.

5.1.2. IBEX

Participants: Ignacio Araya, Clément Carbonnel, Gilles Chabert [correspondant], Benoit Desrochers, Luc Jaulin, Bertrand Neveu, Jordan Ninin, Ignacio Salas Donoso, Gilles Trombettoni.

IBEX (Interval-Based EXplorer) is a C++ library for solving nonlinear constraints over real numbers. The main feature of Ibex is its ability to build solver/paver strategies declaratively through the contractor programming paradigm. It also comes with a black-box solver and a global optimizer.

In 2014 the work on IBEX has focused on the following points.

- Global optimizer:
  - Rigorous mode in the global optimizer (certification of the feasibility of strict equality constraints for the minimum found). This includes Newton-based inflation iteration, Hansen test for underconstrained systems (see Global Optimization using Interval Analysis, E. Hansen, 1992).
  - Unconstrained local search algorithm (quasi-Newton method with trust regions).
  - Rejection test based on first-order conditions (see First Order Rejection Tests For Multiple-Objective Optimization, A. Goldsztejn et al. [42]).
  - Multiple selection technique in exploration (see A new multisection technique in interval methods for global optimization, L.G. Casado, Computing, 2000)
- Contractors:
  - Existentially-quantified constraints, (see Contractor Programming, [8]).
  - Mohc contractor, (see Exploiting Monotonicity in Interval Constraint Propagation, I. Araya et al., [41]).

  - Q-intersection, (see Q-intersection Algorithms for Constraint-Based Robust Parameter Estimation, C. Carbonnel et al., AAAI 2014, [27]).
  - Contractor based on pixel maps (started in Oct 2014, still in progress, see Using set membership methods for robust underwater robot localization, PhD, J. Sliwka).
- Miscellanous
  - Everyday code improvement (around 400 commits in 2014).
  - Symbolic processing features (symbol occurrence splitting, function construction from strings, progress in differentiation with vector/matrix operations).
  - numerous bug fixes (especially in the inner arithmetic routines).

5.1.3. Global Constraint Catalog

Participants: Nicolas Beldiceanu [correspondant], Mats Carlsson, Sophie Demassey, Helmut Simonis.

The global constraint catalog presents and classifies global constraints and describes different aspects with meta data. It consist of

1. a pdf version that can be downloaded from http://sofdem.github.io/gccat/ (at item *working version*) containing 431 constraints, 4070 pages and 1000 figures,
2. an on line version accessible from the previous address,
3. meta data describing the constraints (buton *PL* for each constraint, e.g., alldifferent.pl),
4. an online service (i.e, a *constraint seeker*) which provides a web interface to search for global constraints, given positive and negative ground examples.

This year developments were focusing on:

1. maintaining the content of the catalogue,
2. making more easy the navigation within the pdf version,
3. continuing the redesign of the figures using TikZ: 200 figures were converted and 100 figures remain to be converted, and adding new illustrations (150 figures).
4. updating the web version of the catalogue (see http://sofdem.github.io/gccat/).

5.1.4. AIUR

Participant: Florian Richoux [correspondant].

AIUR (Artificial Intelligence Using Randomness) is an AI for $StarCraft: BroodWar^{tm}$.

The main idea is to be unpredictable by making some stochastic choices. The AI starts a game with a "mood" randomly picked up among 5 moods, dictating some behaviors (aggressive, fast expand, macro-game, ...). In addition, some other choices (productions, timing attacks, early aggressions, ...) are also taken under random conditions.

Learning is an essential part of AIUR. For this, it uses persistent I/O files system to record which moods are efficient against a given opponent, in order to modify the probability distribution for the mood selection. The current system allows both on-line and off-line learning.

AIUR is an open source program under GNU GPL v3 licence, written in C++ (18.000 lines of code). Source and documentations are available at github.com/AIUR-group/AIUR. AIUR finished 4$^{\text{th}}$ to $StarCraft^{tm}$ AI competitions organized at the conferences AIIDE 2014 and CIG 2014.

5.1.5. GHOST

Participant: Florian Richoux [correspondant].

GHOST (General meta-Heuristic Optimization Solving Tool) is a template C++11 library designed for $StarCraft: BroodWar^{tm}$, under the terms of the GNU GPL v3 licence and is about 7500 lines long. GHOST implements a meta-heuristic solver aiming to solve any kind of combinatorial and optimization RTS-related problems represented by a CSP/COP [36]. The solver handles dedicated geometric and assignment constraints in a way that is compatible with very strong real time requirements. The source code as well as documentation pages are available at github.com/richoux/GHOST.

This framework is a deep extension of an ad-hoc solver. Although GHOST has been developed recently (during Summer 2014), it got itself quickly noticed by a French video-game developing company. We are starting discussion about a technology transfer of GHOST.

TEA Project-Team

5. New Software and Platforms

5.1. The Eclipse project POP

Participants: Loïc Besnard, Thierry Gautier, Paul Le Guernic, Jean-Pierre Talpin.

The distribution of project POP [0] is a major achievement of the ESPRESSO project. The Eclipse project POP is a model-driven engineering front-end to our open-source toolset Polychrony. It was finalised in the frame of project OPEES, as a case study: by passing the POLARSYS qualification kit as a computer aided simulation and verification tool. This qualification was implemented by CS Toulouse in conformance with relevant generic (platform independent) qualification documents. Polychrony is now distributed by the Eclipse project POP on the platform of the POLARSYS industrial working group. Team TEA aims at continuing its dissemination to academic partners, as to its principles and features, and industrial partners, as to the services it can offer.

Technically, project POP is composed of the Polychrony toolset, under GPL license, and its Eclipse framework, under EPL license.

The Polychrony toolset. The Polychrony toolset is an Open Source development environment for critical/embedded systems. It is based on Signal, a real-time polychronous dataflow language. It provides a unified model-driven environment to perform design exploration by using top-down and bottom-up design methodologies formally supported by design model transformations from specification to implementation and from synchrony to asynchrony. It can be included in heterogeneous design systems with various input formalisms and output languages.

The Polychrony toolset provides a formal framework:

- to validate a design at different levels, by the way of formal verification and/or simulation,
- to refine descriptions in a top-down approach,
- to abstract properties needed for black-box composition,
- to assemble heterogeneous predefined components (bottom-up with COTS),
- to generate executable code for various architectures.

The Polychrony toolset contains three main components and an experimental interface to GNU Compiler Collection (GCC):

- The Signal toolbox, a batch compiler for the Signal language, and a structured API that provides a set of program transformations. The Signal toolbox can be installed without other components. The Signal toolbox is distributed under GPL V2 license.
- The Signal GUI, a Graphical User Interface to the Signal toolbox (editor + interactive access to compiling functionalities). The Signal GUI is distributed under GPL V2 license.
- The SME/SSME platform, a front-end to the Signal toolbox in the Eclipse environment. The SME/SSME platform is distributed under EPL license.
- GCCst, a back-end to GCC that generates Signal programs (not yet available for download).

In 2013, to be able to use the Signal GUI both as a specific tool and as a graphical view under Eclipse, the code of the Signal GUI has been restructured in three parts: a common part used by both tools (28 classes), a specific part for the Signal GUI (2 classes), a specific part for Eclipse (2 classes). Such a structuration facilitates the maintenance of the products.

[0] *Polychrony on POLARSYS (POP)*, an Eclipse project in the POLARSYS Industry Working Group, 2013. https://www.POLARSYS.org/projects/POLARSYS.pop

The Polychrony toolset also provides:

- libraries of Signal programs,
- a set of Signal program examples,
- user oriented and implementation documentations,
- facilities to generate new versions.

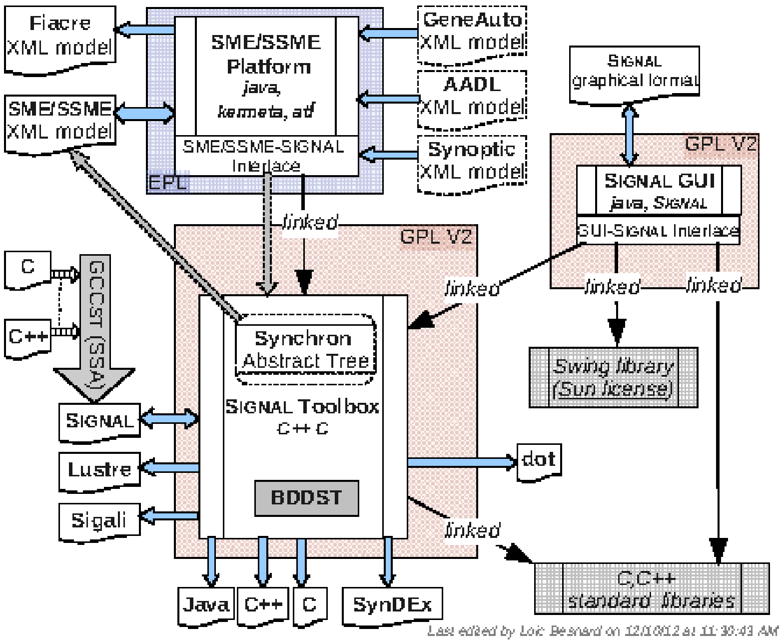


Figure 1. The Polychrony toolset high-level architecture

Dassault Systèmes, supplies a commercial implementation of Polychrony, called RT-Builder, used for industrial scale projects.

As part of its open-source release, the Polychrony toolset not only comprises source code libraries but also an important corpus of structured documentation, whose aim is not only to document each functionality and service, but also to help a potential developer to package a subset of these functionalities and services, and adapt them to developing a new application-specific tool: a new language front-end, a new back-end compiler. This multi-scale, multi-purpose documentation aims to provide different views of the software, from a high-level structural view to low-level descriptions of basic modules. It supports a distribution of the software "by apartment" (a functionality or a set of functionalities) intended for developers who would only be interested by part of the services of the toolset.

The Eclipse POP Framework. We have developed a meta-model and interactive editor of Polychrony in Eclipse. Signal-Meta is the meta-model of the Signal language implemented with Eclipse/Ecore. It describes all syntactic elements specified in [0]: all Signal operators (e.g. arithmetic, clock synchronization), model (e.g. process frame, module), and construction (e.g. iteration, type declaration).

[0] *SIGNAL V4-Inria version: Reference Manual*. Besnard, L., Gautier, T. and Le Guernic, P. http://www.irisa.fr/espresso/Polychrony, 2009

The meta-model primarily aims at making the language and services of the Polychrony environment available to inter-operation and composition with other components (e.g. AADL, Simulink, GeneAuto) within an Eclipse-based development toolchain. Polychrony now comprises the capability to directly import and export Ecore models instead of textual Signal programs, in order to facilitate interaction between components within such a toolchain.

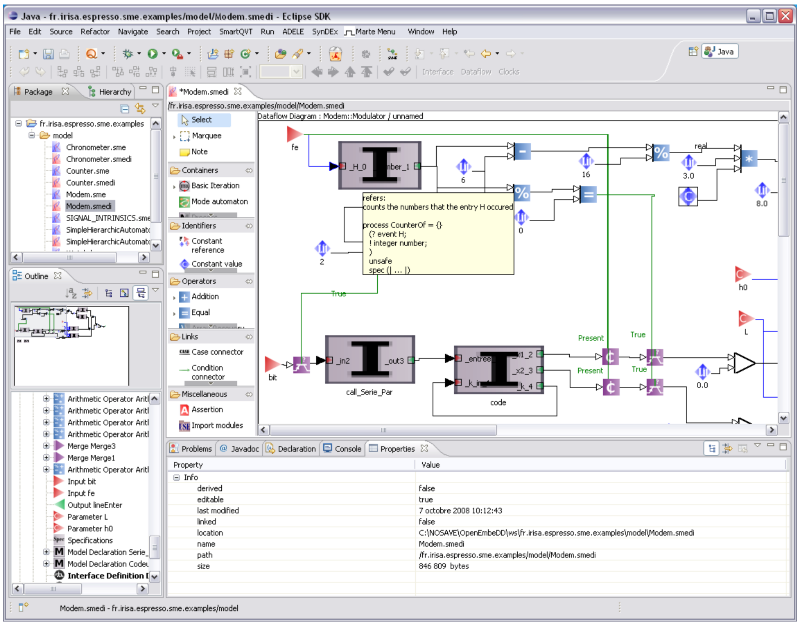


Figure 2. The Eclipse POP Environment

It also provides a graphical modelling framework allowing to design applications using a component-based approach. Application architectures can be easily described by just selecting components via drag and drop, creating some connections between them and specifying their parameters as component attributes. Using the modelling facilities provided with the Topcased framework, we have created a graphical environment for Polychrony called SME (Signal-Meta under Eclipse). To highlight the different parts of the modelling in Signal, we split the modelling of a Signal process in three diagrams: one to model the interface of the process, one to model the computation (or dataflow) part, and one to model all explicit clock relations and dependences. The SME environment is available through the ESPRESSO update site [0]. A new meta-model of Signal, called SSME (Syntactic Signal-Meta under Eclipse), closer to the Signal abstract syntax, has been defined and integrated in the Polychrony toolset.

It should be noted that the Eclipse Foundation does not host code under GPL license. So, the Signal toolbox useful to compile Signal code from Eclipse is hosted on our web server. For this reason, the building of the Signal toolbox, previously managed under Eclipse, has now been exported. The interface of the Signal toolbox for Eclipse is now managed using the CMake tool like the Signal toolbox and the Signal GUI.

5.2. Integrated Modular Avionics design using Polychrony

Participants: Loïc Besnard, Thierry Gautier, Paul Le Guernic, Jean-Pierre Talpin.

[0] *Polychrony Update Site for Eclipse plug-ins*. http://www.irisa.fr/espresso/Polychrony/update, 2009.

The Apex interface, defined in the ARINC standard [0], provides an avionics application software with the set of basic services to access the operating-system and other system-specific resources. Its definition relies on the Integrated Modular Avionics approach (IMA). A main feature in an IMA architecture is that several avionics applications (possibly with different critical levels) can be hosted on a single, shared computer system. Of course, a critical issue is to ensure safe allocation of shared computer resources in order to prevent fault propagations from one hosted application to another. This is addressed through a functional partitioning of the applications with respect to available time and memory resources. The allocation unit that results from this decomposition is the *partition*.

A partition is composed of *processes* which represent the executive units (an ARINC partition/process is akin to a Unix process/task). When a partition is activated, its owned processes run concurrently to perform the functions associated with the partition. The process scheduling policy is priority preemptive. Each partition is allocated to a processor for a fixed time window within a major time frame maintained by the operating system. Suitable mechanisms and devices are provided for communication and synchronization between processes (e.g. *buffer*, *event*, *semaphore*) and partitions (e.g. *ports* and *channels*). The specification of the ARINC 651-653 services in Signal [4] is now part of the Polychrony distribution and offers a complete implementation of the Apex communication, synchronization, process management and partitioning services. Its Signal implementation consists of a library of generic, parameterizable Signal modules.

5.3. Safety-Critical Java Level 1 Code generation from Dataflow Graph Specifications

Participants: Adnan Bouakaz, Thierry Gautier, Jean-Pierre Talpin.

We have proposed a dataflow design model [2] of SCJ/L1 applications [0] in which handlers (periodic and aperiodic actors) communicate only through lock-free channels. Hence, each mission is modeled as a dataflow graph. The presented dataflow design model comes with a development tool integrated in the Eclipse IDE for easing the development of SCJ/L1 applications and enforcing the restrictions imposed by the design model. It consists of a GMF editor where applications are designed graphically and timing and buffering parameters can be synthesized. Indeed, abstract affine scheduling is first applied on the dataflow subgraph, that consists only of periodic actors, to compute timeless scheduling constraints (e.g. relation between the speeds of two actors) and buffering parameters. Then, symbolic fixed-priority schedulability analysis (i.e., synthesis of timing and scheduling parameters of actors) considers both periodic and aperiodic actors.

Through a model-to-text transformation, using Acceleo, the SCJ code for missions, interfaces of handlers, and the mission sequencer is automatically generated in addition to the annotations needed by the memory checker. Channels are implemented as cyclic arrays or cyclical asynchronous buffers; and a fixed amount of memory is hence reused to store the infinite streams of tokens. The user must provide the SCJ code of all the `handleAsyncEvent()` methods. We have integrated the SCJ memory checker [0] in our tool so that potential dangling pointers can be highlighted at compile-time. To enhance functional determinism, we would like to develop an ownership type system to ensure that actors are strongly isolated and communicate only through buffers.

[0] *ARINC Report 651-1: Design Guidance for Integrated Modular Avionics*. Airlines Electronic Engineering Committee, 1997

[0] *Safety critical Java technology specification*. JSR-302, Year = 2010

[0] *Static checking of safety critical Java annotations*. Tang, D. Plsek, A. and Vitek, J. International Workshop on Java Technologies for Real-Time and Embedded Systems, 2010

TEMPO Team

5. New Software and Platforms

5.1. SimSoC

We have continued to work on the SimSoC virtual prototyping framework distributed by Inria. Because of issues in the design of the Power Architecture simulator, we did a redesign of the Power simulator and a new implementation, so that we can simulate in the future both the Power Classic and Power Extended architectures in both 32 bits or 64 bits. We also contributed new extensions as described below.

TOCCATA Project-Team

5. New Software and Platforms

5.1. The Why3 system

Participants: Jean-Christophe Filliâtre [contact], Claude Marché, Guillaume Melquiond, Andrei Paskevich.

Criteria for Software Self-Assessment: A-3-up, SO-4, SM-4, EM-4, SDL-5, OC-4.[0]

Why3 is the next generation of *Why*. *Why3* clearly separates the purely logical specification part from generation of verification conditions for programs. It features a rich library of proof task transformations that can be chained to produce a suitable input for a large set of theorem provers, including SMT solvers, TPTP provers, as well as interactive proof assistants.

It is distributed as open source, under GPL license, at http://why3.lri.fr/. It is also distributed as part of major Linux distributions and in the OPAM packaging system http://opam.ocaml.org/packages/why3/why3.0.85/.

Why3 is used as back-end of our own tools *Krakatoa* and *Jessie*, but also as back-end of the GNATprove tool (Adacore company), and of the WP plugin of *Frama-C*. *Why3* has been used to develop and prove a significant part of the programs of our team gallery http://proval.lri.fr/gallery/index.en.html, and used for teaching (e.g., at the Master Parisien de Recherche en Informatique).

Why3 is used by other academic research groups, e.g. within the CertiCrypt/EasyCrypt project (http://easycrypt.gforge.inria.fr/) for certifying cryptographic programs. The *Why3* web site http://why3.lri.fr lists a few other works done by external researchers and relying on the use of *Why3*.

Two versions were released in 2014: 0.83 released in March and 0.84 in September, plus a few days later a bugfix version 0.85.

5.2. The Alt-Ergo theorem prover

Participants: Sylvain Conchon [contact], Évelyne Contejean, Alain Mebsout, Mohamed Iguernelala.

Criteria for Software Self-Assessment: A-3-up, SO-4, SM-4-up, EM-4, SDL-5, OC-4.

Alt-Ergo is an automated proof engine, dedicated to program verification, whose development started in 2006. It is fully integrated in the program verification tool chain developed in our team. It solves goals that are directly written ina *Why*'s annotation language; this means that *Alt-Ergo* fully supports first order polymorphic logic with quantifiers. *Alt-Ergo* also supports the standard [116] defined by the SMT-lib initiative.

It is currently used in our team to prove correctness of C and Java programs as part of the *Why* platform and the new *Why3* system. It is used as back-end prover in the environments Frama-C and CAVEAT for static analysis of C developed at CEA. In this context, Alt-Ergo has been qualified by Airbus and is integrated in the next generation of Airbus development process. Alt-Ergo is usable as a back-end prover in the SPARK verifier for ADA programs, since Oct 2010, and is also the main back-end prover of the new SPARK2014.

Alt-Ergo is integrated in several other tools and platforms: the Bware platform for discharging VCs generated by Atelier B, the EasyCrypt environment for verifying cryptographic protocols, the Pangolin programming language http://code.google.com/p/pangolin-programming-language/, etc.

Last but not least, Alt-Ergo is the solver used by the Cubicle model checker described below.

[0]self-evaluation following the guidelines (http://www.inria.fr/content/download/11783/409665/version/4/file/SoftwareCriteria-V2-CE.pdf) of the Software Working Group of Inria Evaluation Committee (http://www.inria.fr/institut/organisation/instances/commission-d-evaluation)

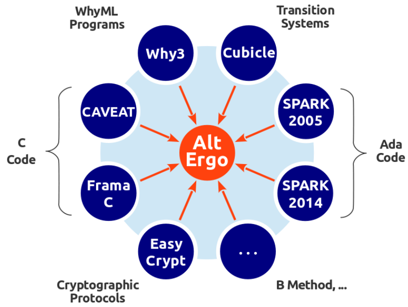


Figure 2.

Alt-Ergo is distributed as open source, under the CeCILL-C license, at URL http://alt-ergo.lri.fr/, and in the OPAM packaging system http://opam.ocaml.org/packages/alt-ergo/alt-ergo.0.95.2/. Latest public version is 0.99.1, released in Dec. 2014. Maintenance is done by the OcamlPro company http://alt-ergo.ocamlpro.com/.

5.3. The Cubicle model checker modulo theories

Participants: Sylvain Conchon [contact], Alain Mebsout.

Partners: A. Goel, S. Krstić (Intel Strategic Cad Labs in Hillsboro, OR, USA), F. Zaïdi (LRI, Université Paris-sud)

Cubicle is an open source model checker for verifying safety properties of array-based systems, which corresponds to a syntactically restricted class of parametrized transition systems with states represented as arrays indexed by an arbitrary number of processes. Cache coherence protocols and mutual exclusion algorithms are typical examples of such systems.

Cubicle model-checks by a symbolic backward-reachability analysis on infinite sets of states represented by specific simple formulas, called cubes. Cubicle is based on ideas introduced by MCMT (http://users.mat.unimi.it/users/ghilardi/mcmt/) from which, in addition to revealing the implementation details, it differs in a more friendly input language and a concurrent architecture. Cubicle is written in OCaml. Its SMT solver is a tightly integrated, lightweight and enhanced version of *Alt-Ergo*; and its parallel implementation relies on the Functory library.

Cubicle is distributed as open source, under the Apache license, at URL http://cubicle.lri.fr/, and in the OPAM packaging system http://opam.ocaml.org/packages/cubicle/cubicle.1.0.1/. Latest version is 1.0.1, released in Nov. 2014.

5.4. The Flocq library

Participants: Sylvie Boldo [contact], Guillaume Melquiond.

Criteria for Software Self-Assessment: A-2, SO-3, SM-3, EM-3, SDL-5, OC-4.

The Flocq library for the *Coq* proof assistant is a comprehensive formalization of floating-point arithmetic: core definitions, axiomatic and computational rounding operations, high-level properties [6]. It provides a framework for developers to formally certify numerical applications.

Flocq is currently used by the CompCert certified compiler for its support of floating-point computations.

It is distributed as open source, under a LGPL license, at http://flocq.gforge.inria.fr/. It was first released in 2010. Current version is 2.4.0 released in Sep. 2014.

5.5. The Gappa tool

Participant: Guillaume Melquiond [contact].

Criteria for Software Self-Assessment: A-3, SO-4, SM-4, EM-3, SDL-5, OC-4.

Given a logical property involving interval enclosures of mathematical expressions, Gappa tries to verify this property and generates a formal proof of its validity. This formal proof can be machine-checked by an independent tool like the *Coq* proof-checker, so as to reach a high level of confidence in the certification [82], [123].

Since these mathematical expressions can contain rounding operators in addition to usual arithmetic operators, Gappa is especially well suited to prove properties that arise when certifying a numerical application, be it floating-point or fixed-point. Gappa makes it easy to compute ranges of variables and bounds on absolute or relative roundoff errors.

Gappa is being used to certify parts of the mathematical libraries of several projects, including CRlibm, FLIP, and CGAL. It is distributed as open source, under a Cecill-B / GPL dual-license, at http://gappa.gforge.inria.fr/. Latest version is 1.1.2 released in October 2014.

Part of the work on this tool was done while in the Arénaire team (Inria Rhône-Alpes), until 2008.

5.6. The Interval package for Coq

Participants: Guillaume Melquiond [contact], Érik Martin-Dorel.

Criteria for Software Self-Assessment: A-3, SO-4, SM-3, EM-3, SDL-4, OC-4.

The Interval package provides several tactics for helping a *Coq* user to prove theorems on enclosures of real-valued expressions. The proofs are performed by an interval kernel which relies on a computable formalization of floating-point arithmetic in *Coq*.

Versions 1.0 and 2.0 were released in 2014. Version 2.0 integrates the CoqApprox library for computing Taylor models, so as to greatly improve performances when bounding univariate expressions [43].

It is distributed as open source, under a CeCILL-C license, at http://coq-interval.gforge.inria.fr/. Latest version is 2.0 released in November 2014.

Part of the work on this library was done while in the Mathematical Components team (Microsoft Research–Inria Joint Research Center).

5.7. The Coquelicot library for real analysis

Participants: Sylvie Boldo [contact], Catherine Lelay, Guillaume Melquiond.

Criteria for Software Self-Assessment: A-3, SO-4, SM-2, EM-3, SDL-4, OC-4.

The Coquelicot library is designed with three principles in mind. The first is the user-friendliness, achieved by implementing methods of automation, but also by avoiding dependent types in order to ease the stating and readability of theorems. This latter part was achieved by defining total function for basic operators, such as limits or integrals. The second principle is the comprehensiveness of the library. By experimenting on several applications, we ensured that the available theorems are enough to cover most cases. We also wanted to be able to extend our library towards more generic settings, such as complex analysis or Euclidean spaces. The third principle is for the Coquelicot library to be a conservative extension of the Coq standard library, so that it can be easily combined with existing developments based on the standard library. Moreover, we achieved this compatibility without adding any additional axiom.

The result is the Coquelicot library available at http://coquelicot.saclay.inria.fr. Latest version is 2.0.1 released in March 2014. It contains about 1,700 theorems and 37,000 lines of Coq.

5.8. The CFML tool for verifying OCaml code

Participant: Arthur Charguéraud [contact].

Criteria for Software Self-Assessment: A-2, SO-4, SM-2, EM-3, SDL-1, OC-4.

The *CFML* tool supports the verification of *OCaml* programs through interactive *Coq* proofs. *CFML* proofs establish the full functional correctness of the code with respect to a specification. They may also be used to formally establish bounds on the asymptotic complexity of the code. The tool is made of two parts: on the one hand, a characteristic formula generator implemented as an *OCaml* program that parses *OCaml* code and produces *Coq* formulae; and, on the other hand, a *Coq* library that provides notation and tactics for manipulating characteristic formulae interactively in *Coq*.

CFML is distributed under the LGPL license, and is available at http://arthur.chargueraud.org/softs/cfml/. It has been continuously extended since its first release in 2010. In particular, in 2014 support for the verification of asymptotic complexity bounds has been added.

5.9. Other Maintained Tools

5.9.1. The ALEA library for randomized algorithms

Participant: Christine Paulin-Mohring [contact].

Criteria for Software Self-Assessment: A-2, SO-3, SM-2, EM-3, SDL-4, OC-4.

The ALEA library is a *Coq* development for modeling randomized functional programs as distributions using a monadic transformation. It contains an axiomatisation of the real interval $[0, 1]$ and its extension to positive real numbers. It introduces definition of distributions and general rules for approximating the probability that a program satisfies a given property.

ALEA is used as a basis of the Certicrypt environment (MSR-Inria joint research center, Imdea Madrid, Inria Sophia-Antipolis) for formal proofs for computational cryptography [55]. It is also experimented in LABRI as a basis to study formal proofs of probabilistic distributed algorithms.

ALEA is distributed as open source, at http://www.lri.fr/~paulin/ALEA. Latest version is 8 released in May 2013. In particular, it includes a module to reason about random variables with values in positive real numbers.

5.9.2. Bibtex2html

Participants: Jean-Christophe Filliâtre [contact], Claude Marché.

Criteria for Software Self-Assessment: A-5, SO-3, SM-3, EM-3, SDL-5, OC-4.

Bibtex2html is a generator of HTML pages of bibliographic references. Distributed as open source since 1997, under the GPL license, at http://www.lri.fr/~filliatr/bibtex2html/. Latest version is 1.98 released in July 2014. Bibtex2html is also distributed as a package in most Linux distributions, and in the OPAM packaging system http://opam.ocaml.org/packages/bibtex2html/bibtex2html.1.98/.

We estimate that between 10000 and 100000 web pages have been generated using Bibtex2html.

5.9.3. The Coccinelle library for term rewriting

Participant: Évelyne Contejean [contact].

Criteria for Software Self-Assessment: A-2, SO-3, SM-2, EM-2, SLD-2, OC-4.

Coccinelle is a *Coq* library for term rewriting. Besides the usual definitions and theorems of term algebras, term rewriting and term ordering, it also models a number of algorithms implemented in the CiME toolbox, such as matching, matching modulo associativity-commutativity, computation of the one-step reducts of a term, recursive path ordering (RPO) comparison between two terms, etc. The RPO algorithm can effectively be run inside *Coq*, and is used in the Color developement (http://color.inria.fr/) as well as for certifying Spike implicit induction theorems in *Coq* (Sorin Stratulat).

Coccinelle is available at http://www.lri.fr/~contejea/Coccinelle, and is distributed under the Cecill-C license.

5.9.4. OCamlgraph

Participants: Jean-Christophe Filliâtre [contact], Sylvain Conchon.

OCamlgraph is a graph library for *OCaml*. It features many graph data structures, together with many graph algorithms. Data structures and algorithms are provided independently of each other, thanks to *OCaml* module system. OCamlgraph is distributed as open source, under the LGPL license, at http://OCamlgraph.lri.fr/. Latest version is 1.8.5, released in March 2014. It is also distributed as a package in several Linux distributions. OCamlgraph is now widely spread among the community of *OCaml* developers, and available as an OPAM package http://opam.ocaml.org/packages/ocamlgraph/ocamlgraph.1.8.5/.

5.9.5. Mlpost

Participant: Jean-Christophe Filliâtre [contact].

Mlpost is a tool to draw scientific figures to be integrated in LaTeX documents. Contrary to other tools such as TikZ or MetaPost, it does not introduce a new programming language; it is instead designed as a library of an existing programming language, namely *OCaml*. Yet it is based on MetaPost internally and thus provides high-quality PostScript figures and powerful features such as intersection points or clipping. Mlpost is distributed as open source, under the LGPL license, at http://mlpost.lri.fr/. Mlpost was presented at JFLA'09 [52].

Mlpost is available as an OPAM package http://opam.ocaml.org/packages/mlpost/mlpost.0.8.1/.

5.9.6. Functory

Participant: Jean-Christophe Filliâtre [contact].

Functory is a distributed computing library for *OCaml*. The main features of this library include (1) a polymorphic API, (2) several implementations to adapt to different deployment scenarios such as sequential, multi-core or network, and (3) a reliable fault-tolerance mechanism. Functory was presented at JFLA 2011 [91] and at TFP 2011 [90].

Functory is distributed as open source, under the LGPL license, at http://functory.lri.fr/, and in the OPAM packaging system http://opam.ocaml.org/packages/functory/functory.0.5/. Latest version is 0.5, release in March 2013.

5.9.7. The Why Environment

Participants: Claude Marché [contact], Jean-Christophe Filliâtre, Guillaume Melquiond, Andrei Paskevich.

Criteria for Software Self-Assessment: A-3, SO-4, SM-3, EM-2, SDL-5-down, OC-4.

The *Why* platform is a set of tools for deductive verification of Java and C source code. In both cases, the requirements are specified as annotations in the source, in a special style of comments. For Java (and Java Card), these specifications are given in JML and are interpreted by the *Krakatoa* tool. Analysis of C code must be done using the external *Frama-C* environment, and its *Jessie* plugin which is distributed in *Why*.

The platform is distributed as open source, under GPL license, at http://why.lri.fr/.

It also distributed as part of major Linux distributions and in the OPAM packaging system http://opam.ocaml.org/packages/why/why.2.34/. Version 2.34 was released in August 2014, to provide a version compatible with both *Frama-C*Neon and *Why3* 0.83.

The internal VC generator and the translators to external provers are no longer under active development, as superseded by the *Why3*system described above. The *Krakatoa* and *Jessie* front-ends are still maintained, although using now by default the *Why3* VC generator. These front-ends are described in a specific web page http://krakatoa.lri.fr/. They are used for teaching (University of Evry, École Polytechnique, etc.), used by several research groups in the world, e.g at Fraunhofer Institute in Berlin [92], at Universidade do Minho in Portugal [50], at Moscow State University, Russia (http://journal.ub.tu-berlin.de/eceasst/article/view/255).

VEGAS Project-Team

4. New Software and Platforms

4.1. QI: Quadrics Intersection

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

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

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

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

4.2. Isotop: Topology and geometry of planar algebraic curves

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

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

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

4.3. CGAL: Computational Geometry Algorithms Library

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

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

4.4. Fast_polynomial: fast polynomial evaluation software

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

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

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

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

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

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

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

VERIDIS Project-Team

5. New Software and Platforms

5.1. The veriT Solver

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

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

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

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

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

5.2. The TLA+ Proof System

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

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

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

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

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

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

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

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

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

5.4. The Redlog Computer Logic System

Participants: Thomas Sturm [contact], Marek Košta.

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

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

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

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

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

APICS Project-Team

5. New Software and Platforms

5.1. RARL2

Participant: Martine Olivi [corresponding participant].

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

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

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

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

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

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

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

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

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

5.2. RGC

Participant: Fabien Seyfert [corresponding participant].

Status: A stable version is maintained.

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

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

5.3. PRESTO-HF

Participant: Fabien Seyfert [corresponding participant].

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

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

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

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

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

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

5.4. Dedale-HF

Participant: Fabien Seyfert [corresponding participant].

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

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

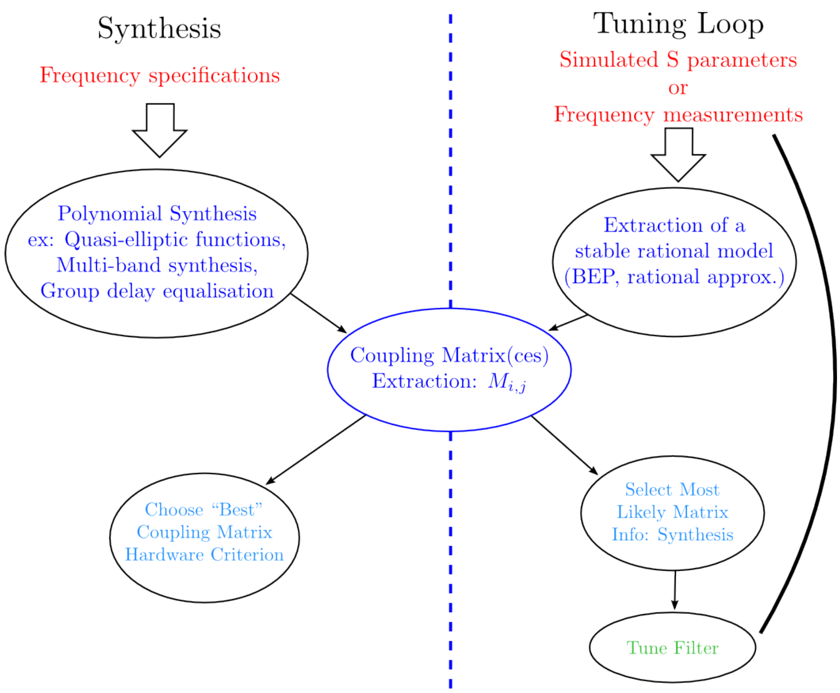


Figure 3. Overall scheme of the design and tuning process of a microwave filter.

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

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

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

5.5. easyFF

Participant: Fabien Seyfert.

Status: A stable version is maintained.

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

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

5.6. FindSources3D

Participant: Juliette Leblond [corresponding participant].

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

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

FindSources3D [0] is a software dedicated to source recovery for the inverse EEG problem, in 3-layer spherical settings, from point-wise data (see http://www-sop.inria.fr/apics/FindSources3D/). Through the algorithm described in [9] and Section 4.2 , it makes use of the software RARL2 (Section 5.1) for the rational approximation step in plane sections.

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

Another release is being prepared, due to strong interest by the German company BESA GmbH [0], which develops EEG software for research and clinical applications. A deeper collaboration with this company started last year. Figure 4 shows good results on a two sources distribution recovered by FindSources3D from values of the potential at electrodes on a sphere (scalp) generated by BESA's simulator. There, the localization error is satisfactory, see [28]. Altogether FindSources3D provides suitable initial guess to heavier dedicated recovery tools, including an estimate of the number of sources see Section 6.1.1 .

5.7. Sollya

Participant: Sylvain Chevillard [corresponding participant].

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

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

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

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

[0] http://www.besa.de/

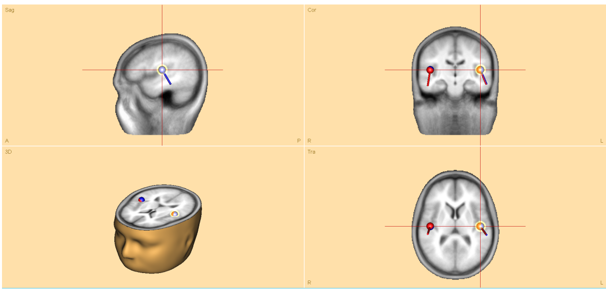


Figure 4. Recovered 2 sources by FindSources3D (courtesy of BESA).

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

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

ASPI Project-Team (section vide)

BACCHUS Team

4. New Software and Platforms

4.1. AeroSol

Participants: Simon Delmas [Cagire], Damien Genet [Bacchus], Maxime Mogé [Cagire], Yann Moguen [Cagire], Vincent Perrier [Corresponding member], Mario Ricchiuto [Bacchus].

The `AeroSol` software is jointly developed in teams Bacchus and Cagire. It is a high order finite element library written in C++. The code has been designed so as to allow for efficient computations, with continuous and discontinuous finite elements methods on hybrid and possibly curvilinear meshes. The work of the team Bacchus is focused on continuous finite elements methods, while the team Cagire is focused on discontinuous Galerkin methods. However, everything is done for sharing the largest part of code we can. More precisely, classes concerning IO, finite elements, quadrature, geometry, time iteration, linear solver, models and interface with `PaMPA` are used by both of the teams. This modularity is achieved by mean of template abstraction for keeping good performances. The distribution of the unknowns is made with the software `PaMPA`, developed within the team Bacchus and the team Castor.

The work performed this year in the `BACCHUS`team has focused on the experimentation of parallelisation solutions on heterogenous machines, and in particular on the study of efficient solution for shared memory parallelism. In particular, in the framework of the PhD of D. Genet, the coupling with runtime systems such as StarPU (Inria team Runtime) and DAGuE (University of Tennesee), has bee compared to a more classical OpenMP implementation. This initial work, done for scalar problems, will be now extended to systems of equations.

4.2. COCA

Participants: Mario Ricchiuto [corresponding member], Gregory Perrot.

`COCA`(CodeOxydationCompositesAutocicatrisants) is a `Fortran 90` code for the simulation of the oxidation process in self-healing composites `COCA` solves the discrete finite element equations relative to the oxidation (chemistry) and flow (potential) models. Time integration is performed with an implicit approach (Backward Euler or second order backward differencing). The linear algebraic systems arising in the discretization are solved with the `MUMPS` library. Physical outputs of interest can be obtained to use `COCA` as a numerical closure for continuous mechanics solvers in order to perform numerical strain tests for self-healing composites.

4.3. RealfluiDS

Participants: Pietro Marco Congedo, Héloïse Beaugendre [corresponding member], Cécile Dobrzynski, Quentin Viville, Leo Nouveau.

`RealfluiDS` is a software dedicated to the simulation of inert or reactive flows. It is also able to simulate multiphase, multimaterial, MHD flows and turbulent flows (using the SA model). There exist 2D and 3D dimensional versions. The 2D version is used to test new ideas that are later implemented in the 3D one. This software implements the more recent residual distribution schemes. The code has been parallelized with and without overlap of the domains. The uncertainty quantification library `RobUQ` has been coupled to the software. A partitioning tool exists in the package, which uses `Scotch`. Recently, the code has been developed for taking into account real-gas effects, in order to use arbitrarily complex equations of state. Further developments concerning multiphase effects are under way.

4.4. MMG3D

Participants: Cécile Dobrzynski [corresponding member], Algiane Froehly.

MMG3D is a tetrahedral fully automatic remesher. Starting from a tetrahedral mesh, it produces quasi-uniform meshes with respect to a metric tensor field. This tensor prescribes a length and a direction for the edges, so that the resulting meshes will be anisotropic. The software is based on local mesh modifications and an anisotropic version of Delaunay kernel is implemented to insert vertices in the mesh. Moreover, MMG3D allows one to deal with rigid body motion and moving meshes. When a displacement is prescribed on a part of the boundary, a final mesh is generated such that the surface points will be moved according this displacement. MMG3D is/was used in gamma3 team, at EPFL (maths department), Dassault Aviation, Lemma (a french SME), Renault etc. MMG3D can be used in FreeFem++ (http://www.freefem.org), a free software which eases the solving of PDEs and in Gmsh (http://geuz.org/gmsh/).

Version 5.0 of MMG3D allows the modification of the surface triangulation based on cubic Bezier patches. A. Froehly, ingenieer in the FUI Rodin, is working on this new version.

More details can be found on http://www.math.u-bordeaux1.fr/~dobj/logiciels/mmg3d.php.

4.5. ORComp

Participants: Pietro Marco Congedo [Corresponding member], Maria-Giovanna Rodio.

The ORComp platform is a simulation tool permitting to design an ORC cycle. Starting from the solar radiation, this plateform computes the cycle providing the best performance with optimal choices of the fluid and the operating conditions. It includes RobUQ, a simulation block of the ORC cycles, the RealfluiDScode for the simulation of the turbine and of the heat exchanger, the software FluidProp (developed at the University of Delft) for computing the fluid thermodynamic properties.

4.6. sDEM

Participants: Pietro Marco Congedo [Corresponding member], Maria-Giovanna Rodio.

The sDEM platform is a simulation tool permitting to simulate multiphase flows with transition modelling. In particular, the code relies on the formulation of a DEM method, the use of a complex thermodynamics, the possibility to model cavitating phenomena. Moreover, the method has been generalized in order to take into account directly uncertainty, thus proposing the so-called Stochastic DEM (sDEM) method. This is one of the first stochastic semi-intrusive scheme, permitting to consider uncertainties in multiphase flows including heat and mass transfer terms. This software is developed together with the University of Zurich.

4.7. PaMPA

Participants: Cédric Lachat, François Pellegrini [Corresponding member], Cécile Dobrzynski.

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. It is based on a distributed data structure that represents meshes as a set of *entities* (elements, faces, edges, nodes, etc.), linked by *relations* (that is, computation dependencies).

PaMPA interfaces with Scotch for mesh redistribution, and with MMG3D for parallel remeshing of tetrahedral elements. Other sequential remeshers can be plugged-in, in order to handle other types of elements.

Version 1.0 of PaMPA allows users to declare distributed meshes, to declare values attached to the entities of the meshes (e.g. temperature attached to elements, pressures to the faces, etc.), to exchange values between overlapping entities located at the boundaries of subdomains assigned to different processors, to iterate over the relations of entities (e.g. iterate over the faces of elements), to remesh in parallel the areas of a mesh that need to be emeshed, and to redistribute evenly the remeshed mesh across the processors of the parallel architecture.

PaMPA is already used as the data structure manager for two solvers being developed at Inria: Plato (team PUMAS) and AeroSol (teams BACCHUS and CAGIRE).

Following expressions of interest from industrial partners, a formal industrialization process of PaMPA has been started, under the auspices of the *Direction du Transfert Technologique* (DTI) at Inria. In this context, much work was directed towards improving the robustness of the code, by including it into a continuous integration framework based on Jenkins.

4.8. RobUQ

Participants: Pietro Marco Congedo [Corresponding member], Maria Giovanna Rodio, Kunkun Tang.

The RobUQ platform has been conceived to solve problems in uncertainty quantification and robust design. It includes the optimization code ALGEN, and the uncertainty quantification code NISP. It includes also some methods for the computation of high-order statistics, efficient strategies for robust optimization, the Simplex2 method. Some methods are developed in partnership with the Stanford University (in the framework of the associated team AQUARIUS). Other methods are developed in the context of ANR UFO.

4.9. Scotch

Participant: François Pellegrini [corresponding member].

parallel graph partitioning, parallel static mapping, parallel sparse matrix block ordering, graph repartitioning, mesh partitioning.

Scotch (http://www.labri.fr/~pelegrin/scotch/) is a software package for parallel and sequential sparse matrix ordering, parallel and sequential graph partitioning, as well as sequential static mapping and remapping, without and with fixed vertices, and mesh and hypergraph partitioning.

The initial purpose of Scotch was to compute high-quality static mappings of valuated graphs representing parallel computations onto target architectures of arbitrary topologies. This allows the mapper to take into account the topology and heterogeneity of the target architecture in terms of processor speed and link bandwidth. This feature, which was meant for the NUMA machines of the 1980's, has not been widely used in the past because machines in the 1990's became UMA again thanks to hardware advances. Now, architectures become NUMA again, and these features are regaining popularity.

The Scotch package consists of two libraries: the sequential Scotch library, and the parallel PT-Scotch library (for "*Parallel Threaded Scotch*") that operates according to the distributed memory paradigm, using MPI. Scotch was the first full 64-bit implementation of a general purpose graph partitioner.

Version 6.0, released on December 2012, offers many new features: static mapping with fixed vertices, static remapping, and static remapping with fixed vertices. Several critical algorithms of the formerly strictly sequential Scotch library can now run in a multi-threaded way. All of these features, which exist only in the sequential version, will be available to the parallel PT-Scotch library in the upcoming release 6.1, the development of which has been pursued this year. Also, Scotch has been integrated into the Jenkins continuous integration framework that is used for other projects of the team, such as PaMPA and AeroSol.

Scotch has been integrated in numerous third-party software, which indirectly contribute to its diffusion. It is natively available in several Linux and Unix distributions, as well as on some vendors platforms (SGI, etc).

4.10. SLOWS

Participants: Luca Arpaia, Andrea Filippini, Maria Kazolea, Mario Ricchiuto [corresponding member].

SLOWS ("*Shallow-water fLOWS*") is a C-platform allowing the simulation of free surface shallow water flows with friction. Arbitrary bathymetries are allowed, defined either by some complex piecewise analytical expression, or by xyz data files, the classical Manning model for friction is used, and an Exner model is implemented for sediment transport. For non-hydrostatic propagation the enhanced Boussinesq equations of Madsen and Sorensen are used. The equations are discretized with a residual based approach which is an adaptation of the schemes developed for aeronautics applications. Due to the inherent unsteadiness of these flows, the time discretization plays an important role. Three different approaches are available, based on conditionally depth-positivity preserving implicit schemes, or on conditionally depth-positivity preserving genuinely explicit discretizations, or on an unconditionally depth-positivity preserving space-time approach. Newton and frozen Newton loops are used to solve the implicit nonlinear equations. The linear algebraic systems arising in the discretization are solved with the MUMPS library. This year implicit and explicit (extrapolated) multistep higher order time integration methods have been implemented, and a mesh adaptation technique based on simple mesh deformation has been also included. The current mid-term objective is to merge SLOWS with the TUCWave code developed by M. Kazolea during her PhD to obtain the first (worldwide) non-hydrostatic dynamically adaptive unstructured mesh Boussinesq code for wave propagation in the near shore region.

4.11. Nomesh

Participants: Cécile Dobrzynski [corresponding member], Algiane Froehly.

Nomesh is a software allowing the generation of third order curved simplicial meshes. Starting from a "classical" mesh with straight elements composed by triangles and/or tetrahedra, we are able to curve the boundary mesh. Starting from a mesh with some curved elements, we can verify if the mesh is valid, that means there is no crossing elements and only positive Jacobian. If the curved mesh is non valid, we modify it using linear elasticity equations until having a valid curved mesh.

BIPOP Project-Team

5. New Software and Platforms

5.1. Nonsmooth dynamics: Siconos

Participants: Vincent Acary, Maurice Brémond, Olivier Huber, Franck Pérignon.

In the framework of the European project Siconos, Bipop was the leader of the Work Package 2 (WP2), dedicated to the numerical methods and the software design for nonsmooth dynamical systems. The aim of this work is to provide a common platform for the simulation, modeling, analysis and control of abstract nonsmooth dynamical systems. Besides usual quality attributes for scientific computing software, we want to provide a common framework for various scientific fields, to be able to rely on the existing developments (numerical algorithms, description and modeling software), to support exchanges and comparisons of methods, to disseminate the know-how to other fields of research and industry, and to take into account the diversity of users (end-users, algorithm developers, framework builders) in building expert interfaces in Python and end-user front-end through Scilab.

After the requirement elicitation phase, the Siconos Software project has been divided into 5 work packages which are identified to software products:

1. SICONOS/NUMERICS This library contains a set of numerical algorithms, already well identified, to solve non smooth dynamical systems. This library is written in low-level languages (C,F77) in order to ensure numerical efficiency and the use of standard libraries (Blas, Lapack, ...)
2. SICONOS/KERNEL This module is an object-oriented structure (C++) for the modeling and the simulation of abstract dynamical systems. It provides the users with a set of classes to describe their nonsmooth dynamical system (dynamical systems, intercations, nonsmooth laws, ...) and to perform a numerical time integration and solving.
3. SICONOS/FRONT-END. This module is mainly an auto-generated wrapper in Python which provides a user-friendly interface to the Siconos libraries. A scilab interface is also provided in the Front-End module.
4. SICONOS/CONTROL This part is devoted to the implementation of control strategies of non smooth dynamical systems.
5. SICONOS/MECHANICS. This part is dedicated to the modeling and the simulation of multi-body systems with 3D contacts, impacts and Coulomb's friction. It uses the Siconos/Kernel as simulation engine but relies on a industrial CAD library (OpenCascade and pythonOCC) to deal with complex body geometries and to compute the contact locations and distances between B-Rep description and on Bullet for contact detection between meshes.

Further informations may be found at http://siconos.gforge.inria.fr/

5.2. Simulation of fibrous materials subject to frictional contact

5.2.1. MECHE: Modeling Entangling within Contacting hair fibErs

Participants: Florence Bertails-Descoubes, Gilles Daviet, Alexandre Derouet-Jourdan, Romain Casati, Laurence Boissieux.

The software MECHE was essentially developed during the MECHE ADT (2009-2011, research engineer: Gilles Daviet), for simulating the dynamics of assemblies of thin rods (such as hair), subject to contact and friction. Currently, this software is extensively used by two PhD students (A. Derouet-Jourdan and R. Casati) and continues to be enriched with new rod models and inversion modules. This software combines a panel of well-accepted models for rods (ranging from reduced coordinates to maximal coordinates models, and including models recently developed by some members of the group) with classical as well as innovative schemes for solving the problem of frictional contact (incorporating the most recent results of the group, as well as the new contact solver we published in [9]). The aim of this software is twofold: first, to compare and analyze the performance of nonsmooth schemes for the frictional contact problem, in terms of realism (capture of dry friction, typically), robustness, and computational efficiency. A first study of this kind was conducted in 2010-2011 onto the different rod models that were available in the software. New studies are planned for evaluating further rod models. Second, we believe such a software will help us understand the behavior of a fibrous material (such as hair) through virtual experiments, thanks to which we hope to identify and understand some important emergent phenomena. A careful validation study against experiments started to be conducted in 2011 in collaboration with physicists from L'Oréal. Once this discrete elements model will be fully validated, our ultimate goal would be to build a continuous macroscopic model for the hair medium relying on nonsmooth laws (which we have started to build in Gilles Daviet's PhD thesis). The core of this software was transferred to L'Oréal in 2011, and to AGT Digital in early 2013, by Gilles Daviet and Florence Bertails-Descoubes. It was also used for generating a number of simulations supporting at least 4 of our research publications.

CAGIRE Team

5. New Software and Platforms

5.1. AeroSol

Participants: Hamza Belkhayat Zougari [Cagire], Simon Delmas [Cagire], Damien Genet [Bacchus], Francois Pellegrini [Bacchus], Vincent Perrier [Cagire, correspondant], Mario Ricchiuto [Bacchus].

The software AeroSol is jointly developed in the team Bacchus and the team Cagire. It is a high order finite element library written in C++. The code design has been carried for being able to perform efficient computations, with continuous and discontinuous finite elements methods on hybrid and possibly curvilinear meshes.

The work of the team Bacchus is focused on continuous finite elements methods, while the team Cagire is focused on discontinuous Galerkin methods. However, everything is done for sharing the largest part of code we can. More precisely, classes concerning IO, finite elements, quadrature, geometry, time iteration, linear solver, models and interface with PaMPAare used by both of the teams. This modularity is achieved by mean of template abstraction for keeping good performances.

The distribution of the unknowns is made with the software PaMPA, developed within the team Bacchus and the team Castor.

At the end of 2013, Aerosol had the following features

- **development environment** use of CMake for compilation (gcc, icc and xlc), CTest for automatic tests and memory checking, lcov and gcov for code coverage reports. Development of a CDash server for collecting the unitary tests and the memory checking. Beginning of the development of an interface for functional tests. Optional linking with HDF5, PAPI, with dense small matrices libraries (BLAS, Eigen)
- **In/Out** link with the XML library for handling with parameter files. Parallel reader for GMSH, with an embedded geometrical pre-partitioner. Writer on the VTK-ASCII legacy format (cell and point centered). Parallel output in vtu and pvtu (Paraview) for cell-centered visualization, and XDMF/HDF5 format for both cell and point centered visualization.
- **Quadrature formula** up to 11th order for Lines, Quadrangles, Hexaedra, Pyramids, Prisms, up to 14th order for tetrahedron, up to 21st order for triangles. Gauss-Lobatto type quadrature formula for lines, triangles, quadrangles and hexaedra.
- **Finite elements** up to fourth degree for Lagrange finite elements and hierarchical orthogonal finite element basis (with Dubiner transform on simplices) on lines, triangles, quadrangles, tetrahedra, prisms, hexaedra and pyramids. Finite element basis that are interpolation basis on Gauss-Legendre points for lines, quadrangles, and hexaedra, and triangle (only 1st and 2nd order)
- **Geometry** elementary geometrical functions for first order lines, triangles, quadrangles, prisms, tetrahedra, hexaedra and pyramids.
- **Time iteration** explicit Runge-Kutta up to fourth order, explicit Strong Stability Preserving schemes up to third order. Optimized CFL time schemes: SSP(2,3) and SSP(3,4). CFL time stepping. Implicit integration with BDF schemes from 2nd to 6th order
- **Linear Solvers** link with the external linear solver UMFPack, PETSc and MUMPS. Internal solver for diagonal and block-diagonal matrices.
- **Memory handling** discontinuous and continuous, sequential and parallel discretizations based on PaMPA for generic meshes, including hybrid meshes.

- **Models** Perfect gas Euler system, real gas Euler system (template based abstraction for a generic equation of state), scalar advection, Waves equation in first order formulation, generic interface for defining space-time models from space models. Diffusive models: isotropic and anisotropic diffusion, compressible Navier-Stokes.
- **Numerical schemes** continuous Galerkin method for the Laplace problem (up to fifth order) with non consistent time iteration or with direct matrix inversion. Explicit and implicit discontinuous Galerkin methods for hyperbolic systems, diffusive and advection-diffusion problems. Beginning of optimization by stocking the geometry for advection problems. SUPG and Residual disribution schemes.
- **Numerical fluxes** centered fluxes, exact Godunov' flux for linear hyperbolic systems, and Lax-Friedrich flux. Riemann solvers for Low Mach flows.
- **Boundary conditions** Periodic boundary conditions, time-dependent inlet and outlet boundary conditions.
- **Parallel computing** Mesh redistribution, computation of Overlap with PaMPA. collective asynchronous communications (PaMPA based). Asynchronous point to point communications. Tests on the cluster Avakas from MCIA, and on Mésocentre de Marseille, and PlaFRIM. Tuer-1 Turing (Blue-Gene).
- **C++/Fortran interface** Tests for binding fortran with C++.
- **Instrumentation** Aerosol can give some traces on memory consumption/problems with an interfacing with the PAPI library. Tests have also been performed with VTUNE and TAU.

This year, the following features were added

- **In/Out** Ability of saving the high order solution and restarting from it. Computation of volumic and probe statistics. Ability of saving averaged layer data in quad and hexa meshes. Ability of defining user defined output visualization variables.
- **Geometry** handling of high order meshes.
- **Time iteration** Newton method for stationary problems. Implicite unstationary time iterator non consistent in time for stationary problems. Implementation of in house GMRES and conjugate gradient based on Jacobian free iterations.
- **Models** scalar advection-diffusion model
- **Numerical schemes** Optimization of DG schemes for advection-diffusion problems: stocking of the geometry and use of BLAS for all the linear phases of the scheme.
- **Numerical fluxes** Development of a new numerical flux accurate for steady and unsteady computations.
- **Boundary conditions** Adiabatic wall and isothermal wall; Steger-Warming based boundary condition.
- **Instrumentation** Tests with Maqao and Scalasca (VIHPS workshop)
- **Validation** Poiseuille, Taylor-Green vortex. Laplace equation on a ring and Poiseuille flow on a ring. Implementation of volumic forcing based on wall dissipation.

CLASSIC Project-Team (section vide)

COMMANDS Project-Team

5. New Software and Platforms

5.1. Bocop

Participants: Pierre Martinon [corresponding author], Frédéric Bonnans, Daphné Giorgi, Olivier Tissot.

Web page: http://bocop.org

The Bocop project aims to develop an open-source toolbox for solving optimal control problems, with collaborations with industrial and academic partners. Optimal control (optimization of dynamical systems governed by differential equations) has numerous applications in transportation, energy, process optimization, and biology. The software reuses some packages from the COIN-OR library, in particular the well-known nonlinear programming solver Ipopt, features a user-friendly interface and can be deployed on Windows / Mac / Linux.

The project is supported by Inria with the recruitment of Vincent Grelard as developer in 2010-2012, Daphné Giorgi (Oct. 2012-Sept. 2014), and Olivier Tissot since October 2014. The first prototype was released at the end of 2011, Bocop is currently at version 2.0.1 and has been downloaded more than 700 times. The software was first succesfully tested on several academic problems, see [55] available on http://bocop.org. Starting in 2012, several research collaborations were initiated in fields such as bio-reactors for energy production ([13], [26]), swimming micro-robots ([71]), and quantum control for medical imaging ([35]). Bocop was also featured during our participation in the Imatch "Optimisation and Control" in october 2013, leading to an ongoing contract with the startup Safety Line, on fuel optimization for civil aircrafts.

Bocop auto-assessment according to Inria notice: A3up4, SO3, SM3, EM3up4, SDL4up5

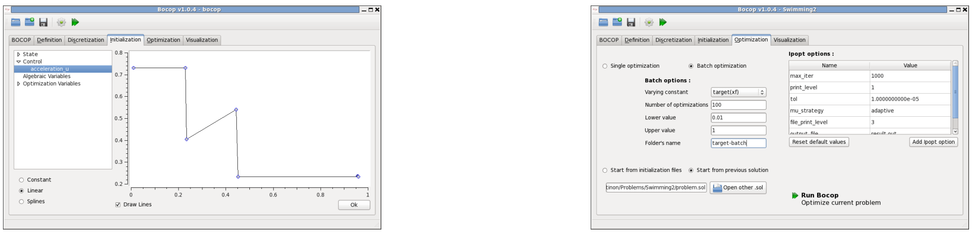


Figure 1. BOCOP

CORIDA Team

5. New Software and Platforms

5.1. Simulation of viscous fluid-structure interactions

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

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

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

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

5.2. Fish locomotion in perfect fluids with potential flow

Participants: Alexandre Munnier [correspondant], Bruno Pinçon.

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

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

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

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

5.3. SUSHI3D : SimUlations of Structures in Hydrodynamic Interactions

Participants: Jean-François Scheid, Takéo Takahashi.

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

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

5.4. The Vir'Volt prototype

Participants: Thomas Chambrion, Bruno Pinçon.

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

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

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

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

CQFD Project-Team

5. New Software and Platforms

5.1. Package PCAmixdata

Mixed data type arise when observations are described by a mixture of numerical and categorical variables. The R package PCAmixdata extends standard multivariate analysis methods to incorporate this type of data. The key techniques included in the package are PCAmix (PCA of a mixture of numerical and categorical variables), PCArot (rotation in PCAmix) and MFAmix (multiple factor analysis with mixed data within a dataset). The MFAmix procedure handles a mixture of numerical and categorical variables within a group - something which was not possible in the standard MFA procedure. We also included techniques to project new observations onto the principal components of the three methods in the new version of the package.

5.2. QuantifQuantile: an R package to estimate conditional quantiles using optimal quantization

QuantifQuantile is an R package that allows to perform quantization-based quantile regression. The different functions of the package allow the user to construct an optimal grid of N quantizers and to estimate conditional quantiles. This estimation requires a data driven selection of the size N of the grid that is implemented in the functions. Illustration of the selection of N is available, and graphical output of the resulting estimated curves or surfaces (depending on the dimension of the covariate) is directly provided via the plot function.

5.3. Biips: Software for Bayesian Inference with Interacting Particle Systems

Biips is a software platform for automatic Bayesian inference with interacting particle systems. Biips allows users to define their statistical model in the probabilistic programming BUGS language, as well as to add custom functions or samplers within this language. Then it runs sequential Monte Carlo based algorithms (particle filters, particle independent Metropolis-Hastings, particle marginal Metropolis-Hastings) in a black-box manner so that to approximate the posterior distribution of interest as well as the marginal likelihood. The software is developed in C++ with interfaces with the softwares R, Matlab and Octave [53].

DEFI Project-Team

5. New Software and Platforms

5.1. RODIN

Participant: Grégoire Allaire [correspondant].

In the framework of the RODIN project we continue to develop with our software partner ESI the codes Topolev and Geolev for topology and geometry shape optimization of mechanical structures using the level set method.

5.2. FreeFem++ Toolboxes

5.2.1. Shape optimization toolbox in FreeFem++

Participants: Grégoire Allaire, Olivier Pantz.

We propose several FreeFem++ routines which allow the users to optimize the thickness, the geometry or the topology of elastic structures. All examples are programmed in two space dimensions. These routines have been written by G. Allaire, B. Boutin, C. Dousset, O. Pantz. A web page of this toolbox is available at http://www.cmap.polytechnique.fr/~allaire/freefem_en.html.

We also have written a C++ code to solve the Hamilton Jacoby equation used in the Level-set shape optimization method. This code has been linked with FreeFem++ routines.

5.2.2. Eddy current problems

Participants: Zixian Jiang, Mohamed Kamel Riahi.

We developed a FreeFem++ toolbox that solves direct and inverse problems for an axisymmetric and 3D eddy current problems related to non destructive testing of deposits on the shell side of PWR fuel tubes. For the 3-D version, one can refer to http://www.cmap.polytechnique.fr/~riahi and also to [15].

5.2.3. Contact managements

Participant: Olivier Pantz.

We have developed a toolbox running under Freefem++ in order to take into account the non-intersection constraints between several deformable bodies. This code has been used to treat contacts between red blood cells in our simulations, but also between genuine non linear elastic structure. It can handle both contacts and self-contacts.
Moreover, a toolbox based on the Penalization method has also been developed.

5.2.4. De-Homogenization

Participant: Olivier Pantz.

We have developed a code under Freefem++ that implements our De-Homogenization method. It has been used to solve the compliance minimization problem of the compliance of an elastic shape. In particular, it enables us to recover well known optimal Michell's trusses for shapes of low density.

5.3. Scilab and Matlab Toolboxes

5.3.1. Shape optimization toolbox in Scilab

Participant: Grégoire Allaire [correspondant].

Together with Georgios Michailidis, we improved a Scilab toolbox for 2-d shape and topology optimization by the level set method which was originally produced by Anton Karrman and myself. The routines, a short user's manual and several examples are available on the web page: http://www.cmap.polytechnique.fr/~allaire/levelset_en.html

5.3.2. Conformal mapping method

Participant: Houssem Haddar [correspondant].

This Scilab toolbox is dedicated to the resolution of inverse 2-D electrostatic problems using the conformal mapping method introduced by Akdumann, Kress and Haddar. The toolbox treats the cases of a simply connected obstacle with Dirichlet, Neumann or impedance boundary conditions or a simply connected inclusion with a constant conductivity. The latest development includes the extension of the method to the inverse scattering problem at low frequencies as introduced by Haddar-Kress (2012).

5.3.3. SAXS Utilities

Participants: Federico Benvenuto [correspondant], Houssem Haddar.

We developed a scilab and matlab toolboxes that post treat SAXS type measurements to identify size distributions of diluted particles. We treat both axisymmetric measuremenst and anisotropic ones. The toolbox also simulates SAXS measurements associated with some canonical geometries.

5.3.4. Direct Solver for periodic media

Participants: Thi Phong Nguyen [correspondant], Houssem Haddar.

This Matlab toolbox solves the scattering from locally perturbed periodic layer using Floquet-Bloch transform and spectral discretization of associated volume integral equation.

5.4. Sampling methods for inverse problems

5.4.1. Samplings-2d

Participant: Houssem Haddar [correspondant].

This software is written in Fortran 90 and is related to forward and inverse problems for the Helmholtz equation in 2-D. It includes three independent components. The first one solves to scattering problem using integral equation approach and supports piecewise-constant dielectrics and obstacles with impedance boundary conditions. The second one contains various samplings methods to solve the inverse scattering problem (LSM, RGLSM(s), Factorization, MuSiC) for near-field or far-field setting. The third component is a set of post processing functionalities to visualize the results
See also the web page http://sourceforge.net/projects/samplings-2d/.

- License: GPL
- Type of human computer interaction: sourceforge
- OS/Middelware: Linux
- Programming language: Fortran
- Documentation: fichier

5.4.2. Samplings-3d

Participant: Houssem Haddar [correspondant].

This software is written in Fortran 90 and is related to forward and inverse problems for the Helmholtz equation in 3-D. It contains equivalent functionalities to samplings-2d in a 3-D setting.

5.4.3. Time domain samplings-2d

Participant: Houssem Haddar [correspondant].

This software is written in Fortran 90 and is related to forward and inverse problems for the time dependent wave equation in 2-D. The forward solver is based on a FDTD method with PMLs. The inverse part is an implementation of the linear sampling method in a near field setting and the factorization method in a far field setting.

5.5. BlochTorreyPDESolver

Participants: Jing-Rebecca Li [correspondant], Dang Van Nguyen.

We developed two numerical codes to solve the multiple-compartments Bloch-Torrey partial differential equation in 2D and 3D to simulate the water proton magnetization of a sample under the influence of diffusion-encoding magnetic field gradient pulses.

We coupled the spatial discretization with an efficient time discretization adapted to diffusive problems called the (explicit) Runge-Kutta-Chebyshev method.

The version of the code using Finite Volume discretization on a Cartesian grid is complete (written by Jing-Rebecca Li). The version of the code using linear Finite Elements discretization is complete (written by Dang Van Nguyen and Jing-Rebecca Li).

See the web page http://www.cmap.polytechnique.fr/~jingrebeccali/ for more details.

DISCO Project-Team

5. New Software and Platforms

5.1. OreModules

Participants: Alban Quadrat [correspondent], Daniel Robertz [Univ. Aachen], Frédéric Chyzak [Inria Rocquencourt, Algorithms Project].

The OREMODULES package [73], based on the commercial Maple package Ore_algebra [74], is dedicated to the study of linear multidimensional systems defined over certain Ore algebras of functional operators (e.g., ordinary or partial differential systems, time-delay systems, discrete systems) and their applications in mathematical systems theory, control theory and mathematical physics. OREMODULES is original because it combines the recent developments of the Gröbner bases over some noncommutative polynomial rings [82], [84] and new algorithms of algebraic analysis in order to effectively check classical properties of module theory (e.g., existence of a non-trivial torsion submodule, torsion-freeness, reflexiveness, projectiveness, stably freeness, freeness), it gives their system-theoretical interpretations (existence of autonomous elements or successive parametrizations, existence of minimal/injective parametrizations or Bézout equations) [90], [89], [72] and it computes important tools of homological algebra (e.g., (minimal) free resolutions, split exact sequences, extension functors, projective or Krull dimensions, Hilbert power series). The abstract language of homological algebra used in the algebraic analysis approach carries over to the implementations in OREMODULES: up to the choice of the domain of functional operators which occurs in a given system, all algorithms are stated and implemented in sufficient generality such that linear systems defined over the Ore algebras developed in the Ore_algebra package are covered at the same time. Applications of the OREMODULES package to mathematical systems theory, control theory and mathematical physics are illustrated in a large library of examples. The binary of the package is freely available at http://wwwb.math.rwth-aachen.de/OreModules/.

5.2. Stafford

Participants: Alban Quadrat [correspondent], Daniel Robertz [Univ. Aachen].

The STAFFORD package of OREMODULES [73] contains an implementation of two constructive versions of Stafford's famous but difficult theorem [96] stating that every ideal over the Weyl algebra $A_n(k)$ (resp., $B_n(k)$) of partial differential operators with polynomial (resp., rational) coefficients over a field k of characteristic 0 (e.g., $k = \mathbb{Q}$, $\mathbb{R}$) can be generated by two generators. Based on this implementation and algorithmic results developed in [92] by the authors of the package, two algorithms which compute bases of free modules over the Weyl algebras $A_n(\mathbb{Q})$ and $B_n(\mathbb{Q})$ have been implemented. The rest of Stafford's results developed in [96] have recently been made constructive in [93] (e.g., computation of unimodular elements, decomposition of modules, Serre's splitting-off theorem, Stafford's reduction, Bass' cancellation theorem, minimal number of generators) and implemented in the STAFFORD package. The development of the STAFFORD package was motivated by applications to linear systems of partial differential equations with polynomial or rational coefficients (e.g., computation of injective parametrization, Monge problem, differential flatness, the reduction and decomposition problems and Serre's reduction problem). To our knowledge, the STAFFORD package is the only implementation of Stafford's theorems nowadays available. The binary of the package is freely available at http://wwwb.math.rwth-aachen.de/OreModules/.

5.3. QuillenSuslin

Participants: Anna Fabiańska [Univ. Aachen], Alban Quadrat [correspondent].

The QUILLEN-SUSLIN package [78] contains an implementation of the famous Quillen-Suslin theorem [94], [97]. In particular, this implementation allows us to compute bases of free modules over a commutative polynomial ring with coefficients in a field (mainly $\mathbb{Q}$) and in a principal ideal domain (mainly $\mathbb{Z}$). The development of the QUILLEN-SUSLIN package was motivated by different constructive applications of the Quillen-Suslin theorem in multidimensional systems theory [78] (e.g., the Lin-Bose conjectures, the computation of (weakly) left/right/doubly coprime factorizations of rational transfer matrices, the computation of injective parametrizations of flat linear multidimensional systems with constant coefficients, the reduction and decomposition problems, Serre's reduction problem). To our knowledge, the QUILLEN-SUSLIN package is the only implementation of the Quillen-Suslin theorem nowadays available. The binary of the package is freely available at http://wwwb.math.rwth-aachen.de/QuillenSuslin.

5.4. OreMorphisms

Participants: Thomas Cluzeau [ENSIL, Univ. Limoges], Alban Quadrat [correspondent].

The OREMORPHISMS package [76] of OREMODULES [72] is dedicated to the implementation of homological algebraic tools such as the computations of homomorphisms between two finitely presented modules over certain noncommutative polynomial algebras (Ore algebras), of kernel, coimage, image and cokernel of homomorphisms, Galois transformations of linear multidimensional systems and idempotents of endomorphism rings. Using the packages STAFFORD and QUILLEN-SUSLIN, the factorization, reduction and decomposition problems can be constructively studied for different classes of linear multidimensional systems. Many linear systems studied in engineering sciences, mathematical physics and control theory have been factorized, reduced and decomposed by means of the OREMORPHISMS package. The binary of the package is freely available at http://pages.saclay.inria.fr/alban.quadrat/OreMorphisms/index.html.

5.5. PurityFiltration

Participant: Alban Quadrat [correspondent].

The PURITYFILTRATION package, built upon the OREMODULES package, is an implementation of a new effective algorithm obtained in [24] which computes the purity/grade filtration [67], [68] of linear functional systems (e.g., partial differential systems, differential time-delay systems, difference systems) and equivalent block-triangular matrices. This package is used to compute closed form solutions of over/underdetermined linear partial differential systems which cannot be integrated by the standard computer algebra systems such as Maple and Mathematica. For more information, see http://pages.saclay.inria.fr/alban.quadrat/OreAlgebraicAnalysis/index.html.

5.6. OreAlgebraicAnalysis

Participants: Thomas Cluzeau [ENSIL, Univ. Limoges], Alban Quadrat [correspondent], Maris Tõnso [Institute of Cybernetics, Univ. Tallinn].

OREALGEBRAICANALYSIS is a Mathematica implementation of algorithms available in the OREMODULES and the OREMORPHISMS packages (developed in Maple). OREALGEBRAICANALYSIS is based on the implementation of Gröbner bases over Ore algebras available in the Mathematica *HolonomicFunctions* package developed by Christoph Koutschan (RICAM). OREALGEBRAICANALYSIS can handle larger classes of Ore algebras than the ones accessible in Maple, and thus we can study larger classes of linear functional systems. Finally, Mathematica internal design allows us to consider classes of systems which could not easily be considered in Maple such as generic linearizations of nonlinear functional systems defined by explicit nonlinear equations and systems containing transcendental functions (e.g., trigonometric functions, special functions). This package has been developed within the PHC Parrot project CASCAC.

5.7. YALTA

Participants: Catherine Bonnet [correspondent], Hugo Cavalera, André R. Fioravanti [UNICAMP], Jim Pioche [SciWorks Technologies].

The YALTA toolbox is a Matlab toolbox dedicated to the study of classical and fractional systems with delay in the frequency-domain. Its objective is to provide basic but important information such as, for instance, the position of the neutral chains of poles and unstable poles, as well as the root locus with respect to the delay of the system. The corresponding algorithms are based on recent theoretical results (see, for instance, [70] and [79]) and on classical continuation methods exploiting the particularities of the problem [80], [81].

For classical delay systems, a Pade2 approximation scheme is available as well as a finite-dimensional approximation of the system.

Recently, some optimizations and features have been added. For instance, the main software function has been split into several procedures, allowing some calculus such as finding the position of the neutral chains to be processed independently of more computationnally expensive ones (for instance determining the root locus with respect to the delay of the system). In parallel, software documentation has been rewritten.

Binaries are freely available at http://yalta-toolbox.gforge.inria.fr/.

The YALTA GUI (graphical user interface) is a graphical application developed in Python that interacts with the Matlab toolbox 5.7 . User actions are performed through intuitive graphic elements (dialog boxes, icons, menus, scroll bars) in order to capitalize on the functionalities of YALTA. This software, still in development, is based on PyQt, a Python binding of the cross-platform GUI toolkit Qt (C++).

DOLPHIN Project-Team

5. New Software and Platforms

5.1. Comparing Continuous Optimizers (Coco) Platform

Participants: Dimo Brockhoff, Arnaud Liefooghe, Thanh-Do Tran.

The Coco Platform (coco.gforge.inria.fr) provides the functionality to automatically benchmark optimization algorithms for unbounded, unconstrained optimization problems in continuous domains. Benchmarking is a vital part of algorithm engineering and a necessary path to recommend algorithms for practical applications. The Coco platform releases algorithm developers and practitioners alike from (re-)writing test functions, logging, and plotting facilities by providing an easy-to-handle interface in several programming languages. The Coco platform has been developed since 2007 and has been used extensively within the “Blackbox Optimization Benchmarking (BBOB)” workshop series since 2009. Overall, 123 algorithms and algorithm variants by contributors from all over the world have been benchmarked with the platform so far and all data is publicly available for the research community (see for example http://coco.gforge.inria.fr/doku.php?id=bbob-2013-algorithms for the submissions to BBOB-2013).

Via the ANR project NumBBO, also Dolphin team members are involved in the development of Coco and it is one of the main purposes of NumBBO to extend the Coco platform towards expensive, large-scale, constrained and multiobjective optimization. In order to facilitate these extensions, a complete overhaul of the platform is currently underway—rewriting the whole functionality from scratch in a single language (ANSI C) which will then be called from all other provided languages (Java, python, Matlab/Octave, R) instead of multiple independent implementations which are highly difficult to maintain. A first release of the new code base is expected in the first half of 2015.

For the rewriting of the source code, we also moved to a publicly available open source code repository at https://code.google.com/p/numbbo/ which, in addition, provides for the first time a bug and feature request tracking system for Coco. As the first Coco-related deliverable of the NumBBO project, the extension towards expensive optimization has been finished this year. Its full functionality will be provided for the first time for the upcoming BBOB special session at the IEEE Congress on Evolutionary Computation (CEC’2015). Coco is also currently extended towards multiobjective optimization in close relation to the PhD topic of Thanh-Do Tran. A first working (but still preliminary) version of the multiobjective Coco has been developed and is expected to be merged with the newly rewritten Coco code just after its first release.

Inria software self-assessment for Coco: [A-4/5, SO-4, SM-3-up4, EM-3, SDL-4-up5][DA-3, CD-3, MS-3, TPM-2]

5.2. MO-Mine

Participants: Clarisse Dhaenens, Benjamin Fisset, Laetitia Jourdan.

The MO-Mine platform (mo-mine.gforge.inria.fr) aims at providing optimization algorithms, and in particular multi-objective approaches, to deal with data-mining classical tasks (Classification, association rules...). The platform is still in development in the context of an Inria ADT.
$MO - Mine_{clust}$ is the first package of the platform and is dedicated to clustering (unsupervised classification). Indeed, it is well-known that clustering may be seen as a bi-objective optimization problem as the goal is both to minimize distances between data belonging to a same cluster, while maximizing distances between data belonging to different clusters. Several models (objective functions used,...) and engines (optimization algorithms) have been implemented. The framework searches, for a given dataset, the best association of model/engine/parameter without specifiing the number of clusters. $MO - Mine_{clust}$ shows very interesting behavior and shows that the choice of the model and the engine has a great importance in the performance of the method and depends on the dataset to analyze.

Inria software self-assessment for $MO - Mine_{clust}$: [A-1, SO-4, SM-3-up4, EM-2, SDL-1,DA-4,CD-4]

5.3. Platforms

Grid'5000 [0] is a French nation-wide computational grid infrastructure composed of several clusters of processors distributed over 11 sites including Lille. Since 2004, the Dolphin team is the scientific leader of the Grid'5000 site located at Lille. The role of the team is mainly threefold consisting in ensuring (1) the maintenance and evolution of the platform, (2) the local and national coordination of the related activities, and (3) the scientific animation around Grid'5000. Regarding the evolution of the infrastructure, this later has been extended in 2014 with a new powerful storage server hosting mainly the NFS system. In addition, the team has contributed to the 5-year CPER project called "data" proposed by the Inria Lille research center. The project includes a part related to the Grid'5000 platform. From the coordination point of view, Dolphin has participated to the monthly meetings of the national Grid'5000 committee. Finally, regarding the scientific animation, first, Dolphin has organized in December 20$^{\text{th}}$, 2014, a training [0] around Grid'5000. Second, Dolphin has participated to the program committee of the Grid'5000 spring school [0] held in Lyon in June 2014. Third, the team has also been involved in the final evaluation of the scientific nation-wide Hemera research project [0] related to Grid'5000.

[0] https://www.grid5000.fr/
[0] http://www.lifl.fr/~melab/HTML/Journee-G5K-Lille.htm
[0] https://www.grid5000.fr/mediawiki/index.php/Grid5000:School2014
[0] https://www.grid5000.fr/Hemera

ECUADOR Project-Team

5. New Software and Platforms

5.1. AIRONUM

Participant: Alain Dervieux [correspondant].

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

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

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

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

5.2. TAPENADE

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

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

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

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

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

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

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

GAMMA3 Project-Team

3. New Software and Platforms

3.1. BLGEOL-V1 software

Participants: Patrick Laug [correspondant], Houman Borouchaki.

BLGEOL-V1 software can generate hex-dominant meshes of geologic structures complying with different geometric constraints: surface topography (valleys, reliefs, rivers), geologic layers and underground workings. First, a reference 2D domain is obtained by projecting all the line constraints into a horizontal plane. Different size specifications are given for rivers, outcrop lines and workings. Using an adaptive methodology, the size variation is bounded by a specified threshold in order to obtain a high quality quad-dominant mesh. Secondly, a hex-dominant mesh of the geological medium is generated by a vertical extrusion, taking into account the surfaces found (interfaces between two layers, top or bottom faces of underground workings). The generation of volume elements follows a global order established on the whole set of surfaces to ensure the conformity of the resulting mesh.

GECO Project-Team

5. New Software and Platforms

5.1. IRHD

We develop a software for reconstruction of corrupted and damaged images, named IRHD (for Image Reconstruction via Hypoelliptic Diffusion). One of the main features of the algorithm on which the software is based is that it does not require any information about the location and character of the corrupted places. Another important advantage is that this method is massively parallelizable; this allows to work with sufficiently large images. Theoretical background of the presented method is based on the model of geometry of vision due to Petitot, Citti and Sarti. The main step is numerical solution of the equation of 3D hypoelliptic diffusion. IRHD is based on Fortran.

GEOSTAT Project-Team

5. New Software and Platforms

5.1. Fluex

Participants: Rémi Paties [correspondent], Hussein Yahia, Joel Sudre.

- Previous software engineer Denis Arrivault has delivered the first Fluex package in December 2013, consisting of a core implementation under Gforge of the Microcanonical Multiscale Formalism in the form of C++ classes, for 1D, 2D 3D and 3D+t general signals. The Fluex project is carried on in 2014 by Rémi Paties. Contact: remi.paties@inria.fr.
- The Fluex project has been supported by the FLUEX ADT.

5.2. Platforms

5.2.1. Plafrim

GEOSTAT has participated financially in the acquisition of a computing server SGI UV2000 for PLAFRIM through funding with the OPTAD project (Conseil Région Aquitaine).

I4S Project-Team

5. New Software and Platforms

5.1. ISTL

Participant: Qinghua Zhang.

ISTL is a software realizing numerical computations of the inverse scattering transform for electrical transmission lines. It provides an efficient solution to experimentally determining the distributed characteristic impedance of an electrical transmission line from the reflection coefficient measured at one end of the line. Its current applications are in the fields of electrical cable fault diagnosis and of civil engineering structure monitoring. In addition to inverse scattering transform algorithms, ISTL includes a numerical simulator generating reflection coefficients of user-specified transmission lines and a graphical user interface. It is registered at Agence pour la Protection des Programmes (APP) under the number IDDN.FR.001.120003.000.S.P.2010.000.30705. See http://people.rennes.inria.fr/Qinghua.Zhang/istl.html.

5.2. PEGASE

Participants: Vincent Le Cam, Mathieu Le Pen, Laurent Mevel, Michael Doehler.

I4S is actually finalizing the setup of a new platform named PEGASE 2.0 as the technological successor of the previous PEGASE platform developed by IFSTTAR.

The new version of PEGASE keeps the best of its previous version in its main vocation, to be a generic high level Wireless Sensor Platofrm.

- What does not change between PEGASE 1 and 2.0: Based on various feedback from application fields, results from real structures monitored by PEGASE, and due to the rapid obsolescence of electronic devices, the design of the new PEGASE platform has been launched in 2013. Some of the main functions of PEGASE does not change but are reinforced.
  - Software genericity: use of a Linux embedded OS to make any application developed independently from the hardware, to make the user able to manage the system without ant physical and heavy operations.
  - Hardware genericity: with a principle of daughter and mother boards, each redundant need is embedded (processing, memory, timing, GPS, energy, etc) which each pluggable daughter board implements a specific function (sensing, 3G, Ethernet, communication, signal processing and relay control).
  - Accurate time synchronization: based on an original GPS and PPS algorithm, PEGASE platform is one of the only board able to time-stamp data from sensors or any event with an accuracy of some micro-seconds Universal Time.
- What's new on PEGASE 2 platform ?

  Previous principles are maintained or extended. Full electronic design from scratch occurred in 2014 to maximise its capacities in terms efficiency, cost, energy consumption, etc. Its main characteristics are
  - Important software evolutions: the platform embedded a real Linux kernel (not μClinux as previously done for memory size questions). This new kernel allows the perception of PEGASE 2 as real PC (without screen and mouse) providing important functions as MMU, software upgrade.
  - Important hardware evolutions and integration

    * A very advanced GPS module (Ublox Neo 6T) to allow more accurate time synchronization (up to 100 nanoseconds)
    * An embedded energy harvesting module (and not from a daughter board as previously done) to recover energy from DC sources if available or a solar cell, while managing the low of dis/charging Lithium-Ion battery and the MMPT algorithms
  - A Single Development Kit (SDK) fully (re) coded in C++ (and not C-object as proposed before) that permit a real capitalization of software developments and knowledges implemented (such as algorithms for SHM). Based on an UML model
- A major evolution in PEGASE concept consists in providing a generic web-tool to monitor PEGASE platform (whatever is the version) and many others wireless and commercial devices. A reproach addressed to previous concept resides in the fact that PEGASE was too focused on providing a generic wireless platform. But a quite big work was still necessary to monitor the devices. This new concept has been already sold to companies (such as SNCF or Cofiroute) and allows:
  - To create one independent instance of the Supervisor by client
  - Each instance of the Supervisor works 100% in the cloud with a secured access (https + login/password). Thus final users can operate it from anywhere in the world : at instrumented site level as at desk or during travel, etc.
  - For each client, the possibility to create an infinity of instrumentation projects
  - For each instrumentation project, to associate as many sensors as required by the application. Sensors can be PEGASE 1 or PEGASE 2 boards as many others : Labjack devices, some National Instruments acquisition boards, Meteo-France sources...

  The list of the devices known by the Supervisor is open and is supposed, year after year, to be completed. Thus, next PEGASE projects aims at providing not only some wireless sensors platform but also a modern, full-clouded, monitoring application. Moreover the 2015 $R\&D$ program plans to add a very interesting function from the scientific point of view: a Matlab plug-in. The idea consists in linking the data flow managed by supervisor directly and automatically to some Matlab sources codes uploaded on the web platform. The Supervisor will compile the original Matlab files. They are dynamically compiled on an embedded Matlab runtime library on the cloud server. Thus, once the the specifications about data format are written and took into account by developers, scientist can dynamically test and operate its Matlab models uploaded on the Supervisor.

IPSO Project-Team (section vide)

MATHERIALS Team (section vide)

MATHRISK Project-Team

5. New Software and Platforms

5.1. PREMIA

Participants: Antonino Zanette, Mathrisk Research Team, Agnès Sulem [correspondant].

5.1.1. Premia: general description

Premia is a software designed for option pricing, hedging and financial model calibration. It is provided with it's C/C++ source code and an extensive scientific documentation. https://www-rocq.inria.fr/mathfi/Premia

The Premia project keeps track of the most recent advances in the field of computational finance in a well-documented way. It focuses on the implementation of numerical analysis techniques for both probabilistic and deterministic numerical methods. An important feature of the platform Premia is the detailed documentation which provides extended references in option pricing.

Premia is thus a powerful tool to assist Research & Development professional teams in their day-to-day duty. It is also a useful support for academics who wish to perform tests on new algorithms or pricing methods without starting from scratch.

Besides being a single entry point for accessible overviews and basic implementations of various numerical methods, the aim of the Premia project is:

1. to be a powerful testing platform for comparing different numerical methods between each other;
2. to build a link between professional financial teams and academic researchers;
3. to provide a useful teaching support for Master and PhD students in mathematical finance.

- AMS: 91B28;65Cxx;65Fxx;65Lxx;65Pxx
- License: Licence Propriétaire (genuine license for the Consortium Premia)
- Type of human computer interaction: Console, interface in Nsp, Web interface
- OS/Middelware: Linux, Mac OS X, Windows
- APP: The development of Premia started in 1999 and 15 are released up to now and registered at the APP agency. Premia 15 has been registered on 18/03/2013 and has the number IDDN.FR.001.190010.012.S.C.2001.000.31000
- Programming language: C/C++ librairie Gtk
- Documentation: the PNL library is interfaced via doxygen
- Size of the software: 280580 lines for the Src part only, that is 11 Mbyte of code, 130400 lines for PNL, 105 Mbyte of PDF files of documentation.
- interfaces : Nsp for Windows/Linux/Mac, Excel, binding Python, and a Web interface.
- Publications: [12], [58], [68], [79], [84], [40], [18].

5.1.2. Content of Premia

Premia contains various numerical algorithms (Finite-differences, trees and Monte-Carlo) for pricing vanilla and exotic options on equities, interest rate, credit and energy derivatives.

1. **Equity derivatives**

   The following models are considered:

   Black-Scholes model (up to dimension 10), stochastic volatility models (Hull-White, Heston, Fouque-Papanicolaou-Sircar), models with jumps (Merton, Kou, Tempered stable processes, Variance gamma, Normal inverse Gaussian), Bates model.

For high dimensional American options, Premia provides the most recent Monte-Carlo algorithms: Longstaff-Schwartz, Barraquand-Martineau, Tsitsklis-Van Roy, Broadie-Glassermann, quantization methods and Malliavin calculus based methods.

Dynamic Hedging for Black-Scholes and jump models is available.

Calibration algorithms for some models with jumps, local volatility and stochastic volatility are implemented.

2. **Interest rate derivatives**

   The following models are considered:

   HJM and Libor Market Models (LMM): affine models, Hull-White, CIR++, Black-Karasinsky, Squared-Gaussian, Li-Ritchken-Sankarasubramanian, Bhar-Chiarella, Jump diffusion LMM, Markov functional LMM, LMM with stochastic volatility.

   Premia provides a calibration toolbox for Libor Market model using a database of swaptions and caps implied volatilities.

3. **Credit derivatives: Credit default swaps (CDS), Collateralized debt obligations (CDO)**

   Reduced form models and copula models are considered.

   Premia provides a toolbox for pricing CDOs using the most recent algorithms (Hull-White, Laurent-Gregory, El Karoui-Jiao, Yang-Zhang, Schönbucher)

4. **Hybrid products**

   A PDE solver for pricing derivatives on hybrid products like options on inflation and interest or change rates is implemented.

5. **Energy derivatives: swing options**

   Mean reverting and jump models are considered.

   Premia provides a toolbox for pricing swing options using finite differences, Monte-Carlo Malliavin-based approach and quantization algorithms.

5.1.3. Premia design

Premia has managed to grow up over a period of fifteen years; this has been possible only because contributing an algorithm to Premia is subject to strict rules, which have become too stringent. To facilitate contributions, a standardized numerical library (PNL) has been developed by J. Lelong under the LGPL since 2009, which offers a wide variety of high level numerical methods for dealing with linear algebra, numerical integration, optimization, random number generators, Fourier and Laplace transforms, and much more. Everyone who wishes to contribute is encouraged to base its code on PNL and providing such a unified numerical library has considerably eased the development of new algorithms which have become over the releases more and more sophisticated. An effort is made to continue and stabilize the development of PNL. J. Ph Chancelier, B. Lapeyre and J. Lelong are using Premia and Nsp for Constructing a Risk Management Benchmark for Testing Parallel Architecture [18].

Development of the PNL in 2014 (J. Lelong) - Release 1.70. and 1.7.1 of the *PNL* library (http://pnl.gforge.inria.fr/).

1. Sampling from new distributions: non central Chi squared, Poisson, Bernoulli.
2. When using quasi Monte Carlo sequences, sampling from any distribution resorts to using the inverse of the cumulative distribution function technique.
3. A CMake module is provided to automatically detect the library when used by third party codes.
4. Add a sparse matrix object with advanced functionalities provided by Blas & Lapack. This new object is handled by the MPI binding.

5. Complete refactoring of the Basis object to considerably speedup the evaluation functions. Multivariate polynomials are represented as tensor products of one variate polynomials. The matrix holding the tensor product now uses a sparse storage which avoids many operations leading to a zero value thus leading to an impressive reduction the computational time.
6. All random number generators are thread–safe.

5.1.4. Algorithms implemented in Premia in 2014

Premia 16 has been delivered to the Consortium Premia in March 2014. In this release we have developed the Haar Wavelets-based approach for quantifying credit portfolio losses, Monte Carlo simulations of Credit Value Adjustment (CVA) using Malliavin techniques, asymptotic and exact pricing options on variance and importance sampling, and multilevel methods for jump models.

It contains the following new algorithms:

5.1.4.1. Commodities, Forex (FX), Insurance, Credit Risk

- Pricing and hedging gap risk. P. Tankov.
  Journal of Computational Finance. Volume 13 Number 3, Spring 2010.
- An Optimal Stochastic Control Framework for Determining the Cost of Hedging of Variable Annuities. P. A. Forsyth K.Vetzal
- Haar Wavelets-Based Approach for Quantifying Credit Portfolio Losses.
  J. J. Masdemont, L. O. Gracia. *Quantitative Finance, to appear*
- Cutting CVA's complexity. P. Henry-Labordère. *Risk Magazine 04 Jul 2012*
- Towards a coherent Monte Carlo simulation of CVA. L. Abbas Turki, A.Bouselmi, M.Mikou, *hal-00873200*
- Stochastic local intensity loss models with interacting particle system.
  A. Alfonsi, C. Labart, J. Lelong *Mathematical Finance, to appear*
- Repricing the Cross Smile: An Analytic Joint Density. P. Austing.

5.1.4.2. Equity Derivatives

- On the Fourier cosine series expansion (COS) method for stochastic control problems. R.F.T. Aalber, C.W. Oosterlee and M.J. Ruijter.
- A multifactor volatility Heston model. J.Da Fonseca M. Grasselli C. Tebaldi.
  Quant. Finance 8 (2008), no. 6, 591—604.
- General approximation schemes for option prices in stochastic volatility models. K.Larsson, *Quantitative Finance Volume 12, Issue 6, 2012*
- A robust tree method for pricing American options with the Cox-Ingersoll-Ross interest rate model.
  E. Appolloni, L. Caramellino and A. Zanette
  IMA Journal of Management Mathematics 2014, to appear.
- A hybrid tree-finite difference approach for the Heston and Bates model model. M. Briani, L. Caramellino and A. Zanette *Preprint* ArXiv 1307.7178
- A Closed-Form Exact Solution for Pricing Variance Swaps with Stochastic Volatility. S. Zhu and G. Lian, *Mathematical Finance, Volume 21, Issue 2, April 2011*
- Asymptotic and exact pricing options on variance. M.Keller-Ressel J.Muhle-Karbe.
  Finance & Stochastics, Volume 17 (2013), issue 1
- Importance sampling and Statistical Romberg Method for jump models.
  M.B. Alaya, A. Kebaier and K. Hajji
- Smart expansion and fast calibration for jump diffusions E. Benhamou, E. Gobet and M. Miri, *Finance Stochastics Volume 13, Number 4, September, 2009*
- Scaling and multiscaling in financial series: a simple model.
  Andreoli, A., Caravenna, F, Dai Pra, P. Posta, G., *Advances in Applied Probability. (2012), 44(4), 1018-1051.*
- Smooth convergence in the binomial model.
  L.B. Chang, K. Palmer.*Finance Stochastics Volume 11, Number 1,2007.*

Maxplus Project-Team

5. New Software and Platforms

5.1. Boîte à outil Maxplus de SCILAB/Maxplus toolbox of Scilab

Trois chercheurs du groupe (S. Gaubert, J.-P. Quadrat, et G. Cohen) ont développé (à partir d'une première version réalisée par M. Mc Gettrick) la *boîte à outils Maxplus* de Scilab, qui est téléchargeable librement parmi les contributions du site Scilab, et qui est maintenant intégrée par défaut dans Scicoslab. Cette boîte à outils implémente l'ensemble du calcul numérique linéaire max-plus, elle comprend en particulier le stockage creux des matrices, et des algorithmes efficaces pour le calcul de la valeur propre basées sur les itérations sur les politiques. Elle a été utilisées par plusieurs chercheurs, voir notamment [76], [147]. Il faut aussi noter que le groupe de L. Hardouin, du LISA/Istia, a complété la boîte à outils Maxplus en interfaçant leur propre librairie C++, qui permet le calcul des séries de transfert de graphes d'événements temporisés.

English version

Three researchers of the team (S. Gaubert, J.-P. Quadrat, and G. Cohen, building on a preliminary version of M. McGettrick) have developed and released the *Maxplus toolbox* of Scilab, which is freely available among the contributions on the Scilab web site, and which is now included by default in Scicoslab. It implements all basic linear algebra functionalities, with a special attention to large sparse matrices, including efficient algorithms for eigenvalue computation based on policy iteration. The software has been used by several researchers in their work, including [76], [147]. It should be noted that the team of L. Hardouin, from LISA/Istia, has completed the toolbox by interfacing their own C++ library computing the transfer series of a timed event graph.

5.2. Itérations sur les politiques pour les jeux stochastiques à somme nulle/Policy iterations for zero sum stochastic games

L'algorithme d'itérations sur les politiques pour les jeux stochastiques à somme nulle pour le cas de paiements ergodiques (gain moyen par unité de temps), et dégénérés de type “multi-chaîne” a été introduit dans [95]. Plusieurs stages ont permis l'implémentation partielle en Scilab, C ou C++, et le test de ce type d'algorithmes (voir le travail de Vishesh Dhingra [112]), ou de son couplage avec la résolution de systèmes linéaires par des méthodes multigrilles algébriques (stage de Shantanu Gangal en 2007). Le travail de thèse de Sylvie Detournay a permis le développement d'un programme complet. Le code écrit par Sylvie Detournay (en C) a été déposé sur InriaGForge. Pour le moment il n'est accessible qu'aux membres de l'équipe.

English version

The policy iteration algorithm for zero sum repeated games with ergodic payoff (i.e. mean payoff per time unit), and in degenerate “multichain” cases, has been introduced in [95]. Several internships allowed us to implement in Scilab, C or C++, and to test such algorithms (see the work of Vishesh Dhingra [112]), or its combinaison with the resolution of linear systems by algebraic multigrid methods (internship of Shantanu Gangal in 2007). The PhD thesis work of Sylvie Detournay allowed us to develop a complete program. The program written by Sylvie Detournay (in C language) has been posted on InriaGForge. For the moment it can only be seen by members of the team.

5.3. TPLib: bibliothèque pour la manipulation de polyèdres tropicaux/TPLib: tropical polyhedra library

TPLib est une bibliothèque écrite en OCaml qui permet de manipuler des polyèdres tropicaux. Elle est distribuée sous license LGPL https://gforge.inria.fr/projects/tplib.

Cette bibliothèque implémente notamment des algorithmes permettant de passer d'une représentation externe d'un polyèdre à une représentation interne, ou inversement (voir §6.3.1 pour plus de détails). Elle permet aussi de réaliser d'autres opérations fondamentales, comme le calcul du complexe polyédral associé à un polyèdre donné (au sens de Develin et Sturmfels [108]), ou le calcul de cônes tangents tropicaux. Enfin, elle fournit toutes les primitives permettant d'utiliser les polyèdres tropicaux en tant que domaine abstrait numérique, afin de déterminer des invariants de programmes ou systèmes faisant intervenir les opérations $\min$ et $\max$ (voir [70]).

TPLib est utilisé dans le logiciel Polymake [128], développé à la Technische Universität Berlin (Allemagne). Ce dernier logiciel constitue une boite à outils permettant de manipuler des nombreux objets mathématiques (polytopes convexes, complexes polyédraux, graphes, matroïdes, polytopes tropicaux).

Le développement d'interfaces avec d'autres logiciels est désormais facilité grâce à la présence de *bindings* dans le langage C. Grâce à cela, un prototype d'interface a été réalisé entre TPLib et l'outil Verify-TAPN (https://launchpad.net/verifytapn), qui permet la vérification de réseaux de Pétri avec arcs temporisés (voir §6.6.4). De même, une interface à la bibliothèque de domaines abstraits numériques APRON [140] est également en cours de développement.

English version

TPLib is a library written in OCaml, which allows to manipulate tropical polyhedra. It is distributed under LGPL https://gforge.inria.fr/projects/tplib.

This library implements algorithms allowing to pass from an external representation of a polyhedron to an internal description, or inversely (see §6.3.1 for more details). Besides, the library allows to perform several fundamental operations over tropical polyhedra, such as computing the associated polyhedral complex (see Develin and Sturmfels [108]), or determining the tropical tangent cone at any point. Finally, it provides all the primitives allowing to use tropical polyhedra as an numerical abstract domain, in order to determine program/system invariants involving the operations $\min$ and $\max$ (see [70]).

TPLib is used in the software Polymake [128], developed in Technische Universität Berlin (Germany). Polymake is a toolbox allowing to manipulate mathematic objects such as convex polytopes, polyhedral complexes, graphs, matroids, and tropical polytopes.

The development of further interfaces is now easier thanks to the distribution of bindings in C language. Using these bindings, a prototype of interface has been created between TPLib and the model-checker VerifyTAPN (https://launchpad.net/verifytapn), which allows the verification of timed-arc Petri Nets (see §6.6.4). An interface to the numerical abstract domain APRON [140] is also under development.

MC2 Team

5. New Software and Platforms

5.1. eLYSe

Participant: Olivier Saut.

eLYse is a numerical platform used for our computations in Biology (tumor growth), micro-fluidics and complex Newtonian fluid flows. The platform is divided in two libraries : one is devoted to the modelling equations and the other one includes the numerical solvers. For example, we are able to treat (in 2D and 3D) transport equations, diffusion equations, Navier-Stokes equations, Maxwell system and the interaction fluid-structure by level-set and penalization methods. The solvers are based on finite volume methods on cartesian grids and allow parallel computations. See also the web page http://www.math.u-bordeaux1.fr/~osaut/pages/eLYSe.html.

- Version: 0.7
- ACM: ACM J.2 J.3 G.1.8 G.1.10
- AMS: AMS65Z05 35Q92
- Keywords: Modélization and numerical simulations, Finite volume methods, Level Set approach, Penalization method
- APP: En cours
- Type of human computer interaction: console
- OS/Middelware: Platform developped on Mac OS X architecture.
- Required library or software: Petsc (http://www.mcs.anl.gov/petsc/petsc-as/) Vtk (http://www.vtk.org/) Blitz++ (http://c2.com/cgi/wiki?BlitzPlusPlus) (optionnel) Boost (http://www.boost.org/)
- Programming language: C++
- Documentation: doxygen.

5.2. Kesaco

Participant: Olivier Saut.

Kesaco is a set of libraries and programs aiming at applications of mathematical modeling in clinical oncology. It features:

- A library of specialized mathematical model describing the growth of different types of cancers (secondary tumors in the lung, gliomas).
- A set of programs useful to validate mathematical models (compute the various behavior they can produce) and to build databases of numerical simulations.
- Segmentation and registration routines to use medical images directly in our numerical codes.
- Calibration methods to recover the parameters of the models using sequences of medical images. Three techniques are implemented (a genetic algorithm, a technique based on reduced order models, a sensitivity technique).

All these routines are adapted to run on a MP architecture. The webpage may be found at http://www.math.u-bordeaux1.fr/~osaut/pages/kesaco.html.

- Version: 0.2
- Keywords: Modélization and numerical simulations
- APP: En cours
- Type of human computer interaction: console
- OS/Middelware: Platform developped on Mac OS X architecture.
- Required library or software: eLYSe, Insight Toolkit (http://www.itk.org)
- Programming language: C++
- Documentation: doxygen.

5.3. NaSCar

Participant: Michel Bergmann [correspondant].

This code is devoted to solve 3D-flows in around moving and deformable bodies. The incompressible Navier-Stokes equations are solved on fixed grids, and the bodies are taken into account thanks to penalization and/or immersed boundary methods. The interface between the fluid and the bodies is tracked with a level set function or in a Lagrangian way. The numerical code is fully second order (time and space). The numerical method is based on projection schemes of Chorin-Temam's type. The code is written in C language and use Petsc (http://www.mcs.anl.gov/petsc/petsc-as/) library for the resolution of large linear systems in parallel.

NaSCar can be used to simulate both hydrodynamic bio-locomation as fish like swimming and aerodynamic flows such wake generated by a wind turbine.

- Version: 1
- Keywords: numerical analyse, fluid mechanics, langage C, PETSc
- Software benefit : simulate a flow around a deformable obstacle, moving into a fluid.
- APP: En cours
- Patent: non
- Type of human computer interaction: human for the moment
- OS/Middelware: unix, linux, mac os
- Required library or software: PETSc item Programming language: C
- Documentation: in progess

5.4. NS-MPI-2D-3D

Participants: Charles-Henri Bruneau [correspondant], Khodor Khadra.

The software NS-MPI-2D-3D is a numerical platform devoted to the computation of the incompressible flow around bodies in two or three dimensions modelled by Stokes, Navier-Stokes or Oldroyd-B equations. It is based on finite differences or finite volumes approximations on cartesian grid using the volume penalization method to handle the obstacles. The resolution is achieved by means of the multigrid method. Dirichlet, periodic or artificial boundary conditions are implemented to solve various problems in closed or open domains.

- Version: 3
- Keywords: Numerical simulation of incompressible flows,
- Type of human computer interaction: console
- OS/Middelware: unix, linux, Mac OS X item Programming language: Fortran 95 and MPI
- Documentation: included

5.5. Other MC2 codes

- Penalization techniques on cartesian grids to solve incompressible Navier-Stokes equations
  - **Vortex**: sequential, Vortex In-Cell (VIC) scheme : hybrid vortex methods based on the combination of Lagrangian mesh-free schemes and Eulerian grid based schemes on the same flow region.
  - Unstructured body fitted meshes
  - **Richards** : 2D Unstructured finite element code, implicit solver, sequential, to solve the transport-diffusion equations through a porous media including tidal forcing and mechanisms of diagenesis.
  - development inside **FluidBox** software in collaboration with **BACCHUS**. 2D-3D unstructured meshes, Stabilized Finite Elements method (SUPG), RANS turbulence model, parallel: Domain Decomposition and MPI.
- Immersed boundary techniques for:
  - **Compressible flows** : 2D-3D finite volume scheme for compressible Euler equations with solid obstacles on cartesian grids. 3D code parallelized with MPI
  - **Elliptic problems :** 2 2D-3D finite difference scheme for elliptic interface problems, parallelized with PETSc
  - Elmo. C++ Code of Finite Differences on cartesian grid parallelized with PETSC to compute the electropermeabilisation of cells in 2D and 3D.

MCTAO Project-Team

5. New Software and Platforms

5.1. Hampath

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

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

MEPHYSTO Team

5. New Software and Platforms

5.1. Platforms

5.1.1. Modulef

The numerical method to approximate the constitutive laws for rubber elasticity derived from polymer physics (as used in [15], [25]) are implemented in the Inria software Modulef (joint work of M. Vidrascu, projetc-team REO, and A. Gloria).

It is based on

- algorithms from stochastic geometry to generate suitable polymer networks;
- Delaunay tessellation algorithms to deal with steric effects (courtesy of the Inria project-team GAMMA2);
- the introduction of 1-dimensional finite elements for the polymer-chains in Modulef.

5.1.2. CMA-ES

To solve the inverse problem for the reconstruction of an explicit constitutive law from in silico experiments in [25], we relied on the Covariance Matrix Adaptation Evolution Strategy developed in the project-team TAO.

https://www.lri.fr/~hansen/cmaesintro.html

5.1.3. FreeFEM++

The numerical methods proposed in [14] for the approximation of homogenized coefficients were implemented in FreeFEM++, a user-friendly PDE-solver.

http://www.freefem.org/ff++/

MISTIS Project-Team

5. New Software and Platforms

5.1. The LOCUS software

Participant: Florence Forbes.

Joint work with: Senan Doyle (start-up creator) and Michel Dojat from Grenoble Institute of Neuroscience and Benoit Scherrer from Harvard Medical School, Boston, MA, USA.

From brain MR images, neuroradiologists are able to delineate tissues such as grey matter and structures such as Thalamus and damaged regions. This delineation is a common task for an expert but unsupervised segmentation is difficult due to a number of artefacts. The LOCUS software (http://locus.gforge.inria.fr) automatically perform this segmentation for healthy brains An image is divided into cubes on each of which a statistical model is applied. This provides a number of local treatments that are then integrated to ensure consistency at a global level, resulting in low sensitivity to artifacts. The statistical model is based on a Markovian approach that enables to capture the relations between tissues and structures, to integrate a priori anatomical knowledge and to handle local estimations and spatial correlations.

The LOCUS software has been developed in the context of a collaboration between Mistis, a computer science team (Magma, LIG) and a Neuroscience methodological team (the Neuroimaging team from Grenoble Institut of Neurosciences, INSERM). This collaboration resulted over the period 2006-2008 into the PhD thesis of B. Scherrer (advised by C. Garbay and M. Dojat) and in a number of publications. In particular, B. Scherrer received a "Young Investigator Award" at the 2008 MICCAI conference.

The originality of this work comes from the successful combination of the teams respective strengths i.e. expertise in distributed computing, in neuroimaging data processing and in statistical methods.

5.2. The P-LOCUS software

Participants: Florence Forbes, Flor Vasseur.

Joint work with: Senan Doyle (start-up creator) and Michel Dojat.

The Locus software was extended to address the delineation of lesions in pathological brains. Its extension P-LOCUS (http://p-locus.com) for lesion detection was realized by S. Doyle with financial support from GRAVIT (Grenoble Alpes Valorisation Innovation Technologies, http://www.gravit-innovation.org/) with the goal to create a Start-up. P-LOCUS software analyses, in few minutes, a 3D MR brain scan and performs fully automatic brain lesion delineation using a combined dataset of various 3D MRI sequences. Its originality comes from:

- it is fully automatic: no external user interaction and no training data required
- the possibility to combine information from several images (MR sequences)
- a statistical Bayesian framework for robustness to image artefacts and a priori knowledge incorporation
- a voxel-based clustering technique that uses Markov random fields (MRF) incorporating information about neighboring voxels for spatial consistency and robustness to imperfect image features (noise).
- the possibility to select and incorporate relevant a priori knowledge via different atlases, e.g. tissue and vascular territory atlases
- a fully integrated preprocessing steps and lesion ROI identification

P-LOCUS software was presented at various conferences and used for the BRATS Challenge on tumor segmentation organized as a satellite challenge of the Miccai conference in Nagoya, Japan. A paper published in IEEE trans. on Medical Imaging reports the challenge results [24]. Results are also shown in [47]. The software has been registered at APP in 2013 and is now undergoing industrial development for the creation of a start-up (Pixyl) expected in January 2015.

5.3. The PyHRF software

Participants: Thomas Perret, Florence Forbes, Thomas Vincent, Aina Frau Pascual.

Joint work with: Philippe Ciuciu and Solveig Badillo from Parietal Team Inria and CEA NeuroSpin, Lotfi Chaari and Laurent Risser from INP Toulouse.

As part of fMRI data analysis, the PyHRF package (http://pyhrf.org) provides a set of tools for addressing the two main issues involved in intra-subject fMRI data analysis: (i) the localization of cerebral regions that elicit evoked activity and (ii) the estimation of the activation dynamics also referenced to as the recovery of the Hemodynamic Response Function (HRF). To tackle these two problems, PyHRF implements the Joint Detection-Estimation framework (JDE) which recovers parcel-level HRFs and embeds an adaptive spatio-temporal regularization scheme of activation maps. With respect to the sole detection issue (i), the classical voxelwise GLM procedure is also available through NIPY, whereas Finite Impulse Response (FIR) and temporally regularized FIR models are implemented to deal with the HRF estimation concern (ii). Several parcellation tools are also integrated such as spatial and functional clusterings. Parcellations may be used for spatial averaging prior to FIR/RFIR analysis or to specify the spatial support of the HRF estimates in the JDE approach. These analysis procedures can be applied either to volumic data sets or to data projected onto the cortical surface. For validation purpose, this package is shipped with artificial and real fMRI data sets. To cope with the high computational needs for inference, PyHRF handles distributing computing by exploiting cluster units as well as multiple cores computers. Finally, a dedicated viewer is available which handles n-dimensional images and provides suitable features for exploring whole brain hemodynamics (display of time series, maps, ROI mask overlay). A paper in Frontiers in Neuroinformatics gives more details on the current PyHRF functionalities [26]. The 2-year engineer position of Thomas Perret is devoted to this software development.

5.4. R packages

Participants: Florence Forbes, Stéphane Girard, Gildas Mazo, Alexis Arnaud.

Joint work with: Charles Bouveyron (Univ. Paris 5) and Stéthane Dépréaux (LJK).

MISTIS is involved in the development of several R packages available on the CRAN archive. They are dedicated to the construction of copulas and to the classification and clustering of data.

- **PBC** (product of bivariate copulas). http://cran.r-project.org/web/packages/PBC/ This R package provides tools for building copulas with the PBC model, a class of multivariate copulas based on Products of Bivariate Copulas. Copulas are a useful tool to model multivariate distributions. While there exist various families of bivariate copulas, much fewer has been done when the dimension is higher. To this aim an interesting class of copulas based on products of transformed copulas has been proposed. However the use of this class for practical high dimensional problems remains challenging. Constraints on the parameters and the product form render inference, and in particular the likelihood computation, difficult. In this R package, we propose a new class of high dimensional copulas based on a product of transformed bivariate copulas. No constraints on the parameters refrain the applicability of the proposed class which is well suited for applications in high dimension. Furthermore the analytic forms of the copulas within this class allow to associate a natural graphical structure (see illustration below) which helps to visualize the dependencies and to compute the likelihood efficiently even in high dimension.

- **FDG** (one-Factor copulas with Durante Generators). http://cran.r-project.org/web/packages/FDGcopulas/ This R package provides tools for building high-dimensional copulas with the FDG model, a class of multivariate copulas based on one-factor copulas. FDG copulas are a class of copulas featuring an interesting balance between flexibility and tractability. This package provides tools to construct, calculate the pairwise dependence coefficients of, simulate from, and fit FDG copulas. The acronym FDG stands for 'one-Factor with Durante Generators', as an FDG copula is a one-factor copula - that is, the variables are independent given a latent factor - whose linking copulas belong to the Durante class of bivariate copulas (also referred to as exchangeable Marshall-Olkin or semilinear copulas).
- **HDclassif** (classification and clustering methods for high dimensional data). http://cran.r-project.org/web/packages/HDclassif/ The HDclassif package is devoted to the clustering and the discriminant analysis of high-dimensional data. The classification methods proposed in the package result from a new parametrization of the Gaussian mixture model which combines the idea of dimension reduction and model constraints on the covariance matrices. The supervised classification method using this parametrization has been called High Dimensional Discriminant Analysis (HDDA). In a similar manner, the associated clustering method has been called High Dimensional Data Clustering (HDDC) and uses the Expectation-Maximization (EM) algorithm for inference. In order to correctly fit the data, both methods estimate the specific subspace and the intrinsic dimension of the groups. Due to the constraints on the covariance matrices, the number of parameters to estimate is significantly lower than other model-based methods and this allows the methods to be stable and efficient in high-dimensional spaces. Experiments on artificial and real datasets show that HDDC and HDDA perform better than existing classical methods on high-dimensional datasets, even with small datasets.
- **robustDA** (robust mixture discriminant analysis). http://cran.r-project.org/web/packages/robustDA/ Robust mixture discriminant analysis allows to build a robust supervised classifier from learning data with label noise. The idea of the proposed method is to confront an unsupervised modeling of the data with the supervised information carried by the labels of the learning data in order to detect inconsistencies. The method is able afterward to build a robust classifier taking into account the detected inconsistencies into the labels. An application to object recognition under weak supervision is presented below.
- **MSST** (Mixtures of multiple scaled Student distributions). The package is not yet available on the CRAN but should be early 2015. It implements more efficiently the models and inference procedures described in [21] and will be used on large data sets of brain MRI in the context of Alexis Arnaud PhD thesis. This is joint work with S. Dépréaux who helped with writing subroutines in C++.

MODAL Project-Team

5. New Software and Platforms

5.1. BlockCluster package for co-clustering

Participants: Serge Iovleff, Vincent Kubicki.

BlockCluster is an R package on top of the coclust C++ library. Maintenance of the CRAN package (http://cran.r-project.org/web/packages/blockcluster/index.html) and user assistance on the forum have been ensured.

5.2. clere package for high dimensional regression

Participants: Christophe Biernacki, Loïc Yengo, Julien Jacques.

The clere package for R proposes variable clustering in high dimensional linear regression. It is available on CRAN (http://cran.r-project.org/web/packages/clere/index.html) and now submitted to an international journal dedicated to software [52].

5.3. Clustericat package for correlated categorical variable

Participants: Christophe Biernacki, Matthieu Marbac-Lourdelle, Vincent Vandewalle.

Clustericat is an R package for model-based clustering of categorical data. In this package, the Conditional Correlated Model (CCM), published in 2014 [24], takes into account the main conditional dependencies between variables through extreme dependence situations (independence and deterministic dependence). Clustericat performs the model selection and provides the best model according to the BIC criterion and the maximum likelihood estimates. It is available online on Rforge (https://r-forge.r-project.org/R/?group_id=1803).

5.4. CoModes package for correlated categorical variables

Participants: Christophe Biernacki, Matthieu Marbac-Lourdelle, Vincent Vandewalle.

CoModes is another R package for model-based clustering of categorical data. In this package, the Conditional Modes Model (CMM), submitted for publication in 2014 [49], takes into account the main conditional dependencies between variables through particular modality crossings (so-called modes). CoModes performs the model selection and provides the best model according to the exact integrated likelihood criterion and the maximum likelihood estimates. It is available online on Rforge (https://r-forge.r-project.org/R/?group_id=1809).

5.5. CorReg package for correlated variables in regression

Participants: Christophe Biernacki, Clément Théry.

The main idea of the CorReg package is to consider some form of sub-regression models, some variables defining others. We can then remove temporarily some of the variables to overcome ill-conditioned matrices inherent in linear regression and then reinject the deleted information, based on the structure that links the variables. The final model therefore takes into account all the variables but without suffering from the consequences of correlations between variables or high dimension. The CorReg package is now available on CRAN (http://cran.r-project.org/web/packages/CorReg/index.html) and graphical functionalities have been added in 2014. It has been presented to a conference [32] and is currently written as a research paper [51]. It is a joint work with Gaétan Loridant.

5.6. HDPenReg package for penalized regressions in high dimension

Participants: Quentin Grimonprez, Serge Iovleff.

HDPenReg is an R-package based on a C++ code dedicated to the estimation of regression model with l1-penalization. It is now available on CRAN (http://cran.r-project.org/web/packages/HDPenReg/index.html). More cross-validation options were added. Maintenance in 2014 concerned bugs correction and documentation updates.

5.7. FunFEM package for functional data

Participant: Julien Jacques.

FunFEM package for R proposes a clustering tool for functional data. The model-based algorithm clusters the functional data into discriminative subspaces. It is available on CRAN (http://cran.r-project.org/web/packages/funFEM/index.html).

5.8. FunHDDC package for functional data

Participant: Julien Jacques.

FunHDDC package for R proposes a clustering tool for functional data. The model-based clustering algorithm considers that functional data live in cluster-specific subspaces. It is available on CRAN (http://cran.r-project.org/web/packages/funHDDC/index.html).

5.9. Galaxy-Modal platform

Participants: Samuel Blanck, Guillemette Marot.

Galaxy is an open, web-based platform for data intensive biomedical research. Galaxy features user friendly interface, workflow management, sharing functionalities and is widely used in the biologist community. The MPAgenomics R package developped by MODAL has been integrated into Galaxy, and the Galaxy-Modal instance has been publicly deployed thanks to the IFB-cloud infrastruture (http://cloud.france-bioinformatique.fr/). An APP repository with Galaxy-Modal source code has been created (reference : Galaxy - MPAgenomics)

5.10. metaMA package for meta-analysis of microarray data

Participant: Guillemette Marot.

metaMA is a specialised software for microarrays. It is an R package which combines either p-values or modified effect sizes from different studies to find differentially expressed genes. The main competitor of metaMA is geneMeta. Compared to geneMeta, metaMA offers an improvement for small sample size datasets since the corresponding modelling is based on shrinkage approaches.

This software is routinely used by biologists from INRA, Jouy en Josas (it has been included in a local analysis pipeline) but its diffusion on the CRAN (http://cran.r-project.org/web/packages/metaMA/index.html) makes it available to a wider community, as attested by the citations of publications related to the methods implemented in the software.

Maintenance in 2014 concerned documentation updates and users assistance.

5.11. metaRNASeq package for meta-analysis of RNA-Seq data

Participant: Guillemette Marot.

This is joint work with Andrea Rau (INRA, Jouy-en-Josas). metaRNASeq is a specialised software for RNA-seq experiments. It is an R package which is an adaptation of the metaMA package presented previously. Both implement the same kind of methods but specificities of the two types of technologies require some adaptations to each one. metaRNASeq is now available on CRAN (http://cran.r-project.org/web/packages/metaRNASeq/index.html).

5.12. MixCluster package for correlated mixed variables

Participants: Christophe Biernacki, Matthieu Marbac-Lourdelle, Vincent Vandewalle.

MixCluster is an R package for model-based clustering of mixed data (continuous, binary, integer). In this package, the model, submitted for publication in 2014 [48], takes into account the main conditional dependencies between variables through Gaussian copula. Mixcluster performs the model selection and provides the best model according to Bayesian approaches. It is available online on Rforge (https://r-forge.r-project.org/R/?group_id=1939).

5.13. MIXMOD and Rmixmod package for mixed data

Participants: Vincent Kubicki, Christophe Biernacki, Serge Iovleff.

MIXMOD (MIXture MODelling) is an important software for the MODAL team since it concerns its main topics: model-based supervised, unsupervised and semi-supervised classification for various data situations. MIXMOD is now a well-distributed software with over 250 downloads/month are recorded for several years. MIXMOD is written in C++ (more than 10 000 lines) and distributed under GNU General Public License. Several other institutions participate in the MIXMOD development since several years: CNRS, Inria Saclay-Île de France, Université de Franche-Comté, Université Lille 1. The software already benefits from several APP depositions and an R package (Rmixmod) has been associated to MIXMOD in 2012. In 2014, it has led to publication in an international journal dedicated to software [23].

5.14. MixtComp package for full mixed data

Participants: Vincent Kubicki, Christophe Biernacki, Serge Iovleff.

MixtComp (Mixture Computation) is another important software fot the MODAL team since it concerns model-based clustering for mixed data. Main difference with the MIXMOD/Rmixmod software is that MixtComp's architecture is able to integrate easily and quickly all new univariate models, under the conditional independence assumption, that will be sequentially available from researches of the MODAL team or others. Currently, central architecture of MixtComp is built and three models (Gaussian, multinomial, Poisson) are implemented with ability to natively manage missing data (completely or by interval). MixtComp stands both as a C++ library and an R package. The code is currently developed internally, and has been field-tested through two contracted partnerships.

5.15. MPAgenomics package for multi-patient analysis of genomic markers

Participants: Quentin Grimonprez, Guillemette Marot, Alain Celisse.

MPAgenomics is an R package for multi-patients analysis of genomics markers. It enables to study several copy number and SNP data profiles at the same time. It offers wrappers from commonly used packages to offer a pipeline for beginners in R. It also proposes a special way of choosing some crucial parameters to change some default values which were not adapted in the original packages. For multi-patients analysis, it wraps some penalized regression methods implemented in HDPenReg.

MPAgenomics is now available on CRAN (http://cran.r-project.org/web/packages/MPAgenomics/index.html). New segmentation methods were added to MPAgenomics. Maintenance in 2014 concerned bugs correction, documentation updates and code factorization.

5.16. RankCluster package to cluster ranking data

Participants: Christophe Biernacki, Quentin Grimonprez, Julien Jacques.

Rankcluster package for R proposes a clustering tool for ranking data. Multivariate and partial rankings can be also taken into account. It is available on CRAN (http://cran.r-project.org/web/packages/Rankcluster/index.html).

Rankcluster now supports tied ranking data. Maintenance in 2014 concerned bugs correction, documentation updates and addition of parallelism.

5.17. rtkpp package: STK++ Integration To R Using Rcpp

Participant: Serge Iovleff.

rtkpp is the integration of the library STK++ (see 5.18) into R. It is using Rcpp. Some functionalities of the Clustering project provided by the library are available in the R environment as R functions. The rtkpp package includes the header files from the STK++ library (currently version 0.8.2). Thus users do not need to install STK++ itself in order to use it. rtkpp is licensed under the GNU GPL version 2 or later and available on CRAN (http://cran.r-project.org/web/packages/rtkpp/index.html).

5.18. STK++ release 0.8.4: The Statistical ToolKit

Participant: Serge Iovleff.

STK++ is a versatile, fast, reliable and elegant collection of C++ classes for statistics, clustering, linear algebra, arrays (with an API Eigen-like), regression, dimension reduction, etc. STK++ is licensed under the GNU LGPL version 2 or later. See: http://www.stkpp.org/

MOKAPLAN Team

5. New Software and Platforms

5.1. ALG2 for Monge Mean-Field Games, Monge problem and Variational problems under divergence constraint

5.1.1. Platforms

A generalisation of the ALG2 algorithm [53] corresponding to the paper [18] ha been implemented in FreeFem++. The scripts and numerical simulations are available at https://team.inria.fr/mokaplan/augmented-lagrangian-simulations/.

We still plan to implement a parallel version on Rocquencourt Inria cluster. We are waiting for FreeFem to be installed on the cluster.

5.2. Mokabajour

5.2.1. Platforms

Following the pioneering work of Caffarelli and Oliker [42], Wang [85] has shown that the inverse problem of freeforming a *convex* reflector which sends a prescribed source to a target intensity is a particular instance of Optimal Mass Transportation. The method developed in [7] has been used by researchers of TU Eindhoven in collaboration with Philips Lightning Labs to compute reflectors [81] in a simplified setting. The industrial motivation is the automatic design of reflector given prescribed source and target illuminance. From the mathematical point of view there is a hierarchy of Optimal Mass Transportation reflector and lenses problems and only the simplest "far field" one can be solved with state of the art Monge-Ampère solvers. We will adapt the Monge-Ampère solvers and also attempt to build real optimized reflector prototypes. We plan on investigating the more complicated near field models and design numerical methods. Finally Monge-Ampère based Optimal Mass Transportation solvers will be made available. This could be used for example in Mesh adaptation.

The web site is under construction https://project.inria.fr/mokabajour/, preliminary results are available.

This ADT (Simon Legrand) on the numerical free forming of specular reflectors started in december. We implement different types of MA solvers in collaboration with Quentin Mérigot (CEREMADE), Boris Thibert (LJK Grenoble) and Vincent Duval. See https://project.inria.fr/mokabajour/.

NACHOS Project-Team

5. New Software and Platforms

5.1. MAXW-DGTD

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

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

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

5.2. MAXW-DGFD

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

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

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

5.3. SISMO-DGTD

Participants: Nathalie Glinsky, Stéphane Lanteri [correspondant].

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

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

NANO-D Project-Team

4. New Software and Platforms

4.1. SAMSON

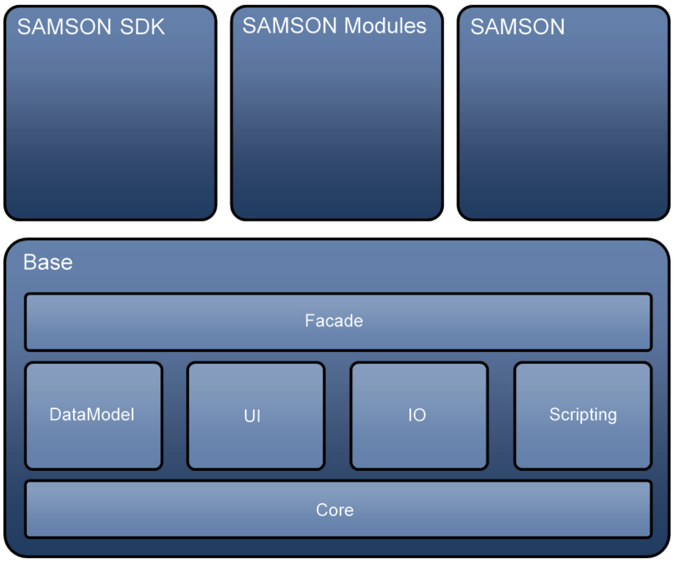


Figure 3. SAMSON's architecture.

A major objective of NANO-D is to try and integrate a variety of adaptive algorithms into a unified framework. As a result, NANO-D is developing SAMSON (Software for Adaptive Modeling and Simulation Of Nanosystems), a software platform aimed at including all developments from the group, in particular those described below.

The objective is to make SAMSON a generic application for computer-aided design of nanosystems, similar to existing applications for macrosystem prototyping (CATIA, SolidWorks, etc.).

The current architecture of SAMSON is visible in Figure 3 . The code is organized into four main parts: a) the Base (in which "Core" contains, in particular, the heart of the adaptive algorithms: signaling mechanisms specifically designed for SAMSON), b) the Software Development Kit (SDK: a subset of the base that will be provided to module developers), c) Modules, and d) the SAMSON application itself.

Similar to the concept of Mathematica *toolboxes*, for example, the goal has been to make it possible to personalize the user interface of SAMSON for potentially many distinct applications. For example, we may want to personalize the interface of SAMSON for crystallography, drug design, protein folding, electronics, material science, nano-engineering, etc., by loading different modules at startup, depending on the user application domain.

NECS Project-Team

5. New Software and Platforms

5.1. GTL – Grenoble Traffic Lab

Participants: C. Canudas de Wit [contact person], I. Bellicot, P. Bellemain, L. Leon Ojeda, D. Pisarski, A. Kibangou, H. Fourati, F. Morbidi, F. Garin, A. Ladino Lopez, P. Grandinetti, E. Lovisari, R. Singhal, A. Andreev.

The Grenoble Traffic Lab (GTL) initiative, led by the NECS team, is a real-time traffic data Center (platform) that collects traffic road infrastructure information in real-time with minimum latency and fast sampling periods. The main elements of the GTL are: a real-time data-base, a show room, and a calibrated micro-simulator of the Grenoble South Ring. Sensed information comes from a dense wireless sensor network deployed on Grenoble South Ring, providing macroscopic traffic signals such as flows, velocities, densities, and magnetic signatures. This sensor network was set in place in collaboration with Inria spin-off Karrus-ITS, local traffic authorities (DIR-CE, CG38, La Metro), and specialized traffic research centers. In addition to real data, the project also uses simulated data, in order to validate models and to test the ramp-metering; the micro-simulator is a commercial software (developed by TSS AIMSUN ©).

More details at http://necs.inrialpes.fr/pages/grenoble-traffic-lab.php

5.2. Source-seeking robot

Participants: R. Fabbiano [contact person], J. Dumon, Y. Gaudfrin.

The source-seeking algorithms developed in the thesis of Ruggero Fabbiano have been implemented in hardware, with a wheeled robot performing 2-dimensional search. The considered scenario is a source of pollutant in the ocean, where the pollutant can be detected thanks to the fact that it is warmer than water, so that data from an infra-red camera can be used by one or multiple helicopters to move along the ocean surface towards the source. In our experimental equipment, the 2-dimensional movement has been performed with a wheeled vehicle, and the camera was a regular camera, taking pictures of a color-coded image from an actual infra-red image of a pollutant leak. Videos of the experiments are available online: http://necs.inrialpes.fr/pages/platforms.php

NON-A Project-Team

5. New Software and Platforms

5.1. SLIM

Multi-robots cooperation can be found as an application in many domains of science and technology: manufacturing, medical robotics, personal assistance, military/security and spatial robots. The market of robots is quickly developing and its capacity is continuously growing. Concerning cooperation of mobile multi-robots, 3 key issues have to be studied: Localization, path planning and robust control, for which Non-A team has worked and proposed new algorithms. Due to the ADT SLIM, we implement our algorithms (localization, path planning and robust control) and integrate them into ROS (Robotic Operating System) as a package, named SLIM.

5.2. Blimp

Scientific research and development on the control of autonomous airship have shown a significant growth in recent years. New applications appear in the areas such as freight carrier, advertising, monitoring, surveillance, transportation, military and scientific research. The control of autonomous airship is a very important problem for the aerial robots research.

The development of Blimp by Non-A is used for experimentation and demonstration of controlling algorithms. The blimp is required to provide some environment information and status of itself, such as surveillance video of surrounding environment, gesture of blimp, altitude of blimp. With these basic information, one could localize blimp with certain algorithm (visual SLAM for example) or implement one controller in order to improve the stability and maneuverability of blimp.

OPALE Project-Team

5. New Software and Platforms

5.1. NUM3SIS

Participant: Régis Duvigneau [correspondant].

The Opale project-team has initiated a few years ago the development of NUM3SIS (http://num3sis.inria.fr), which is is a modular platform devoted to scientific computing and numerical simulation. It is not restricted to a particular application field, but is designed to host complex multidisciplinary simulations. Main application fields are currently Computational Fluid Dynamics (by Opale project-team), Computational Electro-Magnetics (by Nachos project-team) and pedestrian traffic simulation (by Opale project-team). Some components of the platform are also used by the Tosca project-team for CO2 market simulation and wind simulation in collaboration with Ciric (Inria-Chile).
NUM3SIS provides innovative software tools to overcome some limitations encountered by classical monolithic simulation codes. In particular, the platform is based on abstract concepts commonly used in scientific computing, such as mesh, fields, finite-elements, linear solvers etc, that can be implemented in plugins. A fast prototyping of algorithms can be achieved using a visual programing interface. A component is dedicated to deployment on parallel architectures. Moreover, the platform relies on a "store" system to foster exchange of plugins, scripts or data.
This work is being carried out with the support of one engineer in the framework of an ADT (Action de Développement Technologique) program.

5.2. FAMOSA

Participant: Régis Duvigneau [correspondant].

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

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

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

5.3. Plugins for AXEL

Participant: Régis Duvigneau [correspondant].

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

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

5.4. Integration platform for multidiscipline optimization applications

Participants: Toan Nguyen, Laurentiu Trifan.

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

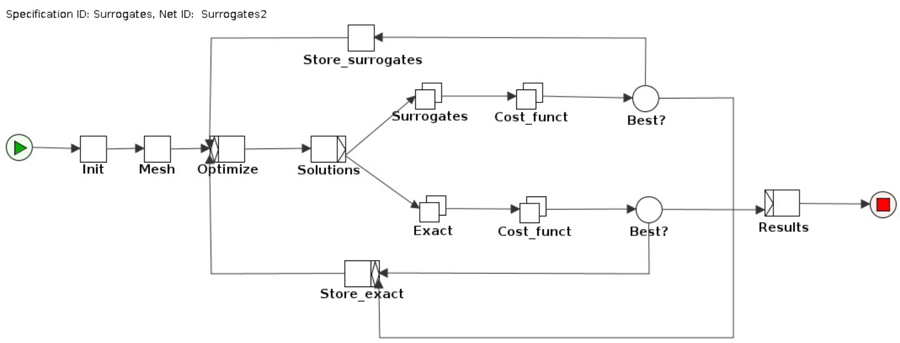


Figure 1. Testcase deployment on the Grid5000 infrastructure.

POEMS Project-Team

5. New Software and Platforms

5.1. Software

5.1.1. XLiFE++

Participants: Eric Lunéville, Nicolas Kielbasiewicz, Colin Chambeyron.

XLIFE++ is a Finite Element library in C++ based on philosophy of the previous library MELINA in Fortran but with new capabilities (boundary elements, discontinuous Galerkin methods, more integrated tools -in particular mesh tools - and high performance computing skills, multithread and GPU computation). It is licensed under LGPL and developed in the context of the European project SIMPOSIUM (FP7/ICT, leader CEA/LIST, from september 2011 to august 2014). There are also academic partners: IRMAR, University of Rennes and LAMA, University of Marne-la-Vallée.

After 3 years of work, the development of the finite element library XLIFE++ reached a milestone in 2014 with the first downloadable public release, after an important effort to improve the user interface and to complete the last major developments necessary for this output: essential boundary conditions, mesh construction, Dirichlet-to-Neumann maps ... among others. In June 2014, a day was organized to present to a wider audience the features of this library. We now provide support to the users (patches, new developments...).

5.1.2. COFFEE

Participant: Stéphanie Chaillat.

COFFEE is a 3D BEM-accelerated FMM solver for linear elastodynamics (full implementation, 30 000 lines of Fortran 90). The 3-D elastodynamic equations are solved with the boundary element method accelerated by the multi-level fast multipole method. The fundamental solutions for the infinite space are used in this implementation. A boundary element-boundary element coupling strategy is also implemented so multi-region problems (strata inside a valley for example) can be solved.

QUANTIC Team (section vide)

REALOPT Project-Team

5. New Software and Platforms

5.1. BaPCod – a generic Branch-and-Price Code

Participants: Issam Tahiri [Software Engineer], François Clautiaux, Boris Detienne, Pierre Pesneau, Ruslan Sadykov, François Vanderbeck [correspondant].

BaPCod is a prototype code that solves Mixed Integer Programs (MIP) by application of decomposition and reformulation approach (relying mostly on Dantzig-Wolfe reformulation techniques). The reformulated problem is solved using a branch-and-price-and-cut (column and cut generation) algorithm. This software platform, made of C++ classes, offers a "*black-box*" implementation that does not require user input and is not application specific. The features are
(i) A modeling language to express a compact integer linear programming model of the application on hand.
(ii) the automation of the Dantzig-Wolfe reformulation process. The user can provide subproblem solvers if available, but he does not need to explicitly define the reformulation, the explicit form of the columns, their reduced cost, or the Lagrangian bounds.
(iii) a default column generation procedure with standard initialization and stabilization [1], [56] [88] [87] [77] and
(iv) a default branching scheme that is generic to all applications [7],
(v) default primal heuristics specially developed for use in a decomposition framework [61], [76], [89].
The prototype software was/is used as background solver in our application studies and local PhD thesis. It also serves as the framework for our comparative study in a Inria assocaited team project and our transfert projects (the prototype enables us to be very responsive in our industrial contact).

See also the web page https://wiki.bordeaux.inria.fr/realopt/pmwiki.php/Project/BaPCod.

REGULARITY Project-Team

5. New Software and Platforms

5.1. FracLab

Participant: Jacques Lévy Véhel [correspondant].

FracLab was developed for two main purposes:

1. propose a general platform allowing research teams to avoid the need to re-code basic and advanced techniques in the processing of signals based on (local) regularity.
2. provide state of the art algorithms allowing both to disseminate new methods in this area and to compare results on a common basis.

FracLab is a general purpose signal and image processing toolbox based on fractal, multifractal and local regularity methods. FracLab can be approached from two different perspectives:

- (multi-) fractal and local regularity analysis: A large number of procedures allow to compute various quantities associated with 1D or 2D signals, such as dimensions, Hölder and 2-microlocal exponents or multifractal spectra.
- Signal/Image processing: Alternatively, one can use FracLab directly to perform many basic tasks in signal processing, including estimation, detection, denoising, modeling, segmentation, classification, and synthesis.

A graphical interface makes FracLab easy to use and intuitive. In addition, various wavelet-related tools are available in FracLab.

FracLab is a free software. It mainly consists of routines developed in MatLab or C-code interfaced with MatLab. It runs under Linux, MacOS and Windows environments. In addition, a "stand-alone" version (*i.e.* which does not require MatLab to run) is available.

Fraclab has been downloaded several thousands of times in the last years by users all around the world. A few dozens laboratories seem to use it regularly, with more than four hundreds registered users. Our ambition is to make it the standard in fractal softwares for signal and image processing applications. We have signs that this is starting to become the case. To date, its use has been acknowledged in roughly three hundreds and fifty research papers in various areas such as astrophysics, chemical engineering, financial modeling, fluid dynamics, internet and road traffic analysis, image and signal processing, geophysics, biomedical applications, computer science, as well as in mathematical studies in analysis and statistics (see http://fraclab.saclay.inria.fr/ for a partial list with papers). In addition, we have opened the development of FracLab so that other teams worldwide may contribute. Additions have been made by groups in Australia, England, France, the USA, and Serbia.

SELECT Project-Team

5. New Software and Platforms

5.1. MIXMOD software

Participants: Gilles Celeux [Correspondant], Erwan Le Pennec, Benjamin Auder.

Mixture model, cluster analysis, discriminant analysis

MIXMOD is being developed in collaboration with Christophe Biernacki, Florent Langrognet (Université de Franche-Comté) and Gérard Govaert (Université de Technologie de Compiègne). MIXMOD (MIXture MODelling) software fits mixture models to a given data set with either a clustering or a discriminant analysis purpose. MIXMOD uses a large variety of algorithms to estimate mixture parameters, e.g., EM, Classification EM, and Stochastic EM. They can be combined to create different strategies that lead to a sensible maximum of the likelihood (or completed likelihood) function. Moreover, different information criteria for choosing a parsimonious model, e.g. the number of mixture component, some of them favoring either a cluster analysis or a discriminant analysis view point, are included. Many Gaussian models for continuous variables and multinomial models for discrete variable are available. Written in C++, MIXMOD is interfaced with MATLAB. The software, the statistical documentation and also the user guide are available on the Internet at the following address: http://www.mixmod.org.

Since 2010, MIXMOD has a proper graphical user interface. A version of MIXMOD in R is now available http://cran.r-project.org/web/packages/Rmixmod/index.html.

Erwan Le Pennec with the help of Serge Cohen has proposed a spatial extension in which the mixture weights can vary spatially.

Benjamin Auder contributes to the informatics improvement of MIXMOD. He implemented an interface to test any mathematical library (Armadillo, Eigen, ...) to replace NEWMAT. He contributed to the continuous integration setup using Jenkins tool and prepared an automated testing framework for unit and non-regression tests.

This year, it has been decide to create MIXMODSTORE which proposes companion programs of MIXMOD. As a matter of fact, the program MixmodCombi of Jean-Patrick Baudry (Université Paris 6) and Gilles Celeux which allows a hierarchical clustering derived from a mixture has been associated to Rmixmod.

5.2. BLOCKCLUSTER software

Participants: Vincent Brault, Gilles Celeux, Christine Keribin.

Mixture model, Block cluster analysis,

Blockcluster is a software devoted on model-based block clustering. It is developed by MODAL team (Inria Lille). With Parmeet Bathia (Inria Lille), Vincent Brault has added a Bayesian point of view for the binary, categorial and continuous datas with the variational Bayes algorithm. It ha been enriched by a full Bayesian version using a Gibbs sampler. This Gibbs sampler coulpled with the variaotional Bayes algorithm provides solutions more stable and less dependent of the starting values of the algorithm. An exact expression of criterion ICL has been provided. This criterion or BIC are used for selecting a relevant block clustering.

SEQUEL Project-Team

5. New Software and Platforms

5.1. Computer Games

Participant: Rémi Coulom.

- ***Crazy Stone*** is a top-level Go-playing program that has been developed by Rémi Coulom since 2005. Crazy Stone won several major international Go tournaments in the past. In 2013, a new version was released in Japan. This new version won the 6th edition of the UEC Cup (the most important international computer-Go tournament). It also won the first edition of the Denseisen, by winning a 4-stone handicap game against 9-dan professional player Yoshio Ishida. It is distributed as a commercial product by *Unbalance Corporation* (Japan). 6-month work in 2013. URL: http://remi.coulom.free.fr/CrazyStone/
- ***Kifu Snap*** is an Android image-recognition app. It can automatically recognize a Go board from a picture, and analyze it with Crazy Stone. It was released on Google Play in November, 2013. 6-month work in 2013. URL: http://remi.coulom.free.fr/kifu-snap/

5.2. Function optimization

Participant: Philippe Preux.

5.2.1. yaStoSOO

We have worked on the efficient implementation of the StoSOO algorithm in order to have a software that can be used for real to optimize real functions, and to be able to experiment with the algorithm, and assess its practical usefulness. This led to yaStoSOO, an implementation in C available on the web at http://www.grappa.univ-lille3.fr/~ppreux/software/StoSOO/. The code is distributed under the GPL licence.

Thanks to this implementation, we were able to compete in the CEC'2014 competition on Real-Parameter Single Objective optimization at which we ranked honorably (10th out of 17 competitor algorithms). More experimental work is under-way.

SIERRA Project-Team

4. New Software and Platforms

4.1. SAG

Participant: Mark Schmidt [correspondent].

SAG: Minimizing Finite Sums with the Stochastic Average Gradient.

The SAG code contains C implements (via Matlab mex files) of the stochastic average gradient (SAG) method detailed below, as well as several related methods, for the problem of L2-regularized logistic regression with a finite training set.

The specific methods available in the package are: SGD: The stochastic gradient method with (user-supplied) step-sizes, (optional) projection step, and (optional) (weighted-)averaging. ASGD: A variant of the above code that supports less features, but efficiently implements uniform averaging on sparse data sets. PCD: A basic primal coordinate descent method with step sizes set according the (user-supplied) Lipschitz constants. DCA: A dual coordinate ascent method with a numerical high-accuracy line-search. SAG: The stochastic average gradient method with a (user-supplied) constant step size. SAGlineSearch: The stochastic average gradient method with the line-search described in the paper. SAG-LipschitzLS: The stochastic average gradient method with the line-search and adaptive non-uniform sampling strategy described in the paper.

4.2. fMRI

Participant: Fabian Pedregosa [correspondent].

We showed that HRF estimation improves sensitivity of fMRI encoding and decoding models and propose a new approach for the estimation of Hemodynamic Response Functions from fMRI data. This is an implementation of the methods described in the paper.

TAO Project-Team

5. New Software and Platforms

5.1. METIS

Participants: Olivier Teytaud [correspondent], Jérémie Decock, Jean-Joseph Christophe, Vincent Berthier, Marie-Liesse Cauwet, Jialin Liu, Sandra Cecilia Astete Morales.

Keywords: Energy, Optimization, Planning.

Many works in Energy Optimization, in particular in the case of high-scale sequential decision making, are based on one software per application, because optimizing the software eventually implies losing generality. Our goal is to develop with Artelys a platform, METIS, which can be used for several applications. In 2012 we interfaced existing codes in Artelys and codes developed in the TAO team; experiments have been performed and test cases have been designed. A main further work is the introduction of generic tools for stochastic dynamic programming into the platform, for comparison and hybridization with other tools from the UCT-SIG.

Our favorite challenge is the hybridization of "classical" tools (based on constraint satisfaction problems, or mixed integer linear programming or mixed integer quadratic programming), which are fast and accurate, with non-linear solvers which can take care of a sophisticated (non-linear) models.

Metis is the Artelys/Tao contribution to Crystal, which is at the heart of the Post project, which is selected by the European Commission for a 4-years project for energy modeling http://www.artelys.com/news/120/90/Energy-The-European-Commission-Chooses-Artelys-Crystal.

5.2. MoGo

Participants: Olivier Teytaud [correspondent], Jean-Baptiste Hoock.

Keywords:

MoGo and its Franco-Taiwanese counterpart MoGoTW is a Monte-Carlo Tree Search program for the game of Go, which made several milestones of computer-Go in the past (first wins against professional players in 19x19; first win with disadvantageous side in 9x9 Go). Recent results include 7 wins out of 12 against professional players (in Brisbane, 2012) in 7x7, and recently an optimization of the random seed which brings a significant improvement in Go and (unpublished) on the difficult case of phantom-Go. However, the work in the UCT-SIG has now shifted to energy management.

5.3. CMA-ES: Covariance Matrix Adaptation Evolution Strategy

Participants: Emmanuel Benazera, Nikolaus Hansen [correspondent].

Keywords: Evolutionary Computation, Stochastic Optimization, Real-parameter Optimization.

The Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [65] is considered to be state-of-the-art in continuous domain evolutionary computation [64], and in stochastic optimization at large. It has been shown to be highly competitive on different problem classes even with deterministic continuous algorithms using numerically computed gradients (see the results published on COCO platform). The algorithm is widely used in research and industry as witnessed by hundreds of published applications. We provide source code for the CMA-ES in C, C++11, Java, Matlab, Octave, Python, and Scilab including the latest variants of the algorithm.

Link: http://www.lri.fr/~hansen/cmaes_inmatlab.html

5.4. COmparing Continuous Optimizers

Participants: Nikolaus Hansen [correspondent], Anne Auger, Marc Schoenauer, Ouassim Ait Elhara, Asma Atamna.

Keywords: Evolutionary Computation, Stochastic Optimization, Real-parameter Optimization, Benchmarking, Derivative Free Optimization.

COCO (COmparing Continuous Optimizers) is a platform for systematic and sound comparisons of real-parameter global optimizers. COCO provides benchmark function testbeds (noiseless and noisy) and tools for processing and visualizing data generated by one or several optimizers. The code for processing experiments is provided in Matlab, C, Java, and Python. The post-processing code is provided in Python. The code is under continuous developement and has been used for the GECCO 2009, 2010, 2012, and 2013 workshops on "Black Box Optimization Benchmarking" (BBOB) (see Section 6.3). It is now undergoing major changes thanks to the ANR project NumBBO that will add constraint handling and multi-objective benchmarks to the existing platform.

Link: http://coco.gforge.inria.fr/ and http://numbbo.gforge.inria.fr/

5.5. MultiBoost

Participant: Balázs Kégl [correspondent].

Keywords: Multi-class, Multi-label Classification.

The MultiBoost package [63] provides a fast C++ implementation of multi-class/multi-label/multi-task boosting algorithms. It is based on AdaBoost.MH but it also implements popular cascade classifiers, Arc-GV, and FilterBoost. The package contains common multi-class base learners (stumps, trees, products, Haar filters). Further base learners and strong learners following the boosting paradigm can be easily implemented in a flexible framework.

Link: http://multiboost.org

5.6. Grid Observatory

Participants: Cécile Germain-Renaud [correspondent], Julien Nauroy, Michèle Sebag.

Keywords: Autonomic Computing, Green Computing.

The Grid Observatory (GO) software suite collects and publishes traces of the EGI (European Grid Initiative) grid usage. With the release and extensions of its portal, the Grid Observatory has made a database of grid usage traces available to the wider computer science community since 2008. These data are stored on the grid, and made accessible through a web portal without the need of grid credentials. The GO is fully integrated with the evolution of EGI monitoring. More than 250 users are currently registered. The acquisition has been extended to the University cloud StratusLab hosted by the VirtualData center.

The Green Computing Observatory (GCO) monitors the VirtualData center; it collects data on energy consumption and publishes the data through the Grid Observatory. These data include the detailed monitoring of the processors and motherboards, as well as global site information. The first results on energy saving opportunities have been presented at the Green Days@Luxembourg meeting.

In order to make the GO data readily consistent and complete, as well as understandable for further exploitation, an original approach has been designed, based on a flexible data schema built in collaboration with the users [27]. Its implementation is developed within the FUI project TIMCO.

Link: http://grid-observatory.org

5.7. Platforms

5.7.1. io.datascience

This Data as a Service (DaaS) platform is developed in the context of the Center for Data Science and the TIMCO project. Its overall goals is to exploit the advances in semantic web techniques for efficient sharing and usage of scientific data. A related specific software is the Tester for Triplestore (TFT) software suite, which benchmarks the compliance of sparql databases wrt the RDF standard and publishes the results through the SparqlScore service. TFT has been selected for the Semantic Web Challenge [42].

Links: https://io.datascience-paris-saclay.fr, https://github.com/BorderCloud/TFT, http://sparqlscore.com

TOSCA Project-Team

5. New Software and Platforms

5.1. Triton

Participant: Antoine Lejay [correspondant].

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

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

- Version: 1.0

5.2. SDM

Participants: Mireille Bossy [correspondant], Sélim Karia.

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

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

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

- Version: 2.0

5.3. CarbonQuant

Participants: Mireille Bossy [correspondant], Sélim Karia.

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

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

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

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

- Version: 2.0

ABS Project-Team

4. New Software and Platforms

4.1. Software

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

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

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

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

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

- `SBL-APPLICATIONS`**:** end-user applications solving specific applied problems.
- `SBL-CORE`**:** low-level generic C++ classes templated by traits classes specifying C++ concepts [0].
- `SBL-MODELS`**:** C++ *models* matching the C++ concepts required to instantiate classes from `SBL-CORE`.
- `SBL-MODULES`**:** C++ classes instantiating classes from the `SBL-CORE` with specific biophysical models from `SBL-MODELS`. A module may be seen as a black box transforming an input into an output. With modules, an application workflow consists of interconnected modules.

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

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

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

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

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

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

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

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

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

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

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

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

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

The releases are scheduled as follows:

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

AMIB Project-Team

4. New Software and Platforms

4.1. Cartaj

Participant: Alain Denise [correspondant].

CARTAJ is a software that automatically predicts the topological family of three-way junctions in RNA molecules, from their secondary structure only : :the sequence and the canonical Watson–Crick pairings. The Cartaj software http://cartaj.lri.fr that implements our method can be used online. It is also meant for being part of RNA modelling softwares and platforms. The methodology and the results of CARTAJ are presented in [59]. More than 300 visits since its release in January 2012.

4.2. DiMoVo

Participant: Julie Bernauer [correspondant].

DIMOVO, *DIscriminate between Multimers and MOnomers by VOronoi tessellation* : Knowing the oligomeric state of a protein is necessary to understand its function. his tool, accessible as a webserver and still used and maintained, provides a reliable discrimination function to obtain the most favorable state of proteins.
Availability : released in 2008.

4.3. VorScore

Participant: Julie Bernauer [correspondant].

VORSCORE, *Voronoi Scoring Function Server* : Scoring is a crucial part of a protein-protein procedure and having a quantitave function to evaluate conformations is mandatory. This server provides access to a geometric knowledge-based evaluation function. It is still maintained and widely used. See Bernauer et al., Bioinformatics, 2007 23(5):555-562 for further details.

4.4. ConQuR-Bio

Participants: Bryan Brancotte, Sarah Cohen-Boulakia [correspondant], Alain Denise.

ConQuR-Bio assists scientists when they query public biological databases. Various reformulations of the user query are generated using medical terminologies (MeSH, OMIM, ...). Such alternative reformulations are then used to rank the query results using a new consensus ranking strategy. The originality of our approach thus lies in using consensus ranking techniques within the context of query reformulation. The ConQuR-Bio system is able to query the Entrez-Gene NCBI database. The benefit of using ConQuR-Bio compared to what is currently provided to users has been demonstrated on a set of biomedical queries.
Availability : http://conqur-bio.lri.fr/

4.5. VARNA (Visualization Application for RNA)

Participants: Yann Ponty [correspondant], Alain Denise.

A lightweight Java Applet dedicated to the quick drawing of an RNA secondary structure. VARNA is open-source and distributed under the terms of the GNU GPL license. Automatically scales up and down to make the most out of a limited space. Can draw multiple structures simultaneously. Accepts a wide range of documented and illustrated options, and offers editing interactions. Exports the final diagrams in various file formats (svg,eps,jpeg,png,xfig) [52]...
VARNA currently ships in its 3.9 version, and consists in ∼50 000 lines of code in ∼250 classes.
Availability : Distributed at http://varna.lri.fr since 2009 under the GPL v3 license.

Impact: Downloaded ∼15k times and cited by ∼250 research manuscripts (source: Google Scholar).

4.6. GenRGenS (GENeration of Random GENomic Sequences)

Participants: Yann Ponty [correspondant], Alain Denise.

A software dedicated to the random generation of sequences. Supports different lasses of models, including weighted context-free grammars, Markov models, PROSITE patterns... [69] GENRGENS currently ships in its 2.0 version, and consists in ∼25 000 lines of code in ∼120 Java classes.
Availability : Distributed at http://www.lri.fr/~genrgens/ since 2006 under the terms of the GPL v3 license.
Impact: Downloaded ∼5k times and cited by ∼60 times (source: Google Scholar).

4.7. GeneValorization

Participants: Bryan Brancotte, Sarah Cohen-Boulakia [correspondant].

High-throughput technologies provide fundamental informations concerning thousands of genes. Most of the current biological research laboratories daily use one or more of these technologies and identify lists of genes. Understanding the results obtained includes accessing to the latest publications concerning individual or multiple genes. Faced to the exponential growth of publications avaliable, this task is becoming particularly difficult to achieve.
Here, we introduce a web-based Java application tool named GeneValorization which aims at making the most of the text-mining effort done downstream to all high throughput technology assays. Regular users come from the Curie Institute, but also the EBI.
Impact : 925 distinct international users have used GeneValorization and about a hundred use it on a regular basis. The tool is on average used once to twice every day.
Availability : it is available at http://bioguide-project.net/gv with Inter Deposit Digital Number (*depot APP*, June 2013).

4.8. HSIM

Participant: Patrick Amar [correspondant].

HSIM (Hyperstructure Simulator) is a simulation tool for studying the dynamics of biochemical processes in a virtual bacteria. The model is given using a language based on probabilistic rewriting rules that mimics the reactions between biochemical species. HSIM is a stochastic automaton that implements an entity-centered model of objects. This kind of modelling approach is an attractive alternative to differential equations for studying the diffusion and interaction of the many different enzymes and metabolites in cells which may be present in either small or large numbers.

The new version of HSIM includes a Stochastic Simulation Algorithm *a la* Gillespie that can be used with the same model in a standalone way or in a mixed way with the entity-centered algorithm. This new version offers also the possibility to export the model in SciLab for a ODE integration. Last, HSIM can export the differential equations system, equivalent to the model, to LaTeX for pretty-printing.

This software is freely available at http://www.lri.fr/~pa/Hsim; A compiled version is available for the Windows, Linux and MacOSX operating systems.

4.9. Pint

Participant: Loïc Paulevé [correspondant].

PINT provides several command-line tools to model, simulate, and analyse the dynamics of automata networks. Its main application domain is systems biology for modelling and analysis of very large interaction networks. Besides a textual language for specifying networks and standard stochastic simulation algorithms, PINT implements static analysis for analysing and controlling the transient reachability. In particular, PINT provides the computation of cut sets for transient reachability, that gives sets of key automata states, whose mutation would prevent the concerned reachability to occur.

PINT has been applied to extremely large biological networks, from 100 to 10,000 interacting components, demonstrating its scalability and potential to handle full databases of interactions.

PINT is distributed under the CeCiLL licence, and is available at http://loicpauleve.name/pint.

ANGE Project-Team

5. New Software and Platforms

5.1. FRESHKISS

Although the Saint-Venant system is the cornerstone of flow modelling in geosciences, this does not mean that the transfer of the efficient dedicated simulation tools is achieved in the geoscience community.

ANGE collaborates with scientists, laboratories and companies that are interested in scientific advances which makes the valuation and the transfer of results easier.

ANGE aims at developing robust and efficient numerical tools. For the simulation of the free surface Navier-Stokes equations, numerical tools have been developed namely FRESHKISS2D[0] and FRESHKISS3D. These tools are used by several scientists typically in the BIOCORE Inria project-team, at EDF and in public research laboratories.

FRESHKISS3D is a numerical code solving the 3D hydrostatic and incompressible Navier-Stokes equations with variable density. This code was initially dedicated to research activities within the team but we now aim at turning it into a numerical tool being used by non-mathematicians. Indeed, there is a demand in research laboratories and companies to use this tool. A young engineer (R. Hamouda) has been hired (ADT In@lgae funded by Inria) and its assignment is to improve/enrich the code and to make it user-friendly. Notice that FRESHKISS3D is used for teaching (master students in geosciences) at university Denis Diderot Paris 7 and IPGP.

5.2. TSUNAMATHS

TSUNAMATHS is an educational platform aiming at simulating historical tsunamis. Real data and mathematical explanations are provided to enable people to better understand the overall process of tsunamis. It is available on the Internet:

http://ange.raoufhamouda.com/tsunami/en_animation.htm

It was presented in the framework of the 2013 UNESCO year of “Mathematics of Planet Earth” and then exhibited at the ICM 2014 session (see § 9.3).

[0] FRESHKISS: FREe Surface Hydrodynamics using KInetic SchemeS

ARAMIS Project-Team

5. New Software and Platforms

5.1. SACHA

Participants: Marie Chupin [Correspondant], Ludovic Fillon.

SACHA ("Segmentation Automatisée Compétitive de l'Hippocampe et de l'Amygdale") is a software for the fully automatic segmentation of the hippocampus and the amygdala from MRI 3D T1 brain scans. It has been validated in various populations including healthy controls and patients with Alzheimer's disease, epilepsy and depression. It has been successfully applied to over 3,000 subjects, both controls, from adolescents to elderly subjects, and patients with different types of pathologies. The current stable version is fully automatic and focused on cross-sectional segmentation. The software can be used both as a command-line program or through a graphical user interface (GUI). The core of the program is coded in C++. It has a dependency to the AIMS library (http://www.brainvisa.info) and preprocessing steps rely on processes in Matlab from SPM (http://www.fil.ion.ucl.ac.uk/spm/). The GUI is coded in Python and is based on BrainVISA (http://www.brainvisa.info). The software has been registered at the APP (French agency for software protection).

5.2. WHASA

Participants: Marie Chupin [Correspondant], Ludovic Fillon, Thomas Samaille.

WHASA ("White matter Hyperintensity Automatic Segmentation Algorithm") is a software for the fully automatic segmentation of age-related white matter hyperintensities from MRI FLAIR and 3D T1 brain scans. It has been validated on a population showing a wide range of lesion load, and is being further evaluated on elderly subjects with few clinical abnormalisties and with different acquisition characteristics. The current stable version is fully automatic and focused on cross-sectional segmentation. The software can be used both as a Matlab command-line or through a graphical user interface (GUI). The core of the program is coded in Matlab. It has a dependency to the SPM environment (http://www.fil.ion.ucl.ac.uk/spm/). The GUI is coded in Python and is based on BrainVISA (http://www.brainvisa.info). The software has been registered at the APP (French agency for software protection).

5.3. Deformetrica

Participants: Stanley Durrleman [Correspondant], Alexandre Routier, Pietro Gori, Marcel Prastawa, Ana Fouquier, Joan Glaunès, Benjamin Charlier, Cedric Doucet.

Deformetrica is a software which estimates diffeomorphic deformations between sets of geometric objects in 2D and 3D. Those deformations are estimated either for the registration of two of such objects sets or for the construction of an atlas from several of such sets (a template model set and deformations mapping the template model to each set). Geometric objects could be grey-level images, surface meshes, polygonal lines or unstructured point sets. The method relies on the metric on currents for the comparison of point sets and the sum of squared differences for the comparison of images.

The software is written in C++ and relies on the ITK and VTK libraries. Core functions are coded with CUDA for their parallelization on GPU. It is a command-line software.

The version 2.1 of the software has been released on December 19, 2014. It is freely accessible to the scientific community at www.deformetrica.org.

5.4. qualiCATI

Participants: Marie Chupin [Correspondant], Hugo Dary, Urielle Thoprakarn, Amadou Tall, David Gay, Nicolas Vibet, Aude Costard, Cyril Poupon, Vincent Perlbarg, Mélanie Pélégrini-Issac, Alexandre Vignaud.

qualiCATI is a software designed for comprehensive quality control of multimodal MRI data acquisition in large multicentre clinical studies. The software is built as a platform receiving several modules, developped by several CATI engineers. The first module is dedicated to acquisition requirement checking and conversion to nifti format. The second module aims at making 3DT1 acquisition quality check more systematic, and relies both on visual inspection and quantitative indices. The third module allows a simultaneous evaluation of the clinical part of the CATI acquisition protocol. The fourth module embeds automatic indices to evaluate resting state fMRI acquisition. The fifth module is dedicated to first preprocessings and quality indices for dMRI. The sixth module is dedicated to qMRI, with visual and automated quality control together with preprocessings. The last module is dedicated to data and project management. QualiCATI requires training for the visual parts, and is closely linked with a team of clinical research assistants. It has been used to analyse about 5000 subjects from about 15 multi centre research projects initiated before or after the CATI started. Other modules will be added in the future to embed new aspects of the MRI protocol proposed by the CATI. The Aramis team is in charge of the second and third modules and jointly in charge of the first module. The software is centered on a graphical user interface (GUI). The whole program is coded in Python within the pyPTK environment developped by Cyril Poupon (Neurospin). It has dependencies to SPM (http://www.fil.ion.ucl.ac.uk/spm/) and brainVISA environments as well as specific tools for DICOM management.

5.5. Brain Networks Toolbox

Participants: Mario Chavez, Fabrizio de Vico Fallani [Correspondant].

Brain Networks Toolbox is a collection of Matlab routines developed to quantify topological metrics of complex brain networks. These routines are associated with published publications with application to real data and freely distributed via the FreeBorn (French Brain Networks) consortium https://sites.google.com/site/fr2eborn/download

ASCLEPIOS Project-Team

4. New Software and Platforms

4.1. SOFA

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

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

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

4.2. MedInria

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

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

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

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

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

4.3. MUSIC

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

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

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

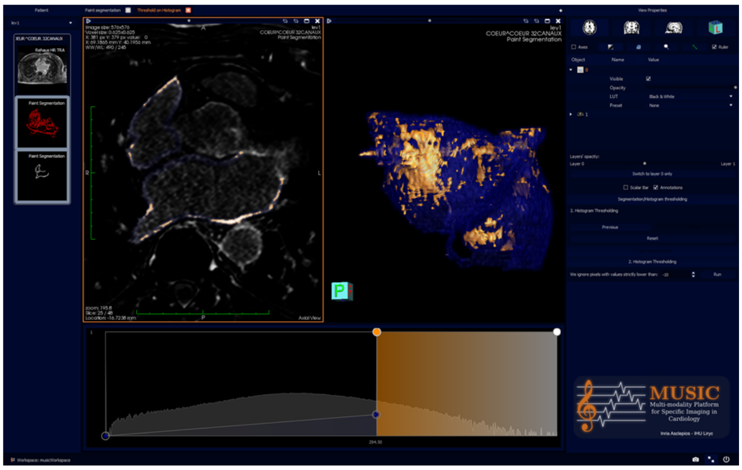


Figure 1. Segmentation of atrial fibrosis using adaptive histogram thresholding based on the MUSIC Software.

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

4.4. VP2HF platform

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

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

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

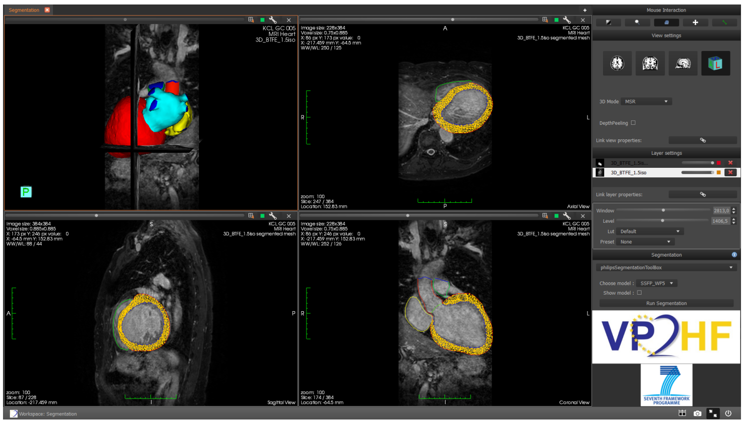


Figure 2. Philips segmentation tool within the VP2HF platform

ATHENA Project-Team

5. New Software and Platforms

5.1. OpenMEEG

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

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

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

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

5.2. High Performance Diffusion MRI

Participants: Aurobrata Ghosh, Théodore Papadopoulo, Rachid Deriche.

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

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

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

5.3. Contributions to the open source dMRI platform DIPY

Participants: Demian Wassermann, Rutger Fick.

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

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

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

5.4. medInria

Participants: Jaime Garcia Guevara, Théodore Papadopoulo.

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

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

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

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

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

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

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

5.5. FindSources3D

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

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

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

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

5.6. CoAdapt P300 Stimulator

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

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

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

BAMBOO Project-Team

5. New Software and Platforms

5.1. AcypiCyc

Participants: Hubert Charles [EPI], Patrice Baa Puyoule [Contact, Patrice.Baa-Puyoulet@lyon.inra.fr], Stefano Colella [Contact, stefano.colella@lyon.inra.fr], Ludovic Cottret, Marie-France Sagot [EPI], Augusto Vellozo [Contact, augusto@cycadsys.org], Amélie Véron.

Database of the metabolic network of *Acyrthosiphon pisum*.
http://acypicyc.cycadsys.org/

5.2. AlViE

Participants: Pierluigi Crescenzi [Contact, pierluigi.crescenzi@unifi.it, ext. member EPI], Giorgio Gambosi, Roberto Grossi, Carlo Nocentini, Tommaso Papini, Walter Verdese.

ALVIE is a post-mortem algorithm visualization Java environment, which is based on the interesting event paradigm. The current distribution of ALVIE includes more than forty visualizations. Almost all visualizations include the representation of the corresponding algorithm C-like pseudo-code. The ALVIE distribution allows a programmer to develop new algorithms with their corresponding visualization: the included Java class library, indeed, makes the creation of a visualization quite an easy task (once the interesting events have been identified).
http://piluc.dsi.unifi.it/alvie/

5.3. Cassis

Participants: Christian Baudet [EPI, Contact, christian.baudet@inria.fr], Christian Gautier [EPI], Claire Lemaitre [Contact, claire.lemaitre@inria.fr], Marie-France Sagot [EPI], Eric Tannier.

Algorithm for precisely detecting genomic rearrangement breakpoints.
http://pbil.univ-lyon1.fr/software/Cassis/

5.4. Coala

Participants: Christian Baudet [EPI, Contact, christian.baudet@inria.fr], Pielrluigi Crescenzi, Bea Donati [EPI, Contact, bea.donati@inria.fr], Christian Gautier [EPI], Catherine Matias, Blerina Sinaimeri [EPI, Contact, blerina.sinaimeri@inria.fr], Marie-France Sagot [EPI, Contact, marie-france.sagot@inria.fr].

COALA stands for "CO-evolution Assessment by a Likelihood-free Approach". It is thus a likelihood-free method for the co-phylogeny reconstruction problem which is based on an Approximative Bayesian Computation (ABC).
http://coala.gforge.inria.fr/

5.5. C3Part & Isofun

Participants: Frédéric Boyer, Yves-Pol Deniélou, Anne Morgat [EPI, ext. member], Marie-France Sagot [EPI], Alain Viari [EPI, Contact, alain.viari@inria.fr].

The C3Part / Isofun package implements a generic approach to the local alignment of two or more graphs representing biological data, such as genomes, metabolic pathways or protein-protein interactions, in order to infer a functional coupling between them. It is based on the notion of "common connected components" between graphs.http://www.inrialpes.fr/helix/people/viari/lxgraph/index.html

5.6. CycADS

Participants: Hubert Charles [EPI], Patrice Baa Puyoule [Contact, Patrice.Baa-Puyoulet@lyon.inra.fr], Stefano Colella [Contact, stefano.colella@lyon.inra.fr], Ludovic Cottret, Marie-France Sagot [EPI], Augusto Vellozo [Contact, augusto@cycadsys.org].

Cyc annotation database system.
http://www.cycadsys.org/

5.7. Eucalypt

Participants: Christian Baudet [EPI, Contact, christian.baudet@inria.fr], Pielrluigi Crescenzi, Bea Donati [Contact, bea.donati@inria.fr], Blerina Sinaimeri, Marie-France Sagot [EPI].

Algorithm for enumerating all optimal (possibly time-unfeasible) mappings of a parasite tree unto a host tree.
http://eucalypt.gforge.inria.fr/

5.8. Gobbolino & Touché

Participants: Vicente Acuña [EPI], Etienne Birmelé, Ludovic Cottret, Pierluigi Crescenzi, Fabien Jourdan, Vincent Lacroix, Alberto Marchetti-Spaccamela [EPI, ext. member], Andrea Marino, Paulo Vieira Milreu [EPI, Contact, pvmilreu@gmail.com], Marie-France Sagot [EPI, Contact, marie-france.sagot@inria.fr], Leen Stougie [EPI, ext. member].

Designed to solve the metabolic stories problem, which consists in finding all maximal directed acyclic subgraphs of a directed graph G whose sources and targets belong to a subset of the nodes of G, called the black nodes. Biologically, stories correspond to alternative metabolic pathways that may explain some stress that affected the metabolites corresponding to the black nodes by changing their concentration (measured by metabolomics experiments).
http://gforge.inria.fr/projects/gobbolino

5.9. KisSNP

Participants: Vincent Lacroix [EPI], Pierre Peterlongo [Contact, pierre.peterlongo@inria.fr], Nadia Pisanti, Marie-France Sagot [EPI], Nicolas Schnel.

Algorithm for identifying SNPs without a reference genome by comparing raw reads. KISSNP has now given birth to DISCOSNP in a work involving V. Lacroix from BAMBOO and the GenScale Inria Team at Rennes (contact: pierre.peterlongo@inria.fr).
http://alcovna.genouest.org/kissnp/, http://colibread.inria.fr/software/discosnp/

5.10. KisSplice & KisSplice2igv7

Participants: Lilia Brinza [EPI], Alice Julien-Laferrière [EPI], Janice Kielbassa, Vincent Lacroix [Contact, EPI], Camille Marchet [EPI], Vincent Miele, Gustavo Sacomoto [EPI], Marie-France Sagot [EPI].

Enables to analyse RNA-seq data with or without a reference genome. It is an exact local transcriptome assembler, which can identify SNPs, indels and alternative splicing events. It can deal with an arbitrary number of biological conditions, and will quantify each variant in each condition. KISSPLICE2IGV is a pipeline that combines the outputs of KISSPLICE to a reference transcriptome (obtained with a full-length transcriptome assembler or a reference database). It provides a visualisation of the events found by KISSPLICE in a longer context using a genome browser (IGV).
http://kissplice.prabi.fr/

5.11. kissDE

Participants: Lilia Brinza [EPI], Janice Kielbassa, Vincent Lacroix [Contact, EPI], Camille Marchet [EPI], Vincent Miele.

KISSDE is an R Package enabling to test if a variant (genomic variant or splice variant) is enriched in a condition. It takes as input a table of read counts obtained from NGS data pre-processing and gives as output a list of condition specific variants. http://kissplice.prabi.fr/tools/kissDE/

5.12. LASAGNE

Participants: Pierluigi Crescenzi [Contact, pierluigi.crescenzi@unifi.it, ext. member EPI], Roberto Grossi, Michel Habib, Claudio Imbrenda, Leonardo Lanzi, Andrea Marino.

LASAGNE is a Java application which allows the user to compute distance measures on graphs by making a clever use either of the breadth-first search or of the Dijkstra algorithm. In particular, the current version of LASAGNE can compute the exact value of the diameter of a graph: the graph can be directed or undirected and it can be weighted or unweighted. Moreover, LASAGNE can compute an approximation of the distance distribution of an undirected unweighted graph. These two features are integrated within a graphical user interface along with other features, such as computing the maximum (strongly) connected component of a graph.
http://piluc.dsi.unifi.it/lasagne/?page_id=142

5.13. MetExplore

Participants: Michael Barrett, Hubert Charles [EPI], Ludovic Cottret [Contact, Ludovic.Cottret@toulouse.inra.fr], Fabien Jourdan, Marie-France Sagot [EPI], Florence Vinson, David Wildridge.

Web server to link metabolomic experiments and genome-scale metabolic networks.
http://metexplore.toulouse.inra.fr/metexplore/

5.14. Migal

Participants: Julien Allali [Contact, julien.allali@labri.fr], Marie-France Sagot [EPI, Contact, marie-france.sagot@inria.fr].

RNA, tree comparison
Algorithm for comparing RNA structures.
http://www-igm.univ-mlv.fr/~allali/logiciels/index.en.php

5.15. Mirinho

Participants: Cyril Fournier [EPI], Susan Higashi [EPI, Contact, susan.higashi@inria.fr], Christian Gautier [EPI], Christine Gaspin, Marie-France Sagot [EPI].

Predicts, at a genome-wide scale, microRNA candidates.
http://mirinho.gforge.inria.fr/

5.16. MotusWEB

Participants: Ludovic Cottret, Fabien Jourdan, Vincent Lacroix [EPI, Contact, vincent.lacroix@univ-lyon1.fr], Odile Rogier, Marie-France Sagot [EPI].

Algorithm for searching and inferring coloured motifs in metabolic networks (web-based version - offers different functionalities from the downloadable version).
http://pbil.univ-lyon1.fr/software/motus_web/

5.17. Motus

Participants: Ludovic Cottret, Fabien Jourdan, Vincent Lacroix [EPI, Contact, vincent.lacroix@univ-lyon1.fr], Odile Rogier, Marie-France Sagot [EPI].

Algorithm for searching and inferring coloured motifs in undirected graphs (downloadable version - offers different functionalities from the web-based version).
http://pbil.univ-lyon1.fr/software/motus/

5.18. PhEVER

Participants: Christian Gautier [EPI], Vincent Lotteau, Leonor Palmeira [Contact, mlpalmeira@ulg.ac.be], Chantal Rabourdin-Combe, Simon Penel.

Database of homologous gene families built from the complete genomes of all available viruses, prokaryotes and eukaryotes and aimed at the detection of virus/virus and virus/host lateral gene transfers.
http://pbil.univ-lyon1.fr/databases/phever/

5.19. PepLine

Participants: Jérôme Garin, Alain Viari [EPI, Contact, alain.viari@inria.fr].

Pipeline for the high-throughput analysis of proteomic data.

5.20. Pitufo and family

Participants: Vicente Acuña [EPI], Ludovic Cottret [Contact, Ludovic.Cottret@toulouse.inra.fr], Alberto Marchetti-Spaccamela [EPI, ext. member], Paulo Vieira Milreu [EPI, Contact, pvmilreu@gmail.com], Marie-France Sagot [EPI], Leen Stougie [EPI, ext. member], Fabio Viduani-Martinez.

Algorithms to enumerate all minimal sets of precursors of target compounds in a metabolic network.
http://sites.google.com/site/pitufosoftware/

5.21. RepSeek

Participants: Guillaume Achaz [Contact, achaz@abi.snv.jussieu.fr], Eric Coissac, Alain Viari [EPI].

Finding approximate repeats in large DNA sequences.
http://wwwabi.snv.jussieu.fr/public/RepSeek/

5.22. Smile

Participants: Laurent Marsan, Marie-France Sagot [EPI, Contact, marie-france.sagot@inria.fr].

Motif inference algorithm taking as input a set of biological sequences.

5.23. UniPathway

Participants: Eric Coissac, Anne Morgat [EPI, Contact, anne.morgat@inria.fr], Alain Viari [EPI].

Database of manually curated pathways developed with the Swiss-Prot group.
http://www.unipathway.org

BEAGLE Project-Team

4. New Software and Platforms

4.1. Aevol (artificial evolution)

Participants: Guillaume Beslon, Jonathan Rouzaud-Cornabas, Carole Knibbe, Priscila Biller, Bérénice Batut.

- Contact: Carole Knibbe (carole.knibbe@inria.fr).
- Aevol is a simulation software dedicated to the study of genome evolution. It allows to carry out *in silico* experimental evolution. Populations of digital organisms reproduce and mutate randomly, with both small mutations and large chromosomic rearrangements, in a steady or varying environment. A curve-fitting task is used to determine the fitness of the organisms and thus their rate of reproduction. The number of genes, their order, their sequences, their intergenic distances are all free to evolve. Thanks to a two-year grant from Inria's Technological Development Department (ADT « aevol »), the development of an improved and parallel version of the software has started in October.
- URL: http://www.aevol.fr

4.2. EvoEvo modelization tool

Participants: Charles Rocabert, Guillaume Beslon, Carole Knibbe.

- Contact: Guillaume Beslon
- In the context of the EvoEvo european project (http://www.evoevo.eu/) we are developing an integrated model of microorganisms evolution. This model will extend the current evolutionary models developped in the team (Aevol and R-Aevol) by adding a metabolic level and an ecosystem level. In 2014, a first version has been developed and released that includes the genomic, genetic and metabolic levels.

4.3. FluoBacTracker

Participants: Hugues Berry, David P Parsons, Magali Vangkeosay.

- Contact: Hugues Berry (hugues.berry@inria.fr)
- FluoBacTracker is a software for automated quantification of bacterial cells in microscopy movies, developed in collaboration with INSERM U1001 and Paris 5 MAP (Applied Mathematics) Labs. The development (started october 2012) has been supported by a 2-year grant (ADT) funded by Inria's Technological Development Department (Sept 2012- July 2014, project name: "MultiPop"). We hope this software will be useful to all the experimental biology labs that tries to derive single-cell data from bacteria growth microscopy movies. Co-developers include Magali Vangkeosay (BEAGLE), David P Parsons (SED, Inria Grenoble) and Xiaohu Song (INSERM U1001).

4.4. Ancestral Genome Reconstructions

Participant: Eric Tannier.

- Contact: Eric Tannier (eric.tannier@inria.fr).
- We participated in the development of a series of softwares for genome organization analysis:
  - ANGES, for ANcestral GEnomeS maps, is a toolkit for ordering ancestral genomic markers in chromosomes. An application note has been published in *Bioinformatics* in 2012 to advertise its first release. It is hosted at SFU in Vancouver, URL: http://paleogenomics.irmacs.sfu.ca/ANGES/, under a GNU license, 2012.
  - DeCo and DeCoLT, for Detection of Co-evolution (with Lateral gene Transfer), reconstruct neighborhood relationships between genes of ancient genomes, in the presence of gene duplications, transfer and losses. Both are hosted at the PRABI, the bioinformatics platform in Lyon, under a Cecill license, 2012 and 2013. URL: http://pbil.univ-lyon1.fr/software/DeCo/ and http://pbil.univ-lyon1.fr/software/DeCoLT/.
  - DCJ2HP provides bayesian samples of rearrangements scenarios between two genomes. It is hosted at the Renyi Institute in Budapest. URL: http://www.renyi.hu/~miklosi/DCJ2HP/

4.5. DMT4SP mining tool

Participant: Christophe Rigotti.

- Contact: Christophe Rigotti (christophe.rigotti@insa-lyon.fr).
- DMT4SP (Data-Mining Tool For Sequential Patterns) – DMT4SP is command-line tool to extract episodes and episode rules over a single sequence or several sequences of events. It allows to specify constraints on the episodes or on the rules. Three kinds of patterns can be extracted: (1) serial episodes, (2) serial episode rules having a single event type in the consequent, and (3) quantitative episodes (aka grouping of "homogeneous" occurrences of serial episodes with respect to the time gap between events). DMT4SP is a prototype that is freely distributed (http://liris.cnrs.fr/~crigotti/dmt4sp.html).

BIGS Project-Team

4. New Software and Platforms

4.1. Online data analysis

Participants: J.-M. Monnez

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

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

4.2. Socio-economic index

Participants: J.-M. Monnez

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

4.3. Angio-Analytics

Participants: T. Bastogne

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

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

Participants: T. Bastogne

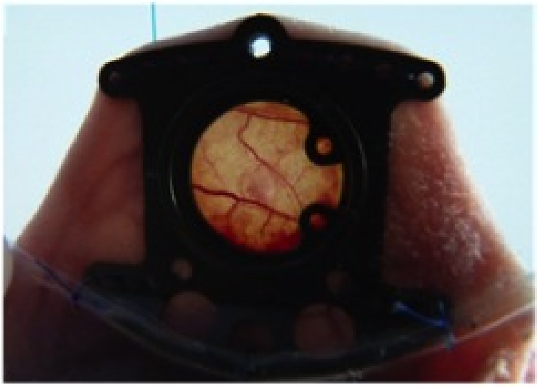


Figure 1. Example of a skinfold chamber placed on a mouse skin

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

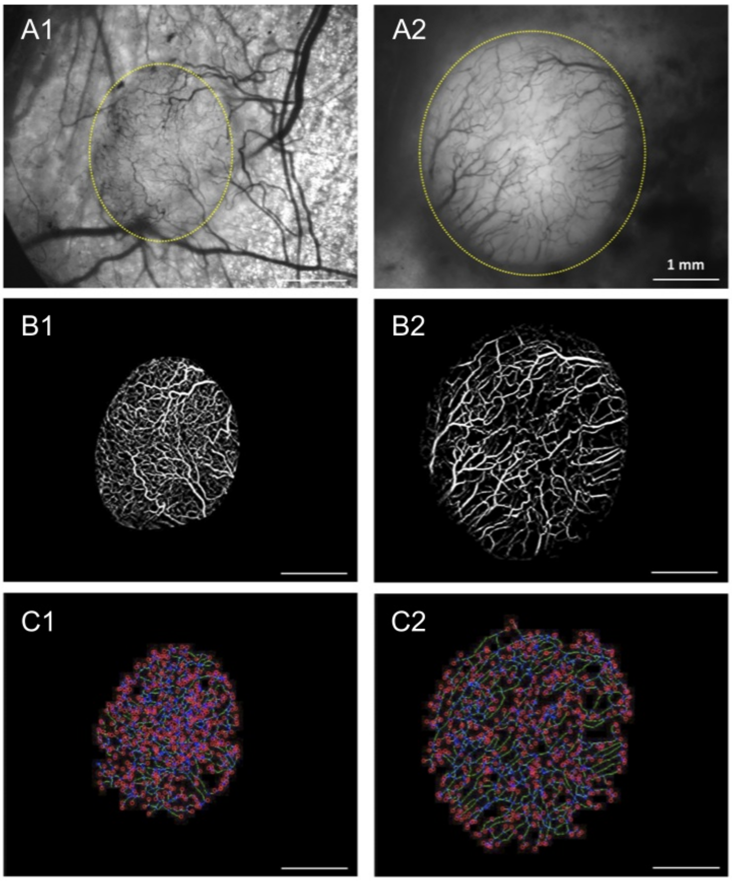


Figure 2. Example of segmentation process on a control (left) and treated tumor (right) at day -7: manual segmentation (ROI) of cancerous tissues is done in yellow on step A, vessel segmentation is performed on step B (vessels are in white), step C presents the quantification (blue and red circles) on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

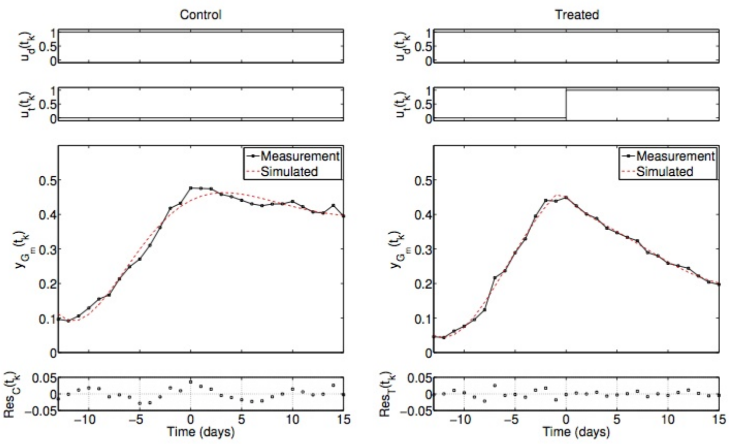


Figure 3. Measurements and estimated outputs for control and treated batches. Input signals and residuals are respectively plotted above and below

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

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

BIOCORE Project-Team

5. New Software and Platforms

5.1. Supervision software

5.1.1. ODIN

Participants: Olivier Bernard, Francesco Novellis.

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

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

5.1.2. In@lgae

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

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

BONSAI Project-Team

5. New Software and Platforms

5.1. SortMeRNA – Metatranscriptome classification

Software web site: http://bioinfo.lille.inria.fr/RNA/sortmerna

Licence: GPL

Objective: *SortMeRNA* is a tool designed to rapidly filter ribosomal RNA fragments from metatranscriptomic data produced by next-generation sequencers. It is available for download from our website, or through the open web-based platform Galaxy. The development version is also available on GitHub. *SortMeRNA* was first released in October 2012. It is now used in production by Genoscope (French National Center for Sequencing) to process all metatranscriptomic data of the Tara Ocean Expedition, and has been integrated in several other computational pipelines (Qiime developed at University of Colorado at Boulder, MetaMetadb developed at University of Tokyo, Leimena pipeline developed at Wageningen University,...).

SortMeRNA is still under development through a partnership with the Knight lab (University of Colorado at Boulder). Version 2.0 has been released in November 2014, and has extended functionalities. It can now perform sequence alignments to any ribosomal RNA database, which allows the user to study the taxonomic content of a microbial sample. This new version has been presented at the international workshops [12], [11].

5.2. Vidjil – Quantifying lymphocyte rearrangements in high-throughput sequencing data

Software web site: http://bioinfo.lille.inria.fr/vidjil/

Objective: **Vidjil** is a platform for high-throughput V(D)J recombinations analysis. containing three components. The Vidjil *algorithm* process high-througput sequencing data to extract V(D)J junctions and gather them into clones. Vidjil starts from a set of reads and detects “windows” overlapping the actual CDR3. This is based on an fast and reliable seed-based heuristic and allows to output all sequenced clones. The analysis is extremely fast because, in the first phase, no alignment is performed with database germline sequences [5]. The Vidjil dynamic *browser* is made for the visualization and analysis of clones and their tracking along the time in a “minimal residual disease” setup or in a immunological study. The browser visualize data processed by the Vidjil algorithm or by other V(D)J analysis pipeline and enables to explore further cluterings. Finally, a *patient database* with a server links the browser and the algorithmic part. The goal is that the clinicians will be able to upload, manage and process their runs on a server hosted in their hospital.

In 2014, the development of Vidjil was supported by the SIRIC OncoLille (Marc Duez). We developed the new patient database and added features both on the browser and on the algorithm (multi-system analysis). Several hospital labs in France and in Europe are testing Vidjil. The Lille hospital plans to use Vidjil in 2015 in a pre-production pipeline.

5.3. Norine – A resource for nonribosomal peptides

Software web site: http://bioinfo.lille.inria.fr/norine/

Objective: **Norine** is a public computational resource that contains a database of NRPs with a web interface and dedicated tools, such as a 2D graph viewer and editor for peptides or comparison of NRPs. Norine was created and is maintained by members of BONSAI team, in tight collaboration with members of the ProBioGEM lab, a microbial laboratory of Lille1 University. Since its creation in 2006, Norine has gained an international recognition as the unique database dedicated to non-ribosomal peptides because of its high quality and manually curated annotations, and has been selected by wwPDB as a reference database. It is queried from all around the world by biologists or biochemists. It receives more than 3000 queries per month.

To enhance the Norine resource, we have recently developed a new module, named MyNorine, which is an open interface for biologists and biochemists dedicated to the submission of new non-ribosomal peptides in Norine database. Up to now, peptides were manually inputted and verified before being added in the database, which could potentially lead to human errors. The goal of MyNorine is to help users during the submission of peptides and monomers, by guiding them during all steps. For that, users, all over the world, can create an account on MyNorine. Thus, they contribute to the Norine resource and become curators (author of a peptide entry is mentioned in the corresponding page of Norine). Submitted peptides/monomers are validated, through a workflow process, by Norine team members, to ensure correct and consistent entries.

5.4. miRkwood –microRNAs in plant genomes

Software web site: http://bioinfo.lille.inria.fr/mirkwood/

Objective: **miRkwood** is a web server for the identification of hairpin precursors of both conserved and non-conserved miRNAs in plant genomes. It is able to face the diversity of plant pre-miRNAs and is optimised to take advantage of their distinctive properties: Sequence length, secondary structure, free energy, miRNA conservation, stability of the miRNA/miRNA* duplex, ... Moreover, it offers an intuitive and comprehensive user interface to navigate in the data, as well as many export options to allow the user to conduct further analyses on a local computer. Ongoing work is concerned with integrating small RNA-seq data.

5.5. ProCARs

Software web site: http://bioinfo.lille.inria.fr/procars

Objective: **ProCARs** is a program used to reconstruct ancestral gene orders as CARs (Contiguous Ancestral Regions) with a progressive homology-based method. The method runs from a phylogeny tree, without branch lengths needed, with a marked ancestor and a block file. The method output CARs as sets of ordered contiguous blocks in the targeted ancestor. ProCARs has been developped with Python 2.7.5.

CARMEN Team

5. New Software and Platforms

5.1. CEPS: a Cardiac ElectroPhysiology Simulator

The Carmen team develops a software code to perform high performance numerical simulations in cardiac electrophysiology using unstructured three-dimensional grids. The software, called CEPS (*Cardiac Electrophysiology Simulation*), is developped as a common tool for researchers in the Carmen team and for our partners and colleagues in scientific computing and biomedical engineering. The goal of CEPS is to easily allow the development of new numerical methods and new physical models. Thanks to the ADT, we are now able to use CEPS for the benchmark named *Second N-version Cardiac Electrophysiology Benchmark Specification actual developments*, see (benchmarck) for more details.

As compared to other existing softwares, CEPS aims at providing a more general framework of integration for new methods or models and a better efficiency in parallel. CEPS is designed to run on massively parallel architectures, and to make use of state-of-the-art and well known computing libraries to achieve realistic and complex heart simulations. CEPS also includes software engineering and and validation tools. We use the platform GForge (ceps) based on Subversion. This allows to keep a history of developments for developers and users.

Some of our collaborators actively participate to the testing and discussion for the development of CEPS, namely:

- C. Pierre, LMA University of Pau et des Pays de l'Adour;
- R. Turpault, IMB University of Bordeaux;

5.2. PROPAG

The workhorse for our applied simulation studies of the whole human heart is PROPAG, a code that has its origins at the Université de Montréal in Canada, and has been further developed by the Institute of Computational Science in Lugano, Switzerland. PROPAG is highly configurable and works with arbitrary model geometries. It runs efficiently on high-performance computing systems with many thousands of cores, including a "difficult" system such as the BlueGene/Q "Turing" at IDRIS. It is particularly useful for whole-heart studies, which typically rely on very large model sizes (in the order of 10^8 elements), several different membrane models and cell types in a single simulation run, and several regionally varying parameters.

PROPAG is presently used in our group to study the relation between the substrate, complexity, and electrocardiographic features of atrial fibrillation and of cardiomyopathy-related ventricular arrhythmia, providing the efficiency and flexibility that is required to handle the complex anatomical structures that are involved.

5.3. YAPI: A new project for the development of a platform for the simulation of the electrophysiology cardiac with CEPS

Many of our projects rely on realistic or even patient-tailored meshes to represent the anatomy of the human heart and torso. The construction of such meshes provides challenges on many levels, from the delineation of the anatomical structures in medical images to the construction of high-quality meshes. The construction of such meshes provides challenges on many levels, from the delineation of the anatomical structures in medical images to the construction of high-quality meshes. We presently use a variety of in-house and public software packages to perform this work and are able to produce meshes of sufficient quality, but we strive for an important streamlining of this work. We have initiated a discussion with several groups inside and outside Inria who have similar needs or can offer solutions. We specifically investigate the possibility to build a common software which combines and complements our present solutions. The new code should make various methods easily accessible and automate the work as much as possible. Because accuracy and mesh quality are important requirements, the new code should also provide convenient options for human intervention where algorithms fall short. For example, manual segmentation and mesh editing should be as easy and efficient as they are in medical-imaging tools and 3D-editing software, respectively, but well integrated into the workflow.

CASTOR Project-Team

5. New Software and Platforms

5.1. Free boundary equilibrium codes

5.1.1. CEDRES++

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

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

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

5.1.2. FEEQS.M

Participant: Holger Heumann.

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

5.2. Equinox

Participants: Jacques Blum, Cédric Boulbe, Blaise Faugeras.

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

5.3. VacTH

Participants: Jacques Blum, Cédric Boulbe, Blaise Faugeras.

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

5.4. FBGKI

Participants: Sébastian Minjeaud, Richard Pasquetti.

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

5.5. Platforms

5.5.1. FluidBox

Participants: Boniface Nkonga [contact], Hervé Guillard.

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

5.5.2. Plato

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

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

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

5.5.3. Jorek-Inria

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

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

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

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

CLIME Project-Team

5. New Software and Platforms

5.1. Data assimilation library: Verdandi

Participants: Nicolas Claude, Vivien Mallet, Dominique Chapelle [M3DISIM], Philippe Moireau [M3DISIM].

The leading idea is to develop a data assimilation library intended to be generic, at least for high-dimensional systems. Data assimilation methods, developed and used by several teams at Inria, are generic enough to be coded independently of the system to which they are applied. Therefore these methods can be put together in a library aiming at:

- making easier the application of methods to a great number of problems,
- making the developments perennial and sharing them,
- improving the broadcast of data assimilation works.

An object-oriented language (C++) has been chosen for the core of the library. A high-level interface to Python is automatically built. The design study raised many questions, related to high dimensional scientific computing, the limits of the object contents and their interfaces. The chosen object-oriented design is mainly based on three class hierarchies: the methods, the observation managers and the models. Several base facilities have also been included, for message exchanges between the objects, output saves, logging capabilities, computing with sparse matrices.

In 2014, version 1.6 was released with a lot of new unit tests, within the Google Test framework. The extended Kalman filter now supports model error. For users of C++11, a native random perturbation manager has been added and allows to circumvent the use of Newran. The overall compatibility with Clang has been reinforced. The documentation was significantly improved, especially about the installation under Windows and Linux.

5.2. Image processing library: Heimdali

Participants: David Froger [SED], Dominique Béréziat, Isabelle Herlin.

The initial aim of the image processing library Heimdali was to replace an internal Inria library (named Inrimage) by a library based on standard and open source tools, and mostly dedicated to satellite acquisitions.

The leading idea of the library is to allow the following issues:

- making easier the sharing and development of image assimilation softwares. For that purpose, the installation is easily achieved with the package manager Conda.
- developing generic tools for image processing and assimilation based on ITK (Insight Segmentation and Registration Toolkit http://www.itk.org). In reverse providing tools to ITK and contribute to the ITK community. Our software corresponds to issues related to satellite acquisitions but could be of interest for processing medical image sequences.

The main components of Heimdali concern:

- the pre/post processing of image sequences,
- the image assimilation with numerical models,
- the visualization of image sequences.

In 2014, prototypes of the two first items have been defined. The development of the whole library should be available in 2015.

5.3. Polyphemus

Participants: Sylvain Doré, Vivien Mallet, Yelva Roustan [CEREA].

Polyphemus (see the web site http://cerea.enpc.fr/polyphemus/) is a modeling system for air quality. As such, it is designed to yield up-to-date simulations in a reliable framework: data assimilation, ensemble forecast and daily forecasts. Its completeness makes it suitable for use in many applications: photochemistry, aerosols, radionuclides, etc. It is able to handle simulations from local to continental scales, with several physical models. It is divided into three main parts:

- libraries that gather data processing tools (SeldonData), physical parameterizations (AtmoData) and post-processing abilities (AtmoPy);
- programs for physical pre-processing and chemistry-transport models (Polair3D, Castor, two Gaussian models, a Lagrangian model);
- model drivers and observation modules for model coupling, ensemble forecasting and data assimilation.

Fig. 1 depicts a typical result produced by Polyphemus.

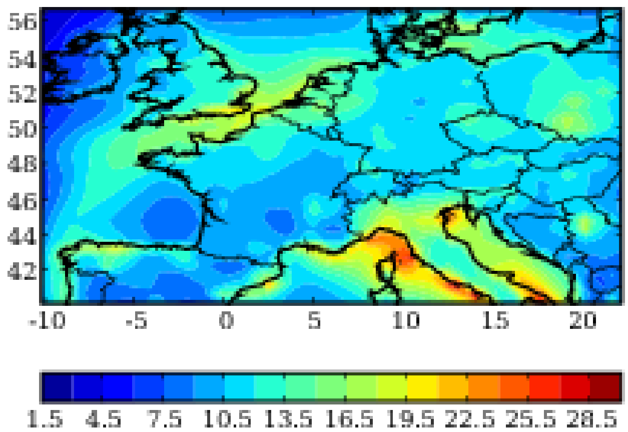


Figure 1. Map of the relative standard deviation (or spread, %) of an ensemble built with Polyphemus (ozone simulations, $\mu g\ m^{-3}$). The standard deviations are averaged over the summer of 2001. They provide an estimation of the simulation uncertainties.

Clime is involved in the overall design of the system and in the development of advanced methods in model coupling, data assimilation and uncertainty quantification (through model drivers and post-processing).

In 2014, Polyphemus was developed to better handle in-cloud and below-cloud scavenging. The interface of its Eulerian model, Polair3D, was extended to allow for detailed sensitivity analysis.

COFFEE Project-Team

5. New Software and Platforms

5.1. NS2DDV

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

5.2. Compass

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

5.3. SimBiof

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

5.4. AP_PartFlow

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

DEMAR Project-Team

4. New Software and Platforms

4.1. New Software and Platforms

4.1.1. RdP to VHDL tool

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

Our SENIS (Stimulation Electrique Neurale dIStribuee) based FES architecture relies on distributed stimulation units (DSU) which are interconnected by means of a 2-wire based network. A DSU is a complex digital system since 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 units digital part and to ease its prototyping on programmable digital devices (e.g. FPGA), we developed an approach for high level hardware component programming (HILECOP). To support the modularity and the reusability of sub-parts of complex hardware systems, the HILECOP methodology is based on components. An HILECOP component has: a Petri Net (PN) based behavior, a set of functions whose execution is controlled by the PN, and a set of variables and signals. Its interface contains places and transitions from which its PN model can be interconnected as well as signals it exports or imports. The interconnection of those components, from a behavioral point 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 Eclipse-based version of HILECOP (registered at the french Agence de Protection des Programmes (APP)) is regulary updated.

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

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

4.1.2. SENISManager

Participants: Robin Passama, David Andreu.

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

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

4.1.3. Synergy Neurostimulation Software

Participants: David Andreu, Amandine Pantel, Arthur Hiairrassary.

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

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

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

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

Participants: Melissa Dali, Olivier Rossel, David Guiraud.

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

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

There are two programs that perform the following functions:

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

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

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

IDDN.FR.001.490036.000.S.P.2014.000.31230

4.1.5. SENSBIOTK

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

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

Many "ready to use" examples with relative data and scripts : goniometer, compass, pedometer, motion capture validation...
https://github.com/sensbio/sensbiotk

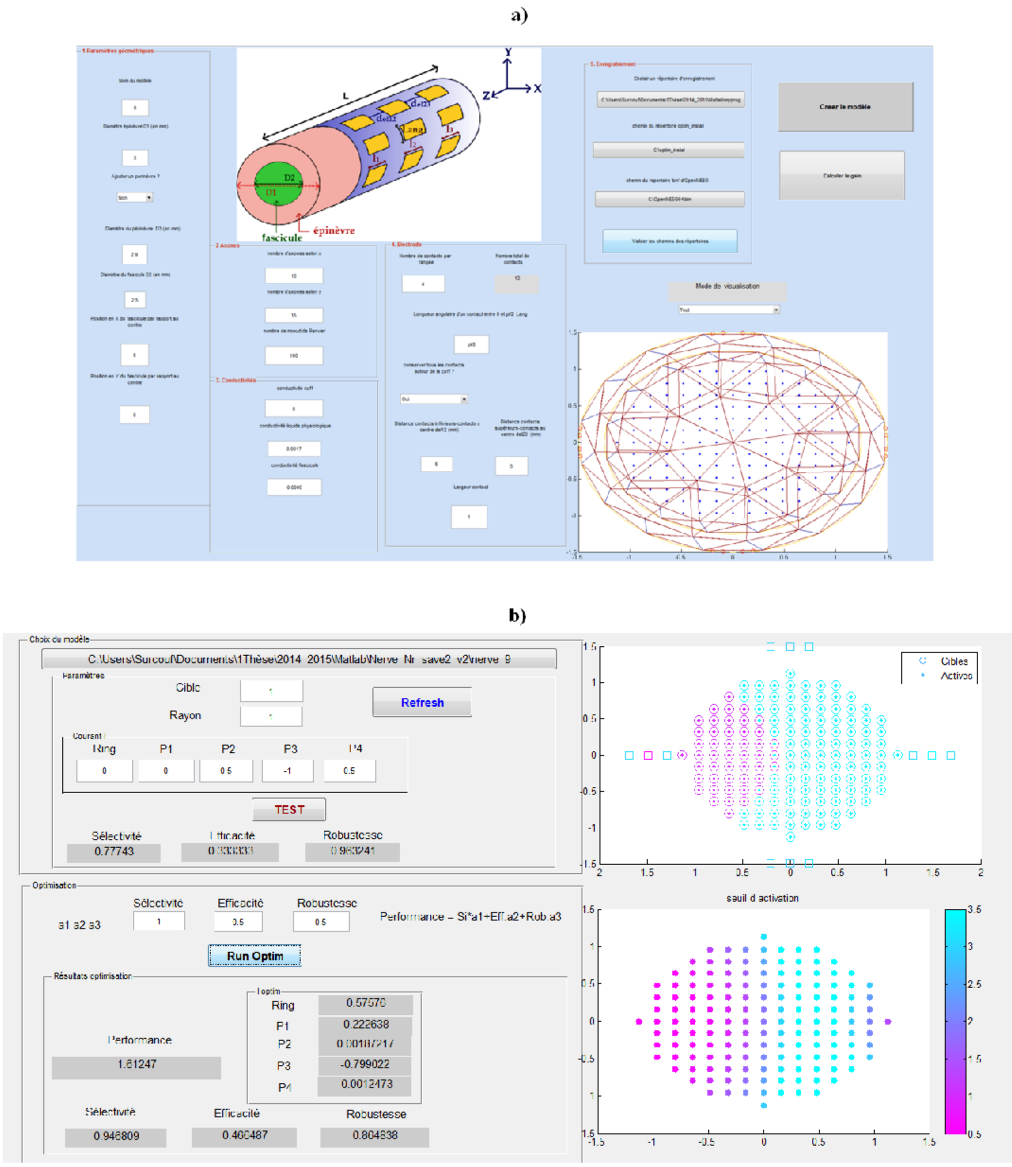


Figure 1. MOS2SENS interface. a) interface for the calculation of the electric field induced by electrical stimulation, b) interface for the configuration optimization

DRACULA Project-Team

5. New Software and Platforms

5.1. CelDyn

Participants: Laurent Pujo-Menjouet, Alen Tosenberger, Vitaly Volpert [correspondant].

Software "Celdyn" is developed in order to model cell population dynamics for biological applications. Cells are represented either as soft spheres or they can have more complex structure. Cells can divide, move, interact with each other or with the surrounding medium. Different cell types can be introduced. When cells divide, the types of daughter cells are specified. A user interface is developed.

DYLISS Project-Team

5. New Software and Platforms

5.1. Platforms and toolboxes

Among others, a goal of the team is to facilitate interplays between tools for biological data analysis and integration. Our tools are based on formal systems. They aim at guiding the user to progressively reduce the space of models (families of sequences of genes or proteins, families of keys actors involved in a system response, dynamical models) which are compatible with both knowledge and experimental observations.

Most of our tools are available both as stand-alone software and through portals such as Mobyle or Galaxy interfaces. Tools are developed in collaboration with the GenOuest resource and data center hosted in the IRISA laboratory, including their computer facilities [more info].

We present here three toolboxes which each contain complementary tools with respect to their targeted sub-domain of bioinformatics.

5.1.1. Integrative Biology: (constraint-based) toolbox for network filtering

The goal is to offer a toolbox for the reconstruction of networks from genome, literature and large-scale observation data (expression data, metabolomics...) in order to elucidate the main regulators of an observed phenotype. Most of the optimization issues are addressed with Answer Set Programming.

MeMap and MeMerge. We develop a workflow for the **Au**tomatic **Re**construction of **Me**tabolic networks (AuReMe). In this workflow, we use heterogeneous sources of data with identifiers from different namespaces. MeMap (**Me**tabolic network **Map**ping) consists in mapping identifiers from different namespaces to a unified namespace. Then, MeMerge (**Me**tabolic network **Merge**) merges two metabolic networks previously mapped on the same namespace. [web server].

meneco [*input*: draft metabolic network & metabolic profiles. *output*: metabolic network]. It is a qualitative approach to elaborate the biosynthetic capacities of metabolic networks. In fact, large-scale metabolic networks as well as measured datasets suffer from substantial incompleteness. Moreover, traditional formal approaches to biosynthesis require kinetic information, which is rarely available. Our approach builds upon formal systems for analyzing large-scale metabolic networks. Mapping its principles into Answer Set Programming allows us to address various biologically relevant problems [57] [50] [python package][web server].

shogen [*input*: genome & metabolic network. *output* : functional regulatory modules]. This software is able to identify genome portions which contain a large density of genes coding for enzymes that regulate successive reactions of metabolic pathways [48] [python package].

lombarde [*input*: genome, modules & several gene-expression datasets. *output*: oriented regulation network]. This tool is useful to enhance key causalities within a regulatory transcriptional network when it is challenged by several environmental perturbations [26] [web server].

bioquali [*input*: signed regulation network & one gene-expression dataset. *output*: consistency-checking and gene-expression prediction]. It is a plugin of the Cytoscape environment. BioQuali analyses regulatory networks and expression datasets by checking a global consistency between the regulatory model and the expression data. It diagnoses a regulatory network searching for the regulations that are not consistent with the expression data, and it outputs a set of genes which predicted expression is decided in order to explain the expression inputed data. It also provides the visualization of this analysis with a friendly environment to encourage users of different disciplines to analyze their regulatory networks [5] [web server][cytoscape plugin] .

ingranalyze [*input*: signed regulation network & one gene-expression dataset. *output*: network repair gene-expression prediction] This tool is an extension to the bioquali tool. It proposes a range of different operations for altering experimental data and/or a biological network in order to re-establish their mutual consistency, an indispensable prerequisite for automated prediction. For accomplishing repair and prediction, we take advantage of the distinguished modeling and reasoning capacities of Answer Set Programming [4] [Python package][web server].

Unifier. [*input*: sbml file with Palsson's metabolites identifiers *output*: sbml file with standard identifiers for metabolites]. This software is a Decision Support Tool to help biologists to normalize a file, containing Palsson's identifiers to refer to reactions and metabolites, using well known identifiers. Submit a list of Palsson identifiers to retrieve the corresponding database entries. Typically it maps with Metacyc identifiers but it would be used with Kegg or other databases later. A Unifier web service will be soon available.

NetWikiMaker. This tool generates (half) automatically a wiki on our reconstruction workflow. It contains information and data about the network reconstruction process such as different versions of draft metabolic networks files, parameters of tools, log files. It also displays the reactions, genes and metabolites that the workflow has found to be involved in the metabolic network, and provides a powerful search tool.

5.1.2. Dynamics and invariant-based prediction

We develop tools predicting some characteristics of a biological system behavior from incomplete sets of parameters or observations.

cadbiom. Based on Guarded transition semantic, this software provides a formal framework to help the modeling of biological systems such as cell signaling network. It allows investigating synchronization events in biological networks. [software][web server].

caspo: Cell ASP Optimizer This soft provides an easy to use software for learning Boolean logic models describing the immediate-early response of protein signaling networks. Given a network describing causal interactions, and a phospho-proteomics dataset, caspo is able to searches for optimal Boolean logic models explaining the dataset. Optimality includes both the size of the boolean network and the distance of predictions to real-data observations. It is useful to boolean networks inference, cancer research, drug discovery, and experimental design. It is used in the CellNOpt environment [0]. [python package][web server].

nutritionAnalyzer. This tool is dedicated to the computation of allocation for an extremal flux distribution. It allows quantifying the precursor composition of each system output (AIO) and to discuss the biological relevance of a set of flux in a given metabolic network by computing the extremal values of AIO coefficients. This approach enables to discriminate diets without making any assumption on the internal behaviour of the system [14][webserver][software and doc].

POGG. The POGG software allows scoring the importance and sensibility of regulatory interactions with a biological system with respect to the observation of a time-series quantitative phenotype. This is done by solving nonlinear problems to infer and explore the family of weighted Markov chains having a relevant asymptotic behavior at the population scale. Its possible application fields are systems biology, sensitive interactions, maximal entropy models, natural language processing. It results from our collaboration with the LINA-Nantes [1][matlab package].

5.1.3. Sequence annotation

We develop tools for discovery and search of complex pattern signatures within biological sequences, with a focus on protein sequences.

Logol Logol is a swiss-army-knife for pattern matching on DNA/RNA/Protein sequences, using a high-level grammar to permit a large expressivity. Allowed patterns can consist in a combination of motifs, structures (stem-loops, repeats), indels etc. It allows pseudo-knot identification, context sensitive grammatical formalism and full genome analysis. Possible fields of application are the detection of mutated binding sites or stem-loop identification (e.g. in CRISPR [0] [9]) [software]

[0] http://www.cellnopt.org/

Protomata learner This tool is a grammatical inference framework suitable for learning the specific signature of a functional protein family from unaligned sequences by partial and local multiple alignment and automata modeling. It performs a syntactic characterization of proteins by identification of conservation blocks on sequence subsets and modelling of their succession. Possible fields of application are new members discovery or study (for instance, for site-directed mutagenesis) of, possibly non-homologous, functional families and subfamilies such as enzymatic, signaling or transporting proteins [49][3] [web server]

5.1.4. Integration of toolboxes and platforms in webservices

Most of our software were designed as "bricks" that can combined through workflow application such as Mobyle. It worths considering them into larger dedicated environments to benefit from the expertise of other research groups.

Web servers In collaboration with the GenOuest ressource center, most our tools are made available through several web portals.

- The **mobyle@GenOuest portal** is the generic web server of our ressource center. It hosts the ingranalysis, meneco, caspo, lombarde and shogun tools [website].
- The **Mobyle@Biotempo server** is a mobyle portal for system biology with formal approaches. It hosts the memap, memerge, meneco, ingranalysis, cadbiom and pogg tools [website].
- The **GenOuest galaxy portal** now provides access to most tools for integrative biology and sequence annotation (access on demand).

Dr Motif This resource aims at the integration of different software commonly used in pattern discovery and matching. This resource also integrates Dyliss pattern search and discovery software [website].

ASP4biology and BioASP It is a meta-package to create a powerful environment of biological data integration and analysis in system biology, based on knowledge representation and combinatorial optimization technologies (ASP). It provides a collection of python applications which encapsulates ASP tools and several encodings making them easy to use by non-expert users out-of-the-box. [Python package][website].

ASP encodings repository This suite comprises projects related to applications of Answer Set Programming using Potassco systems (the Potsdam Answer Set Solving Collection, bundles tools for Answer Set Programming developed at the University of Potsdam). These are usually a set of encodings possibly including auxiliary software and scripts [respository].

5.2. New tools for integrative biology

Participants: Anne Siegel [contact], Jeanne Cambefort [contact], Guillaume Collet, Damien Eveillard, Sylvain Prigent, Marie Chevallier.

The tools MeMap and MeMerge were complemented with new tools in order to analyze reference networks from litterature database and to vizualize the product of reconstructed metabolic networks.

Unifier. [*input*: SBML file with Palsson's metabolites identifiers *output*: sbml file with standard identifiers for metabolites]. This software is a Decision Support Tool to help biologists to normalize a file, containing Palsson's identifiers to refer to reactions and metabolites, using well known identifiers. Submit a list of Palsson identifiers to retrieve the corresponding database entries. Typically it maps with Metacyc identifiers but it would be used with Kegg or other databases later. A Unifier web service will be soon available.

NetWikiMaker. This tool generates (half) automatically a wiki on our reconstruction workflow. It contains information and data about the network reconstruction process such as different versions of draft metabolic networks files, parameters of tools, log files. It also displays the reactions, genes and metabolites that the workflow has found to be involved in the metabolic network, and provides a powerful search tool.

[0] http://crispi.genouest.org/

5.3. New tools for dynamics

Participants: Jérémie Bourdon [contact], Jeanne Cambefort [contact], Damien Eveillard, Anne Siegel, Nathalie Théret, Santiago Videla [contact].

In 2014, the tool caspo was extended to new functionnalities.

caspo: Cell ASP Optimizer In the new version of caspo, *automated inference* of logical networks from experimental data allows for identifying admissible large-scale logic models saving a lot of efforts and without any a priori bias. Next, once a family a logical networks has been identified, one can suggest or *design new experiments* in order to reduce the uncertainty provided by this family. Finally, one can look for *intervention strategies* (i.e. inclusion minimal sets of knock-ins and knock-outs) that force a set of target species or compounds into a desired steady state. Altogether, this constitutes a pipeline for automated reasoning on logical signaling networks. Hence, the aim of caspo is to implement such a pipeline providing a powerful and easy-to-use software tool for systems biologists. [doc and download as a python package][web server].

FLUMINANCE Project-Team

5. New Software and Platforms

5.1. DenseMotion software - Estimation of 2D dense motion fields

Participant: Etienne Mémin.

This code allows the computation from two consecutive images of a dense motion field. The estimator is expressed as a global energy function minimization. The code enables the choice of different data models and different regularization functionals depending on the targeted application. Generic motion estimators for video sequences or fluid flows dedicated estimators can be set up. This software allows in addition the users to specify additional correlation based matching measurements. It enables also the inclusion of a temporal smoothing prior relying on a velocity vorticity formulation of the Navier-Stoke equation for Fluid motion analysis applications. The different variants of this code correspond to research studies that have been published in IEEE transaction on Pattern Analysis and machine Intelligence, Experiments in Fluids, IEEE transaction on Image Processing, IEEE transaction on Geo-Science end Remote Sensing. The binary of this code can be freely downloaded on the FLUID web site http://fluid.irisa.fr.

5.2. 2DLayeredMotion software - Estimation of 2D independent mesoscale layered atmospheric motion fields

Participant: Etienne Mémin.

This software enables to estimate a stack of 2D horizontal wind fields corresponding to a mesoscale dynamics of atmospheric pressure layers. This estimator is formulated as the minimization of a global energy function. It relies on a vertical decomposition of the atmosphere into pressure layers. This estimator uses pressure data and classification clouds maps and top of clouds pressure maps (or infra-red images). All these images are routinely supplied by the EUMETSAT consortium which handles the Meteosat and MSG satellite data distribution. The energy function relies on a data model built from the integration of the mass conservation on each layer. The estimator also includes a simplified and filtered shallow water dynamical model as temporal smoother and second-order div-curl spatial regularizer. The estimator may also incorporate correlation-based vector fields as additional observations. These correlation vectors are also routinely provided by the Eumetsat consortium. This code corresponds to research studies published in IEEE transaction on Geo-Science and Remote Sensing. It can be freely downloaded on the FLUID web site http://fluid.irisa.fr.

5.3. 3DLayeredMotion software - Estimation of 3D interconnected layered atmospheric motion fields

Participant: Etienne Mémin.

This software extends the previous 2D version. It allows (for the first time to our knowledge) the recovery of 3D wind fields from satellite image sequences. As with the previous techniques, the atmosphere is decomposed into a stack of pressure layers. The estimation relies also on pressure data and classification clouds maps and top of clouds pressure maps. In order to recover the 3D missing velocity information, physical knowledge on 3D mass exchanges between layers has been introduced in the data model. The corresponding data model appears to be a generalization of the previous data model constructed from a vertical integration of the continuity equation. This research study has been published in IEEE trans. on Geo-Science and Remote Sensing. The binary of this code can be freely downloaded on the FLUID web site http://fluid.irisa.fr.

5.4. Low-Order-Motion - Estimation of low order representation of fluid motion

Participants: Anne Cuzol, Etienne Mémin.

This code enables the estimation of a low order representation of a fluid motion field from two consecutive images.The fluid motion representation is obtained using a discretization of the vorticity and divergence maps through regularized Dirac measure. The irrotational and solenoidal components of the motion fields are expressed as linear combinations of basis functions obtained through the Biot-Savart law. The coefficient values and the basis function parameters are formalized as the minimizer of a functional relying on an intensity variation model obtained from an integrated version of the mass conservation principle of fluid mechanics. Different versions of this estimation are available. The code which includes a Matlab user interface can be downloaded on the FLUID web site http://fluid.irisa.fr. This program corresponds to a research study that has been published in the International Journal on computer Vision.

GALEN Project-Team

5. New Software and Platforms

5.1. Deformable Registration Software

Participant: Nikos Paragios [Correspondant].

deformable image and volume registration, is a deformable registration platform in C++ for the medical imaging community (publicly available at http://www.mrf-registration.net) developed mainly at Ecole Centrale, Technical University of Munich and University of Crete. This is the first publicly available platform which contains most of the existing metrics to perform registration under the same concept. The platform is used for clinical research from approximately 3,000 users worldwide.

5.2. Dense image and surface descriptors

Participant: Iasonas Kokkinos [Correspondant].

Scale-Invariant Descriptor, Scale-Invariant Heat Kernel Signatures DISD (publicly available at http://vision.mas.ecp.fr/Personnel/iasonas/descriptors.html) implements the SID, SI-HKS and ISC descriptors. SID (Scale-Invariant Descriptor) is a densely computable, scale- and rotation- invariant descriptor. We use a log-polar grid around every point to turn rotation/scalings into translation, and then use the Fourier Transform Modulus (FTM) to achieve invariance. SI-HKS (Scale-Invariant Heat Kernel Signatures) extract scale-invariant shape signatures by exploiting the fact that surface scaling amounts to multiplication and scaling of a properly sampled HKS descriptor. We apply the FTM trick on HKS to achieve invariance to scale changes. ISC (Intrinsic Shape Context) constructs a net-like grid around every surface point by shooting outwards and tracking geodesics. This allows us to build a meta-descriptor on top of HKS/SI-HKS that takes neighborhood into account, while being invariant to surface isometries.

5.3. Ranking with High-Order Information

Participant: Puneet Dokania [Correspondant].

Average precision optimization, high-order information, ranking The software (publicly available at http://cvn.ecp.fr/projects/ranking-highorder/) provides a convenient API for learning to rank with high-order information. The samples are ranked according to a scorethat is proportional to the difference of max-marginals of the positive and the negative class. The parameters of the score function are computed by minimizing an upper bound on the average precision loss. The software also provides an instantiation of the API for ranking samples according to their relevance to an action, using the poselet features.

5.4. Efficient bounding-based object detection

Participant: Iasonas Kokkinos [Correspondant].

branch-and-bound, parts detection, segmentation, DPMS implements branch-and-bound object detection, cutting down the complexity of detection from linear in the number of pixels to logarithmic (publicly available at http://vision.mas.ecp.fr/Personnel/iasonas/dpms.html). The results delivered are identical to those of the standard deformable part model detector, but are available in 5 to 20 times less time. This website has been visited 1500 times in 10 months.

5.5. Fast Primal Dual Strategies for Optimization of Markov Random Fields

Participant: Nikos Komodakis [Correspondant].

discrete optimization, Markov random field, duality, graph cuts, FASTPD is an optimization platform in C++ for the computer vision and medical imaging community (publicly available at http://www.csd.uoc.gr/~komod/FastPD/) developed mainly at Ecole Centrale and University of Crete. This is the most efficient publicly available platform in terms of a compromise of computational efficiency and ability to converge to a good minimum for the optimization of generic MRFs. The platform is used from approximately 1,500 users worldwide.

5.6. imaGe-based Procedural Modeling Using Shape Grammars

Participant: Iasonas Kokkinos [Correspondant].

procedural modeling, image-based building reconstruction, shape grammars GRAPES is a generic image parsing library based on re-inforcement learning (publicly available at http://vision.mas.ecp.fr/Personnel/teboul/grapesPage/index.php). It can handle grammars (binary-split, four-color, Hausmannian) and image-based rewards (Gaussian mixtures, Randomized Forests) of varying complexity while being modular and computationally efficient both in terms of grammar and image rewards. The platform is used from approximately 500 users worldwide.

5.7. Learning-based symmetry detection

Participant: Stavros Tsogkas [Correspondant].

Scale-Invariant Descriptor, Scale-Invariant Heat Kernel Signatures LBSD (publicly available at http://cvn.ecp.fr/personnel/tsogkas/code.html implements the learning-based approach to symmetry detection. It includes the code for running a detector, alongside with the ground-truth symmetry annotations that we have introduced for the Berkeley Segmentation Dataset (BSD) benchmark.

5.8. Texture Analysis Using Modulation Features and Generative Models

Participant: Iasonas Kokkinos [Correspondant].

Texture, modulation, generative models, segmentation, TEXMEG is a front-end for texture analysis and edge detection platform in Matlab that relies on Gabor filtering and image demodulation (publicly available at http://cvsp.cs.ntua.gr/software/texture/). Includes frequency- and time- based definition of Gabor- and other Quadrature-pair filterbanks, demodulation with the Regularized Energy Separation Algorithm and Texture/Edge/Smooth classification based on MDL criterion. The platform is used from approximately 250 users worldwide.

GENSCALE Project-Team

5. New Software and Platforms

5.1. Next Generation Sequencing

Participants: Alexan Andrieux, Gaëtan Benoit, Charles Deltel, Erwan Drezen, Dominique Lavenier, Claire Lemaitre, Antoine Limasset, Pierre Peterlongo, Chloé Riou, Guillaume Rizk.

GATB: Genome Analysis Tool Box

The GATB software toolbox aims to lighten the design of NGS algorithms. It offers a panel of high-level optimized building blocks to speed-up the development of NGS tools related to genome assembly and/or genome analysis. The underlying data structure is the de Bruijn graph, and the general parallelism model is multithreading. The GATB library targets standard computing resources such as current multicore processor (laptop computer, small server) with a few GB of memory. From high-level API, NGS programming designers can rapidly elaborate their own software based on domain state-of-the-art algorithms and data structures. The GATB library is written in C++ and is available under the GNU Affero GPL License. [contact: D. Lavenier] https://gatb.inria.fr

Mapsembler: targetted assembly

The Mapsembler tool enables the micro assembly of one or several area(s) of interest. It takes as input one or more read set(s) and a one or more sequences fragments used as "starters" of each micro-assembly. This task provides a way to check the existence/absence of an area for which the user has an *a priori* interest. Moreover, for each extended "starter", the output is either a flat fasta sequence or a portion of the assembly graph. In this latter case, Mapsembler offers a visualization interface on which each graph (including the read coverage per read set) can be visualized, annotated, and manipulated. [contact: P. Peterlongo] http://colibread.inria.fr/mapsembler2/

Leon: NGS data compressor

Leon is a lossless compression software that achieves compression of DNA sequences of high throughput sequencing data, without the need of a reference genome. Techniques are derived from assembly principles that better exploit NGS data redundancy. A reference is built de novo from the set of reads as a probabilistic de-Bruijn graph stored in a Bloom filter. Each read is encoded as a path in this graph, storing only an anchoring kmer and a list of bifurcations indicating which path to follow in the graph. This new method will allow to have compressed read files containing its underlying de-Bruijn Graph, thus directly re-usable by many tools relying on this structure. Leon achieved encoding of a *C. elegans* reads set with 0.7 bits/base, outperforming state of the art reference-free methods. Leon is available under the GNU Affero GPL License. [contact: C. Lemaitre] https://gatb.inria.fr/software/leon/

Bloocoo: read corrector

Bloocoo is a k-mer spectrum-based read error corrector, designed to correct large datasets with a very low memory footprint. It uses the disk streaming k-mer counting algorithm contained in the GATB library, and inserts solid k-mers in a bloom-filter. The correction procedure is similar to state-of-the-art approaches. Bloocoo yields similar results while requiring far less memory: as an example, it can correct whole human genome re-sequencing reads at 70 x coverage with less than 4GB of memory [32]. [contact: C. Lemaitre] https://gatb.inria.fr/bloocoo-read-corrector/

MindTheGap: insertion variant detection

MindTheGap is a software that performs detection and assembly of DNA insertion variants in NGS read datasets with respect to a reference genome. It takes as input a set of reads and a reference genome. It outputs two sets of FASTA sequences: one is the set of breakpoints of detected insertion sites, the other is the set of assembled insertions for each breakpoint. For each breakpoint, MindTheGap either returns a single insertion sequence (when there is no assembly ambiguity), or a set of candidate insertion sequences (due to ambiguities) or nothing at all (when the insertion is too complex to be assembled). MindTheGap performs de novo assembly using the de Bruijn Graph implementation of GATB. Hence, the computational resources required to run MindTheGap are significantly lower than that of other assemblers. [contact: C. Lemaitre] http://mindthegap.genouest.org/

TakeABreak: de novo inversion variant discovery
TakeABreak is a tool that can detect inversion breakpoints directly from raw NGS reads, without the need of any reference genome and without de novo assembling the genomes. Its implementation is based on the Genome Assembly Tool Box (GATB) library, and has a very limited memory impact allowing its usage on common desktop computers and acceptable runtime (Illumina reads simulated at 80x coverage from human chromosome 22 can be treated in less than two hours, with less than 1GB of memory). TakeABreak is available under the GNU Affero GPL License. [contact: C. Lemaitre] http://colibread.inria.fr/software/takeabreak/

discoSnp: de novo SNP discovery
The discoSnp tool detects isolated SNPs given one, two or more raw read set(s) without using any reference genome. discoSnp ranks predictions and outputs quality and coverage per allele. Compared to finding isolated SNPs using a state-of-the-art assembly and mapping approach, discoSnp requires significantly less computational resources, shows similar precision and recall values, and highly ranked predictions are less likely to be false positives. [contact: P. Peterlongo] http://colibread.inria.fr/discosnp/

5.2. High throughput sequence comparisons

Participants: Sébastien Brillet, Erwan Drezen, Dominique Lavenier, Pierre Peterlongo, Ivaylo Petrov.

KLAST: bank-to-bank alignment search tool
KLAST is a fast, accurate and NGS scalable bank-to-bank sequence similarity search tool providing significant accelerations of seeds-based heuristic comparison methods, such as the Blast suite. KLAST is a new optimized implementation of the PLAST algorithm to which several improvements have been made in 2014. KLAST is fully designed to compare query and subject comprised of large sets of DNA, RNA and protein sequences. It is significantly faster than original PLAST, while providing comparable sensitivity to BLAST and SSearch algorithms. KLAST contains a fully integrated data-filtering engine capable of selecting relevant hits with user-defined criteria (E-Value, identity, coverage, alignment length, etc.). Klast is developed with the Korilog Company and an academic version is now freely available for the scientific community [contact: D. Lavenier]. [34] https://koriscale.inria.fr/klast/

COMMET: de novo comparison of metagenomic datasets
Commet is an extension of the Comparead tool that proposes to compute similarity between set of raw non assembled (and usually non-assemblable with current state of the art assemblers) reads. Commet enables to factorize computations when n read sets have to be compared all together. Moreover, Commet proposes a new representation of sub-read sets that has the main advantages to save huge disk space and to enable efficient logical operations between sub-read sets. [contact: P. Peterlongo] https://colibread.inria.fr/software/commet/

5.3. 3D Protein structures

Participants: Douglas Goncalves, Antonio Mucherino.

MD-jeep version 0.2
MD-jeep is the result of a strong collaboration among Antonio Mucherino, Leo Liberti, Carlile Lavor and Nelson Maculan. Over the years, PhD and postdoc students under our supervision have also been contributing to this research topic. The new method for the computation of atomic coordinates in MD-jeep v.0.2 was developed in collaboration with Douglas Soares Gonçalves [13], who was a postdoc student in Rennes for one year [contact: A. Mucherino]. http://www.antoniomucherino.it/en/mdjeep.php

IBIS Project-Team

4. New Software and Platforms

4.1. Genetic Network Analyzer (GNA)

Participants: Hidde de Jong [Correspondent], Michel Page, François Rechenmann.

Keywords. Gene regulatory networks, qualitative simulation, model checking

GENETIC NETWORK ANALYZER (GNA) is the implementation of methods for the qualitative modeling and simulation of gene regulatory networks developed in the IBIS project. The input of GNA consists of a model of the regulatory network in the form of a system of piecewise-linear differential equations (PLDEs), supplemented by inequality constraints on the parameters and initial conditions. From this information, GNA generates a state transition graph summarizing the qualitative dynamics of the system. In order to analyze large graphs, GNA allows the user to specify properties of the qualitative dynamics of a network in temporal logic, using high-level query templates, and to verify these properties on the state transition graph by means of standard model-checking tools, either locally installed or accessible through a remote web server. GNA is currently distributed by the company Genostar, but remains freely available for academic research purposes. The current version is GNA 8.7. In comparison with the previously distributed versions, GNA 8.7 has the following additional functionalities: (1) it supports the editing and visualization of regulatory networks, in an SBGN-compatible format, (2) it semi-automatically generates a prototype model from the network structure, thus accelerating the modeling process, and (3) it allows models to be exported in the SBML Qual standard. For more information, see http://www-helix.inrialpes.fr/gna.

4.2. WellReader, WellFARE, and WellInverter

Participants: Johannes Geiselmann, Hidde de Jong [Correspondent], Michel Page, Delphine Ropers, Valentin Zulkower.

Keywords. Gene expression, reporter gene data

WELLREADER is a program for the analysis of gene expression data obtained by means of fluorescent and luminescent reporter genes. WELLREADER reads data files in an XML format or in a format produced by microplate readers, and allows the user to detect outliers, perform background corrections and spline fits, compute promoter activities and protein concentrations, and compare expression profiles across different conditions. WELLREADER has been written in MATLAB and is available under an LGPL licence, both as source code (M files) and compiled code (platform-specific binary files). For more information, see: http://ibis.inrialpes.fr/article957.html.

In the past year, we developed novel approaches towards the analysis of reporter gene data, based on regularized linear inversion (Section 5.3). The linear inversion methods were implemented in the Python package WELLFARE, relying on the scientific Python libraries NumPy and SciPy. In addition, the package provides utilities for parsing data files and removing possible outliers from the absorbance and fluorescence signals. The WELLFARE package is available under an LGPL license, but has also been integrated into a web application called WELLINVERTER, which provides a graphical user interface allowing access to the linear inversion methods through a web browser (Figure 5). The user can upload data files by means of WELLINVERTER, remove outliers and subtract background, and launch the procedures for computing growth rates, promoter activities, and protein concentrations. For more information, see: http://ibis.inrialpes.fr/article1080.html?menu=menu4.

KALIFFE Project-Team

5. New Software and Platforms

5.1. New Softwares

5.1.1. Hope : High Order Program for Energy

This software is focused on the numerical simulation of 2D transport equation using fully deterministic methods (high order finite difference solvers, semi-Lagrangian methods).

Numerical simulation of guiding center model [9]

We consider the diocotron instability for an annular electron layer. This plasma instability is created by two sheets of charge slipping past each other and is the analog of the Kelvin-Helmholtz instability in fluid mechanics. We propose a comparison of two different numerical methods : the mixed method (top): this method uses alternatively a semi-Lagrangian and finite difference method with fifth order Hermite WENO reconstruction. The choice is made automatically according to a good preservation of mass (the finite difference method is conservative). the semi-Lagrangian (bottom): this method is based on a cubic spline interpolation for the reconstruction of the distribution function.

Numerical simulation in a D shape [9]

This simulation illustrates an instability development of the solution to the guiding-center model in a D-shaped domain. We present the difference between the perturbed density and the steady state density. An instability develops and generates small filaments. It correspond to the motion of the density in the transverse plane of the tokamak.

Figure 2 illustrates the evolution of density governed by the guiding-center model. We present the difference between the perturbed density and the steady state density, *i.e.*$\delta\rho(t) = \overline{\rho}(t) - \overline{\rho}_0$. We observe that the difference of density $\delta\rho$ revolves, and small filaments appear at time $t = 200$. Until the time $t = 300$, we can clearly identify the filaments.

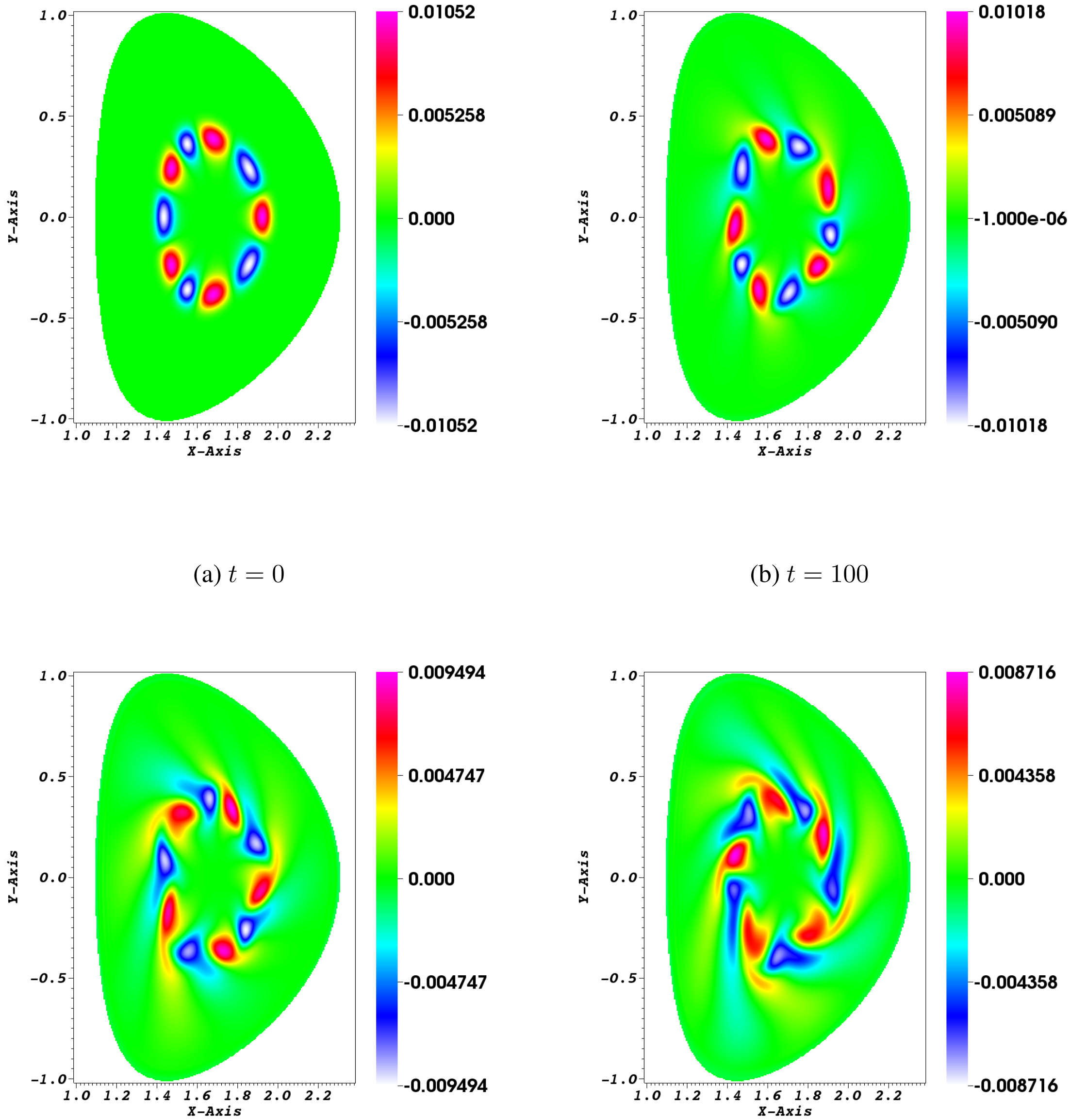
5.1.2. Towards 4D numerical simulations

The discretization of the Drift-Kinetic model can be developed very similarly as the one for the guiding-center model. Here, we present some principle discretization steps.

The Vlasov equation of system can be split into three equations :

$$\left\{\begin{array}{l}\dfrac{\partial f}{\partial t} + \mathbf{U}\cdot\nabla_{\mathbf{x}_\perp} f = 0,\\ \dfrac{\partial f}{\partial t} + v_\parallel \partial_z f = 0,\\ \dfrac{\partial f}{\partial t} + E_\parallel \partial_{v_\parallel} f = 0.\end{array}\right.$$

This test represents a snapshot of the charge density when an instability occurs (ion turbulence simulation). This simulation has been realized by different methods but in cylindrical coordinates, here we perform numerical simulation in Cartesian coordinates on a uniform grid. The discretization of the Drift-Kinetic model can be developed very similarly as the one for the guiding-center model.



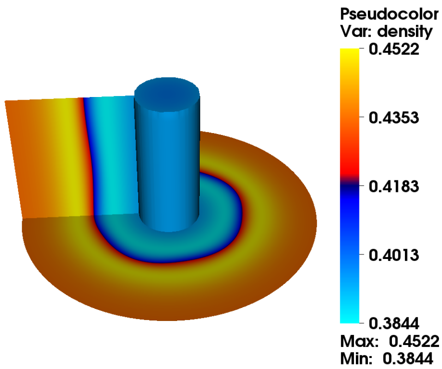
(a) $t = 0$

(b) $t = 100$

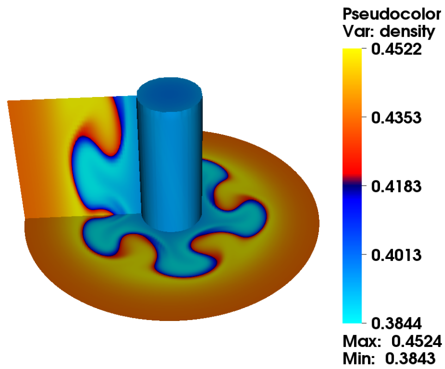
(c) $t = 200$

(d) $t = 300$

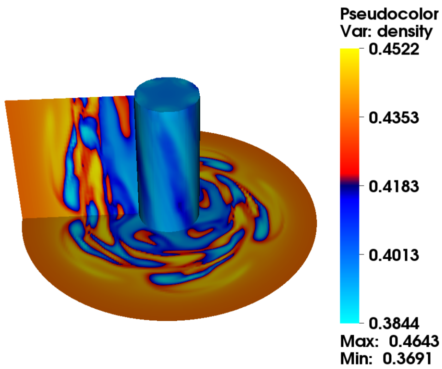
Figure 2. Instability simulation for guiding-center model in D-shaped domain. The difference between the perturbed density and the steady state density is presented, i.e. $\delta\rho(t) = \overline{\rho}(t) - \overline{\rho}_0$.



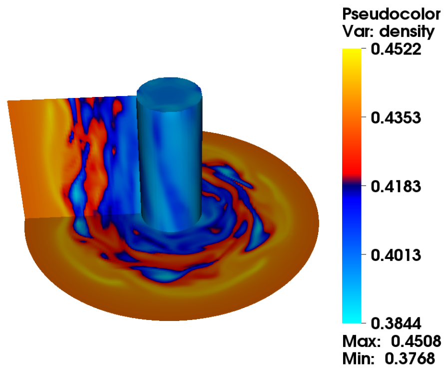
(a) $t = 2000$



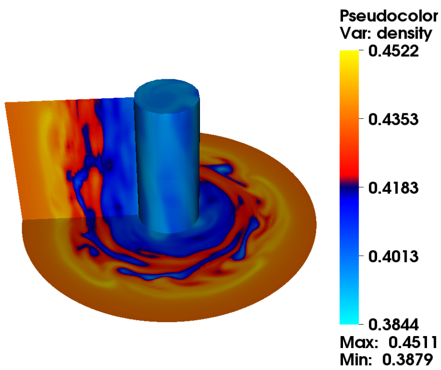
(b) $t = 3000$



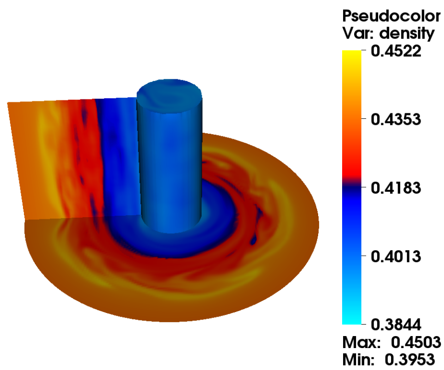
(c) $t = 4000$



(d) $t = 5000$



(e) $t = 6000$



(f) $t = 8000$

Figure 3. Evolution of ion turbulence. The distribution function is shown for the velocity $v_{\|} = 0$*. The mesh size is* $n_x = n_y = 128, n_z = 32, n_v = 65$*. Mixed Semi-Lagrangian/finite difference method is used.*

LEMON Team

5. New Software and Platforms

5.1. SW2D

Participant: Vincent Guinot.

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

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

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

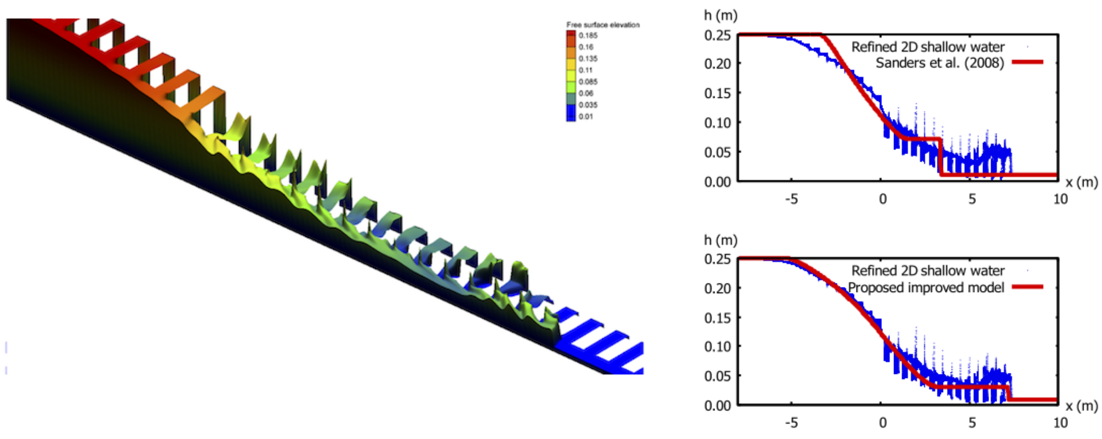


Figure 2. Propagation of a flood wave into a channel with lateral storage. Refined 2D simulation using the SW2D computational code

5.2. Stochastic Downscaling Method

Participant: Antoine Rousseau.

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

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

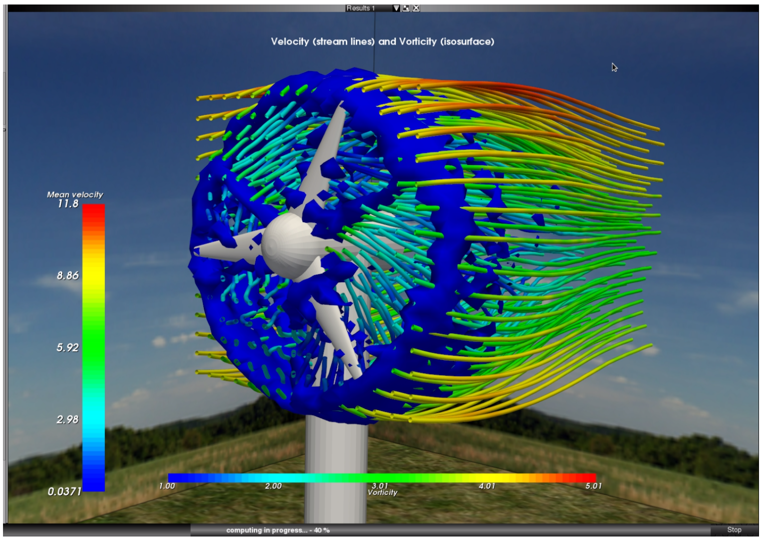


Figure 3. Velocity streamlines and vorticity around a wind mill (artistic view). WINDPOS Project.

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

5.3. Action Dépollution

Participants: Antoine Rousseau, Alexis Pacholik.

Action Dépollution (see website in french) is a serious game made for learning how to purify fast and well a water reservoir, such as lakes. In the scope of the international initiative Mathematics of Planet Earth, this game shows an application of mathematics related to environmental education and sustainable development. The player can act as a researcher, that compares different strategies and looks for the best solution. The conception has been achieved in collaboration with the Inria project-team MODEMIC, and the realization with the help of the start-up Funkadelichik, sponsored by the french consortium Cap'Maths and Inria (Direction de la Communication).
This work is in connection with the INRA/Inria patent [19].

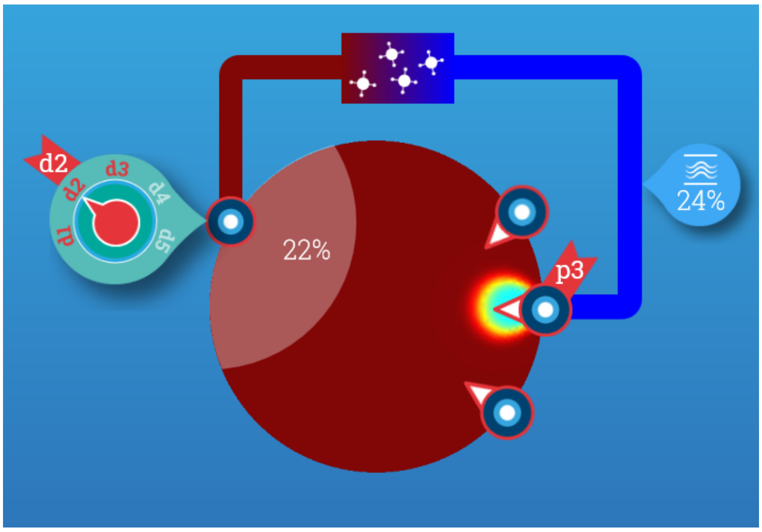


Figure 4. Player interface. Serious game Action Dépollution.

LIFEWARE Team

5. New Software and Platforms

5.1. The Biochemical Abstract Machine

Participants: François Fages, François-Marie Floch, Thierry Martinez, Sylvain Soliman, Pauline Traynard.

5.1.1. BIOCHAM V3.6

The Biochemical Abstract Machine (BIOCHAM) is a software environment for modeling and analyzing biochemical reaction systems, making simulations, performing static analyses, specifying behaviors in temporal logic. It is distributed under the GPL license since 2002.

The new features of version v3.6 released in October 2014 include:

- Hybrid boolean-stochastic-continuous models and simulations
- Quantitative temporal logic patterns with dedicated solvers (based on [20])
- Trace simplifications (based on [19])
- `export_sbml3` added
- `curate_model` and `curate_sbml` added (based on [8])
- Refinements of the types for command arguments (no effect on Biocham models)
- Bug fixes

5.1.2. BIOCHAM-web

BIOCHAM-WEB is a web service which makes it possible to try BIOCHAM on line without any installation, through a spreadsheet. A new version BIOCHAM-web was released in March 2014 and is kept evolving since.

That web service will evolve to a complete graphical user interface, named BIOCHAM-gui, and will replace the current graphical user interface.

5.1.3. BIOCHAM-parallel

A (non-distributed) parallel version of BIOCHAM is also maintained for our own use on a cluster of 10000 cores at GENCI CINES. This version speeds-up (linearly in the number of processors) the search of parameter values by parallelizing the evaluation of the fitness function (computed by numerical integration) for both the different parameter sets and the different conditions (perturbations, gene knock down or stress).

5.2. ClpZinc

Participants: François Fages, Thierry Martinez, Philippe Morignot, Sylvain Soliman.

CLP2ZINC is a rule-based modeling language for constraint programming. It extends the MiniZinc modeling language with Horn clauses which can be used to express search strategies as constraints in the model. This system is developed in the framework of the ANR Net-WMS-2 project and is a follow-up of the RULES2CP modeling language.

5.3. CellStar: Long-term tracking of single cells from brightfield microscopy images

Participants: Grégory Batt, Pascal Hersen, Artémis Llamosi, Szymon Stoma.

In close collaboration with Kirill Batmanov, Cédric Lhoussaine and Cristian Versari from the LIFL (CNRS/Lille Univ), we developed CELLSTAR, a tool-chain for image processing and analysis dedicated to segmentation and tracking of yeast cells in brightfield time-lapse microscopy movies. To estimate algorithm quality we developed a benchmark made of manually-verified images illustrating various situations. On this benchmark, CELLSTAR outperformed 5 other state-of-the-art methods. The tool-chain is implemented in MATLAB and is provided together with the Python YEAST IMAGE TOOLKIT benchmark tool.

5.4. Other software

The team also develops several software primarily for internal use. Some of them are specific to particular hardware and are not distributed. Some others are general purpose and currently on the web page for free downloading.

M3DISIM Team

5. New Software and Platforms

5.1. FELISCE

Participants: Dominique Chapelle, Sébastien Gilles [correspondant], Sébastien Imperiale, Philippe Moireau.

FELISCE – standing for "Finite Elements for LIfe SCiences and Engineering" – is a new finite element code which the MACS and REO teams have decided to jointly develop in order to build up on their respective experiences concerning finite element simulations. One specific objective of this code is to provide in a unified software environment all the state-of-the-art tools needed to perform simulations of the complex cardiovascular models considered in the two teams – namely involving fluid and solid mechanics, electrophysiology, and the various associated coupling phenomena. FELISCE is written in C++, and may be later released as an opensource library. See https://gforge.inria.fr/projects/felisce/.

In FELISCE we have prepared a branch called HappyHeart, which aims at providing a user-friendly interface able to deal efficiently with complex cardiovascular simulations. Started in 2013, the code is already quite large (about 55 000 lines of code in almost 700 different files) and its core is about to be complete in early 2015. It includes among others full HPC functionalities, high-order finite elements, physics coupling and topology capabilities. Our purpose will then be to use the library to implement the sophisticated cardiovascular models of the team and couple them with Verdandi (data assimilation library) to provide patient-specific simulations.

- Software benefit: HappyHeart is a multiphysics HPC FEM Library with cardiac simulation concerns
- Type of human computer interaction: Command line and configuration files.
- OS/Middleware: MacOS, Linux.
- Required library or software: OpenMpi (parallelism), Petsc (linear algebra), Seldon (linear algebra), Parmetis (partitioner), Mumps (solver), Ops (input parameter file management), STL and Yuni (generic C++ libraries).
- Programming language: C++ 11/14.
- Documentation: Doxygen and user's manual in English.

5.2. HeartLab

Participants: Matthieu Caruel, Dominique Chapelle, Alessandro Felder, Philippe Moireau [correspondant].

The heartLab software is a library written in (64-bit compatible) Matlab and C (mex functions) designed to perform both simulation and estimation (based on various types of measurements, e.g. images) of the heart mechanical behavior. Started in 2006, it is already quite large (about 60,000 lines), and is used within various collaborations.

The code relies on OpenFEM – to which the team has previously contributed, see http://www.openfem.net – for the finite element computations, and the implementation was performed with a particular concern for modularity, since modeling and estimation use the same finite element operators. This modularity also allows to couple the code with other FEM solvers, such as LifeV and Mistral developed in the Reo team-project. In particular, we are now able to include perfusion and electrical coupling with LifeV using PVM, and fluid-structure interaction using Mistral.

We also included geometric data and tools in the code to define cardiac anatomical models compatible with the simulation requirements in terms of mesh quality, fiber direction data defined within each element, and the referencing necessary for handling boundary conditions and estimation, in particular. These geometries are analytical or come from computerized tomography (CT) or magnetic resonance (MR) image data of humans or animals.

We incorporated numerous non-linear data assimilation observation operators based on medical imaging post-processing to be able to now perform estimation with a large variety of medical imaging modalities. And recently we have worked on generalized micro-macro cardiac law using stochastic formulations.

5.3. Verdandi

Participants: Aurora Armiento [Mamba team], Dominique Chapelle, Annabelle Collin, Vivien Mallet [Clime team], Karine Mauffrey, Philippe Moireau [correspondant].

Verdandi is an opensource (LGPL) software library aiming at providing data assimilation methods and related tools. Mainly targeted at large systems arising from the discretization of PDEs, it is intentionally devised as generic, which allows for applications in a wide range of problems (biology and medicine, environment, image processing...). See also the web page http://verdandi.gforge.inria.fr/, with a complete documentation in English. The first stable version (1.0) was released in June 2012 and contains most of the major data assimilation algorithms of both variational and sequential types. The current version (1.6) contains additional estimation algorithm and parallel capabilities. Note that some specific developments are performed with particular regard to cardiac modeling applications, as Verdandi is partly funded by – and distributed within – the VPH-Share and VP2HF projects and is now referenced in the peer-reviewed article [4].

- ACM: Mathematical software
- AMS: System theory; control
- Software benefit: Verdandi is the only *generic* data assimilation library
- License: LGPL (2.1 or any later version)
- Type of human computer interaction: Command line and configuration files
- OS/Middelware: Linux, MacOS ou Windows
- Required library or software: Seldon (LGPL, http://seldon.sourceforge.net/)
- Programming language: C++, ISO/IEC 14882: I998(E) Python, version 2.6
- Documentation: Doxygen and utilisation manual in English

Moreover a Matlab module called VerdandinMatlab is developed in the team for pedagogical and test purposes.

MAGIQUE-3D Project-Team

5. New Software and Platforms

5.1. Hou10ni

Participant: Julien Diaz [correspondant].

This software, written in FORTRAN 90, simulates the propagation of waves in heterogeneous 2D and 3D media in time-domain and in frequency domain. It is based on an Interior Penalty Discontinuous Galerkin Method (IPDGM) and allows for the use of meshes composed of cells of various order (p-adaptivity in space).

This year, we have implemented the 3D version for the simulation of elastodynamic waves. This version handles polynomials of arbitrary order while the previous one was only able to deal with polynomials of degree up to three.

We have also improved the parallelism by coupling the code to a mesh partitioner and we have totally rewritten the code to handle MPI parallelism both for the construction of the matrices and for the time scheme.

5.2. Montjoie

Participant: Marc Duruflé [correspondant].

`Montjoie` is a code developed by Marc Duruflé with contributions of students, including Juliette Chabassier during her PhD. It provides a C++ framework for solving partial differential equations on unstructured meshes with finite element-like methods (continuous finite element, discontinuous Galerkin formulation, edge elements and facet elements). The handling of mixed elements (tetrahedra, prisms, pyramids and hexahedra) has been implemented for these different types of finite elements methods in the context of Morgane Bergot's PhD. Several applications are currently available : wave equation, elastodynamics, aeroacoustics, Maxwell's equations. In 2014, the implementation of linearized Euler equations and Galbrun's equation has been improved and extended to the axisymmetric case. Raman effect has been implemented in the 1-D non-linear Schrödinger equation.

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

5.3. Elasticus

Participant: Simon Ettouati.

Within the framework of the strategic action DIP, Magique-3D collaborates with Total to develop a computing platform, DIVA, meant to produce accurate images of the subsurface. To achieve this, approximate solutions of the first-order wave problem are computed thanks to a Discontinuous Galerkin (DG) Method. It is increasingly difficult to include new numerical schemes developped in the team in the industrial and highly parallel environment of Total.

Elasticus is a sequential library, independent of DIVA and developped in Fortran, to simulate wave propagation in geophysical environment, based on a DG method. It is meant to help PhD students and post-doctoral fellows to easily implement their algorithms in the library. Thus, readability of the code is privileged to optimization of its performances. Developped features should be easily transferred in the computing platform of Total. Contrary to DIVA which only computes approximate solutions with P1, P2 and P3 elements, Elasticus manages arbitrary orders for the spatial discretization with DG method. Matrices on the reference element for arbitrary orders are computed thanks to a library developped by J. Diaz.

5.4. DIVA-DG

Participants: Lionel Boillot, Marie Bonnasse-Gahot, Théophile Chaumont-Frelet, Jérôme Luquel.

DIVA-DG is the simulation code that we develop in collaboration with our partner Total. This year we have implemented

- 2D/3D anisotropic elastic Absorbing Boundary Conditions for time-domain.
- 2D elastic imaging conditions.
- 2D multiscale strategy to take into account fine scale heterogeneities on coarse meshes in frequency domain.
- Hybrydized Discontinuous Galekin method for 2D elastodynamic in frequency domain.

MAGNOME Project-Team

5. New Software and Platforms

5.1. Magus: Genome exploration and analysis

Participants: David James Sherman [correspondant], Pascal Durrens, Florian Lajus, Xavier Calcas.

The MAGUS genome annotation system integrates genome sequences and sequences features, *in silico* analyses, and views of external data resources into a familiar user interface requiring only a Web navigator. MAGUS implements annotation workflows and enforces curation standards to guarantee consistency and integrity. As a novel feature the system provides a workflow for simultaneous annotation of related genomes through the use of protein families identified by *in silico* analyses; this results in an n-fold increase in curation speed, compared to curation of individual genes. This allows us to maintain standards of high-quality manual annotation while efficiently using the time of volunteer curators. MAGUS can be used on small installations with a web server and a relational database on a single machine, or scaled out in clusters or elastic clouds using Apache Cassandra for NoSQL data storage and Apache Hadoop for Map-Reduce (figure 1). For more information see the MAGUS Gforge web site. [0] MAGUS 2.0 was developed in an Inria Technology Development Action (ADT) and is distributed with an open-source license.

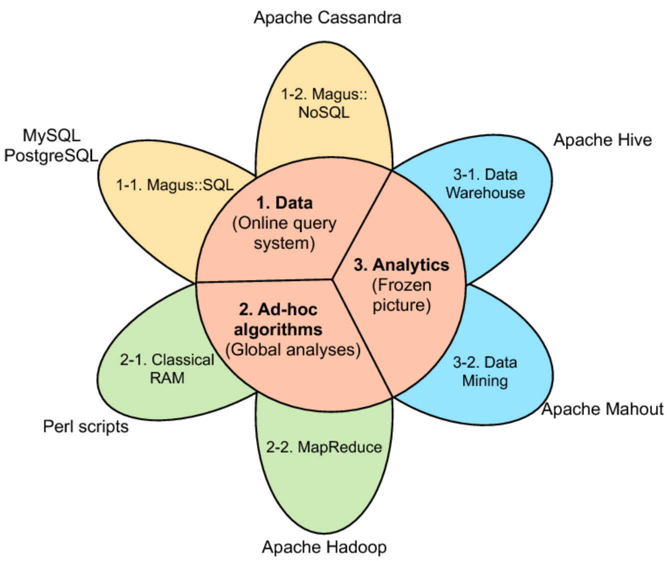


Figure 1. General architecture of the Tsvetok system implemented in MAGUS, showing the role of the NoSQL (Apache Cassandra) and Map-Reduce (Apache Hadoop) paradigms

5.2. Pantograph: Inference of metabolic networks

Participants: David James Sherman [correspondant], Pascal Durrens, Anna Zhukova.

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

Pantograph is a software tool developed by Nicolás Loira for his thesis, that infers whole-genome metabolic models for eukaryote cell factories from reference models and genome comparison. A novel feature of Pantograph is that it uses expert knowledge implicitly encoded in the scaffold's gene associations, and explicitly transfers this knowledge to the new model. Pantograph is available under an open-source license. For more information see the Pantograph Gforge web site. [0].

5.3. Mimoza: Generalizing and Visualizing Metabolic Models

Participants: David James Sherman [correspondant], Anna Zhukova.

Mimoza uses metabolic model generalization and cartographic paradigms to allow human experts to explore a metabolic model in a hierarchical manner. The software creates an zoomable representation of a model submitted by the user in SBML [0] format. The most general view represents the compartments of the model; the next view shows the visualization of generalized versions of reactions and metabolites in each compartment (see section 6.3); and the most detailed view visualizes the initial model with the generalization-based layout (where similar metabolites and reactions are placed next to each other). The zoomable representation is implemented using the Leaflet [0] JavaScript library for mobile-friendly interactive maps. Users can click on reactions and compounds to see the information about their annotations. The resulting map can be explored on-line, or downloaded in a COMBINE archive. The software and examples are available at http://mimoza.bordeaux.inria.fr.

5.4. Génolevures On Line: Comparative Genomics of Yeasts

Participants: Pascal Durrens [correspondant], David James Sherman.

The Génolevures online database provides archival data for exploring the annotated genome sequences of more than 20 genomes, determined and manually annotated by the Génolevures Consortium to facilitate comparative genomic studies of hemiascomycetous yeasts. Data are presented with a focus on relations between genes and genomes: conservation of genes and gene families, speciation, chromosomal reorganization and synteny. Génolevures online uses our open-source MAGUS system for genome navigation, with project-specific extensions developed by MAGNOME. For more information see the Génolevures web site. [0]

[0]http://pathtastic.gforge.inria.fr
[0]http://sbml.org
[0]http://leafletjs.com
[0]http://www.genolevures.org/

MAMBA Team

5. New Software and Platforms

5.1. Logiciels

5.1.1. TiQuant

Systems biology and medicine on histological scales require quantification of images from histological image modalities such as confocal laser scanning or bright field microscopy. The latter can be used to calibrate the initial state of a mathematical model, and to evaluate its explanatory value, which hitherto has been little recognized ([7]). We generated a software for image analysis of histological material and demonstrated its use in analysing liver confocal micrografts, called TiQuant (Tissue Quantifier) [10]. The software is part of an analysis chain detailing protocols of imaging, image processing and analysis in liver tissue, permitting 3D reconstructions of liver lobules down to a resolution of less than a micrometer. (This work is a collaboration with the group of JG Hengstler, IfADo, Germany)

5.1.2. TiSim

We advanced the complementary software TiSim (Tissue Simulator) that will soon be provided. TiSim permits agent-based simulations of multicellular systems and can be directly fed by processed image data from TiQuant.

MASAIE Project-Team (section vide)

MNEMOSYNE Project-Team

5. New Software and Platforms

5.1. Positioning

Our previous works in the domain of well-defined distributed asynchronous adaptive computations [43], [40], [45] have already made us define a library (DANA [39]), closely related to both the notion of artificial neural networks and cellular automata. From a conceptual point of view, the computational paradigm supporting the library is grounded on the notion of a unit that is essentially a (vector of) potential that can vary along time under the influence of other units and learning. Those units can be organized into layers, maps and networks.

We also gather in the middleware EnaS (that stands for *Event Neural Assembly Simulation*; cf. http://gforge.inria.fr/projects/enas) our numerical and theoretical developments, allowing to simulate and analyze so called "event neural assemblies".

We will also have to interact with the High Performance Computing (HPC) community, since having large scale simulations at that mesoscopic level is an important challenge in our systemic view of computational neuroscience. Our approach implies to emulate the dynamics of thousands, or even millions, of integrated computational units, each of them playing the role of a whole elementary neural circuit (e.g. the microcolumn for the cortex). Mesoscopic models are considered in such an integrative approach, in order to exhibit global dynamical effect that would be hardly reachable by compartment models involving membrane equations or even spiking neuron networks.

The vast majority of high performance computing softwares for computational neuroscience addresses sub-neural or neural models [30], but coarser grained population models are also demanding for large scale simulations, with fully distributed computations, without global memory or time reference, as it is specified in (*cf.* § 3.2).

5.2. Dana

Participant: Nicolas Rougier.

DANA [39] is a python framework (http://dana.loria.fr) whose computational paradigm is grounded on the notion of a unit that is essentially a set of time dependent values varying under the influence of other units via adaptive weighted connections. The evolutions of a unit's value are defined by a set of differential equations expressed in standard mathematical notation which greatly ease their definition. The units are organized into groups that form a model. Each unit can be connected to any other unit (including itself) using a weighted connection. The DANA framework offers a set of core objects needed to design and run such models. The modeler only has to define the equations of a unit as well as the equations governing the training of the connections. The simulation is completely transparent to the modeler and is handled by DANA. This allows DANA to be used for a wide range of numerical and distributed models as long as they fit the proposed framework (e.g. cellular automata, reaction-diffusion system, decentralized neural networks, recurrent neural networks, kernel-based image processing, etc.).

5.3. Virtual Enaction

Participants: Frédéric Alexandre, André Garenne, Nicolas Rougier, Thierry Viéville.

The computational models studied in this project have applications that extend far beyond what is possible to experiment yet in human or non-human primate subjects. Real robotics experimentations are also impaired by rather heavy technological constraints; for instance, it is not easy to dismantle a given embedded system in the course of emerging ideas. The only versatile environment in which such complex behaviors can be studied both globally and at the level of details of the available modeling is a virtual environment, as in video games, Such a system can be implemented as “brainy-bot” (a programmed player based on our knowledge of the brain architecture) which goal is to survive in a complete manipulable environment.

In order to attain this rather ambitious objective we both (i) deploy an existing open-source video game middleware in order to be able to shape the survival situation to be studied and (ii) revisit the existing models in order to be able to integrate them as an effective brainy-bot. It consists of a platform associated to a scenario that is the closest possible to a survival situation (foraging, predator-prey relationship, partner approach to reproduction) and in which it is easy to integrate an artificial agent with sensory inputs (visual, touch and smell), emotional and somatosensory cues (hunger, thirst, fear, ..) and motor outputs (movement, gesture, ..) connected to a "brain" whose architecture will correspond to the major anatomical regions involved in the issues of learning and action selection (cortex areas detailed here, basal ganglia, hippocampus, and areas dedicated to sensorimotor processes). The internal game clock can be slowed down enough to be able to run non trivial brainy-bot implementations. This platform [13] has already being used by two students of the team and is now a new deliverable of the KEOpS project.

MODEMIC Project-Team

5. New Software and Platforms

5.1. Action Depollution

Participant: Alain Rapaport.

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

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

5.2. VITELBIO (VIrtual TELluric BIOreactors)

Participants: Jérôme Harmand, Alain Rapaport.

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

5.3. Mass-structured chemostat simulators

Participants: Fabien Campillo, Coralie Fritsch.

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

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

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

MOISE Project-Team

5. New Software and Platforms

5.1. Adaptive Grid Refinement

Participants: Laurent Debreu, Marc Honnorat.

AGRIF (Adaptive Grid Refinement In Fortran, [85], [83]) is a Fortran 90 package for the integration of full adaptive mesh refinement (AMR) features within a multidimensional finite difference model written in Fortran. Its main objective is to simplify the integration of AMR potentialities within an existing model with minimal changes. Capabilities of this package include the management of an arbitrary number of grids, horizontal and/or vertical refinements, dynamic regridding, parallelization of the grids interactions on distributed memory computers. AGRIF requires the model to be discretized on a structured grid, like it is typically done in ocean or atmosphere modelling. As an example, AGRIF is currently used in the following ocean models: MARS (a coastal model developed at IFREMER-France), ROMS (a regional model developed jointly at Rutgers and UCLA universities), NEMO ocean modelling system (a general circulation model used by the French and European scientific community) and HYCOM (a regional model developed jointly by University of Miami and the French Navy).
Recent applications produced by the NEMO-AGRIF system are described in [12],[19]. AGRIF is licensed under a GNU (GPL) license and can be downloaded at its web site (http://ljk.imag.fr/MOISE/AGRIF/index.html).

5.2. NEMOVAR

Participant: Arthur Vidard.

NEMOVAR is a state-of-the-art multi-incremental variational data assimilation system dedicated to the European ocean modelling platform NEMO for research and operational applications. It is co-developed by MOISE, CERFACS (FR), ECMWF (EU) and MetOffice (UK) under the CeCILL license, written in Fortran and Python. It is now in use in both ECMWF and MetOffice for their operational oceanic forecasting systems. It has also been used for specific studies in collaboration with Mercator-Ocean, LPO, LOCEAN and LEGI in France and University of Namur in Belgium. It has been adopted as the ocean analysis component in the FP7 project ERA-Clim2 (01/2014-12/2016).

Previously part of NEMOVAR, NEMO-TAM (Tangent and adjoint models for NEMO) that have been developed by the MOISE team will be now distributed directly by the NEMO consortium. The first official tagged release including NEMO-TAM has been published early 2013.

5.3. R Packages for Uncertainty Quantification

Participants: Laurent Gilquin, Céline Helbert.

Laurent Gilquin is one of the authors of the R package sensitivity (see http://cran.r-project.org/web/packages/sensitivity/index.html). This package is useful for conducting sensitivity analysis of complex computer codes.

Céline Helbert is now the maintainer of the packages DiceDesign (see http://cran.r-project.org/web/packages/DiceDesign/index.html) and DiceEval (see http://cran.r-project.org/web/packages/DiceEval/index.html). These packages are useful for conducting design and analysis of computer experiments.

MORPHEME Project-Team

4. New Software and Platforms

4.1. New Software

4.1.1. Stracking

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

4.2. Platforms

4.2.1. Biological Image Platform (PIB)

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

MYCENAE Project-Team

5. New Software and Platforms

5.1. Platforms

5.1.1. DynPeak

In collaboration with the SED (George Rosca) and Serge Steer (SISYPHE), we have deployed a web resource version of our algorithm for the detection of peaks in pulsatile hormone patterns, DynPeak, that is accessible at the https://dynpeak.inria.fr url.

NEUROMATHCOMP Project-Team

4. New Software and Platforms

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

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

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

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

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

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

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

4.2. Event Neural Assembly Simulation

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

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

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

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

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

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

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

Website: `https://enas.inria.fr/`

NEUROSYS Team

5. New Software and Platforms

5.1. Software

5.1.1. Visualization

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

5.2. Platforms

5.2.1. OpenViBE

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

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

NUMED Project-Team

4. New Software and Platforms

4.1. SimPHyT

SimPHyt has been developed by Morgan Martinet (junior engineer). SimPHyt is an implementation in Python of the low grad glioma model developped by Benjamin Ribba. The aim is to predict the evolution of the glioma size of patients. It is used by Dr François Ducray in Pierre Wertheimer Hospital in Lyon.

4.2. SETIS

We are currently developing the SETIS software which is a GUI allowing to treat DICOM medical images to extract pathological data. These data can then be exported and used in a SAEM software (including Monolix (Inria & Lixoft)) for the parameters' estimation of models in the context of population approaches. As an example SETIS can be used to segment and compute the tumor size of a patients from MRI scans taken at different times. The software is sufficiently general to be used in various situations by clinicians (already done by our colleagues in Lyon Hospital). It will be freely distributed and is based on open source technology, so that it can easily be adapted to specific needs by other users.

SETIS is filed under APP number IDDN.FR.001.150013.000.S.A.2014.000.21000.

4.3. OptimChemo

Participants: Violaine Louvet [correspondant], Emmanuel Grenier, Paul Vigneaux, Ehouarn Maguet.

OptimChemo is a userfriendly software designed to study numerically the effect of multiple chemotherapies on simple models of tumour growth and to optimize chemotherapy schedules.

4.4. Simstab

Stability prediction of vaccine, property of Sanofi, developper by E. Grenier

4.5. Bingham flows

A 1D and 2D code with a new method for the computation of viscoplatic flows with free-surface. It essentially couples Optimization methods and Well-Balanced Finite-Volumes schemes for viscous shallow-water equations (induced by the viscoplastic nature of the fluid). Currently applied to avalanches of dense snow, it is a private code currently actively developed (in C++). One of the key feature is that its well-balanced property allows to obtained the stationary states which are linked to the stopping of the snow avalanche for this highly non-linear type of fluid.

PARIETAL Project-Team

5. New Software and Platforms

5.1. Scikit learn

Participants: Olivier Grisel [correspondant], Gaël Varoquaux, Bertrand Thirion, Michael Eickenberg, Loïc Estève, Alexandre Gramfort, Fabian Pedregosa Izquierdo.

Scikit-learn is an open-source machine learning toolkit written in Python/C that provides generic tools to learn information for the classification of various kinds of data, such as images or texts. It is tightly associated to the scientific Python software suite (Numpy/Scipy) for which it aims at providing a complementary toolkit for machine learning (classification, clustering, dimension reduction, regression). There is an important focus on code quality (API consistency, code readability, tests, documentation and examples), and on efficiency, as the scikit-learn compares favorably to state-of-the-art modules developed in R in terms of computation time or memory requirements. Scikit-learn is currently developed by more than 60 contributors, but the core developer team has been with the Parietal Inria team at Saclay-Île-de-France since January 2010. The scikit-learn has recently become the reference machine learning library in Python.

- Version: 0.15.2
- Programming language: Python, C/Cython

5.2. Nilearn

Participants: Gaël Varoquaux [correspondant], Bertrand Thirion, Loïc Estève, Alexandre Abraham, Michael Eickenberg, Alexandre Gramfort, Fabian Pedregosa Izquierdo, Elvis Dohmatob, Virgile Fritsch.

NiLearn is the neuroimaging library that adapts the concepts and tools of scikit-learn to neuroimaging problems. As a pure Python library, it depends on scikit-learn and nibabel, the main Python library for neuroimaging I/O. It is an open-source project, available under BSD license. The two key components of NiLearn are *i)* the analysis of functional connectivity (spatial decompositions and covariance learning) and *ii)* the most common tools for multivariate pattern analysis. A great deal of efforts has been put on the efficiency of the procedures both in terms of memory cost and computation time. NiLearn is maintained both through the help of Inria: a developer funded by Saclay CRI in 2012-2013, a 2013-2014 ADT and through the NiConnect project.

- Version: 0.1
- Programming language: Python

5.3. Mayavi

Participant: Gaël Varoquaux [Correspondant].

Mayavi is the most used scientific 3D visualization Python software (http://mayavi.sourceforge.net/). It has been developed by Prabhu Ramachandran (IIT Bombay) and Gaël Varoquaux (PARIETAL, Inria Saclay). Mayavi can be used as a visualization tool, through interactive command line or as a library. It is distributed under Linux through Ubuntu, Debian, Fedora and Mandriva, as well as in PythonXY and EPD Python scientific distributions. Mayavi is used by several software platforms, such as PDE solvers (fipy, sfepy), molecule visualization tools (http://pyrx.scripps.edu) and brain connectivity analysis tools (connectomeViewer).

See also the web page http://mayavi.sourceforge.net/ and the following paper http://hal.inria.fr/inria-00528985/en.

- Version: 3.4.0

5.4. Nipy

Participants: Bertrand Thirion [correspondant], Elvis Dohmatob, Gaël Varoquaux.

Nipy is an open-source Python library for neuroimaging data analysis, developed mainly at Berkeley, Stanford, MIT and Neurospin. It is open to any contributors and aims at developing code and tools sharing. Some parts of the library are completely developed by Parietal. It is devoted to algorithmic solutions for various issues in neuroimaging data analysis. The Nipy project is available, under BSD license, and within NeuroDebian.

See also the web page http://nipy.org.

- Version: 0.3

5.5. PyHRF

Participants: Philippe Ciuciu [correspondant], Aina Frau Pascual, Salma Torkhani.

PyHRF is a set of tools for within-subject fMRI data analysis, focused on the characterization of the hemodynamics. Within the chain of fMRI data processing, these tools provide alternatives to the classical within-subject GLM estimation step. The inputs are preprocessed within-subject data and the outputs are statistical maps and/or fitted HRFs. The package is mainly written in Python and provides the implementation of the two following methods:

- The joint-detection estimation (JDE) approach, that divides the brain into functionally homogeneous regions and provides one HRF estimate per region as well as response levels specific to each voxel and each experimental condition. This method embeds a temporal regularization on the estimated HRFs and an adaptive spatial regularization on the response levels.
- The Regularized Finite Impulse Response (RFIR) approach, that provides HRF estimates for each voxel and experimental conditions. This method embeds a temporal regularization on the HRF shapes, but proceeds independently across voxels (no spatial model).

The development of PyHRF is now funded by an Inria ADT, in collaboration with MISTIS.

- Version: 0.1
- Keywords: Hemodynamic response function; estimation; detection; fMRI
- License: BSD 4
- Multiplatform: Windows - Linux - MacOSX
- Programming language: Python

POMDAPI Project-Team

4. New Software and Platforms

4.1. FreeFem++

Participants: Martin Vohralík, Martin Čermák, Zuqi Tang.

The scientific calculation code FreeFem++ is an example of a complex software numerical simulation tool. It in particular encompasses all specifications of the problem, the choice and implementation of the numerical method, the choice and implementation of the linearization method (nonlinear solver), and the choice and implementation of the method of solution of the associated linear systems (linear solver). In the post-doc stays of M. Čermák and Z. Tang, we integrated there the most recent advances of the theory of a posteriori error estimation and of adaptive algorithms. In particular, adaptive stopping criteria for the linear and nonlinear solvers were implemented.

Version 3.33

Programming language: C++

http://www.freefem.org/ff++/

4.2. Oqla, Qpalm

Participants: Jean Charles Gilbert, Émilie Joannopoulos.

OQLA and QPALM aim at minimizing a large scale convex quadratic function on a polyhedron by an augmented Lagrangian method. The original features of the approach are its capacity to solve the problem without factorization, which makes them adapted to large scale problems, and to deal with unbounded and infeasible problems. In case the problem is infeasible, the codes solve the *closest feasible problem* with a global linear rate of convergence [3]. In case the problem is unbounded, the solvers build a feasible direction of unboundedness for the closest feasible problem. The solvers OQLA and QPALM only differ by their programming language; they are documented in [16], [14], [15].

Versions: 0.6 (OQLA), 0.5 (QPALM)

Programming languages: `C++` (OQLA), `Matlab` (QPALM)

4.3. Ref-image

Participants: Hend Ben Ameur, François Clément, Pierre Weis.

Ref-image is an image segmentation program using optimal control techniques. Slogan is “no gestalt inside”. Ref-image implements the refinement indicator algorithm, specialized to the case of the inversion of the identity map. It is a first step towards the implementation of a generic inversion platform using the refinement indicator algorithm.

Version: 1.1+pl0 (2014/02/28)

Programming language: OCaml

http://refinement.inria.fr/ref-image/

4.4. Ref-indic

Participants: Hend Ben Ameur, François Clément, Pierre Weis.

Ref-indic is an adaptive parameterization platform using refinement indicators. Slogan is “details only where they are worth it”. Ref-indic implements a generic version of the refinement indicator algorithm that can dock specific programs provided they conform to the generic algorithm API.

Version: 1.4+pl0 (2014/07/01)

Programming language: OCaml

http://refinement.inria.fr/ref-indic/

4.5. Sklml

Participants: François Clément, Pierre Weis.

Sklml is a functional parallel skeleton compiler and programming system for OCaml programs. Slogan is “easy coarse grain parallelization”.

Version: 1.1+pl0 (2014/01/21)

Programming language: OCaml

http://sklml.inria.fr/

POPIX Team

5. New Software and Platforms

5.1. Monolix

Participants: Marc Lavielle, Célia Barthélémy.

MONOLIX is an easy, fast and powerful tool for parameter estimation in nonlinear mixed-effect models, model diagnosis and assessment, and advanced graphical representation. It is a platform of reference for model-based drug development. Pharmacometricians and biostatisticians can rely on MONOLIX for population analysis and to model PK/PD and other complex biochemical and physiological processes.

MONOLIX was developed by Inria until June 2011. The start-up Lixoft now develops and supports MONOLIX. POPIX collaborates closely with Lixoft to convert research results into new user features available in MONOLIX.

5.2. MLXtran

Participant: Marc Lavielle.

MONOLIX is associated with MLXtran, a powerful and immediately readable declarative language for describing complex pharmacometric and statistical models. MLXtran can be used and interfaced with various environments, e.g., R, Matlab, etc.

POPIX collaborates closely with Lixoft on the definition of the specifications and the syntax of MLXtran. Implementation is then ensured by Lixoft.

5.3. Clinical trial simulator

Participants: Marc Lavielle, Fazia Bellal, Célia Barthélémy.

A clinical trial simulator (CTS) enables effective implementation of the learn-and-confirm paradigm in drug development. Through simulations the anticipated success rate of a future trial can be estimated. For various reasons industry has not embraced currently available software for trial simulation. A new tool is essential for Model Based Drug Development (MBDD).

POPIX is responsible for developing a new CTS within the DDMoRe project (see below). A new version of the CTS is available as a R package since December 2014. The capabilities of this new version comprise:

- Flexible study designs used in Phase 2 of clinical drug development: parallel group studies, crossover studies, complex treatments defined as a combination of different treatments
- Simulation of patients sampled from a joint distribution or using an external data file
- Simulation of exposure to the investigated drug and several types of drug effects related to drug exposure (continuous, categorical, count, time-to-event)
- Inter individual and intra individual variability models
- Graphics and statistical tests

5.4. MLXplore

Participant: Marc Lavielle.

MLXplore is a graphical and interactive software for the exploration and visualization of complex pharmacometric models. MLXplore also includes the ability to study the statistical variability of the models, and to model and study complex administration designs.

MLXplore does not require MONOLIX, although they make for a powerful combination, enabling to use the same, human-readable model description, to finely explore the properties of the model on the one hand, and on the other hand use the same model for advanced parameter estimation in the context of population analysis and mixed effect statistics.

MLXplore is an ideal tool to learn about pharmacometric models and population analysis, and is used extensively in the online wiki WikiPopix created by POPIX, found at: https://wiki.inria.fr/popix. MLXplore is developed by Lixoft but POPIX collaborates closely with Lixoft on on the definition of the specifications of MLXplore.

REO Project-Team

5. New Software and Platforms

5.1. FELiScE

Participants: Grégory Arbia, Cédric Doucet, Miguel Ángel Fernández Varela, Justine Fouchet-Incaux, Benoit Fabrèges, Axel Fourmont, Jean-Frédéric Gerbeau [correspondant], Mikel Landajuela Larma, Damiano Lombardi, Elisa Schenone, Saverio Smaldone, Marina Vidrascu, Irène Vignon-Clementel, Vincent Martin.

FELiScE – standing for "Finite Elements for Life Sciences and Engineering" – is a finite element code which the M3DISYM and REO project-teams have decided to jointly develop in order to build up on their respective experiences concerning finite element simulations. One specific objective of this code is to provide in a unified software environment all the state-of-the-art tools needed to perform simulations of the complex respiratory and cardiovascular models considered in the two teams – namely involving fluid and solid mechanics, electrophysiology, and the various associated coupling phenomena. FELISCE is written in C++, and may be later released as an opensource library.

It was registered in July 2014 at the *Agence pour la Protection des Programmes* under the Inter Deposit Digital Number IDDN.FR.001.350015.000.S.P.2014.000.10000.

Gforge web site: https://gforge.inria.fr/projects/felisce/

5.2. LiFE-V library

Participants: Miguel Ángel Fernández Varela [correspondant], Jean-Frédéric Gerbeau.

LiFE-V [0] is a finite element library providing implementations of state of the art mathematical and numerical methods. It serves both as a research and production library. LiFE-V is the joint collaboration between three institutions: Ecole Polytechnique Fédérale de Lausanne (CMCS) in Switzerland, Politecnico di Milano (MOX) in Italy and Inria (REO) in France. It is a free software under LGPL license.

5.3. SHELDDON

Participant: Marina Vidrascu [correspondant].

SHELDDON (SHELls and structural Dynamics with DOmain decomposition in Nonlinear analysis) is a finite element library based on the Modulef package which contains shell elements, nonlinear procedures and PVM subroutines used in domain decomposition or coupling methods, in particular fluid-structure interaction.

Gforge web site: https://gforge.inria.fr/projects/shelddon

[0] http://www.lifev.org/

SAGE Project-Team

5. New Software and Platforms

5.1. Platforms

5.1.1. Platform H2OLab

Participants: Jean-Raynald de Dreuzy, Jocelyne Erhel [correspondant], Grégoire Lecourt, Géraldine Pichot.

The software platform H2OLab is devoted to stochastic simulations of groundwater flow and contaminant transport in highly heterogeneous porous and fractured geological media. It contains a database which is interfaced through the web portal H2OWeb. It contains also software modules which can be used through the interface H2OGuilde. The platform H2OLab is an essential tool for the dissemination of scientific results. Currently, software and database are shared by the partners of the h2mno4 project (see 7.2.1). Software integrated in the platform and registered at APP are GW-UTIL, GW-NUM, PARADIS, MP-FRAC.

See also the web page http://h2olab.inria.fr.

5.2. Hydrogeology

5.2.1. GRT3D

Participants: Édouard Canot, Jocelyne Erhel [correspondant].

- Version: version 2.0, April 2014
- APP: registered
- Programming language: C
- Abstract: Reactive transport modeling has become an essential tool for understanding complex environmental problems. It is an important issue for MoMaS and C2S@EXA partners (see sections 7.2.5 , 7.2.3), in particular Andra. We have developed a method coupling transport and chemistry, based on a method of lines such that spatial discretization leads to a semi-discrete system of algebraic differential equations (DAE system). The main advantage is to use a complex DAE solver, which controls simultaneously the timestep and the convergence of Newton algorithm. The approach SIA uses a fixed-point method to solve the nonlinear system at each timestep, whereas the approach SNIA uses an explicit scheme.

  The software suite GRT3D has four executable modules:
  - SIA1D: Sequential Iterative Approach for 1D domains;
  - GDAE1D: Global DAE approach for 1D domains;
  - SNIA3D: Sequential Non Iterative Approach for 1D, 2D or 3D domains.
  - GDAE3D: Global DAE approach for 1D, 2D or 3D domains. This module has three variants: the original one with logarithms, an optimized one still with logarithms, an optimized one which does not use logarithms.
- Current work: extension of the chemistry module and parallelization.

5.2.2. SBM

Participant: Géraldine Pichot [correspondant].

- Version: version 1.0, November 2013
- Programming language: C
- Abstract: SBM (Skew Brownian Motion) is a code developed with A. Lejay (Inria, Nancy). This code allows exact or approximated simulations of the Skew Brownian Motion. This code is used for the simulation, with a Monte-Carlo approach, of a 1D diffusion process with a discontinuous diffusion coefficient. Several benchmark tests are also implemented.
- Current work: paper about benchmarking results 5.2.2 .

5.2.3. GENFIELD

Participants: Jean-Raynald de Dreuzy, Jocelyne Erhel, Grégoire Lecourt, Géraldine Pichot [correspondant].

- Version: version 1.0, December 2014
- Programming language: C++
- Abstract: GENFIELD allows the generation of log-normal correlated fields. It is based on a spectral method and uses the FFTW library. Parallelism is implemented using MPI communications. GENFIELD is used in hydrogeology to model natural fields, like hydraulic conductivity or porosity fields.
- Current work: paper about algorithms 6.4.7 .

5.3. High Performance Scientific Computing

5.3.1. PALMTREE

Participants: Lionel Lenôtre [correspondant], Géraldine Pichot.

- Version: version 1.0, November 2013
- Programming language: C++
- Abstract: We present an easy-to-use package for the parallelization of Lagrangian methods for partial differential equations. In addition to the reduction of computation time, the code aims at satisfying three properties:
  - simplicity: the user just has to add the algorithm governing the behaviour of the particles.
  - portability: the possibility to use the package with any compiler and OS.
  - action-replay: the ability of the package to replay a selected batch of particles.

  The last property allows the user to replay and capture the whole sample path for selected particles of a batch. This feature is very useful for debugging and catching some relevant information.
- Current work: paper about performance results.

5.3.2. MUESLI

Participant: Édouard Canot [corresponding author].

Muesli is a library designed to help in coding scientific problems in Fortran using a vector-oriented syntax like Matlab. One of its aims is therefore to speed-up the development process. It contains all the necessary materials to work numerically with a dynamic array (dynamic in size, shape, type, and storage structure), called mfArray. Muesli includes all or some parts of the following numerical libraries: Blas and Lapack, Arpack, Minpack, Slatec, Sparskit, SuiteSparse, Metis, Triangle, RngStreams, and other routines based on ACM algorithms.

The key points of Muesli is to efficiently solve large ODE/DAE systems (which come from, e.g., PDE problems after using the method of lines) or large non-linear minimization problems (where Jacobian matrices can be provided in a sparse format). The user can easily monitor the whole integration process and have access to tools to fix the singularity of the system of equations.

The lastest version of Muesli is 2.9.5 (2014-10-03). More information can be found at: http://people.irisa.fr/Edouard.Canot/muesli.

5.3.3. Zohour

Participant: Édouard Canot [correspondant].

Zohour is a node-based adaptive 2D mesh algorithm, written in Fortran 2003. A basic rectangular, regular set of nodes is recursively refined. Then the cells come from the Voronoi tessellation. While the domain is currently limited to a rectangular shape, its strength is three-fold:

- first, computing the flux via a Finite Element or Finite Volume method is both simple and accurate because each cell-side of cells is the bisection of two nodes;
- second, the transition between zones of different levels of refinement is more progressive than other methods, leading to a smaller number of nodes for the whole mesh;
- third, during successive refinements when dealing with a transient problem, interpolation is needed only by the new nodes, limiting the numerical errors.

It is planned for use in the HeMaTiS code (5.4.1) in order to get a refined mesh zone around the phase change surface.

See also the web page http://people.irisa.fr/Edouard.Canot/zohour.

5.4. Heat diffusion in soils

5.4.1. HeMaTiS

Participants: Édouard Canot [correspondant], Salwa Mansour.

HeMaTiS (**He**at and **Ma**ss **T**ransfer **i**n **S**oils) is a set of Finite Volume programs (variants concern different geometrical configurations: 1D, 1D-radial, 2D, 3D-axisymmetric) for computing the transient heat diffusion in soils when there is a phase change of water. Currently, the soil is modelled by a heterogeneous porous medium having constant thermo-physical properties, and the porous medium is saturated with water. The phase change is treated by means of the Apparent Heat Capacity method. In the near future, we plan to use an unsaturated model (but limited to small water content), and an effective thermal conductivity which depends on the local humidity (this latter law may reveal hysteresis behaviour). The software is written in Fortran 95 and is based on the Muesli library (5.3.2). A Computer Algebra System (Maple or Maxima) is used to compute the Jacobian matrix.

5.4.2. TPIP

Participants: Édouard Canot [correspondant], Salwa Mansour.

TPIP (**T**hermal **P**roperties by **I**nverse **P**roblem) is a program which aims at estimating the thermo-physical of a saturated porous medium after a strong heating which leads to the phase change of the water contained in the pores, knowing the experimental heating curves history at few selected points. The least-square criterion is used, in which sensitivity coefficients are the solution of a huge, complex PDE system in order to take into account the phase change of water. These equations for the sensitivity coefficients are therefore obtained via a Computer Algebra System (Maple or Maxima). In many aspects, the forward problem is similar to the HeMaTiS code (5.4.1), and like it, is based on Muesli (5.3.2). Two different minimization algorithms may be used, Damped Gauss-Newton or Levenberg-Marquardt. A special procedure has been applied in order to obtain a robust convergence, by changing some parameters of the forward problem during the iterations.

5.4.3. GLiMuH

Participants: Édouard Canot [correspondant], Salwa Mansour.

The GLiMuH code (**G**rains with **Li**quid **M**eniscus **u**nder **H**eating) is devoted to the understanding of how heat diffuses in an assembly of solid grains separated by air and water. In the pendular regime, the quantity of water is very small, leading to liquid bridges between the grains. In the current approximation, the grains are spherical in shape, and the numerical simulation is done in a 3D axisymmetric coordinate system. The shape of the liquid/gas interface is computed by integrating a differential algebraic system of equations, with a given quantity of water per unit volume of the porous medium, and under the constraint of a given contact angle between the liquid/gas interface and the solid boundaries. The numerical results allow us to estimate the effective thermal conductivity of a real wet granular medium, which is required to establish more realistic models for the HeMaTiS code (5.4.1).

SERPICO Project-Team

5. New Software and Platforms

5.1. Software for live cell imaging

Participants: Charles Kervrann [(contact)], Patrick Bouthemy, Thierry Pécot.

Motion2d: Parametric motion model estimation

The MOTION2D software written in C++ (APP deposit number: FR.001.520021.001.S.A.1998.000.21000 / release 1.3.11, January 2005) and JAVA (plug-in IMAGEJ (http://rsbweb.nih.gov/ij/) is a multi-platform object-oriented library to estimate 2D parametric motion models in an image sequence. It can handle several types of motion models, namely, constant (translation), affine, and quadratic models. Moreover, it includes the possibility of accounting for a global variation of illumination and more recently for temporal image intensity decay (e.g. due to photo-bleaching decay in fluorescence microscopy). The use of such motion models has been proved adequate and efficient for solving problems such as optic flow computation, motion segmentation, detection of independent moving objects, object tracking, or camera motion estimation, and in numerous application domains (video surveillance, visual servoing for robots, video coding, video indexing), including biological imaging (image stack registration, motion compensation in videomicroscopy). Motion2D is an extended and optimized implementation of the robust, multi-resolution and incremental estimation method (exploiting only the spatio-temporal derivatives of the image intensity function) [48]. Real-time processing is achievable for motion models involving up to six parameters. Motion2D can be applied to the entire image or to any pre-defined window or region in the image.

Free academic software distribution: Motion2D Free Edition is the version of Motion2D available for development of Free and Open Source software only. More information on Motion2D can be found at http://www.irisa.fr/vista/Motion2D and the software can be downloaded at the same Web address (about 1650 downloads registered).

On-line demo: Mobyle@SERPICO http://mobyle-serpico.rennes.inria.fr/cgi-bin/portal.py#forms::Motion2D.

Collaborator: Fabien Spindler (Inria Lagadic team).

ND-Safir and Fast2D-SAFIR: Image denoising software

The ND-SAFIR software (APP deposit number: IDDN.FR.001.190033.002.S.A.2007.000.21000 / new release 3.0 in 2013) written in C++, JAVA and MATLAB, removes additive Gaussian and non-Gaussian noise in still 2D or 3D images or in 2D or 3D image sequences (without any motion computation) [4]. The method is unsupervised and is based on a pointwise selection of small image patches of fixed size (a data-driven adapted way) in spatial or space-time neighbourhood of each pixel (or voxel). The main idea is to modify each pixel (or voxel) using the weighted sum of intensities within an adaptive 2D or 3D (or 2D or 3D + time) neighbourhood and to use image patches to take into account complex spatial interactions. The neighbourhood size is selected at each spatial or space-time position according to a bias-variance criterion. The algorithm requires no tuning of control parameters (already calibrated with statistical arguments) and no library of image patches. The method has been applied to real noisy images (old photographs, JPEG-coded images, videos, ...) and is exploited in different biomedical application domains (time-lapse fluorescence microscopy, video-microscopy, MRI imagery, X-ray imagery, ultrasound imagery, ...).

The FAST-2D-SAFIR software (APP deposit number: IDDN.FR.001.190033.001.S.A.2007.000.21000) written in C++ removes mixed Gaussian-Poisson noise in large 2D images, typically $10^3 \times 10^3$ pixels, in a few seconds. The method is unsupervised and is a simplified version of the method related to the SAFIR-nD software. The software dedicated to microarrays image denoising, was licensed to the INNOPSYS company which develops scanners for disease diagnosis and multiple applications (gene expression, genotyping, aCGH, ChIP-chip, microRNA, ...).

On-line demo: Mobyle@SERPICO http://mobyle-serpico.rennes.inria.fr/cgi-bin/portal.py#forms::NDSafir.
Free download binaries: Binaries of the software ND-SAFIR are freely and electronically distributed. Developed in standard C/C++ under Linux using the CImg library, it has been tested over several platforms such as Linux/Unix, Windows XP and Mac OS.
Academic licence agreements: Institut Curie, CNRS, ENS Ulm, Oxford University, Weizmann Institute, UCSF San-Francisco, Harvard University, Berkeley University, Stanford University, Princeton University, Georgia-Tech, Kyoto UNiversity, IMCB Singapore ...
Commercial licence agreements: Innopsys, Roper Scientfic, Photmetrics, Nikon (2015).
Collaborators: Jérôme Boulanger and Jean Salamero (UMR 144 CNRS-Institut Curie, STED team), Peter Elbau (RICAM Linz, Austria) and Jean-Baptiste Sibarita (UMR 5091, University of Bordeaux 2).

HullkGround: Background subtraction by convex hull estimation

The HULLKGROUND software (APP deposit number: IDDN.FR.001.400005.000.S.P.2009.000.21000) written in JAVA (plug-in IMAGEJ) decomposes a fluorescence microscopy image sequence into two dynamic components: i) an image sequence showing mobile objects; ii) an image sequence showing the slightly moving background. Each temporal signal of the sequence is processed individually and analyzed with computational geometry tools. The convex hull is estimated automatically for each pixel and subtracted to the original signal. The method is unsupervised, requires no parameter tuning and is a simplified version of the α shapes-based scale-space method [35].

On-line demo: Mobyle@SERPICO http://mobyle-serpico.rennes.inria.fr/cgi-bin/portal.py#forms::Hullkground.
Collaborators: Anatole Chessel and Jean Salamero (UMR 144 CNRS-Institut Curie, STED team).

5.2. Software for cryo-electron tomography

Participant: Charles Kervrann [(contact)].

TubuleJ: Straightening of microtubule cryo-EM projection views

The TUBULEJ software (APP deposit number: IDDN.FR.001.240023.000.S.P.2011.000.21000) written in JAVA (plug-in IMAGEJ) is devoted to the analysis of microtubules and helical structures in 2D cryo-electron microscope images. The software straightens curved microtubule images by estimating automatically points locations on the microtubule axis. The estimation of microtubule principal axis relies on microtubule cylindrical shape analyzed in the Fourier domain. A user-friendly interface enables to filter straight fiber images by selecting manually the layer lines of interest in the Fourier domain. This software can be used to generate a set of 2D projection views from a single microtubule projection view and a few parameters of this microtubule structure. These projection views are then back projected, by using the IMOD plug-in (http://rsbweb.nih.gov/ij/), to reconstruct 3D microtubules.

On-line demo: see http://equipes.igdr.univ-rennes1.fr/en/tips/Software/TubuleJ/.

Collaborators: Sophie Blestel and Denis Chrétien (UMR 6290, CNRS, University of Rennes 1).

Cryo-Seg: Segmentation of tomograms in cryo-electron microscopy

The CRYO-SEG software written in C++ and JAVA (plug-in MAGEJ) has been developed to detect microtubule structures and helical structures in 2D cryo-electron microscope images. Cryo-electron tomography allows 3D observation of biological specimens in their hydrated state. Segmentation is formulated as Maximum A Posteriori estimation problem and exploits image patches to take into account spatial contexts (Markov Random Fields). Because of the contrast anisotropy in the specimen thickness direction, the whole tomogram is segmented section by section, with an automatic update of reference patches. This algorithm has been evaluated on synthetic data and on cryo-electron tomograms of in vitro microtubules [19]. On real data, this segmentation method extracts the most contrasted regions of microtubules, and 3D visualization is improved.

Collaborators: Sophie Blestel and Denis Chrétien (UMR 6290, CNRS-University of Rennes 1).

5.3. Image Processing software distribution and Mobyle plateform

Participants: Tinaherinantenaina Rakotoarivelo, Thierry Pécot [(contact)], Charles Kervrann.

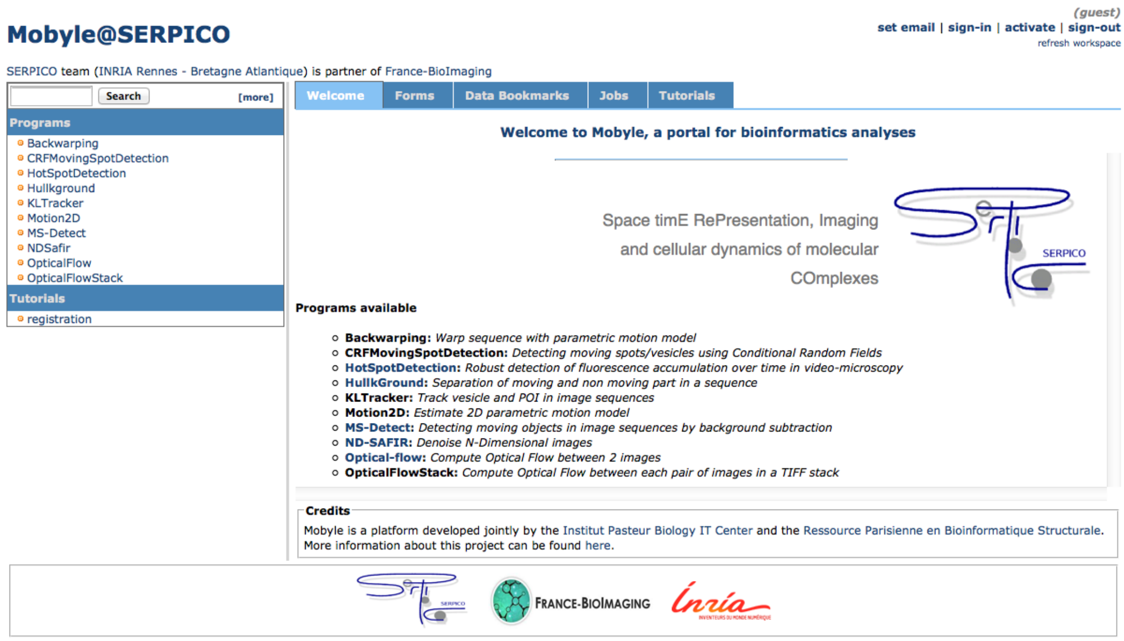


Figure 2. Mobyle@SERPICO web portal.

The objective is to disseminate the distribution of SERPICO image processing software for biologist users:

- *Free binaries:* software packages have been compiled for the main operating systems (Linux, MacOS, Windows) using CMake (see http://www.cmake.org/). They are freely available on the team website under a proprietary license (e.g. ND-SAFIR and HULLKGROUND are distributed this way at http://serpico.rennes.inria.fr/doku.php?id=software:index).
- *Mobyle@SERPICO web portal*: An on-line version of the image processing algorithms has been developped using the Mobyle framework (Institut Pasteur, see http://mobyle.pasteur.fr/). The main role of this web portal (see Fig. 2) is to demonstrate the performance of the programs developed by the team: C-CRAFT[13], ATLAS[23], HOTSPOTDETECTION[51], HULLKGROUND[35], KL-TRACKER[50], MOTION2D[49], MS-DETECT[37], ND-SAFIR[4] and OPTICALFLOW. The web interface makes our image processing methods available for biologist users at Mobyle@SERPICO

(http://mobyle-serpico.rennes.inria.fr/cgi-bin/portal.py#welcome) without any installation or configuration on their own. The size of submitted images is limited to 200 MegaBytes per user and all the results are kept 15 days. The web portal and calculations run on a server with 2 CPU x 8 cores, 64 GigaBytes of RAM.

- IMAGEJ *plug-ins*: IMAGEJ (see http://rsb.info.nih.gov/ij/) is a widely used image visualization and analysis software for biologist users. We have developed IMAGEJ plug-in JAVA versions of the following software: ND-SAFIR [4], HULLKGROUND [35], MOTION2D [49], HOTSPOTDETECTION [51]. The C-CRAFT algorithm [13] has been developped for the image processing ICY platform (http://icy.bioimageanalysis.org/).
- *Institut Curie CID iManage database*: The microscopy facility of Institut Curie has co-developped a commercial database system (CID iManage/Strand Avadis company). The database can be searched via meta-data and includes menu selections that enable to run remote processing from a cluster. We have integrated ND-SAFIR and HULLKGROUND in the interface environment to allow the database users to process their images easily, and store associated results and parameters used.

Collaborators: Charles Deltel (Inria Rennes SED) and Perrine Paul-Gilloteaux (UMR 144 CNRS-Institut Curie, STED team and PICT-IBiSA).

SHACRA Project-Team

4. New Software and Platforms

4.1. SOFA

4.1.1. Description of the SOFA framework

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

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

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

4.1.2. Consortium

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

4.1.3. SOFA Day after ISBMS'14

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

4.1.4. A new website

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

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

SISTM Team

5. New Software and Platforms

5.1. New Software

5.1.1. *TcGSA*

An *R* package for the gene set analysis of longitudinal gene expression data sets. Under development, and soon to be available on the CRAN website, this package implements a Time-course Gene Set Analysis method and provides useful plotting functions facilitating the interpretation of the results.

5.2. Upgraded Software

5.2.1. *NIMROD*

We have written a specific program called NIMROD for estimating parameter of ODE based population models. It has been regularly updated. For instance, we have adapted the program for parallel computing, in collaboration with the MCIA (Mésocentre de calcul intensif Aquitain) facility, which makes available a large computer with more than 3000 cores. This program is described in [43]. Although the program is available on the ISPED website [0], it is not user-friendly and needs further improvement to be more widely used. By now, the users are the current or previous (Jérémie Guedj, Julia Drylewicz, Mélanie Prague) members of the team and close collaborators (Andrew Yates). Furthermore, as a validation step, it would need a head-to-head comparison with other available softwares. We bet that our program can be very competitive for parameter identification in ODE models with more than two compartments.

5.2.2. *marqLevAlg*

An *R* package for function optimization. Available on CRAN, this package performs a minimization of function based on the Marquardt-Levenberg algorithm. This package is really useful when the surface to optimize is non-strictly convex or far from a quadratic function. A new convergence criterion, the relative distance to maximum (RDM), allows the user to have a better confidence in the stopping points, other than basic algorithm stabilization.

5.2.3. *VSURF*

An *R* package for Variable Selection Using Random Forests. Available on CRAN, this package performs an automatic (meaning completely data-driven) variable selection procedure. Originally designed to deal with high dimensional data, it can also be applied to standard datasets.

5.2.4. *R2GUESS*

R2GUESS package is a wrapper of the GUESS (Graphical processing Unit Evolutionary Stochastic Search) program. GUESS is a computationally optimised C++ implementation of a fully Bayesian variable selection approach that can analyse, in a genome-wide context, single and multiple responses in an integrated way. The program uses packages from the GNU Scientific Library (GSL) and offers the possibility to re-route computationally intensive linear algebra operations towards the Graphical Processing Unit (GPU) through the use of proprietary CULA-dense library.

[0] http://etudes.isped.u-bordeaux2.fr/BIOSTATISTIQUE/NIMROD/documentation/html/index.html.

SISYPHE Project-Team

4. New Software and Platforms

4.1. The Cardiovascular Waves Analysis toolbox for Scilab

Participants: Lisa Guigue, Claire Médigue, Michel Sorine, Serge Steer.

The work about Heart Failure with preserved Ejection Fraction required the development of a set of tools for ECG signal manipulation and analysis. These tools, developed by Serge Steer, have been included in a Scilab toolbox named Cardiovascular Waves Analysis toolbox that will be available soon as a Scilab module. It extends the former Cardiovascular Toolbox and provides functions for:
- ECG reading multi channel ECG files in various formats (ISHNE, MIT, TMS32),
- Handling huge ECG files obtained through Holter devices,
- ECG pretreatment (filtering, subsampling, power line interference and base line wander removal),
- ECG events detection (P, Q, R, S, T) peaks, onset and end, based on former tools developed by Qinghua Zhang,
- Cardiovascular signals analysis using various approaches like frequency or time-frequency analysis, complex demodulation, non parametric, multilevel and multifractal methods,
- Specialized plotting facilities.

STEEP Team

5. New Software and Platforms

5.1. REDEM: REDuction Of GHG EMission software

Participant: Emmanuel Prados.

REDEM software (REDuction of EMissions) is a tool designed for the benchmarking of national GHG emission reduction trajectories. We have developed REDEM in collaboration with EDDEN Laboratory (Patrick Criqui and Constantin Ilasca). The actual version of the software is implemented in Visual Basic under Microsoft Excel in order to facilitate handling and diffusion to climate/energy economists. The work related to this software has been published in [5].

5.2. Wassily

Participants: Julien Alapetite, Jean-Yves Courtonne.

In collaboration with the association "Groupe de Réflexion sur les Empreintes Ecologiques Locales" (eco-data.fr), STEEP contributes to the development of Wassily (in tribute to Wassily Leontief who first designed the relevant concepts), to perform input-output analyses applied to environmental issues (see section 4.2). The purpose of this software is to automatize most of the work of standard input-output analysis and to visualize the results in a user-friendly way in order to efficiently address the related key environmental questions.

The software is structured in three different modules:

- the database module stores all the input-output data coming from Eurostat, OCDE, Insee or other sources.
- the computation module performs the input-output calculations
- the visualization module displays the results in a synthetic manner.

The database module is based on the SQlite format and makes use of SQL to manipulate the various tables involved in the process. The goal of this module is to provide a normalized data interface for the computation module, from various types of input-output data which are often stored as Excel sheet on web sites.
The computation module is based on QT and C++ and deals mostly with matrix manipulation.
The visualization module is based on a JavaScript library called D3 and allows the user to visualize the results in a number of different ways, such as bar charts, pie charts, sankey diagrams to name a few. The integration between the C++ and JavaScript pieces of code is performed with QTScript.

5.3. QGIS_Tranus_Reports

Participants: Patricio Inzaghi, Emmanuel Prados, Peter Sturm.

This software allows to graphically visualise data output by the TRANUS LUTI model (and possibly, of any other data of the same structure). In particular, this concerns any data items defined per zone of a modelled territory (productions, indicators, etc.). The software is designed as a plugin for the geographical information system platform QGIS and can be run interactively as well as by the command line or by a call from within another software. The interactive mode (within QGIS) allows the user to define graphical outputs to be generated from TRANUS output files (type of graphs to be generated – 2D or 3D – color coding to be used, choice of data to be displayed, etc.). Visualisation of data is done in the form of 2D graphs or 3D models defined using java-script. The software is about to be registered with the APP.

TONUS Team

5. New Software and Platforms

5.1. SeLaLib

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

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

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

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

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

5.2. CLAC

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

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

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

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

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

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

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

VIRTUAL PLANTS Project-Team

4. New Software and Platforms

4.1. OpenAlea

4.1.1. OpenAlea 2.0

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

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

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

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

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

4.1.2. OpenAleaLab

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

This research theme is supported by the Inria ADT OpenAlea.

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

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

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

4.1.3. Similarity and Provenance in OpenAlea workflows

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

This research theme is supported by IBC and Inria.

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

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

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

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

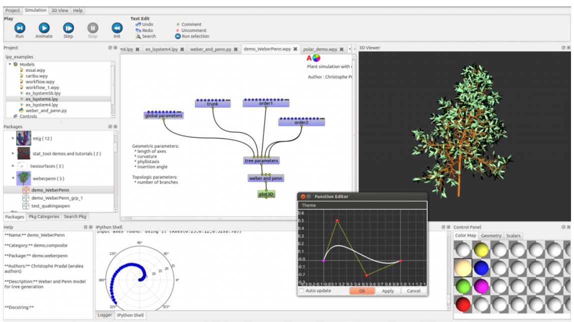


Figure 1. OpenAleaLab - A highly modular environment for modeling morphogenesis.

VISAGES Project-Team

5. New Software and Platforms

5.1. Shanoir

Participants: Justine Guillaumont, Michael Kain, Yao Yao, Christian Barillot.

Shanoir (Sharing NeurOImaging Resources) is an open source neuroinformatics platform designed to archive, structure, manage, visualize and share neuroimaging data with an emphasis on multi-centric collaborative research projects (Figure 2). It provides a user-friendly interface, a secure web access and offers an intuitive workflow to facilitate the collecting and retrieving of neuroimaging data from multiple sources and a wizard to make the completion of metadata easy. Shanoir comes along many features of neuroimaging data management systems along with research-oriented data imaging organization and enhanced data accessibility, support multi-centers clinical studies on subjects or group of subjects and other functionalities such as anonymization of data. For a better distribution/replication of stored data on a Shanoir server an export and import function on base of XML has been developed for the usage of server administrators.

Shanoir APP registration number is: IDDN.FR.001.520021.003.S.A.2008.000.31230

See also the web page http://www.shanoir.org

- Keywords: neuroimaging, ontology, sharing neuroimages
- Version: 0.5
- Software benefit: full featured neuroimaging management system with additionnal web services
- APP: IDDN.FR.001.520021.000.S.P.2008.000.31230
- License: Licence QPL
- Type of human computer interaction: Online web application, web service (SOAP messages based)
- OS/Middleware: Windows, Mac et Linux.
- Required library or software: Java 1.6, JBoss server, JBoss Seam, JSF, JPA Hibernate, EJB, Richfaces, Faceless, Ajax4JSF, Dcmtk, Dcm4chee.
- Programming language: Java / J2EE
- Documentation: see the website

5.2. ShanoirUploader

Participants: Justine Guillaumont, Michael Kain, Christian Barillot.

The ShanoirUploader (Fig. 3) is a desktop application on base of JavaWebStart (JWS). The application can be downloaded and installed using an internet browser. It interacts with a PACS to query and retrieve the data stored on any PACS. After this the ShanoirUploader sends the data to a Shanoir server instance to import these data into a Shanoir server instance. This application bypasses the situation, that in most of the clinical network infrastructures a server to server connection is complicated to set up between the PACS and a Shanoir server instance.

An APP registration is in progress. See also the web page http://shanoir.gforge.inria.fr as the ShanoirUploader documentation is integrated on this page.

- Keywords: neuroimaging, ontology, sharing neuroimages
- Version: 0.1
- Software benefit: offers a great solution to query a PACS server, download the data and send the data to a Shanoir server
- License: no defined license for the moment
- Type of human computer interaction: desktop application on base of JavaWebStart (JWS), web service (SOAP messages based)
- OS/Middleware: Linux, Windows and Mac
- Required library or software : Java SDK, installed on client machine
- Programming language: Java
- Documentation : see the website

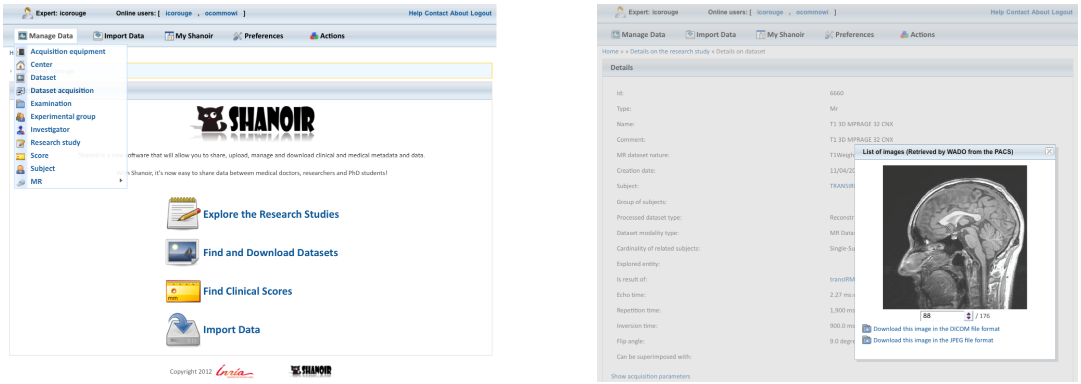


Figure 2. The SHANOIR software is a web application to share, archive, search and visualize neuroimaging data.

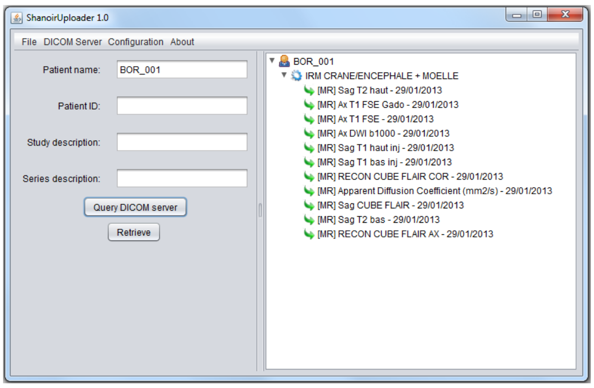


Figure 3. The ShanoirUploader software is a desktop application designed to interact with a PACS to query and retrieve the data stored on any PACS.

5.3. iShanoir

Participants: Michael Kain, Christian Barillot.

iShanoir (Fig. 4) is an iOS application, designed for iPhone and iPad. On base of this application a Shanoir server can be accessed. For this the Shanoir SOAP web-services are called. iShanoir can be used to access and navigate in the data tree structure, stored on a Shanoir server. iShanoir displays as well additional meta data corresponding to the data entities in the tree structure. On base of these informations image files (NIfTI and DICOM) can be selected and downloaded on a local iPhone/iPad in a temporary cache. From this cache the files can be opened and displayed with a corresponding viewer, the user already has to have installed on his device. This project is the result of the internship of Hélène Gérome in the team. An APP registration is in progress.

See also the web page http://shanoir.gforge.inria.fr as the iShanoir documentation is integrated on this page.

- Keywords: neuroimaging, ontology, sharing neuroimages
- Version: 0.1
- Software benefit: offers access to data stored on a Shanoir server from native iOS devices, like iPhones and iPads
- License: no defined license for the moment
- Type of human computer interaction: mobile iOS Cocoa Touch application with web service connection
- OS/Middleware: iOS
- Required library or software: none
- Programming language: Objective-C
- Documentation : see the website

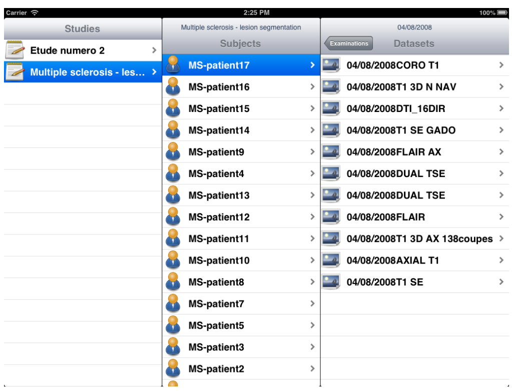


Figure 4. The iShanoir software is a desktop application designed to...

5.4. AutoMRI

Participants: Fang Cao, Isabelle Corouge, Pierre Maurel, Elise Bannier.

AutoMRI Based on MATLAB and the SPM8 toolbox, autoMRI provides complete pipelines to pre-process and analyze various types of images (anatomical, functional, perfusion, metabolic, relaxometry, vascular). This software is highly configurable in order to fit to a wide range of needs. Pre-processing includes segmentation of anatomical data, as well as co-registration, spatial normalization and atlas building of all data types. The analysis pipelines perform either within-group analysis or between-group or one subject-versus-group comparison and produce statistical maps of regions with significant differences. These pipelines can be applied to structural data to exhibit patterns of atrophy or lesions, to ASL (both pulsed or pseudo-continuous sequences) or PET data to detect perfusion or metabolic abnormalities, to relaxometry data to detect deviations from a template, to functional data - either BOLD or ASL - to outline brain activations related to block or event-related paradigms. In addition to the standard General Linear Model approach, the ASL pipelines implement an a contrario approach and, for patient-specific perfusion study, an heteroscedastic variance model. Besides, the vascular pipeline processes 4D MRA data and enables accurate assessment of hemodynamic patterns (Figure 5).

- Keywords: fMRI, MRI, ASL, fASL, SPM, automation
- Software benefit: Automatic MRI data analysis based on SPM. Once the parameters are set, the analysis is performed without human interaction.
- APP: Part in IDDN.FR.001.130017.000.S.A.2012.000.31230
- License: Part under CeCILL
- Type of human computer interaction: Matlab function (script, no GUI)
- OS/Middleware: Windows, OS X, Linux
- Required library or software: Matlab, SPM, SPM toolboxes : Marsbar, LI-toolbox, NS
- Programming language: Matlab
- Documentation: available at https://gforge.inria.fr/projects/autofmri/ and https://gforge.inria.fr/projects/asl/

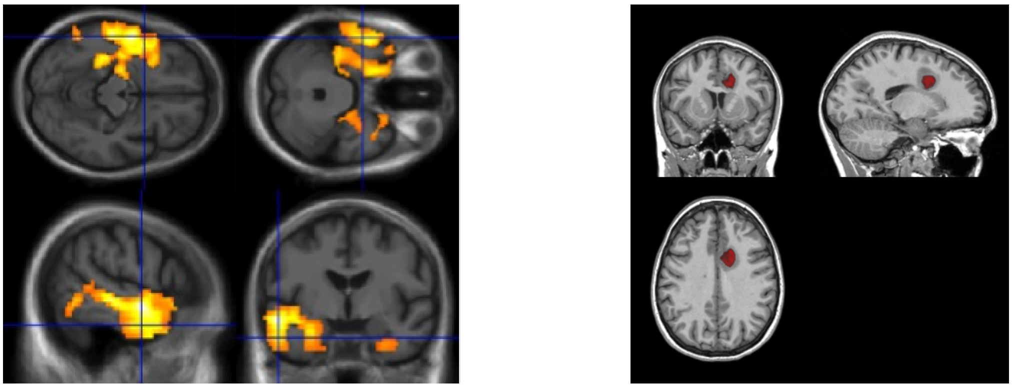


Figure 5. Illustrations of results obtained with autoMRI: Conjunction map showing areas of hypoperfusion and hypometabolism in semantic dementia (right), Detection of relaxometry defect in an MS patient (left).

5.5. medInria

Participants: René-Paul Debroize, Guillaume Pasquier, Laurence Catanese, Olivier Commowick.

medInria is a national Inria project shared between 4 Inria teams (Asclepios, Athena, Parietal and Visages). It aims at creating an easily extensible platform for the distribution of research algorithms developed at Inria for medical image processing. This project has been funded by the D2T (ADT MedInria-NT) in 2010 and renewed in 2012. The Visages team leads this Inria national project and participates in the development of the common core architecture and features of the software as well as in the development of specific plugins for the team's algorithm. medInria 2.2.1 has been released in September 2014 for the main distribution platforms. medInria core API source code is also released under a BSD license.

See also Figure 6 and the web page http://med.inria.fr

- Keywords: medical imaging, diffusion imaging, registration, filtering, user-friendly interface
- Software benefit: user-friendly interface to cutting-edge research tools for research clinicians. Straightforward to add functionalities through plugins.
- License: core: BSD, plugins: choice of each team.
- Type of human computer interaction: Qt-based GUI
- OS/Middleware: Windows, Mac et Linux.
- Required library or software : Qt, DTK, ITK, VTK.
- Programming language: C++

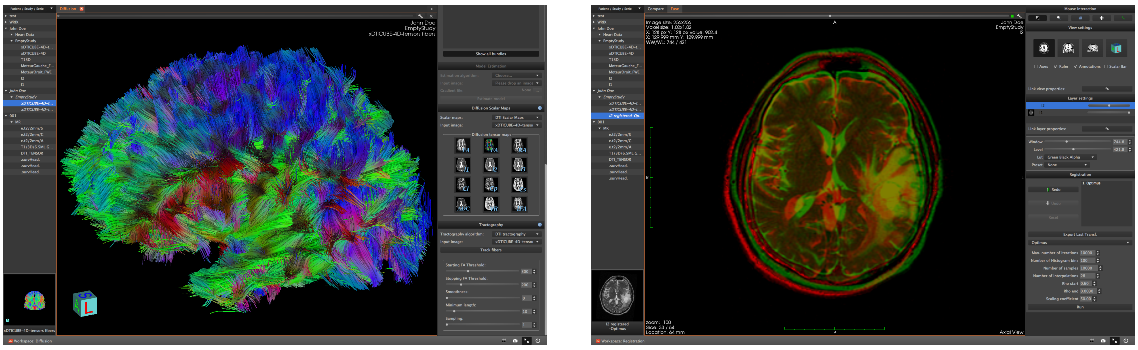


Figure 6. The medInria software platform: Fused view of registered images (right), Tractography overlapped with 3D image (left)

5.6. Anima

Participants: Fang Cao, Laurence Catanese, Olivier Commowick, René-Paul Debroize, Florent Leray, Renaud Hédouin, Guillaume Pasquier.

Anima is a set of libraries and tools developed by the team as a common repository of research algorithms. As of now, it contains tools for image registration, statistical analysis (group comparison, patient to group comparison), diffusion imaging (model estimation, tractography, etc.), quantitative MRI processing (quantitative relaxation times estimation, MR simulation), image denoising and filtering, and segmentation tools. All of these tools are based on stable libraries (ITK, VTK), making it simple to maintain.

- Keywords: medical imaging, diffusion imaging, registration, filtering, relaxometry
- Software benefit: New methodological image processing, common place for team code
- Type of human computer interaction: C++ API
- OS/Middleware: Windows, Mac and Linux.
- Required library or software : ITK, VTK.
- Programming language: C++

5.7. Integration of EEG and fMRI

Participants: Marsel Mano, Lorraine Perronnet.

Related to the project Hemisfer there have been development of new functions, scripts and demos for the acquisition and processing of the EEG and fMRI data in Real-time. These include:

- Functions for fMRI header info reader, volume reader, motion correction, slice time correction nifty output conversion, real time fMRI initialization, real time fMRI processing, z-score calculation, volume smoother, alignment, etc., functions for real time EEG data acquisition, filtering, power calculation and display.
- Scripts for various protocols used in offline fMRI experiments, real time processing loop for EEG and fMRI.
- Demo for real time acquisition of the EEG and fMRI data, demo for real time processing efficiency of the fMRI data, demo for the real time processing of EEG data, real time z-Score for fMRI data.
- Several small aux functions for I/O interfaces (e.g. com, serial)

In the current stage the prototype also relies on various other free toolboxes (e.g. SPM, pnet)

- Keywords: medical imaging, EEG, fMRI
- Software benefit: integration of EEG and fMRI processing
- Type of human computer interaction: C++ API, shell scripts
- OS/Middleware: Windows, Mac and Linux.
- Required library or software : SPM, pnet.
- Programming language: C++, shell scripts

5.8. Platforms

5.8.1. The Neurinfo Platform

VISAGES is the founding actor of a new experimental research platform which was installed in August 2009 at the University Hospital of Rennes. The University of Rennes 1, Inria, Inserm for the academic side, and the University Hospital of Rennes and the Cancer Institute "Eugene Marquis" for the clinical side, are partners of this neuroinformatics platform called NeurINFO (http://www.neurinfo.org). This platform has been supported under the "Contrat de Projets Etat-Région" (C. Barillot is the PI) and has received a total amount of 5.1 Meuros for the period 2007–2013. European (FEDER), National (through Ministry of research, Inria, Inserm and ANR) and local councils (Brittany Region, Ille et Vilaine, and Rennes Metropolis) have joined their effort to support this operation for a total amount of 5070 keuros (600keuros for the infrastructures, 3670keuros for the equipments and 800keuros for the functioning). This application was set up through the Regional PIMATGI initiative coordinated by INSERM in Brittany (C. Roux). The overall PIMATGI initiative served for the financing of three distinct, but complementary, platforms: NeurINFO, TheraFONC as a technical platform dedicated to therapy guided by functional imaging especially in the oncology domain (Inserm U 650 - LaTIM, Dir. Ch. Roux, Brest), and TherA-Image as a platform dedicated to image guided mini-invasive surgery and therapy especially in the domain of cardio-vascular diseases (U 642 -LTSI, Dir. L. Senhadji, Rennes).

Concerning the NeurINFO Platform, the activity domain is a continuum between methodological and technological research built around specific clinical research projects. The ambition is to do innovation in science, technology and medical technology transfer for the implementation on the clinical field. On the medical field, the translational research domain mainly concerns medical imaging and more specifically the clinical neurosciences. Among them are multiple sclerosis, epilepsy, neurodegenerative, neurodevelopmental and psychiatric diseases, surgical procedures of brain lesions, neuro-oncology and radiotherapy planning. Beyond these CNS applications, the platform is also open to alternative applications. Neurinfo ambitions to support the emergence of research projects based on their level of innovation, their pluri-disciplinarity and their ability to foster collaborations between different actors (public and private research entities, different medical specialties, different scientific profiles). In this context, a new research 3T MRI system (Siemens Verio system) was acquired in summer 2009 in order to develop the clinical research in the domain of morphological, functional, structural and cellular in-vivo imaging. In 2014 a new equipment for simultaneous recording of EEG and MRI images has been acquired from Brain Product. Visages and its partners in the Neurinfo project are committed to use this new research platform for developing new regional, national and international collaborations around fundamental and applied clinical research projects dealing with in-vivo medical imaging. In 2014, the two engineers running the platform (Elise Bannier and Isabelle Corouge), members of the Visages team, moved from temporary employment contracts to open-ended research engineers contracts.

ALGORILLE Project-Team

5. New Software and Platforms

5.1. Introduction

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

5.2. Implementing parallel models

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

5.2.1. ORWL and P99

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

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

5.2.2. parXXL

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

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

5.2.3. musl

Participant: Jens Gustedt.

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

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

5.3. Parallel developments for numerical scientific application

Participant: Sylvain Contassot-Vivier.

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

5.4. Distem

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

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

5.5. SimGrid

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

5.5.1. Core distribution

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

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

5.5.2. SimGridMC

Participants: Martin Quinson, Marion Guthmuller, Gabriel Corona.

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

5.5.3. SCHIaaS

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

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

5.6. Kadeploy

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

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

5.7. XPFlow

Participants: Tomasz Buchert, Lucas Nussbaum [correspondant].

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

5.8. Grid'5000 testbed

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

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

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

ALPINES Project-Team

5. New Software and Platforms

5.1. Platforms

5.1.1. FreeFem++, http://www.freefem.org/ff++/

FreeFem++ is a PDE solver based on a flexible language that allows a large number of problems to be expressed (elasticity, fluids, etc) with different finite element approximations on different meshes. There are more than 2000 users, and on the mailing list there are 430 members. Among those, we are aware of at least 10 industrial companies, 8 french companies and 2 non-french companies. It is used for teaching at Ecole Polytechnique, Ecole Centrale, Ecole des Ponts, Ecole des Mines, University Paris 11, University Paris Dauphine, La Rochelle, Nancy, Metz, Lyon, etc. Outside France, it is used for example at universities in Japan (Tokyo, Kyoto, Hiroshima, there is a userguide FreeFem++ in japan), Spain (Sevilla, BCAM, userguide available in spanish), UK (Oxford), Slovenia, Switzerland (EPFL, ETH), China. For every new version, there are 350 regression tests, and we provide a rapid correction of reported bugs. The licence of FreeFem++ is LGPL.

5.1.2. Library for preconditioned iterative methods

In the project-team we develop a library that integrates the direction preserving and low rank approximation preconditioners for both approached factorizations and domain decomposition like methods. It will be available through `FreeFem++` and also as a stand alone library, and we expect to have one version of this library available in 2014.

5.1.3. HPDDM, https://github.com/hpddm

HPDDM is an efficient implementation of various domain decomposition methods (DDM) such as one- and two-level Restricted Additive Schwarz methods, the Finite Element Tearing and Interconnecting (FETI) method, and the Balancing Domain Decomposition (BDD) method. These methods can be enhanced with deflation vectors computed automatically by the framework using methods developed by members of the team:

- Generalized Eigenvalue problems on the Overlap (GenEO), an approach first introduced in the PhD of Nicole Spillane.
- local Dirichlet-to-Neumann operators, an approach first introduced in a paper by Nataf et al. and recently revisited by Conen et al.

This code has been proven to be efficient for solving various elliptic problems such as scalar diffusion equations, the system of linear elasticity, but also frequency domain problems like the Helmholtz equation. A comparison with modern multigrid methods can be found in the thesis of Pierre Jolivet.

HPDDM is a header-only library written in C++11 with MPI and OpenMP for parallelism. While its interface relies on plain old data objects, it requires a modern C++ compiler: g++ 4.7.3 and above, clang++ 3.3 and above, icpc 15.0.0.090 and above. HPDDM has to be linked against BLAS and LAPACK (as found in OpenBLAS, in the Accelerate framework on OS X, in IBM ESSL, or in Intel MKL) as well as a direct solver like MUMPS, SuiteSparse, MKL PARDISO, or PaStiX. At compilation, just define before including HPDDM.hpp one of these preprocessor macros MUMPSSUB, SUITESPARSESUB, MKL_PARDISOSUB, or PASTIXSUB (resp. DMUMPS, DSUITESPARSE, DMKL_PARDISO, or DPASTIX) to use the corresponding solver inside each subdomain (resp. for the coarse operator). Additionally, an eigenvalue solver is recommended. There is an existing interface to ARPACK. Other (eigen)solvers can be easily added using the existing interfaces. For building robust two-level methods, an interface with a discretization kernel like FreeFem++ or Feel++ is also needed. It can then be used to provide, for example, elementary matrices, that the GenEO approach requires. As such HPDDM is not an algebraic solver, unless only looking at one-level methods. Note that for substructuring methods, this is more of a limitation of the mathematical approach than of HPDDM itself.

ASAP Project-Team

5. New Software and Platforms

5.1. MediEgo: A recommendation solution for webmasters

Participants: Jacques Falcou, Arnaud Jégou, Xavier Lucas, Anne-Marie Kermarrec, Jean-François Verdonck.

Contact:	Anne-Marie Kermarrec
Licence:	Proprietary
Presentation:	Recommendation solution for webmasters
Status:	Beta version, IDDN.FR.001.490030.000.S.P.2013.000.30000 on 09/12/2013

MEDIEGO is a solution for content recommendation based on the users navigation history. The solution 1) collects the usages of the Web users and store them in a profile; 2) uses this profile to associate to each user her most similar users; 3) leverages this implicit network of close users in order to infer their preferences and recommend advertisements and recommendations. MEDIEGO achieves scalability using a sampling method, which provides very good results at a drastically reduced cost. The MEDIEGO recommendation engine is built in collaboration with Sébastien Campion. We have demonstrated the software at the conference Le Web 14 (9-11 Dec 2014) in collaboration with France Television.

5.2. AllYours-P2P: A distributed news recommender (former WhatsUp)

Participants: Heverson Borba Ribeiro, Raziel Carvajal Gomez, Davide Frey, Arnaud Jégou, Anne-Marie Kermarrec.

Contact:	Davide Frey
Licence:	AGPL 3.0
Presentation:	A distributed news recommender
Status:	Beta version, IDDN.FR.001.500002.000.S.P.2013.000.30000 on 09/12/2013

Within the context of the AllYours EIT/ICT-Labs project, we refined the implementation of WhatsUp into the Peer-to-Peer AllYours application. AllYours-P2P is a peer-to-peer based news recommender system that organizes users into an implicit social network based on their explicit opinions. In AllYours-P2P the recommendation process is collaboratively performed by connected users. Every user runs a symmetric piece of software responsible for storing user interests and calculating the affinity between a user and its neighborhood. The local computed similarity then is used to keep virtual connections to other users whose interests are alike and remove connection to the ones that are not. As a result, an interest-based overlay is built and users converge to groups of similar interests within which news are disseminated. The AllYours-P2P software consists of two parts, running on each peer: an embedded application server, based on Jetty, and a web interface accessible from any web browser. The back-end is written in Java, while the user interface comprises HTML and Javascript code. AllYours-P2P is currently available in three different platforms: Mac OSx (10.5 or later), Windows (Vista and Windows 7) and Linux (Ubuntu 10.4 or later). We have tested Allyours-p2p in a real life environment with a set of invited users in Italy from Sep to Nov 2014. These test were a part of joint project between ASAP Team and its Italian partner Trentorise.

Currently the implementation of Allyours-p2p includes approximately 21K lines of code.

5.3. HyRec: A hybrid recommender system

Participants: Davide Frey, Anne-Marie Kermarrec.

Contact:	Davide Frey
Licence:	Proprietary
Status:	Beta version, IDDN.FR.001.500007.000.S.P.2013.000.30000 on 09/12/2013

This work leads to the development of HyRec, a hybrid recommender system. The motivation of this work is to explore solutions that could in some sense democratize personalization by making it accessible to any content provider company without generating huge investments. HyRec implements a user-based collaborative filtering scheme and offloads CPU-intensive recommendation tasks to front-end client browsers, while retaining storage and orchestration tasks within back-end servers. HyRec seeks to provide the scalability of p2p approaches without forcing content providers to give up the control of the system. This software has been developed in collaboration with Antoine Boutet (Univ. Saint-Étienne) and Rhicheek Patra (EPFL).

5.4. GossipLib: A library for gossip-based applications

Participants: Heverson Borba Ribeiro, Davide Frey, Anne-Marie Kermarrec.

Contact:	Heverson Borba Ribeiro, Davide Frey
Licence:	AGPL 3.0
Presentation:	Library for gossip protocols
Status:	Alpha version, IDDN.FR.001.500001.000.S.P.2013.000.10000 on 09/12/2013

GossipLib is a library consisting of a set of JAVA classes aimed to facilitate the development of gossip-based application in a large-scale setting. It provides developers with a set of support classes that constitute a solid starting point for building any gossip-based application. GossipLib is designed to facilitate code reuse and testing of distributed application, and provides also the implementation of a number of standard gossip protocols that may be used out of the box or extended to build more complex protocols and applications. These include for example the peer-sampling protocols for overlay management. GossipLib also provides facility for the configuration and deployment of applications as final-product but also as research prototype in environments like PlanetLab, clusters, network emulators, and even as event-based simulation. The code developed with GossipLib can be run both as a real application and in simulation. Currently the implementation of GossipLib includes approximately 9K lines of code, and is used in several projects by ASAP, including HEAP, AllYours-P2P, and Behave.

5.5. YALPS: A library for p2p applications

Participants: Heverson Borba Ribeiro, Davide Frey, Arnaud Jégou, Anne-Marie Kermarrec.

Contact:	Heverson Borba Ribeiro, Davide Frey
Licence:	Open Source
Presentation:	Library for p2p applications
Status:	Beta version, IDDN.FR.001.500003.000.S.P.2013.000.10000 on 09/12/2013

YALPS is an open-source Java library designed to facilitate the development, deployment, and testing of distributed applications. Applications written using YALPS can be run both in simulation and in real-world mode without changing a line of code or even recompiling the sources. A simple change in a configuration file will load the application in the proper environment. A number of features make YALPS useful both for the design and evaluation of research prototypes and for the development of applications to be released to the public. Specifically, YALPS makes it possible to run the same application as a simulation or in a real deployment. Applications communicate by means of application-defined messages which are then routed either through UDP/TCP or through YALPS's simulation infrastructure. In both cases, YALPS's communication layer offers features for testing and evaluating distributed protocols and applications. Communication channels can be tuned to incorporate message losses or to constrain their outgoing bandwidth. Finally, YALPS includes facilities to support operation in the presence of NATs and firewalls using relaying and NAT-traversal techniques. The implementation of YALPS includes approximately 16K lines of code, and is used in several projects by ASAP, including HEAP, AllYours-P2P, and Behave. This work was done in collaboration with Maxime Monod (EPFL).

5.6. HEAP: Heterogeneity-aware gossip protocol

Participants: Davide Frey, Arnaud Jégou, Anne-Marie Kermarrec.

Contact:	Davide Frey
Licence:	Open Source
Presentation:	Java Application
Status:	Release & ongoing development

This work has been done in collaboration with Vivien Quéma (CNRS Grenoble), Maxime Monod and Rachid Guerraoui (EPFL), and has lead to the development of a video streaming platform based on HEAP, *HEterogeneity-Aware gossip Protocol*. The platform is particularly suited for environment characterized by heterogeneous bandwidth capabilities such as those comprising ADSL edge nodes. HEAP is, in fact, able to dynamically leverage the most capable nodes and increase their contribution to the protocol, while decreasing by the same proportion that of less capable nodes. During the last few months, we have integrated HEAP with the ability to dynamically measure the available bandwidth of nodes, thereby making it independent of the input of the user.

5.7. Brow2Brow: Browser-to-browser serverless toolboxes

Participants: Raziel Carvajal Gomez, Davide Frey, Anne-Marie Kermarrec.

Brow2Brow is an "Action de Development Technologique", i.e. a collaborative development project that aims at providing a middleware and software library for browser-to-browser applications. Brow2Brow involves the ASAP team as well as the DICE Team from Inria Grenoble (Antenne de Lyon). The project seeks to provide an alternative to the current model followed by Web2.0 applications by exploiting the recently introduced WebRTC standard. Existing Web 2.0 applications collect data on browsers and send it to servers that store and process it. The goal of Brow2Brow is to provide an alternative approach where browsers can themselves proceed to collaborative data processing. This will make it possible avoid data concentration at a single server. The project has resulted so far in the development of WebGC, a library for gossip-based applications on browsers.

5.8. WebGC: Web-based Gossip Communication

Participants: Raziel Carvajal Gomez, Davide Frey, Anne-Marie Kermarrec.

Contact:	Raziel Carvajal Gomez, Davide Frey
License:	Not-yet released
Presentation:	Library for Gossip protocols within Web Browsers
Status:	Ongoing development

WebGC is a library for gossip-based communication between web-browsers. It has been developed in collaboration with Mathieu Simonin in the context of the Brow2Brow ADT project. WebGC builds on the recent WebRTC standard as well as on PeerJS, an open-source project that provides primitives for data transfer on top of WebRTC.

The library currently includes the implementation of two peer sampling protocols, CYCLON and the generic peer-sampling protocol from [7], as well as a clustering protocol [1]. All protocols implement a common GOSSIPPROTOCOL "interface"—since Javascript does not natively support interfaces, we adopt the interface pattern. A COORDINATOR makes it possible to stack these protocols on top of each other to implement applications.

ASCOLA Project-Team

5. New Software and Platforms

5.1. btrCloud (and Entropy)

Participants: Jean-Marc Menaud [correspondent], Guillaume Le Louët, Frédéric Dumont.

Orchestration, virtualization, energy, autonomic system, placement, cloud computing, cluster, data center, scheduler, grid

btrCloud is a virtual machine manager for clusters and provides a complete solution for the management and optimization of virtualized data centers. btrCloud (acronym of better cloud) is composed of three parts.

The analysis function enables operatives and people in charge to monitor and analyze how a data-center works — be it on a daily basis, on the long run, or in order to predict future trends. This feature includes boards for performance evaluation and analysis as well as trends estimation.

btrCloud, by the integration of btrScript, provides (semi-)automated VM lifecycle management, including provisioning, resource pool management, VM tracking, cost accounting, and scheduled deprovisioning. Key features include a thin client interface, template-based provisioning, approval workflows, and policy-based VM placement.

Finally, several kinds of optimizations are currently available, such as energy and load balancing. The former can help save up to around 20% of the data-center energy consumption. The latter provides optimized quality of service properties for applications that are hosted in the virtualized datacenters.

btrCloud is available at http://www.btrcloud.org.

5.2. EScala and JEScala

Participants: Jacques Noyé [correspondent], Jurgen Van Ham.

AOP, inheritance, event-based programming, events, declarative events, asynchronous events, join operator, Scala

EScala is an extension of the programming language Scala with support for events as object members. EScala combines ideas of event-driven, aspect-oriented and functional reactive programming.

Events are natural abstractions for describing interactive behavior as part of an object interface. In conventional object-oriented languages, events are implemented indirectly, typically using the Observer pattern. C# eliminates the corresponding glue code and directly supports events as object members. However, events are still *explicitly* triggered at specific locations within the program.

EScala goes much further. First, it also supports *implicit* events. Akin to join points in aspect-oriented languages, these events are implicitly produced at specific execution points, such as the beginning or the end of the execution of a method. Second, *declarative events* make it possible to compose events using logical operators as well as to filter them and alter their content.

EScala events are fully integrated with object-oriented features. An event is defined in the context of its owner object. Event definitions are inherited in subclasses and event uses are late-bound. Unlike typical aspect-oriented languages, EScala preserves object-oriented encapsulation and modular reasoning.

JEScala extends EScala with support for concurrent programming (see Sec. 6.2). Events can be declared as *asynchronous* so that their handling takes place concurrently. A new composition operator, the *join* operator, inspired by the join calculus, can also be used to synchronize the concurrent activities created by asynchronous events and communicate between them.

This is joint work with the Software Technology Group at TU Darmstadt.

Prototype implementations of these languages are available through http://www.stg.tu-darmstadt.de/research.

5.3. CSLA

Participants: Thomas Ledoux [correspondent], Yousri Kouki.

Service-level agreement, Cloud computing, elasticity

Verifying non-functional properties like performance, dependability, energy consumption and economical costs of Cloud is challenging today due to ad hoc management in terms of Quality-of-Service (QoS). We believe that a differentiating element between Cloud computing environments will be the QoS and the Service-Level Agreement (SLA) provided by the Cloud.

CSLA, the Cloud Service Level Agreement language, allows the definition of SLA properties for arbitrary Cloud services (XaaS). CSLA addresses QoS uncertainty in unpredictable and dynamic environment and provides a cost model of Cloud computing. Besides the standard formal definition of contracts – comprising validity, parties, services definition and guarantees/violations – CSLA is enriched with features, such as QoS degradation and an advanced penalty model, thus introducing fine-grained language support for Cloud elasticity management [27][26].

CSLA is available at http://www.emn.fr/z-info/csla.

5.4. SAdapt

Participants: Ronan-Alexandre Cherrueau [correspondent], Mario Südholt.

Service-oriented systems, distributed programming, event-based programming, workflow patterns

The SAdapt tool provides an implementation of workflow adaptation patterns and allows the transformation of service-oriented systems implemented using Apache's CXF service infrastructure in terms of high-level declarative service transformations. The transformations are defined using an expressive language that supports matching of the execution of service-based systems in terms of flexible patterns over service compositions.

The SAdapt tool has partially been developed and is employed in the A4Cloud EU project (see Sec. 8.3) as a basis for our work on the enforcement of accountability properties in complex cloud-based systems.

The SAdapt tool and its application, notably to the security hardening of service systems that use OAuth 2 for the authorization of resource accesses is available at http://a4cloud.gforge.inria.fr/doku.php?id=start:advservcomp.

In 2014, we have used and extended the tool in order to investigate accountability properties of service-based applications, see Sec. 6.3 .

5.5. SimGrid/VMPlaces

Participants: Takahiro Hirofuchi, Adrien Lebre [correspondent], Jonathan Pastor, Flavien Quesnel, Mario Südholt.

Simulation, Virtualization, Cloud computing, VM placement

SimGrid is a toolkit for the simulation of algorithms executed on large-scale distributed systems. Developed for more than a decade, it has been used in a large number of studies described in more than 100 publications. In 2013, ASCOLA with the support of the SimGrid core-developers, designed and implemented additional capabilities, in particular the Virtual Machine abstraction, enabling to address Cloud Computing related concerns.

Developed, first, in an experimental repository, the integration of these extensions into the master branch of SimGrid has been achieved during Summer 2014. The principal role of ASCOLA is now to ensure the maintenance of this portion of the code with respect to the evolutions of the SimGrid toolkit (such as for instance the recent port of the SimGrid kernel in C++).

Although the virtualization extensions are recent, several projects leveraging them have been already proposed. [0] Among them, ASCOLA is working on dedicated framework to evaluate and compare VM placement algorithms. Entitled VMPlaces, this framework is composed of two major components: the injector and the VM placement algorithm. The injector is the generic part of the framework (i.e. the one you can directly use) while the VM placement algorithm is the part you want to study (or compare with available algorithms). Currently, the VMPlaceS is released with three algorithms:

- Entropy, a centralized approach using a constraint programming approach to solve the placement/reconfiguration VM problem
- Snooze, a hierarchical approach where each manager of a group invokes Entropy to solve the placement/reconfiguration VM problem. Note that in the original implementation of Snooze, it is using a specific heuristic to solve the placement/reconfiguration VM problem. As the sake of simplicity, we have simply reused the entropy scheduling code.
- DVMS, a distributed approach that dynamically partitions the system and invokes Entropy on each partition.

SimGrid is available at http://simgrid.gforge.inria.fr.
VMPlaces is available at http://beyondtheclouds.github.io/VMPlaceS/

[0] The list of the projects is available at : http://simgrid.gforge.inria.fr/contrib/clouds-sg-doc.html

ATLANMOD Project-Team

5. New Software and Platforms

5.1. The ATL Model Transformation Language

URL: http://www.eclipse.org/atl/

With an eye on the normative work of the OMG (MOF, OCL, QVT, etc.), a new conceptual framework has been developed based on a second generation model transformation language called ATL. Although ATL influenced the OMG standard, the approach is more general as discussed in [48]. In 2004 IBM gave an Eclipse innovation award to the ATL project. In 2007 Eclipse recognized ATL as one central solution for model transformation and promoted it to the M2M project (see *Eclipse.org/m2m*). There are more than 200 industrial and academic sites using ATL today, and several Ph.D. thesis in the world are based on this work.

In 2011 we started a new evolution phase for ATL. Our mid-term plan is making of ATL the leading solution for building autonomous reactive transformation systems, i.e. transformation networks that can autonomously manage a set of dataflows among the application models.

Following this line, we first implemented a new refinement mode for ATL, to support in-place transformations. This extension allows the dynamic manipulation of models while keeping them connected to runtime applications. Next, we presented a lazy execution algorithm for ATL. With it, the elements of the target model are generated only when and if they are accessed. This extension allows to build reactive transformation systems that react to requests of model elements, by triggering the necessary computation. Our lazy version of ATL enables also transformations that generate infinite target models, extending the application space of the model-transformation paradigm.

The latest (still ongoing) work in this direction is the development of a full reactive ATL engine, able to activate the minimal computation for responding to updates or request on the involved models. This engine is studied to scale up with large ATL networks. In this line we also introduced an algorithm for simplifying ATL transformation chains.

Performing just the required work on model transformation improves scalability, an open issue the previous described works contribute to solve.n efficient execution, as in the the lazy and reactive scenarios, may help with scalability problems by focusing the tasks in the required part of a very large transformation. However, this is not always the case and we might have to perform operations in the whole model. In this scenario, a solution for the scalability problem would be to take advantage of multi-core architectures that are very popular today, to improve computation times in the transformation of very large models. In this sense, a first step explores the strong parallelization properties rule-based languages like ATL have. A new prototype implementation of a parallel ATL engine has been developed showing how transformations can be developed without taking into account concurrency concerns, and such a transformation engine can automatically parallelize operations improving execution times.

Aligned with this research line we propose in recent works an approach to automatically parallelize the computation of model transformation using Cloud infrastructures. For this, we take advantage of a well-known distributed programming model: *MapReduce*. In this sense, we introduce an algorithm aligning both execution semantics of ATL and MapReduce. Based on this, a new prototype tool has been developed [0] showing in several experiments the scalability of the solution.

5.2. MoDisco (Model Discovery)

URL: http://www.eclipse.org/MoDisco/

[0] https://github.com/atlanmod/ATL_MR

MoDisco is an open source Eclipse project that provides a generic and extensible framework dedicated to the elaboration of Model Driven Reverse Engineering (MDRE) solutions. Gathering contributions from both academics and industrials, the goal of the project is to federate common efforts in the model-based transformation of legacy software systems implemented using different technologies (e.g. Java, COBOL, C). The first principle is to discover models out of legacy artifacts, representing appropriately all the relevant information, to be then used as part of reverse engineering processes for software understanding, evolution or modernization. Targeted scenarios include software (technical or architectural) migration of large legacy systems, but also retro-documentation, refactoring, quality assurance, etc. Within this context, MoDisco has collaborations with the OMG Architecture Driven Modernization (ADM) Task Force, for which the project provides several reference implementations of its standards: Knowledge Discovery Metamodel (KDM), Software Measurement Metamodel (SMM) and Abstract Syntax Tree Metamodel (ASTM).

The MoDisco framework is composed of a set of Eclipse plugins, and relies on the de-facto standard Eclipse Modeling Framework (EMF) for model handling. Thanks to its modular architecture, it allows completely covering the three steps of a standard MDRE approach: 1) Discovery (i.e. extracting a complete model of the source code), 2) Understanding (i.e. browsing and providing views on this model for a given purpose) and 3) Transformation (evolving the model towards a new technology, architecture, etc). More specifically, as part of its *Infrastructure* layer, MoDisco offers the set of generic (i.e.; legacy technology-independent) reusable components really useful to build the core of MDRE solutions: Discovery Manager and Workflow for MDRE task orchestration, Model Browser for advanced navigation in complex models, model extension and customization capabilities for understanding (e.g. views definition), etc. As part of its *Technologies* layer, it provides an advanced support for the Java, JEE and XML technologies, including complete metamodels, corresponding model discoverers, transformations, code generators, customizations, query libraries, etc.

MoDisco (or some of its components) is being used by different partners including other academics, industrials (e.g. Sodifrance on several of their real modernization projects for their customers) or Eclipse projects (e.g. Eclipse-MDT Papyrus as developed by CEA). Moreover, the Eclipse-EMFT EMF Facet project has been initiated as a MoDisco spin-off, in order to externalize some features which are not actually specific to reverse engineering problems and thus may be reused in many different contexts (cf. corresponding EMF Facet section).

The initiative continues to be developed within the context of the European FP7-ICT project named ARTIST [0], and also to a lower extent within the context of the French FUI 13 project named TEAP.

5.3. Community-driven language development

URL: http://atlanmod.github.io/collaboro

Software development processes are collaborative in nature. Neglecting the key role of end-users leads to software that does not satisfy their needs. This collaboration becomes specially important when creating Domain-Specific Languages (DSLs), which are (modeling) languages specifically designed to carry out the tasks of a particular domain. While end-users are actually the experts of the domain for which a DSL is developed, their participation in the DSL specification process is still rather limited nowadays.

Thus, Collaboro is an approach to make language development processes more participative, meaning that both developers and users of the language can collaborate together to design it and make it evolve. Since the very first implementation of the Collaboro toolset was released, it has evolved to provide support to both Eclipse-based and web-based clients.

The Eclipse-based client has been developed as a plugin in the platform while the web-based client includes two components: (1) the server-side part, which offers a set of services to access to the main functionalities of Collaboro; and the client-side part, which allows both end-users and developers to take part of the DSML development process from their browsers. The server-side component has been developed as a Java web application which uses a set of Servlets providing the required services. On the other hand, the client-side component has been developed as an AngularJS-enabled website and provides.

[0] http://www.artist-project.eu/

The Collaboro clients provide access to the following features:

- Version view to navigate through the Proposals of a version of a language. For each Proposal, the solutions and comments are shown.
- Collaboration view to show the data related to a Collaboration selected in the version view. This view also shows the changes to apply if the selected element is a Solution.
- The user can login to the Collaboro system and create proposals, solutions and comments by right-clicking in the version view. The user can also vote for/against the collaborations.
- Decision engine based on a total agreement (i.e., all the community users must vote for the collaboration). The decision engine can be launch by using the menu bar.
- Notation engine and Notation view to render SVG snapshots of the DSL concrete syntax.
- Support for example-driven development of DSMLs, thus incoporating a graphical editor which allows end-users to draw examples of the DSML they are developing.

5.4. JSON Discoverer

URL: http://atlanmod.github.io/json-discoverer/

Given a set of JSON documents, the tool (distributed as an open source Eclipse plugin contributed to MoDisco) returns a model describing their implicit schema. We follow an iterative process where new JSON documents (from the same or different services within the API) contribute to enrich the generated model. The model helps to both understand single services and to infer possible relationships between them, thus suggesting possible compositions and providing an overall view of the application domain. The tool has also been released as a web site, thus allowing any web developer to use our approach without the need of installing Eclipse.

5.5. EMF-REST

URL: http://emf-rest.com/

EMF is *the modeling framework* of the Eclipse community. While EMF is able to automatically generate Java APIs from Ecore models, it is still missing support to deal with Web APIs such as RESTful ones that could boost the use of modeling techniques in the Web. However, the creation of RESTful APIs requires from developers not only an investment in implementation but also a good understanding of the REST Principles to apply them correctly. We therefore created EMF-REST, a tool that empowers EMF to get Truly RESTFul APIs from Ecore models, thus allowing web developers to generate JSON-based Web APIs for their applications. It generates both a JavaScript API to work with models as Javascript Objects in the client-side (without any EMF dependency) and REST services in the server-side based on the Java JAX-RS specification.

5.6. EMF Views (Model Views)

URL: https://github.com/atlanmod/emfviews

The Eclipse Modeling Framework (EMF) is widely used in the Eclipse community: defining domain models and generating corresponding source code, modeling software architectures, specifying DSL concepts or simply representing software/user data in different contexts. This implies that any software project involves a large number of heterogeneous but interrelated EMF models. To make matters worse, not all participants in the project should have the same kind of access/views on the models. Some users only need to see some parts of one model, others have to get the full model extended with data from another model, or simply access to a combination of information coming from different interconnected models. Up to now, creating such perspectives transparently in EMF was almost impossible. Based on the unquestionable success/usefulness of database views to solve similar problems in databases, EMF Views aims to bring the same concept to the modeling world. Thanks to the three main constructs (inspired from SQL) offered by the tool, designers can create new model views: SELECTing a subset of elements from a model, PROJECTing only some of the properties of those elements and/or JOINing them with elements from other models. A model view is a special type of model whose instances are directly computed at runtime based on the model view definition and concerned actual model(s). EMF Views has been initially developed in the context of the TEAP industrial project http://www.teap-project.org/ that ended in November 2014, by showing different possible applications of model views including:

- Software architect/developer views relating UML design models and Java code models (cf. Eclipse MoDisco).
- Enterprise architect views linking (BPMN) business process models, (ReqIF) requirements models and (TOGAF) architecture models.
- View transformation using dedicated technologies (e.g. Eclipse ATL).
- Report generation from views, etc.

The EMF Views prototype is currently being re-used and further developed, in a (meta)model extension context this time, within the ongoing MoNoGe industrial project. The objective of this present work is to propose a simple base generic (meta)model extension mechanism, relying on EMF Views capabilities, that could be deployed in different scenarios where (meta)model extension is required (e.g. metamodel evolution, model integration, etc.)

A presentation of EMF Views tooks place at EclipseCon 2014 [0], held in San Francisco, California, U.S.A

5.7. EMFtoCSP

URL: http://code.google.com/a/eclipselabs.org/p/emftocsp/

EMFtoCSP is a tool for the verification of precisely defined conceptual models and metamodels. For these models, the definition of the general model structure (using UML or EMF) is supplemented by OCL constraints. The Eclipse Modeling Development Tools (MDT [0]) provides mature tool support for such OCL-annotated models with respect to model definition, transformation, and validation.

However, an additional important task that is not supported by Eclipse MDT is the assurance of model quality. A systematical assessment of the correctness of such models is a key issue to ensure the quality of the final application. EMFtoCSP fills this gap by provided support for automated model verification in Eclipse.

Essentially, the EMFtoCSP is a sophisticated bounded model finder that yields instances of the model that conform not only to the structural definition of the model (e.g. the multiplicity constraints), but also to the OCL constraints. Based on this core, several correctness properties can be verified:

1. Satisfiability – is the model able to express our domain? For this check, the minimal number of instances and links can be specified to ensure non-trivial instances.
2. Unsatisfiability – is the model unable to express undesirable states? To verify this, we add further constraints to the model that state undesired conditions. Then we can check if is it impossible to instantiate the amended model.
3. Constraint subsumption – is one constraint already implied by others (and could therefore be removed)?
4. Constraint redundancy – do different constraints express the same fact (and could therefore be removed)?

To solve these search problems, EMFtoCSP translates the EMF/OCL (resp. UML/OCL) model into a constraint satisfaction problem and employs the Eclipse CLP solver [0]to solve it. This way, constraint propagation is exploited to tackle the (generally NP-hard) search.

The tool is a continuation of the UMLtoCSP approach [45] developed previously by Jordi Cabot, Robert Clarisó and Daniel Riera. It provides a generic plugin framework for Eclipse to solve OCL-annotated models using constraint logic programming. Apart from already supported Ecore and UML metamodels, further metamodels can be added easily in the future. Similarly, other constraint solving back-ends can be integrated. It is provided under the Eclipse Public License.

[0] https://www.eclipsecon.org/na2014/session/modeling-symposium
[0] http://www.eclipse.org/modeling/mdt/?project=ocl
[0] http://eclipseclp.org/

5.8. NeoEMF

URL: http://www.neo4emf.com

NeoEMF (a relaunch of the tool formerly known as Neo4EMF) is an open source software distributed under the terms of the Eclipse Public License that provides a backend-agnostic persistence solution for big, complex and highly interconnected EMF models. NeoEMF is a model repository and persistence framework allowing on-demand loading, storage, and unloading of large-scale EMF models.

NeoEMF is designed to allow the easy integration of custom backends depending on user needs. By default, NeoEMF is bundled with out-of-the-box support for graph databases (based on the blueprints API [0] and key-value stores (based on MapDB [0]). Blueprints is an abstraction layer for graph storages that allows changing the actual database used without affecting the application code. The blueprints-based back-end allows the integration of NeoEMF and Neo4j—among other databases—providing in NeoEMF the full set of features already implemented in Neo4EMF. MapDB is an efficient key-value store that provides concurrent Maps, Sets and Queues backed by disk storage or off-heap memory.

In terms of performance, NeoEMF eases data access and storage not only in a manner to reduce time and memory usage but also to allow big models to fit into small memory. This is provided through the following features:

- Lazy-loading mechanism. Model objects are loaded on demand while needed. In its basic configuration, model objects act as a proxy that ocuppy little memory, and fields are only retrieved when accessed.
- Caching. NeoEMF relies on database caches to retrieve EObjects, but in some situation this is not enough. For this reason, the architecture on NeoEMF allows the easy implementation of domain-specific cache strategies based on the decorator pattern.
- Auto-commit. In back-ends in which transaction data is stored on the heap, it is possible to use the auto-commit feature to split large transaction into several small ones.
- Dirty saving. The dirty saving feature is an step forward on the auto-commit strategy. It allows to safely handle big transactions by spliting them into small ones by saving partial changes made on models to disk. In case of transaction failure or cancellation, the partial model changes can be reverted and the model is restored to its original state.

A session about NeoEMF took place at eclipseCon France 2014 [0], held in Toulouse, France.

Works are still going over NeoEMF (within the context of the project ITM Factory - FUI14) to provide more utilities such as backend-aware query languages (which allows improving performance by taking advantage of the backend built-in query languages), concurrent access, model distribution, and other Ecore utilities.

5.9. GitHub Label Analyzer

URL: http://atlanmod.github.io/gila/

Reporting bugs, asking for new features and in general giving any kind of feedback is the easiest way to contribute to an Open-Source Software (OSS) project. In GitHub, the largest code hosting service for OSS, this feedback is typically expressed as new issues for the project managed by an issue-tracking system available in each new project repository. Among other features, the issue tracker allows creating and assigning labels to issues with the goal of helping the project community to better classify and manage those issues (e.g., facilitating the identification of issues for top priority components or candidate developers that could solve them). Nevertheless, as the project grows a manual browsing of the project issues is no longer feasible.

---

[0] https://github.com/tinkerpop/blueprints/
[0] https://github.com/jankotek/MapDB
[0] https://www.eclipsecon.org/france2014/session/neo4emf-when-big-models-are-no-longer-issue

We believe that visualization techniques could be applied here to overcome this challenge. In particular, we have created GiLA, a tool to better understand how labels are being used in GitHub projects, with the aim of providing more insights into how such projects are being managed. GiLA provides three visualizations addressing three different viewpoints, specifically:

V1 Label usage, which helps to identify the most used labels and which ones are commonly used together.

V2 User involvement, which allows discovering the most active and knowledgeable users around each label.

V3 Typical Label timeline, which provides some insights about how issues under that label evolve over time (e.g., time to be treated).

The tool can be used to explore these viewpoints on all the original projects (i.e., projects that are not a fork of a previous project) in GitHub. We believe that the results favour not only a better comprehension of the project but also help in its advancement, e.g., by helping to quickly identify experts on a particular topic/label.

AVALON Project-Team

5. New Software and Platforms

5.1. BitDew/Active Data

Participants: Gilles Fedak [correspondant], Anthony Simonet.

BITDEW is an open source middleware implementing a set of distributed services for large scale data management on Desktop Grids and Clouds. BITDEW relies on five abstractions to manage the data : i) replication indicates how many occurrences of a data should be available at the same time on the network, ii) fault-tolerance controls the policy in presence of hardware failures, iii) lifetime is an attribute absolute or relative to the existence of other data, which decides of the life cycle of a data in the system, iv) affinity drives movement of data according to dependency rules, v) protocol gives the runtime environment hints about the protocol to distribute the data (http, ftp, or bittorrent). Programmers define for every data these simple criteria, and let the BITDEW runtime environment manage operations of data creation, deletion, movement, replication, and fault-tolerance operation.

BITDEW is distributed open source under the GPLv3 or Cecill licence at the user's choice. 10 releases were produced over the last two years, and it has been downloaded approximately 6,000 times on the Inria forge. Known users are Université Paris-XI, Université Paris-XIII, University of Florida (USA), Cardiff University (UK) and University of Sfax (Tunisia). In terms of support, the development of BitDew is partly funded by the Inria ADT BitDew and by the ANR MapReduce projects. Thanks to this support, we have developed and released the first prototype of the MapReduce programming model for Desktop Grids on top of BitDew. In 2012, 8 versions of the software have been released, including the version 1.2.0 considered as a stable release of BitDew with many advanced features. Our most current extension focuses on Active Data, which is a data-centric and event-driven programming model combined with a runtime environment, which allows to expose and manage data set life cycle. Active Data strength is to facilitate the development of applications that handle dynamic data sets distributed on heterogeneous systems and infrastructures.

5.2. DIET

Participants: Daniel Balouek Thomert, Eddy Caron [correspondant], Frédéric Desprez, Maurice Faye, Arnaud Lefray, Guillaume Verger, Jonathan Rouzaud-Cornabas, Lamiel Toch, Huaxi Zhang.

Huge problems can now be processed over the Internet thanks to Grid and Cloud middleware systems. The use of on-the-shelf applications is needed by scientists of other disciplines. Moreover, the computational power and memory needs of such applications may of course not be met by every workstation. Thus, the RPC paradigm seems to be a good candidate to build Problem Solving Environments on the Grid or Cloud. The aim of the DIET project (http://graal.ens-lyon.fr/DIET) is to develop a set of tools to build computational servers accessible through a GridRPC API.

Moreover, the aim of a middleware system such as DIET is to provide a transparent access to a pool of computational servers. DIET focuses on offering such a service at a very large scale. A client which has a problem to solve should be able to obtain a reference to the server that is best suited for it. DIET is designed to take into account the data location when scheduling jobs. Data are kept as long as possible on (or near to) the computational servers in order to minimize transfer times. This kind of optimization is mandatory when performing job scheduling on a wide-area network. DIET is built upon *Server Daemons*. The scheduler is scattered across a hierarchy of *Local Agents* and *Master Agents*. Applications targeted for the DIET platform are now able to exert a degree of control over the scheduling subsystem via *plug-in schedulers*. As the applications that are to be deployed on the Grid vary greatly in terms of performance demands, the DIET plug-in scheduler facility permits the application designer to express application needs and features in order that they be taken into account when application tasks are scheduled. These features are invoked at runtime after a user has submitted a service request to the MA, which broadcasts the request to its agent hierarchy.

DIET provide a support for Cloud architecture. and it takes benefits from virtualized resources. As cloud resources are dynamic, we have on-going research in the field of automatic and elastic deployment for middleware systems. DIET will be able to extend and reduce the amount on aggregated resources and adjust itself when resources fail.

In the context of the Seed4C project, we have studied how secured our platform, authenticated and secured interactions between the different parts of our middleware and between our middleware and its users. By the way, we have added the SSL support into the DIET communication layer. We have worked to show how to securely use public cloud storage without taking the risk of losing confidentiality of data stored on them.

We have started a work to design a plug-in schedulers into DIET to deal with energy management. Using this scheduler we have obtain a significatif gain close to 25% with a minor weakening of performance (6%). Moreover we have experimented some dynamic resources management through DIET based on the energy criteria.

5.3. Sam4c

Participants: Eddy Caron, Arnaud Lefray [correspondant], Jonathan Rouzaud-Cornabas.

Sam4C (https://gforge.inria.fr/projects/sam4c/) -Security-Aware Models for Clouds- is a graphical and textual editor to model Cloud applications (as virtual machines, processes, files and communications) and describe its security policy. Sam4C is suitable to represent any static application without deadline or execution time such as n-tiers or parallel applications. This editor is generated in Java from an EMF -Eclipse Modeling Framework- metamodel to simplify any modifications or extensions. The application model and the associated security policy are compiled in a single XML file which serves as input for an external Cloud security-aware scheduler. Alongside with this editor, Cloud architecture models and provisioning algorithms are provided for simulation (in the current version) or real deployments (in future versions). During this step of development this software is private and available only for Seed4C project members. The design of Sam4c is a joint effort with INSA Centre Val de Loire.

5.4. SimGrid

Participants: Jonathan Rouzaud-Cornabas, Frédéric Suter [correspondant].

SIMGRID is a toolkit for the simulation of distributed applications in heterogeneous distributed environments. The specific goal of the project is to facilitate research in the area of parallel and distributed large scale systems, such as Grids, P2P systems and clouds. Its use cases encompass heuristic evaluation, application prototyping or even real application development and tuning. SIMGRID has an active user community of more than one hundred members, and is available under GPLv3 from http://simgrid.gforge.inria.fr/.

5.5. HLCMi, L^{2}C, & Gluon++

Participants: Hélène Coullon, Vincent Lanore, Christian Perez [correspondant], Jérôme Richard.

HLCMI (http://hlcm.gforge.inria.fr) is an implementation of the HLCM component model. HLCM is a generic extensible component model with respect to component implementations and interaction concerns. Moreover, HLCM is abstract; it is its specialization—such as HLCM/L^{2}C—that defines the primitive elements of the model, such as the primitive components and the primitive interactions.

HLCMI is making use of Model-driven Engineering (MDE) methodology to generate a concrete assembly from an high level description. It is based on the Eclipse Modeling Framework (EMF). HLCMI contains 700 Emfatic lines to describe its models and 7000 JAVA lines for utility and model transformation purposes. HLCMI is a general framework that supports several HLCM specializations: HLCM/CCM, HLCM/JAVA, HLCM/L^{2}C and HLCM/Charm++ (known as Gluon++).

L^{2}C (http://hlcm.gforge.inria.fr) is a *Low Level Component* model implementation targeting at use-cases where overhead matters such as High-Performance Computing. L^{2}C does not offer network transparency neither language transparency. Instead, L^{2}C lets the user choose between various kinds of interactions between components, some with ultra low overhead and others that support network transport. L^{2}C is extensible as additional interaction kinds can be added quite easily. L^{2}C currently supports C++, FORTRAN 2013, MPI and CORBA interactions.

Gluon++ (http://hlcm.gforge.inria.fr) is a thin component model layer added on top of Charm++ (http://charm.cs.uiuc.edu/). It defines chare components as a Charm++ chare with minimal metadata, C++ components as a C++ class with minimal metadata, (asynchronous) entry method calls between components, and plain C++ method calls between components.

L^{2}C and Gluon++ are implemented in the LLCMc++ framework (http://hlcm.gforge.inria.fr). It is distributed under a LGPL licence and represents 6400 lines of C++.

5.6. Execo

Participants: Matthieu Imbert [correspondant], Laurent Pouilloux.

`Execo`(http://execo.gforge.inria.fr) is a Python library designed for rapid prototyping of experiments on distributed systems, automatization of system administration tasks (such as deployment and configuration of distributed middleware), and creation of reproducible experiments scripts. It allows easy and asynchronous management of thousands of local or remote unix processes and offers tools for easy usage of the Grid'5000 platform services.

`Execo` currently has more than 20 users in and outside the AVALON team, who rely on it to automate experimental workflows. It was used to develop one of the two contenders who won the 2014 Grid'5000 Large Scale Deployment Challenge. It is used as a building block in the Grid'5000 metrology service and has been used to produce experimental results involved in numerous papers and reports.

It is distributed under GPLv3 and it is made of 7200 lines of code.

5.7. Kwapi

Participants: Laurent Lefèvre [correspondant], François Rossigneux, Jean-Patrick Gelas, Laurent Pouilloux.

Kwapi (https://launchpad.net/kwapi) is a software framework dealing with energy monitoring of large scale infrastructures through heterogeneous energy sensors. Kwapi has been designed inside the FSN XLCloud project for Openstack infrastructures. Through the support of Hemera Inria project, kwapi has been extended and deployed in production mode to support easy and large scale energy profiling of the Grid5000 resources.

5.8. Platforms

5.8.1. Grid'5000

Participants: Frédéric Desprez, Simon Delamare, Laurent Lefèvre, David Loup, Christian Perez, Marc Pinhède, Laurent Pouilloux.

The GRID'5000 experimental platform (http://www.grid5000.fr) is a scientific instrument to support computer science research related to distributed systems, including parallel processing, high performance computing, cloud computing, operating systems, peer-to-peer systems and networks. It is distributed on 10 sites in France and Luxembourg, including Lyon. GRID'5000 is a unique platform as it offers to researchers many and varied hardware resources and a complete software stack to conduct complex experiments, ensure reproducibility and ease understanding of results.

Not only GRID'5000 is heavily used for Avalon research, but several team members are also involved in GRID'5000 direction:

- Frédéric Desprez is leading the "Groupement d'Intérêt Scientifique Groupement Grille 5K" which drives GRID'5000.
- Laurent Lefèvre is responsible of the GRID'5000 Lyon platform and member of the GRID'5000 direction committee.
- Christian Perez is leading the Hemera initiative (https://www.grid5000.fr/Hemera) and he is a member of the GRID'5000 direction committee.
- Simon Delamare is the plateform's operational manager.

Avalon also provides an important effort for Grid'5000 operation and development by hosting several enginneers belonging to Grid'5000 technical team (Marc Pinhède, David Loup) or HEMERA IPL (Laurent Pouilloux).

CIDRE Project-Team

5. New Software and Platforms

5.1. Intrusion Detection and Privacy

Members of the team have developed several intrusion detectors and security tools: **Blare** implements our approach of illegal information flow detection at the OS level for a single node and a set of nodes; **GNG** is an intrusion detection system that correlates different sources (such as different logs) in order to identify attacks against the system. The attack scenarios are defined using the Attack Description Langage (**ADeLe**) proposed by our team; **Netzob** is an open-source tool for reverse engineering, traffic generation and fuzzing of communication protocols; a log visualization tool called **ELVIS** (Extensible Log VISualization) has been implemented in order to test our approches for log exploration.

In addition, the team participate to the development of **GEPETO** (GEoPrivacy-Enhancing TOolkit), an open source software for managing location data (in cooperation with the CNRS Lab. LAAS, Toulouse). GEPETO can be used to visualize, sanitize, perform inference attacks, and measure the utility of a particular geolocated dataset.

These tools are still under development in the team. Nevertheless, there are not new. For more details, please see previous activity reports.

COAST Team

4. New Software and Platforms

4.1. Rivage

Participant: Claudia-Lavinia Ignat [contact].

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

4.2. Replication Benchmarker

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

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

4.3. BeGoood

Participant: Gérôme Canals.

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

4.4. MUTE

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

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

COATI Project-Team

5. New Software and Platforms

5.1. JourneyPlanner

Participant: Marco Biazzini [correspondant].

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

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

5.2. Software updates

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

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

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

5.3. Platforms

5.3.1. BigGraphs

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

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

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

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

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

CTRL-A Exploratory Action

5. New Software and Platforms

5.1. Heptagon/BZR programming language

Participants: Gwenaël Delaval [Contact person], Eric Rutten.

We want to produce results concretely usable by third parties, either in cooperative projects, or by free diffusion of tools. One perspective is to build tool boxes for the design of continuous control solutions for computing systems: it will be explored in the future. A readily available result concerns discrete control and programming.

HEPTAGON is a dataflow synchronous language, inspired from LUCID SYNCHRONE[0]. Its compiler is meant to be simple and modular, allowing this language to be a good support for the prototyping of compilation methods of synchronous languages.

HEPTAGON has been used to build BZR [0], which is an extension of the former with contracts constructs. These contracts allow to express dynamic temporal properties on the inputs and outputs of HEPTAGON node. These properties are then enforced, within the compilation of a BZR program, by discrete controller synthesis, using the SIGALI tool [0]. The synthesized controller is itself generated in HEPTAGON, allowing its analysis and compilation towards different target languages (C, Java, VHDL).

Prospects about Heptagon/BZR lie in: support for programming methodologies in BZR: debug, diagnosis, abstraction and composition, ...; extensions towards use of more expressive synthesis tools ; integration of target code into execution platforms (Fractal, reconfigurable FPGA, ...)

[0]http://www.di.ens.fr/~pouzet/lucid-synchrone
[0]http://bzr.inria.fr
[0]http://www.irisa.fr/vertecs/Logiciels/sigali.html

DANTE Team

5. New Software and Platforms

5.1. Sensor Network Tools: drivers, OS and more

As outcomes of the Equipped FIT IoT-LAB, ANR SensLAB project and the Inria ADT SensTOOLS and SensAS, several softwares (from low level drivers to OSes) were delivered and made available to the research community. The main goal is to lower the cost of developing/deploying a large scale wireless sensor network application. All software are gathered under the IoT-LAB web site: https://www.iot-lab.info web page where one can find:

- low C-level drivers to all hardware components;
- ports of the main OS, mainly FreeRTOS, Contiki, TinyOS, Riot, Linux;
- ports and development of higher level library like routing, localization.

IoT-LAB software is licensed under a CeCILL License. IoT-LAB users are welcome to contribute code, papers, tutorials or experiments reports.

5.2. Queueing Systems

Online tool: http://queueing-systems.ens-lyon.fr

This tool aims at providing a simple web interface to promote the use of our proposed solutions to numerically solve classical queueing systems. It is a joint project between Thomas Begin (DANTE) and Pr. Brandwajn (UCSC). This tool supported since 2011 attracts each month hundreds of visitors from all around the world. Its current implementation includes the solution to:

- a queue with multiple servers, general arrivals, exponential services and a possibly finite buffer (*i.e.*, $Ph/M/c/N$-like queue) (refer to [32] for more details);
- a single server queue with Poisson arrivals, general services and a possibly finite buffer (*i.e.*, $M/Ph/1/N$-like queue);
- a queue with multiple servers, general service times and Poisson arrivals (*i.e.*, $M/Ph/c/N$-like queue) based on a recent work that was published in 2014 in Performance Evaluation [4]. Associated URL is: http://queueing-systems.ens-lyon.fr

DIANA Team

4. New Software and Platforms

4.1. ns-3

Participants: Walid Dabbous [correspondant], Thierry Turletti.

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

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

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

4.2. DCE

Participants: Thierry Turletti [correspondant], Walid Dabbous.

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

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

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

4.3. NEPI

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

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

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

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

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

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

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

4.4. OpenLISP

Participant: Damien Saucez [correspondant].

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

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

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

4.5. ACQUA

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

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

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

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

4.6. ElectroSmart

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

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

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

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

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

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

4.7. Platforms

4.7.1. Reproducible research laboratory ($\mathrm{R^2LAB}$)

Scientific evaluation of network protocols requires that experiment results must be reproducible before they can be considered as valid. This is particularly difficult to obtain in the wireless networking domain, where characteristics of wireless channels are known to be variable, unpredictable and hardly controllable. Indeed, anechoic chambers with RF absorbers preventing radio waves reflections and with Faraday cage blocking external interferences represent an ideal environment for experiments reproducibility. This year witnessed the realization of such experimental platform (called $\mathrm{R^2LAB}$ or Reproducible Research Laboratory) at Inria Sophia-Antipolis, in the context of the FIT 'Equipment of Excellence' project. The objectives of this platform are twofold : on the one hand, we need to achieve highly controllable wireless experiments (e.g. control plane for 5G), and to this end, the testbed features an anechoic chamber. On the other hand, we need to make it possible to deploy experiments that have demanding resource requirements, as this is typically the case with e.g. ICN-based research, or when involving simulation. For that reason, the platform features some powerful servers of its own; in addition, these experiments can be either hybrid-experiments (as NEPI will be deployed) or federated experiments through several testbeds such as PlanetLab. As the final objective is to provide an environment to easily run realistic and reproducible wireless experiments and simulations, it is important to be able to increase the testbed realism by injecting noise and interfering signals in a controllable way. Experimentation results done in $\mathrm{R^2LAB}$ could also be used to augment the realism of propagation models in simulators, which are able to run large scale scenarios. We are currently deploying the wireless nodes. The next step will to extend the testbed to support software defined networking (SDN) and LTE experimentations and to install specific tools for operating the platform.

DIONYSOS Project-Team

4. New Software and Platforms

4.1. T3devKit testing toolkit and IPv6 test suites

Participant: César Viho.

We have built a toolkit for easing executing tests written in the standardized TTCN-3 test specification language. This toolkit is made of a C++ library together with a highly customizable CoDec generator that allows fast development of external components (that are required to execute a test suite) such as CoDec (for message Coding/Decoding), System and Platform Adapters. It also provides a framework for representing and manipulating TTCN-3 events so as to ease the production of test reports. The toolkit addresses issues that are not yet covered by ETSI standards while being fully compatible with the existing standard interfaces: TRI (Test Runtime Interfaces) and TCI (Test Control Interfaces), it has been tested with four TTCN-3 environments (IBM, Elvior, Danet and Go4IT) and on three different platforms (Linux, Windows and Cygwin). It is publicly released under the CeCILL-C License.

All these tools with associated test suites (for RIPng, DHCPv6 and examples for DNS) are freely available at http://www.irisa.fr/tipi.

4.2. Interoperability Assessment

Participant: César Viho.

Our experience in interoperability assessment (since 1996) and in using the TTCN-3 standard allowed us to develop a tool (called `ttproto`) that helps in: (i) experimenting new concepts for long term evolution of the TTCN-3 standard and (ii) facilitating new approaches and methods for interoperability assessment. For instance, new passive approaches that we developed have been implemented and validated using `ttproto`. This tool `ttproto` has been used to develop test suites for 6LoWPAN-ND (IPv6 for Low Power Networks) and CoAP (Constrained Application Protocol). The CoAP test suites have been successfully used for two Plugtest interoperability events organized by ETSI, IPSO Alliance and the FP7 PROBE-IT project. The tool `ttproto` and the test suites indicated above are freely available at http://www.irisa.fr/tipi.

4.3. Performance and dependability evaluation

Participants: Gerardo Rubino, Bruno Sericola, Bruno Tuffin.

We develop software tools for the evaluation of two classes of models: Markov models and reliability networks. The main objective is to quantify dependability aspects of the behaviors of the modeled systems, but other aspects of the systems can be handled (performance, performability, vulnerability). The tools are specialized libraries implementing numerical, Monte Carlo and Quasi-Monte Carlo algorithms.

One of these libraries has been developed for the Celar (DGA), and its goal is the evaluation of dependability and vulnerability metrics of wide area communication networks (WANs). The algorithms in this library can also evaluate the sensitivities of the implemented dependability measures with respect to the parameters characterizing the behavior of the components of the networks (nodes, lines).

We are also developing tools with the objective of building Markovian models and to compute bounds of asymptotic metrics such as the asymptotic availability of standard metrics of models in equilibrium, loss probabilities, blocking probabilities, mean backlogs, etc. A set of functions designed for dependability analysis is being built under the name `DependLib`.

Pierre L'Ecuyer is also developing in Montreal a library, *Stochastic Simulation in Java* (SSJ), providing facilities for generating uniform and nonuniform random variates, computing different measures related to probability distributions, performing goodness-of-fit tests, applying quasi-Monte Carlo methods, collecting (elementary) statistics, and programming discrete-event simulations with both events and processes.

DIVERSE Project-Team

5. New Software and Platforms

5.1. Kermeta

Participants: Benoit Combemale [correspondant], Olivier Barais, Arnaud Blouin, Didier Vojtisek, Benoit Baudry, Thomas Degueule, David Mendez, Erwan Bousse, Francois Tanguy, Fabien Coulon.

Nowadays, object-oriented meta-languages such as MOF (Meta-Object Facility) are increasingly used to specify domain-specific languages in the model-driven engineering community. However, these meta-languages focus on structural specifications and have no built-in support for specifications of operational semantics. Integrated with the Ecore industrial standard and aligned with the EMOF 2.0 OMG standard, the Kermeta language consists in a extension to these meta-languages to support behavior definition. The language adds precise action specifications with static type checking and genericity at the meta level. Based on object-orientation and aspect orientation concepts, the Kermeta language adds model specific concepts.

Kermeta is used in several use cases:

- to give a precise semantic of the behavior of a metamodel, which then can be simulated;
- to act as a model transformation language;
- to act as a constraint language.

In 2014, we have continued the refactoring of Kermeta to leverage on Xtend. The Kermeta action language is now defined as an extension of Xtend, by proposing model-specific features (e.g., model type, containment, opposite) and an open class mechanism for aspect weaving. The main objective of this new refactoring was to benefit from the non model–specific features of Xtend (including the basics of the action language and its respective tooling such as editor, type checker and compiler), and to focus in our development on innovative solutions for MDE.

More precisely, in addition to an Xtend extension dedicated to model manipulation, we started to integrate in Kermeta various facilities to support software language engineering (slicing, pruning, reuse, variability management, etc). In 2014, we improved this software language engineering feature (currently named k3sle/Melange) in order to offer a functional model typing system allowing safe model polymorphism. This system enables reuse of algorithms and transformations accross different metamodels, as well as language inheritance, evolution and interoperability.

Moreover, while this version of Kermeta is a DSML development workbench that provides good support for developping independent DSMLs, little or no support is provided for integrated use of multiple DSMLs. The lack of support for explicitly relating concepts expressed in different DSMLs makes it very difficult for developers to reason about information spread across models describing different system aspects.

According to Google Scholar [0], the Kermeta platform was used or cited in more than 1300 papers.

5.2. FAMILIAR

Participants: Mathieu Acher [correspondant], Olivier Barais, Guillaume Bécan, Aymeric Hervieu, Julien Richard-Foy, Sana Ben Nasr, Edward Mauricio Alferez Salinas, João Ferreira Filho, Didier Vojtisek, Benoit Baudry.

[0] http://scholar.google.fr/scholar?q=kermeta+model

Modeling and reasoning about configuration options is crucial for the effective management of configurable software systems and product lines. The FAMILIAR project provides dedicated languages, APIs, and comprehensive environments for that purpose. Specifically, FAMILIAR provides support for feature models (by far the most popular notation). The feature models formalism has been studied for more than 20 years [98], and it is widely used in the industry [100]. FAMILIAR (for FeAture Model scrIpt Language for manIpulation and Automatic Reasoning) provides a scripting language for importing, exporting, composing, decomposing, editing, configuring, computing "diffs", refactoring, reverse engineering, testing, and reasoning about (multiple) feature models. For interoperability, many bridges with existing feature modeling languages are implemented. All these operations can be combined to perform complex variability management tasks: extraction of feature models from software artifacts [87], product line evolution [89], management of multiple models [88] [75], [76], model-based validation of SPLs [22], large scale configuration of feature models [122], etc. The level of maturity of the FAMILIAR platform is TRL 3 (*i.e.* New technology tested Prototype built and functionality demonstrated through testing over a limited range of operating conditions. These tests can be done on a scaled version if scalable).

Main innovative features:

- reverse engineering of variability models from multiple kinds of artefacts;
- composition of multiple variability models (e.g., for combining different sources of variability);
- slicing of variability model (e.g., for scheduling a configuration process in different steps);
- connection with the Common Variability Language (CVL);
- support of advanced variability constructs (e.g., attributes, multi-features, meta-information);
- Web-based, comprehensive environment (WebFML [42]).

Impact:

The results are connected to the CVL standardization initiative. From a research perspective, FAMILIAR helps to support all the research activity on variability modeling (e.g., design of new operators, benchmarking). Several tutorials and tool demonstrations [42], [25] have been performed at SPLC (the major conference in software product lines), at ECOOP, at CIEL and MODELS in 2012 and 2013. FAMILIAR is also used in the context of teaching activities. From an industrial perspective, the languages and tools have already been applied in practical contexts in different application domains (medical imaging, video surveillance, system engineering, web configurators, etc.) and for various purposes. This platform is also used for supporting industrial transfer activity with companies such as Thales. FAMILIAR is involved in several research projects (e.g., in the Merge ITEA project, in the MOTIV project, in the VaryMDE project).

FAMILIAR is distributed under the terms of the LGPL and EPL open source license.

See also the web page familiar-project.github.com.

- Version: 1.3
- Programming language: Java, Scala

5.3. Kevoree

Participants: Olivier Barais [correspondant], Johan Bourcier, Noel Plouzeau, Benoit Baudry, Maxime Tricoire, Jacky Bourgeois, Inti G. Herrera, Ivan Paez, Francisco Acosta, Mohamed Boussaa.

Kevoree is an open-source models@runtime platform [0] to properly support the dynamic adaptation of distributed systems. Models@runtime basically pushes the idea of reflection [132] one step further by considering the reflection layer as a real model that can be uncoupled from the running architecture (e.g. for reasoning, validation, and simulation purposes) and later automatically resynchronized with its running instance.

[0] http://www.kevoree.org

Kevoree has been influenced by previous work that we carried out in the DiVA project [132] and the Entimid project [135]. With Kevoree we push our vision of models@runtime [131] farther. In particular, Kevoree provides a proper support for distributed models@runtime. To this aim we introduced the *Node* concept to model the infrastructure topology and the *Group* concept to model semantics of inter node communication during synchronization of the reflection model among nodes. Kevoree includes a *Channel* concept to allow for multiple communication semantics between remote*Components* deployed on heterogeneous nodes. All Kevoree concepts (Component, Channel, Node, Group) obey the object type design pattern to separate deployment artifacts from running artifacts. Kevoree supports multiple kinds of very different execution node technology (e.g. Java, Android, MiniCloud, FreeBSD, Arduino, ...).

Kevoree is distributed under the terms of the LGPL open source license.

Main competitors:

- the Fractal/Frascati eco-system [0].
- SpringSource Dynamic Module [0]
- GCM-Proactive [0]
- OSGi [0]
- Chef [0]
- Vagran [0]

Main innovative features:

- distributed models@runtime platform (with a distributed reflection model and an extensible models@runtime dissemination set of strategies).
- Support for heterogeneous node type (from Cyber Physical System with few resources until cloud computing infrastructure).
- Fully automated provisioning model to correctly deploy software modules and their dependencies.
- Communication and concurrency access between software modules expressed at the model level (not in the module implementation).

Impact:

A tutorial have been performed at the Comparch conference in July 2014 and at the Middleware conference in december 2014. See also the web page http://www.kevoree.org.

In 2014, we mainly created a new implementation in JavaScript and we created an implementation for system containers for driving resources using Kevoree. We also use Kevoree in the context of Mohammed's PhD to create testing infrastructure on-demand.

- Version: 5.1.4
- Programming language: Java, Scala, Kermeta, Kotlin, Javascript

[0] http://frascati.ow2.org
[0] http://spring.io/
[0] http://proactive.inria.fr/
[0] http://www.osgi.org
[0] http://wiki.opscode.com/display/chef/Deploy+Resource
[0] http://vagrantup.com/

DYOGENE Project-Team

5. New Software and Platforms

5.1. Clones: CLOsed queueing Networks Exact Sampling

Clones is a Matlab toolbox for exact sampling of closed queueing networks.

Details can be found in [18] (best tool-paper award).

Available at: http://www.di.ens.fr/~rovetta/Clones/index.html

FOCUS Project-Team

5. New Software and Platforms

5.1. Jolie

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

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

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

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

5.2. Others

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

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

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

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

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

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

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

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

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

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

The sofware below have not undergone substantial modifications during 2014.

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

FUN Project-Team

4. New Software and Platforms

4.1. New ALE module for ASPIRERFID middleware.

Participants: Rim Driss [correspondant], Nathalie Mitton, Ibrahim Amadou, Julien Vandaele.

AspireRFID middleware is a modular OW2 open source RFID middleware. It is compliant with EPC Global standards. This new module integrates the modifications of the new standard release, including new RP and LLRP definitions and fixing bugs. This module has been implemented in the framework of the MIAOU project.

- Version: 1.0. . APP number: IDDN.FR.001.100017.000.S.P.2012.000.10000

4.2. T-SCAN.

Participants: Gabriele Sabatino [correspondant], Nathalie Mitton.

T-Scan is an interface ensuring the translation from a SGTIN tag format to an ONS hostname format according to the EPCGlobal standards. It allows the sending of a DNS request to look up the EPC-IS aides to which the product belongs in order to access the data relative to that product. This module has been implemented in the framework of the TRACAVERRE project.

- Version: 1.0 . March 1st 2014. N IDDN abrégé : 14-440017-000

4.3. GOLIATH 1.0

Participants: Nathalie Mitton [correspondant], Salvatore Guzzo Bonifacio [correspondant].

GOLIATH (Generic Optimized LIghtweight communication stack for Ambient TecHnologies) is a full protocol stack for wireless sensor networks. This module has been implemented in the framework of the ETIPOPS project.

See also the web page https://gforge.inria.fr/projects/goliath/.

4.4. ETINODE-CONTIKI-PORT

Participants: Salvatore Guzzo Bonifacio [correspondant], Roudy Dagher, Nathalie Mitton.

Contiki is an open source embedded OS for Internet of Things (IoT). It is light and portable to different hardware architectures. It embeds communication stacks for IoT Il embarque aussi des piles de communication pour l'internet des objets. This driver allows the running of Contiki OS over Etinode-MSP430. The code dalso allows the use of radio chip and embedded sensors. This module has been implemented in the framework of the ETIPOPS project.

- Version: 1.0 .

4.5. ETINODE-DRIVERS

Participants: Salvatore Guzzo Bonifacio [correspondant], Roudy Dagher, Nathalie Mitton.

These drivers for Etinode-MSP430 control the different embedded sensors and hardware components available on an Etinode-MSP430 node such as gyroscope, accelerometer and barometric sensor. This module has been implemented in the framework of the ETIPOPS project.

- Version: 1.0 .

4.6. FIT IoT-Lab

Participants: Raymond Borenstein, Nathalie Mitton [correspondant], Anne-Sophie Tonneau, Julien Vandaele, Roberto Quilez.

FIT IoT-LAB is a very large scale open testbed that features over 2700 wireless sensor nodes and more than 200 robots spread across six different sites in France. Nodes are either fixed or mobile and can be allocated in various topologies throughout all sites. A variety of wireless sensors are available, with different processor architectures (MSP430, STM32 and Cortex-A8) and different wireless chips (802.15.4 PHY at 800 MHz or 2.4 GHz). In addition, "open nodes" can receive custom wireless sensors for inclusion in IoT-LAB testbed. This platform is completely open and can be used by any one wishing to run experiment on wireless sensors and robots.

The Lille site displays 3 subsets of the platforms:

- Euratechnologies : this site features 256 WSN430 sensor nodes operating in the 2.4GHz band. 64 nodes are mobile, embedded on mobile trains.
- Haute Borne : this site features 256 M3 sensor nodes operating in the 2.4GHz band and 64 mobile robots (32 turtlebots and 32 wifibots) completely remotely programmable.
- Opennodes : this site will feature (opening beginning 2015) 64 hardware open slots to allow any one to plug his own hardware and benefits from the platform debugging and monitoring tools.

GANG Project-Team (section vide)

HIEPACS Project-Team

5. New Software and Platforms

5.1. Introduction

We describe in this section the software that we are developing. The first list will be the main milestones of our project. The other software developments will be conducted in collaboration with academic partners or in collaboration with some industrial partners in the context of their private R&D or production activities. For all these software developments, we will use first the various (very) large parallel platforms available through GENCI in France (CCRT, CINES and IDRIS Computational Centers), and next the high-end parallel platforms that will be available via European and US initiatives or projects such that PRACE.

5.2. MaPHyS

Participant: Emmanuel Agullo [corresponding member].

MaPHyS (Massivelly Parallel Hybrid Solver) is a software package that implements a parallel linear solver coupling direct and iterative approaches. The underlying idea is to apply to general unstructured linear systems domain decomposition ideas developed for the solution of linear systems arising from PDEs. The interface problem, associated with the so called Schur complement system, is solved using a block preconditioner with overlap between the blocks that is referred to as Algebraic Additive Schwarz.

The MaPHyS package is very much a first outcome of the research activity described in Section 3.3 . Finally, MaPHyS is a preconditioner that can be used to speed-up the convergence of any Krylov subspace method. We forsee to either embed in MaPHyS some Krylov solvers or to release them as standalone packages, in particular for the block variants that will be some outcome of the studies discussed in Section 3.3 .

MaPHyS can be found at http://maphys.gforge.inria.fr.

5.3. PaStiX

Participant: Pierre Ramet [corresponding member].

Complete and incomplete supernodal sparse parallel factorizations.

PaStiX (Parallel Sparse matriX package) is a scientific library that provides a high performance parallel solver for very large sparse linear systems based on block direct and block ILU(k) iterative methods. Numerical algorithms are implemented in single or double precision (real or complex): LLt (Cholesky), LDLt (Crout) and LU with static pivoting (for non symmetric matrices having a symmetric pattern).

The PaStiX library uses the graph partitioning and sparse matrix block ordering package Scotch. PaStiX is based on an efficient static scheduling and memory manager, in order to solve 3D problems with more than 50 million of unknowns. The mapping and scheduling algorithm handles a combination of 1D and 2D block distributions. This algorithm computes an efficient static scheduling of the block computations for our supernodal parallel solver which uses a local aggregation of contribution blocks. This can be done by taking into account very precisely the computational costs of the BLAS 3 primitives, the communication costs and the cost of local aggregations. We also improved this static computation and communication scheduling algorithm to anticipate the sending of partially aggregated blocks, in order to free memory dynamically. By doing this, we are able to reduce the aggregated memory overhead, while keeping good performance.

Another important point is that our study is suitable for any heterogeneous parallel/distributed architecture when its performance is predictable, such as clusters of multicore nodes. In particular, we now offer a high performance version with a low memory overhead for multicore node architectures, which fully exploits the advantage of shared memory by using an hybrid MPI-thread implementation.

Direct methods are numerically robust methods, but the very large three dimensional problems may lead to systems that would require a huge amount of memory despite any memory optimization. A studied approach consists in defining an adaptive blockwise incomplete factorization that is much more accurate (and numerically more robust) than the scalar incomplete factorizations commonly used to precondition iterative solvers. Such incomplete factorization can take advantage of the latest breakthroughs in sparse direct methods and particularly should be very competitive in CPU time (effective power used from processors and good scalability) while avoiding the memory limitation encountered by direct methods.

PaStiX is publicly available at http://pastix.gforge.inria.fr under the Inria CeCILL licence.

5.4. HIPS

Participant: Pierre Ramet [corresponding member].

Multilevel method, domain decomposition, Schur complement, parallel iterative solver.

HIPS (Hierarchical Iterative Parallel Solver) is a scientific library that provides an efficient parallel iterative solver for very large sparse linear systems.

The key point of the methods implemented in HIPS is to define an ordering and a partition of the unknowns that relies on a form of nested dissection ordering in which cross points in the separators play a special role (Hierarchical Interface Decomposition ordering). The subgraphs obtained by nested dissection correspond to the unknowns that are eliminated using a direct method and the Schur complement system on the remaining of the unknowns (that correspond to the interface between the sub-graphs viewed as sub-domains) is solved using an iterative method (GMRES or Conjugate Gradient at the time being). This special ordering and partitioning allows for the use of dense block algorithms both in the direct and iterative part of the solver and provides a high degree of parallelism to these algorithms. The code provides a hybrid method which blends direct and iterative solvers. HIPS exploits the partitioning and multistage ILU techniques to enable a highly parallel scheme where several subdomains can be assigned to the same process. It also provides a scalar preconditioner based on the multistage ILUT factorization.

HIPS can be used as a standalone program that reads a sparse linear system from a file ; it also provides an interface to be called from any C, C++ or Fortran code. It handles symmetric, unsymmetric, real or complex matrices. Thus, HIPS is a software library that provides several methods to build an efficient preconditioner in almost all situations.

HIPS is publicly available at http://hips.gforge.inria.fr under the Inria CeCILL licence.

5.5. MetaPart

Participant: Aurélien Esnard [corresponding member].

MetaPart is a library that adresses the challenge of (dynamic) load balancing for emerging complex parallel simulations, such as multi-physics or multi-scale coupling applications. First, it offers a uniform API over state-of-the-art (hyper-) graph partitioning software packages such as Scotch, PaToH, METIS, Zoltan, Mondriaan, etc. etc. Based upon this API, it provides a framework that facilitates the development and the evaluation of high-level partitioning methods, such as MxN repartitioning or coupling-aware partitionining (co-partitioning).

The framework is publicy available at Inria Gforge: http://metapart.gforge.inria.fr.

5.6. MPICPL

Participant: Aurélien Esnard [corresponding member].

MPICPL (MPI CouPLing) is a software library dedicated to the coupling of parallel legacy codes, that are based on the well-known MPI standard. It proposes a lightweight and comprehensive programing interface that simplifies the coupling of several MPI codes (2, 3 or more). MPICPL facilitates the deployment of these codes thanks to the *mpicplrun* tool and it interconnects them automatically through standard MPI inter-communicators. Moreover, it generates the universe communicator, that merges the world communicators of all coupled-codes. The coupling infrastructure is described by a simple XML file, that is just loaded by the *mpicplrun* tool.

MPICPL was developed by HIEPACS for the purpose of the ANR NOSSI. It uses advanced features of MPI2 standard. The framework is publicy available at Inria Gforge: http://mpicpl.gforge.inria.fr.

5.7. ScalFMM

Participant: Olivier Coulaud [corresponding member].

ScalFMM (Parallel Fast Multipole Library for Large Scale Simulations) is a software library to simulate N-body interactions using the Fast Multipole Method.

ScalFMM intends to offer all the functionalities needed to perform large parallel simulations while enabling an easy customization of the simulation components: kernels, particles and cells. It works in parallel in a shared/distributed memory model using OpenMP and MPI. The software architecture has been designed with two major objectives: being easy to maintain and easy to understand. There are two main parts: 1) the management of the octree and the parallelization of the method ; 2) the kernels. This new architecture allows us to easily add new FMM algorithm or kernels and new paradigm of parallelization. The code is extremely documented and the naming convention fully respected. Driven by its user-oriented philosophy, ScalFMM is using CMAKE as a compiler/installer tool. Even if ScalFMM is written in C++ it will support a C and fortran API soon.

The library offers two methods to compute interactions between bodies when the potential decays like $1/r$. The first method is the classical FMM based on spherical harmonic expansions and the second is the Black-Box method which is an independent kernel formulation (introduced by E. Darve at Stanford). With this method, we can now easily add new non oscillatory kernels in our library. For the classical method, two approaches are used to decrease the complexity of the operators. We consider either matrix formulation that allows us to use BLAS routines or rotation matrix to speed up the M2L operator.

The ScalFMM package is available at http://scalfmm.gforge.inria.fr

5.8. ViTE

Participant: Mathieu Faverge [corresponding member].

Visualization, Execution trace

ViTE is a trace explorer. It is a tool made to visualize execution traces of large parallel programs. It supports Pajé, a trace format created by Inria Grenoble, and OTF and OTF2 formats, developed by the University of Dresden and allows the programmer a simpler way to analyse, debug and/or profile large parallel applications. It is an open source software licenced under CeCILL-A.

The ViTE software is available at http://vite.gforge.inria.fr and has been developed in collaboration with the Inria Bordeaux - Sud-Ouest SED team, Telecom SudParis and Inria Grenoble.

In the same context we also contribute to the EZtrace and GTG libraries in collaboration with F. Trahay from Telecom SudParis. EZTrace (http://eztrace.gforge.inria.fr) is a tool that aims at generating automatically execution trace from HPC programs. It generates execution trace files thanks to the GTG library (http://gtg.gforge.inria.fr) that can be later interpreted by visualization tools such as ViTE.

5.9. Other software

For the materials physics applications, a lot of development will be done in the context of ANR projects (NOSSI and OPTIDIS, see Section 4.1) in collaboration with LaBRI, CPMOH, IPREM, EPFL and with CEA Saclay and Bruyère-le-Châtel.

- **FAST**
  Participant: Olivier Coulaud [corresponding member].

  FAST is a linear response time dependent density functional program for computing the electronic absorption spectrum of molecular systems. It uses an O(N^3) linear response method based on finite numerical atomic orbitals and deflation of linear dependence in atomic orbital product space. This version is designed to work with data produced by the SIESTA DFT code. The code produces as principal output a numerical absorption spectrum (complex part of the polarisability, loosely called the polarisability below) and a list of transition energies and oscillator strengths deduced from fitting Lorentzians to the numerical spectrum. Considering the absence of hybrid functionals in SIESTA and that concerning calculation of spectra, generalized gradient Hamiltonians are not usually considered to be notably better than the local density approximation, the present release of FAST works only with LDA, which despite its limitations, has provided useful results on the systems to which the present authors have applied it. The FAST library is available at http://people.bordeaux.inria.fr/coulaud/Softwares/FAST/index.html.

- **OptiDis**
  Participant: Olivier Coulaud [corresponding member].

  OptiDis is a new code for large scale dislocation dynamics simulations. Its aim is to simulate real life dislocation densities (up until 5.10^{22} dislocations/m^{-2}) in order to understand plastic deformation and study strain hardening. The main application is to observe and understand plastic deformation on irradiated zirconium. Zirconium alloys is the first containment barrier against the dissemination of radioactive elements. More precisely, with neutron irradiated zirconium alloys we are talking of channeling mechanism, which means to stick with the reality,more than tens of thousands of induced loops so 10^8 degrees of freedom in the simulation.

  The code is based on Numodis code developed at CEA Saclay and the ScalFMM library developed in our Inria project. The code is written in C++ language and using the last features of C++11. One of the main aspects is the hybrid parallelism MPI/OpenMP that gives the software the ability to scale on large cluster while the computation load rises. In order to achieve that, we use different levels of parallelism. First of all, the simulation box is spread over MPI processes, we then use a thinner level for threads, dividing the domain using an Octree representation. All theses parts are driven by the ScalFMM library. On the last level our data are stored in an adaptive structure absorbing dynamic of this kind of simulation and handling well task parallelism.

The two following packages are mainly designed and developed in the context of a US initiative led by ICL and to which we closely collaborate through the associate team MORSE.

- **PLASMA**
  Participant: Mathieu Faverge [corresponding member].

  The PLASMA (Parallel Linear Algebra for Scalable Multi-core Architectures) project aims at addressing the critical and highly disruptive situation that is facing the Linear Algebra and High Performance Computing community due to the introduction of multi-core architectures.

  The PLASMA ultimate goal is to create software frameworks that enable programmers to simplify the process of developing applications that can achieve both high performance and portability across a range of new architectures.

  The development of programming models that enforce asynchronous, out of order scheduling of operations is the concept used as the basis for the definition of a scalable yet highly efficient software framework for Computational Linear Algebra applications.

The PLASMA library is available at http://icl.cs.utk.edu/plasma.

- **PaRSEC/DPLASMA**
  Participant: Mathieu Faverge [corresponding member].

  PaRSEC Parallel Runtime Scheduling and Execution Controller, is a generic framework for architecture aware scheduling and management of micro-tasks on distributed many-core heterogeneous architectures. Applications we consider can be expressed as a Direct Acyclic Graph of tasks with labeled edges designating data dependencies. DAGs are represented in a compact problem-size independent format that can be queried on-demand to discover data dependencies in a totally distributed fashion. PaRSEC assigns computation threads to the cores, overlaps communications and computations and uses a dynamic, fully-distributed scheduler based on architectural features such as NUMA nodes and algorithmic features such as data reuse.

  The framework includes libraries, a runtime system, and development tools to help application developers tackle the difficult task of porting their applications to highly heterogeneous and diverse environments.

  DPLASMA (Distributed Parallel Linear Algebra Software for Multicore Architectures) is the leading implementation of a dense linear algebra package for distributed heterogeneous systems. It is designed to deliver sustained performance for distributed systems where each node featuring multiple sockets of multicore processors, and if available, accelerators like GPUs or Intel Xeon Phi. DPLASMA achieves this objective through the state of the art PaRSEC runtime, porting the PLASMA algorithms to the distributed memory realm.

  The PaRSEC runtime and the DPLASMA library are available at http://icl.cs.utk.edu/parsec.

5.10. Platforms

5.10.1. PlaFRIM: an experimental parallel computing platform

PLAFRIM is an experimental platform for research in modeling, simulations and high performance computing. This platform has been set up from 2009 under the leadership of Inria Bordeaux Sud-Ouest in collaboration with computer science and mathematics laboratories, respectively Labri and IMB with a strong support in the region Aquitaine.

It aggregates different kinds of computational resources for research and development purposes. The latest technologies in terms of processors, memories and architecture are added when they are available on the market. It is now more than 1,000 cores (excluding GPU and Xeon Phi) that are available for all research teams of Inria Bordeaux, Labri and IMB. This computer is in particular used by all the engineers who work in HiePACS and are advised by F. Rue from the SED.

The PlaFRIM platform initiative is coordinated by O. Coulaud.

HIPERCOM2 Team

5. New Software and Platforms

5.1. OPERA and OCARI Software

Participants: Cédric Adjih, Ichrak Amdouni, Pascale Minet, Ridha Soua, Erwan Livolant.

The OPERA software was developed by the Hipercom2 team in the OCARI project (see https://ocari.org/). It includes EOLSR, an energy efficient routing protocol and OSERENA, a coloring algorithm optimized for dense wireless networks. It was registered by the APP. In 2013, OPERA has been made available for download as an open software from the InriaGForge site: https://gforge.inria.fr/scm/?group_id=4665

In 2014, OPERA has been ported on a more powerful platform based on the Atmel transceiver AT86RF233 and on a 32 bits microcontroler Cortex M3.

More details and documentation about this software are available in the website made by the Hipercom2 team: http://opera.gforge.inria.fr/index.html

Erwan Livolant, Pascale Minet from Inria as well as Tuan Dang from EDF and Maurice Sellin from DCNS showed the wireless sensor network OCARI during the Inria-Industry Meeting devoted to Telecommunications organized by Inria at Rocquencourt in November 2014. Two types of demonstration were done: one illustrating the internal behavior of OCARI and the other one illustrating a fire detection in a DCNS ship.

5.2. SAHARA Software

Participants: Erwan Livolant, Pascale Minet, Ridha Soua.

The software module DimTool developed in 2014 by the Hipercom2 team in the SAHARA project has been registered by the APP in January 2014. It helps to dimension the network parameters of the SAHARA wireless sensor network and evaluate the feasibility of a given application.

5.3. CONNEXION Software

Participants: Ines Khoufi, Pascale Minet, Erwan Livolant.

In 2014, we developped two softwares to compute the positions of:

- sensor nodes that ensure full coverage of a 2-D area with irregular shape and containing obstacles.
- relay nodes that maintain a robust connectivity of each point of interest with the sink. The area may contain obstacles.

With regard to the wireless sensor network OCARI, in 2014 we developed the interface of wireless temperature sensors PT100 with the OCARI stack. We also gave our support to CEA for the development of the OCARI interface of smoke detectors. With Telecom ParisTech, we interconnected OCARI with the industrial facility backbone based on OPC/UA via a gateway implemented on a Raspberry Pi. More precisely, we developed the serial interface between the OCARI network coordinator and the OPC/UA server.

5.4. NS3 Network Coding Software

Participants: Cédric Adjih, Ichrak Amdouni, Hana Baccouch.

One output of the GETRF project, was the creation of a solution for Network Coding, called DragonNet. DragonNet is a complete modular solution. This solution is responsible of: coding, decoding, maintaining necessary information and the associated signaling. It is designed to be extensible. A variant of DragonNet has been specified for wireless sensor networks and implemented.

As a follow-up to the ADT MOBSIM (and the previous module EyWifi), DragonNet was also integrated as a module for the ns-3 simulation tool.

5.5. FIT IoT-LAB Platforms

Participants: Cédric Adjih, Alaeddine Weslati, Vincent Ladeveze, Ichrak Amdouni.

This is a joint work with Emmanuel Baccelli from Inria Saclay.

Period: 2011 - 2021

Partners: Inria (Lille, Sophia-Antipolis, Grenoble), INSA, UPMC, Institut Télécom Paris, Institut Télécom Evry, LSIIT Strasbourg.

- Deployment: During the year 2014, the practical deployment has been finished, at the location planned for this testbed of the EQUIPEX FIT: the basement of building 1 at Rocquencourt. Ten external nodes have also been integrated.

  The testbed is offering most of the planned 344 open nodes, including 120 WSN430 nodes, 200 Cortex A8 based nodes, 24 Cortex M3.
- Opening: The official opening of the Rocquencourt was done in November 2014, at that point all IoT-LAB (and OneLab) users could run experiments in the M3 and A8 nodes from the site. See https://www.iot-lab.info/deployment/rocquencourt/ for more information.
- Support of external projects: Support for RIOT-OS and OpenWSN projects has been developed for IoT-Lab hardware and is being tested.
  - RIOT-OS, a joint work between Inria and FU-Berlin to create an Operating System for the Internet of Things.
  - OpenWSN, an open-source protocol stack for Internet of Things developed by UC Berkley.
- Demonstration: the project IoT-LAB in general had been demonstrated in several events during the year
  - March 2014: Description of IoT-LAB, and test of IoT-LAB nodes at the IETF 6TiSCH plugfest during IETF-89 in London https://bitbucket.org/6tisch/meetings/wiki/140306a_ietf89_london_plugfest
  - July 2014: Demonstration of IoT-LAB (Contiki RPL experiments) at:
    * the Bits-n-Bytes event of IETF-90 in Toronto http://www.ietf.org/meeting/90/ietf-90-bits-n-bites.html
    * the LLN plugfest of IETF-90 in Toronto https://bitbucket.org/6tisch/meetings/wiki/140720a_ietf90_toronto_plugfest
  - November 2014: Demonstration of IoT-LAB for use of scientific experiments, with network coding at MASS 2014 (http://mass2014.eecs.utk.edu/)

Moreover, during 2014, Ichrak Amdouni was in charge of:

- Testing the support of new switches in FIT IoT-LAB Paris-Rocquencourt site.
- Experimenting network coding protocols in the FIT IoT-LAB platform.

INDES Project-Team

5. New Software and Platforms

5.1. Introduction

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

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

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

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

5.2. Language-based Security

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

5.2.1. JavaScript Library iflowtypes.js

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

5.2.2. JavaScript Library iflowsigs.js

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

5.3. Web programming

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

5.3.1. The HOP web programming environment

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

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

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

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

5.4. Old software

5.4.1. Camloo

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

5.4.2. Skribe

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

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

5.4.3. Scheme2JS

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

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

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

5.4.4. The FunLoft language

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

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

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

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

5.4.5. The Bigloo compiler

The programming environment for the Bigloo compiler [7] is available on the Inria Web site at the following URL: http://www-sop.inria.fr/teams/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.

5.4.6. CFlow

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

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

5.4.7. FHE type-checker

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

5.4.8. Mashic compiler

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

5.4.9. IFJS compiler

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

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

INFINE Team

5. New Software and Platforms

5.1. Software

5.1.1. MACACOapp

Participant: Aline Carneiro Viana.

MACACOapp (https://macaco.inria.fr/macacoapp/) is developped in the context of the EU CHIST-ERA MACACO project (https://macaco.inria.fr/). It consists in a mobile phone application that periodically samples phone's information on the mobility (through, e.g., GPS sensor, accelerometer and WiFi/Bluetooth/Cellular environment, connectivity type) and on the data traffic it generates (through, e.g., Internet browser history and applications data consumption). The information collected will be time-stamped and will be periodically sent to the central servers for analysis and visualization. We expect that (1) the collected information will allow us studying the correlation between mobility and content demand patterns and that (2) the results of this analysis will allow us inferring the best times and places to transfer content from/to users' phones location and/or from/to the wireless infrastructure closest to the users' phones location. Users will be also invited to fill a non-mandatory questionnaire relevant to this study. Our questionnaire collects information about the personality traits and application preferences of people. We expect that the information collected from questionnaire will allow us to analyse the correlation between users personality traits and their application preferences and interests. User's application preferences and interests will be inferred from the Internet browsing history and running app information obtained from the MACACO App.

The data collection and on-the-phone storage of MACACOapp is designed in accordance with the state-of-the-art best practices in application development. The data collected on the phone is encrypted and inaccessible by any other application installed on the same phone or to any other third party, even in case your phone gets lost or stolen. Moreover, any user's identity information available in the collected data or in the questionnaire will be completely and irreversibly anonymised before its transfer to the central servers. The on-the-phone collected data and questionnaire data will be transferred via a secure transmission protocol to the central servers. This application is in phase of test.

5.1.2. RIOT

Participants: Emmanuel Baccelli, Oliver Hahm.

RIOT (http://www.riot-os.org) is a nano operating system for the Internet of Things. While requiring as low as 1,5kB of RAM and 5kB or ROM, RIOT offers real time and energy efficiency capabilities, as well as a single API (partially POSIX compliant) across heterogeneous 8-bit, 16-bit and 32-bit low-hardware. This API is developer-friendly in that it enables multi-threading, standard C and C++ application programming and the use of standard debugging tools (which was not possible so far for embedded programming). On top of this, RIOT includes several network stacks, such as a standard IPv6/6LoWPAN stack and a information-centric network stack (based on CCN).

RIOT is developed by an international community of open-source developers that was co-founded by Inria and Freie Universitaet Berlin. The goal of RIOT is to provide a powerful, free, open-source IoT software platform that can be used like Linux is for less constrained machines, both (i) in the context of research and/or teaching, as well as (ii) in industrial contexts.

5.1.3. DragonNet

Participants: Cédric Adjih, Ichrak Amdouni, Hana Baccouch, Antonia Masucci.

DragonNet is a generic framework for network coding in wireless networks. It is a initially result of the GETRF project of the Hipercom2 team.

It is based on intra-flow coding where the source divides the flow in a sequence of payloads of equal size (padding may be used). The design keys of DragonNet are simplicity and universality; DragonNet does not use explicit or implicit knowledge about the topology (such as the direction or distance to the source, the loss rate of the links, ...). Hence, it is perfectly suited to the most dynamic wireless networks. The protocol is distributed and requires minimal coordination. DragonNet architecture is modular, it is based on 5 building blocks (LIB, SIG, Protocol, SEW and DRAGON). Each block is almost independent. This makes DragonNet generic and hence adaptable to many application scenarios. DragonNet derives from a prior protocol called DRAGONCAST. Indeed, DragonNet shares the same principles and theoretical overview of DRAGONCAST. It enriches DRAGONCAST by the information base and signaling required to perform broadcast in wireless networks and in wireless sensor networks in particular.

5.2. Platforms

5.2.1. FIT IoT-LAB

Participants: Cedric Adjih, Emmanuel Baccelli, Ichrak Amdouni.

FIT IoT-LAB is a platform built to help foster the development, tuning and experimentation of protocols and applications for the Internet of Things and wireless sensor networks. IoT-LAB provides both dedicated IoT hardware deployments, a front-end webportal and backend management software. Using these elements, IoT-LAB enables users to share access to this IoT hardware, set-up and manage experiments. Remote use, and large scale experiments on concrete IoT deployments are thus made possible.

The Infine team is now managing the IoT-LAB site currently located in Rocquencourt, and which was publically opened in November 2014. It consists of the following:

- A set of GPS repeaters are relaying the GPS signal indoor (used for time synchronization)
- 200 A8 nodes, all equipped with GPS (10 deployed outside – identifiers between 166 and 175)
- 24 M3 nodes
- 120 WSN430 nodes

This platform was developped as part of the Equipex FIT (see section 8.1.1).

KerData Project-Team

5. New Software and Platforms

5.1. Major Software

5.1.1. BlobSeer

Participants: Loïc Cloatre, Alexandru Costan, Gabriel Antoniu, Luc Bougé.

Contact: Gabriel Antoniu.

Presentation: BlobSeer is the core software platform for most current projects of the KerData team. It is a data storage service specifically designed to deal with the requirements of large-scale, data-intensive distributed applications that abstract data as huge sequences of bytes, called BLOBs (Binary Large OBjects). It provides a versatile versioning interface for manipulating BLOBs that enables reading, writing and appending to them.

BlobSeer offers both scalability and performance with respect to a series of issues typically associated with the data-intensive context: *scalable aggregation of storage space* from the participating nodes with minimal overhead, ability to store *huge data objects*, *efficient fine-grain access* to data subsets, *high throughput in spite of heavy access concurrency*, as well as *fault-tolerance*. This year we have mainly focused on the deployment in production of the BlobSeer software on IBM's cluster at Montpellier, in the context of the ANR MapReduce project. To this end, several bugs were solved, and several optimizations were brought to the communication layer of BlobSeer. To showcase the benefits of BlobSeer on this platform we focused on the Terasort benchmark. Currently, preliminary tests on Grid5000 with this benchmark show that BlobSeer performs better than HDFS for block sizes lower than 2 MB. We have also improved the continuous integration process of BlobSeer by deploying daily builds and automatic tests on Grid5000.

Users: Work is currently in progress in several formalized projects (see previous section) to integrate and leverage BlobSeer as a data storage back-end in the reference cloud environments: a) Microsoft Azure; b) the Nimbus cloud toolkit developed at Argonne National Lab (USA); and c) the OpenNebula IaaS cloud toolkit developed at UCM (Madrid).

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

License: GNU Lesser General Public License (LGPL) version 3.

Status: This software is available on Inria's forge. Version 1.0 (released late 2010) registered with APP: IDDN.FR.001.310009.000.S.P.000.10700.

A *Technology Research Action* (ADT, *Action de recherche technologique*) started in November 2012 for two years, aiming at robustifying the BlobSeer software and making it a safely distributable product. This project is funded by Inria *Technological Development Office* (D2T, *Direction du Développement Technologique*). Loïc Cloatre has been hired as a senior engineer for the second year of this project, as a successor of Zhe Li, starting in February 2014.

5.1.2. Damaris

Participants: Matthieu Dorier, Orçun Yildiz, Lokman Rahmani, Shadi Ibrahim, Gabriel Antoniu.

Contact: Gabriel Antoniu.

Presentation: Damaris is a middleware for multicore SMP nodes enabling them to handle data transfers for storage and visualization efficiently. The key idea is to dedicate one or a few cores of each SMP node to the application I/O. It is developed within the framework of a collaboration between KerData and the *Joint Laboratory for Petascale Computing* (JLPC). Damaris enables efficient asynchronous I/O, hiding all I/O related overheads such as data compression and post-processing, as well as direct (*in-situ*) interactive visualization of the generated data. Version 1.0 was released in November 2014 and enables other approaches such as the use of dedicated nodes instead of dedicated cores.

Users: Damaris has been preliminarily evaluated at NCSA/UIUC (Urbana-Champaign, IL, USA) with the CM1 tornado simulation code. CM1 is one of the target applications of the Blue Waters supercomputer in production at, in the framework of the Inria-UIUC-ANL Joint Lab (JLPC). Damaris now has external users, including (to our knowledge) visualization specialists from NCSA and researchers from the France/Brazil Associated research team on Parallel Computing (joint team between Inria/LIG Grenoble and the UFRGS in Brazil). Damaris has been successfully integrated into four large-scale simulations (CM1, OLAM, Nek5000, GTC).

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

License: GNU Lesser General Public License (LGPL) version 3.

Status: This software is available on Inria's forge and registered with APP. Registration of the latest version with APP is in progress.

5.2. New Software

5.2.1. Omnisc'IO

Participants: Matthieu Dorier, Shadi Ibrahim, Gabriel Antoniu.

Contact: Matthieu Dorier

Presentation: Omnisc'IO is a middleware integrated in the POSIX and MPI-I/O stacks to observe, model and predict the I/O behavior of any HPC application transparently. It is based on formal grammars, implementing a modified version of the Sequitur algorithm. Omnisc'IO has been used on Grid'5000 with the CM1 atmospheric simulation, the LAMMPS molecular dynamics simulation, the GTC fusion simulation and the Nek5000 CFD simulation. Omnisc'IO was subject to a publication at SC14.

Users: Omnisc'IO is currently used only within the KerData team.

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

License: GNU Lesser General Public License (LGPL) version 3.

Status: This software is available on Inria's forge. Registration with APP is in progress.

5.2.2. Darshan-Web

Participants: Matthieu Dorier, Thomas Bouguet.

Contact: Matthieu Dorier

Presentation: Darshan-Web is a web interface for Darshan-Ruby, based on Ruby on Rails and AJAX technologies. It allows to navigate through many Darshan log files and display graphs on demand, directly on a web brother. A demo of Darshan-Web is available at http://darshan-web.irisa.fr/, which includes 2 months of logs from ANL's Intrepid supercomputer. The code of this demo is available and can be installed and used by the community.

Users: The KerData team is currently seeking potential users, in particular from Argonne National Laboratory, and will push the development further according to potential users' feedback.

URL: http://darshan-ruby.gforge.inria.fr/

License: GNU Lesser General Public License (LGPL) version 3.

Status: Prototype and demo available on demand.

5.2.3. JetStream

Participants: Radu Tudoran, Alexandru Costan, Gabriel Antoniu.

Contact: Alexandru Costan

Presentation: JetStream is a middleware solution for batch-based, high-performance streaming across cloud data centers. JetStream implements a set of context-aware strategies for optimizing batch-based streaming, being able to self-adapt to changing conditions. Additionally, the system provides multi-route streaming across cloud data centers for aggregating bandwidth by leveraging the network parallelism. It enables easy deployment across .Net frameworks and seamless binding with event processing engines such as StreamInsight.

Users: JetStream is currently used at Microsoft Research ATLE Munich for the management of the Azure cloud infrastructure.

License: Microsoft Public License.

Status: Prototype and demo available.

5.2.4. OverFlow

Participants: Radu Tudoran, Alexandru Costan, Gabriel Antoniu.

Contact: Alexandru Costan.

Presentation: OverFlow is a uniform data management system for scientific workflows running across geographically distributed sites, aiming to reap economic benefits from this geo-diversity. The software is environment-aware, as it monitors and models the global cloud infrastructure, offering high and predictable data handling performance for transfer cost and time, within and across sites. OverFlow proposes a set of pluggable services, grouped in a data-scientist cloud kit. They provide the applications with the possibility to monitor the underlying infrastructure, to exploit smart data compression, deduplication and geo-replication, to evaluate data management costs, to set a tradeoff between money and time, and optimize the transfer strategy accordingly.

Users: Currently, OverFlow is used for data transfers by the Microsoft Research ATLE Munich team as well as for synthetic benchmarks at the Politehnica University of Bucharest.

License: GNU Lesser General Public License (LGPL) version 3.

Status: Registration of the latest version with APP is in progress

5.2.5. iHadoop

Participants: Tien Dat Phan, Shadi Ibrahim.

Contact: Shadi Ibrahim

Presentation: *iHadoop* is a Hadoop simulator developed in Java on top of SimGrid to simulate the behavior of Hadoop and therefore accurately predict the performance of Hadoop in normal scenarios and under failures.

Users: iHadoop is an internal software prototype, which was initially developed to validate our idea for exploring the behavior of Hadoop under failures. iHadoop has preliminarily evaluated within our group and it has shown very high accuracy when predicating the execution time of a Map-Reduce application. We intend to integrate iHadoop within the SimGrid distribution and make it available to the SimGrid community.

License: GNU Lesser General Public License (LGPL) version 3.

Status: Available on Inria’s forge. Registration with APP is in progress.

MADYNES Project-Team

5. New Software and Platforms

5.1. Escape

Participants: Thibault Cholez [contact], Shbair Wazen.

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

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

5.2. MPIGate

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

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

5.3. AA4MM

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

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

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

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

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

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

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

5.4. Platforms

5.4.1. Android Security platform

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

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

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

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

5.4.2. IoT platform

Participants: Emmanuel Nataf [contact], Thibault Cholez.

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

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

Based on this platform, we developed several innovative applications:

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

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

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

5.4.3. SCADA platform

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

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

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

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

MAESTRO Project-Team

5. New Software and Platforms

5.1. New Software

5.1.1. ns-3

Participants: Sara Alouf, Abdulhalim Dandoush, Giovanni Neglia.

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

5.2. Platforms

5.2.1. Marmote

Participants: Alain Jean-Marie, Issam Rabhi.

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

MESCAL Project-Team

5. New Software and Platforms

5.1. Tools to visualize and analyze traces of execution of distributed applications

Participants: Jean-Marc Vincent [correspondent], Arnaud Legrand.

The Pajé (http://paje.sourceforge.net/) generic tool provides interactive and scalable behavioral visualizations of parallel and distributed applications, helping to capture the dynamics of their executions; because of its genericity, it can be used unchanged in a large variety of contexts. Pajé Next Generation

Pajé Next Generation (https://github.com/schnorr/pajeng) is a re-implementation (in C++) and direct heir of the well-known Paje visualization tool for the analysis of execution traces (in the Paje File Format) through trace visualization (space/time view). The tool is released under the GNU General Public License 3. PajeNG comprises the libpaje library, the space-time visualization tool in pajeng and a set of auxiliary tools to manage Paje trace files (such as pj_dump and pj_validate). It was started as part of the french INFRA-SONGS ANR project. Development has continued at INF/UFRGS. Viva

Viva (https://github.com/schnorr/viva) is an open-source tool used to analyze traces (in the Paje File Format) registered during the execution of parallel or distributed applications. The tool also serves as a sandbox to the development of new visualization techniques. Current features include: Temporal integration using dynamic time-intervals Spatial aggregation through hierarchical traces Interactive Graph Visualization with a force-directed algorithm, with viva Squarified Treemap to compare processes behavior on scale, with vv_treemap

Framesoc (http://soctrace-inria.github.io/framesoc/) is the core software infrastructure of the SoC-Trace project. It provides a graphical user environment for execution-trace analysis, featuring interactive analysis views as Gantt charts or statistics views. It provides also a software library to store generic trace data, play with them, and build other analysis tools (e.g., Ocelotl). This software is developed in partnership with Nanosim. Ocelotl

Ocelotl (http://soctrace-inria.github.io/ocelotl/): Multidimensional Overviews for Huge Trace Analysis is an innovative visualization tool, which provides overviews for execution trace analysis by using a data aggregation technique. This technique enables to find anomalies in huge traces containing up to several billions of events, while keeping a fast computation time and providing a simple representation that does not overload the user.

5.2. Simulation and performance evaluation tools

Participants: Arnaud Legrand [correspondent], Luka Stanisic, Augustin Degomme, Jean-Marc Vincent, Florence Perronnin.

5.2.1. SimGrid

(see http://simgrid.gforge.inria.fr/) is a toolkit that provides core functionalities for the simulation of distributed applications in heterogeneous distributed environments. The specific goal of the project is to facilitate research in the area of distributed and parallel application scheduling on distributed computing platforms ranging from simple network of workstations to Computational Grids.

5.2.2. Perfect simulator

– Ψ^2 (https://gforge.inria.fr/projects/psi/) is a simulation software of markovian models. It be able to simulate discrete and continuous time models to provide a perfect sampling of the stationary distribution or directly a sampling of functional of this distribution by using *coupling from the past* The simulation kernel is based on the CFTP algorithm, and the internal simulation of transitions on the Aliasing method.

5.2.3. PEPS

– The main objective of PEPS (http://www-id.imag.fr/Logiciels/peps/) is to facilitate the solution of large discrete event systems, in situations where classical methods fail. PEPS may be applied to the modelling of computer systems, telecommunication systems, road traffic, or manufacturing systems. Development has continued at INF/UFRGS.

5.2.4. GameSeer

(http://mescal.imag.fr/membres/panayotis.mertikopoulos/publications.html) is a tool for students and researchers in game theory that uses Mathematica to generate phase portraits for normal form games under a variety of (user-customizable) evolutionary dynamics. The whole point behind GameSeer is to provide a dynamic graphical interface that allows the user to employ Mathematica's vast numerical capabilities from a simple and intuitive front-end. So, even if you've never used Mathematica before, you should be able to generate fully editable and customizable portraits quickly and painlessly.

5.3. Tools for cluster management and software development

Participant: Olivier Richard [correspondent].

The KA-Tools (http://ka-tools.imag.fr/) is a software suite developed by MESCAL for exploitation of clusters and grids. It uses a parallelization technique based on spanning trees with a recursive starting of programs on nodes. Industrial collaborations were carried out with Mandrake, BULL, HP and Microsoft.

5.3.1. KA-Deploy

(http://kadeploy3.gforge.inria.fr/) is a fast and scalable deployment system for clusters and grids. It provides a set of tools for cloning, configuring (post installation) and managing a set of nodes. Currently it can successfully deploy Linux, *BSD, Windows and Solaris on x86 and 64 bits computers. Kameleon

5.3.2. Kameleon

(http://kameleon.imag.fr/) is a simple but powerful tool to generate customized appliances. With Kameleon, you make your recipe that describes how to create step by step your own distribution. At start Kameleon is used to create custom kvm, docker, VirtualBox, ..., but as it is designed to be very generic you can probably do a lot more than that.

5.4. Infrastructure Middleware and scheduler

Participant: Olivier Richard [correspondent].

5.4.1. OAR

– The OAR project (see http://oar.imag.fr) focuses on robust and highly scalable batch scheduling for clusters and grids. Its main objectives are the validation of grid administration tools such as TAKTUK, the development of new paradigms for grid scheduling and the experimentation of various scheduling algorithms and policies.

The grid development of OAR has already started with the integration of best effort jobs whose purpose is to take advantage of idle times of the resources. Managing such jobs requires a support of the whole system from the highest level (the scheduler has to know which tasks can be canceled) down to the lowest level (the execution layer has to be able to cancel awkward jobs). OAR is perfectly suited to such developments thanks to its highly modular architecture. Moreover, this development is used for the CiGri grid middleware project.

The OAR system can also be viewed as a platform for the experimentation of new scheduling algorithms. Current developments focus on the integration of theoretical batch scheduling results into the system so that they can be validated experimentally.

5.4.2. CiGri

(http://cigri.imag.fr/) is a middleware which gathers the unused computing resource from intranet infrastructure and makes it available for the processing of large set of tasks. It manages the execution of large sets of parametric tasks on lightweight grid by submitting individual jobs to each batch scheduler. It is s associated to the OAR resource management system (batch scheduler). Users can easily monitor and control their set of jobs through a web portal. CiGri provides mechanisms to identify job error causes, to isolate faulty components and to resubmit jobs in a safer context.

5.4.3. ComputeMode

(http://computemode.imag.fr/) is an software infrastructure that allows to extends or create a Grid through the aggregation of unused computing resources. For instance, a virtual cluster can be built using anyone's PC while not in use. Indeed, most PCs in large companies or university campus are idle at night, on weekends, and during vacations, training periods or business trips.

5.5. Platforms

5.5.1. Grid'5000

The MESCAL project-team is involved in development and management of Grid'5000 platform. The Digitalis and IDPot clusters are integrated in Grid'5000 as well as of CIMENT.

5.5.2. The ICluster-2, the IDPot and the new Digitalis Platforms

The MESCAL project-team manages a cluster computing center on the Grenoble campus. The center manages different architectures: a 48 bi-processors PC (ID-POT), and the center is involved with a cluster based on 110 bi-processors Itanium2 (ICluster-2) and another based on 34 bi-processor quad-core XEON (Digitalis) located at Inria. The three of them are integrated in the Grid'5000 grid platform.

More than 60 research projects in France have used the architectures, especially the 204 processors Icluster-2. Half of them have run typical numerical applications on this machine, the remainder has worked on middleware and new technology for cluster and grid computing. The Digitalis cluster is also meant to replace the Grimage platform in which the MOAIS project-team is very involved.

5.5.3. The Bull Machine

In the context of our collaboration with Bull the MESCAL project-team exploits a Novascale NUMA machine. The configuration is based on 8 Itanium II processors at 1.5 Ghz and 16 GB of RAM. This platform is mainly used by the Bull PhD students. This machine is also connected to the CIMENT Grid.

MIMOVE Team

5. New Software and Platforms

5.1. Introduction

In order to validate our research results, our research activities encompass the development of related prototypes as surveyed below.

5.2. iCONNECT: Emergent Middleware Enablers

Participant: Valérie Issarny [correspondent].

As part of our research work on Emergent Middleware, we have implemented Enablers (or Enabler functionalities) that make part of the overall CONNECT architecture realizing Emergent Middleware in practice [4]. The focus of our work is on the: *Discovery enabler* that builds on our extensive background in the area of interoperable pervasive service discovery; and *Synthesis enabler* that synthesizes mediators that allow Networked Systems (NSs) that have compatible functionalities to interact despite mismatching interfaces and/or behaviors.

The Discovery Enabler is the component of the overall CONNECT architecture that handles discovery of networked systems, stores their descriptions (NS models), and performs an initial phase of matchmaking to determine which pairs of systems are likely to be able to interoperate. Such pairs are then passed to the Synthesis Enabler so that mediators can be generated. The Discovery Enabler is written in Java and implements several legacy discovery protocols including DPWS and UPnP.

The Synthesis Enabler assumes semantically-annotated system descriptions *à la* OWL-S, which are made available by the Discovery Enabler, together with a domain ontology, and produces the mediators that enable functionally compatible networked systems to interoperate. The semantically-annotated interfaces of the NSs that need to communicate are processed to compute the semantic mapping between their respective operations using a constraint solver. The resulting mapping serves generating a mediator that coordinates the behaviors of the NSs and guarantees their successful interaction. Only when the mediator includes all the details about the communication of NSs, can interoperability be achieved, which calls for the adequate concretization of synthesized mediators.

The *concretization of mediators* bridges the gap between the application level, which provides the abstraction necessary to reason about interoperability and synthesize mediators, and the middleware-level, which provides the techniques necessary to implement these mediators. Concretization entails the instantiation of the data structures expected by each NS and their delivery according to the interaction pattern defined by the middleware, based on which the NS is implemented. Therefore, we have been developing a mediation engine that, besides executing the data translations specified by the mediator, generates composed parsers and composers, which can process complex messages, by relying on existing libraries associated with standard protocols and state-of-the-art middleware solutions.

The data structures defined within OWL-S system descriptions, that is, the types of the inputs and outputs of the OWL-S atomic process were previously defined manually. As part of the prototype implementation that allows the mediator engine to generate composed parsers and composers, we made an extension that enables the inference of correct data-types [17], relieving developers from the time-consuming task of defining them. This prototype takes the form of a JAVA library that can optionally be used by the Synthesis Enabler whenever abstract data structures are unavailable. This library can be obtained from the iCONNECT GIT repository https://gforge.inria.fr/git/iconnect/iconnect.git (under the subproject `mtc`). While the underlying source code closely follows the formal mechanisms (such as tree automata) and algorithms presented in [17], we further concerned ourselves with making this library usable for non-expert developers by adhering to well-established standards. Specifically, data types, which are internally modeled as top-down tree automata, are transformed both on the input and output to RelaxNG (http://relaxng.org/) or XSD (www.w3.org/TR/xmlschema-1/).

The iCONNECT software has been released in the OW2 open source community (http://forge.ow2.org/projects/iconnect/), as part of FISSi, the Future Internet Software and Services initiative (http://www.ow2.org/view/Future_Internet/). OW2 and FISSi will give to this effort the required visibility in order to attract users and developers of the open source community.

5.3. XSB: eXtensible Service Bus for the Future Internet

Participant: Nikolaos Georgantas [correspondent].

The eXtensible Service Bus (XSB) is a development and runtime environment dedicated to complex distributed applications of the Future Internet. Such applications will be based, to a large extent, on the open integration of extremely heterogeneous systems, such as lightweight embedded systems (e.g., sensors, actuators and networks of them), mobile systems (e.g., smartphone applications), and resource-rich IT systems (e.g., systems hosted on enterprise servers and Cloud infrastructures). Such heterogeneous systems are supported by enabling middleware platforms, particularly for their interaction. With regard to middleware-supported interaction, the client-service (CS), publish-subscribe (PS), and tuple space (TS) paradigms are among the most widely employed ones, with numerous related middleware platforms, such as: Web Services, Java RMI for CS; JMS, SIENA for PS; and JavaSpaces, Lime for TS. XSB then provides support for the seamless integration of heterogeneous interaction paradigms (CS, PS and TS).

In a nutshell, our systematic interoperability approach implemented by the proposed XSB is carried out in two stages. First, a middleware platform is abstracted under a corresponding interaction paradigm among the three base ones, i.e., CS, PS and TS. To this aim, we have elicited a connector model for each paradigm, which comprehensively covers its essential semantics. Then, these three models are abstracted further into a single generic application (GA) connector model, which encompasses their common interaction semantics. Based on GA, we build abstract connector converters that enable interconnecting the base interaction paradigms.

Following the above, XSB is an abstract service bus that prescribes only the high-level semantics of the common bus protocol, which is the GA semantics. Furthermore, we provide an implementation of the XSB, building upon existing SOA and ESB realizations. XSB features richer interaction semantics than common ESBs to deal effectively with the increased Future Internet heterogeneity. Moreover, from its very conception, XSB incorporates special consideration for the cross-integration of heterogeneous interaction paradigms. Services relying on different interaction paradigms can be plugged into XSB by employing binding components (BCs) that adapt between their native middleware and the common bus protocol. This adaptation is based on the abstractions, and in particular on the conversion between the native middleware, the corresponding CS/PS/TS abstraction, and the GA abstraction.

Furthermore, we provide a companion implementation, named Light Service Bus (LSB), targeting the Internet of Things (IoT) domain. LSB forms a concrete access solution for IoT systems as it is able to cope with the diversity of the involved interaction protocols and take care of the specifics of IoT services, such as resource constraints, dynamic environments, data orientation, etc. It is implemented to be lightweight in nature and uses REST as the common protocol/bus in place of an ESB solution. In LSB, we confirm the wide use of the aforementioned interaction paradigms (CS/PS/TS) but also underline the existence of an additional paradigm focused on continuous interaction known as Streaming (STR).

Both the XSB and LSB solutions are available for download under open source licenses at http://xsb.inria.fr and http://websvn.ow2.org/listing.php?repname=choreos&path=%2Ftrunk%2Fextensible-service-access%2Flsb%2FLsbBindingComponents%2F respectively.

5.4. MobIoT: Service-oriented Middleware for the Mobile IoT

Participant: Valérie Issarny [correspondent].

MobIoT is a service-oriented middleware aimed at the mobile Internet of Things (IoT), which in particular deals with the ultra-large scale, heterogeneity and dynamics of the target networking environment. MobIoT offers novel probabilistic service discovery and composition approaches, and wraps legacy access protocols to be seamlessly executed by the middleware. The middleware exposes two levels of service abstractions: Thing as a service (on the service provider side); and Things measurements/actions as a service (on the service consumer side).

Key features of MobIoT lie in: (i) the exploitation of ontologies to overcome the heterogeneity of the Things network, (ii) the introduction of probabilistic approaches for both registering and retrieving networked things so as to filter out the ones that are redundant with already known alternatives, and finally, (iii) the exploitation of Thing services composition for responding to user queries asking information about the physical world so as to ease interaction with such a complex and dynamic networking environment.

MobIoT is implemented using Java and the Android platform, and consists of two complementary components: The MobIoT Mobile middleware and the MobIoT Web Service. The MobIoT Mobile middleware is deployed on mobile devices (e.g., smartphones, tablets, sensor devices). It wraps: (i) the Query component that enables the querying of the physical world, (ii) the Registration component that deals with the probabilistic registration of local sensors and actuators, (iii) the domain ontology that allows reasoning about the features of Things, and (iv) the Sensor Access component that enables the sensor data retrieval and exposure. The MobIoT Web Service wraps: (i) the Registry component that keeps tracks of the registered services, (ii) the probabilistic Lookup component that allows retrieving relevant services in a scalable way, and (iii) the Composition & Estimation component to answer queries over the physical world using available Thing services, and finally domain and devices ontologies.

The MobIoT middleware is available for download under an open source license at http://mobiot.ow2.org.

5.5. Srijan: Data-driven Macroprogramming for Sensor Networks

Participant: Animesh Pathak [correspondent].

Macroprogramming is an application development technique for wireless sensor networks (WSNs) where the developer specifies the behavior of the system, as opposed to that of the constituent nodes. As part of our work in this domain, we are working on *Srijan*, a toolkit that enables application development for WSNs in a graphical manner using data-driven macroprogramming.

It can be used in various stages of application development, *viz.,*

1. Specification of the application as a task graph,
2. Customization of the auto-generated source files with domain-specific imperative code,
3. Specification of the target system structure,
4. Compilation of the macroprogram into individual customized runtimes for each constituent node of the target system, and finally
5. Deployment of the auto generated node-level code in an over-the-air manner to the nodes in the target system.

The current implementation of *Srijan* targets both the Sun SPOT sensor nodes and larger nodes with J2SE. Most recently, *Srijan* also includes rudimentary support for incorporating Web services in the application being designed.

The software is released under open source license, and available as an Eclipse plug-in at http://code.google.com/p/srijan-toolkit/.

5.6. Dioptase and Spinel: Lightweight Streaming Middleware for the IoT

Participant: Valérie Issarny [correspondent].

Dioptase is a service-oriented middleware for developing stream-based applications which produce, process, store and consume data streams in complex environments such as the Internet of Things (IoT) and Wireless Sensor and Actuator Networks (WSAN). Dioptase leverages a novel service-oriented architecture for continuous processing, in order to deal with the large scale and the heterogeneity of the IoT. Once Dioptase is deployed onto a device, it enables developers to manage it as a generic pool of resources that can execute tasks provided over time. Those tasks are described as compositions of both standard continuous processing operators and customized computations written using a new lightweight stream processing language, called *DiSPL*.

The IoT infrastructure is composed of various devices (sensors, registries, proxies, clusters, etc.), and Dioptase is intended to be deployed onto all of them. To this end, Dioptase is highly modular and can be customized depending on the targeted devices and their roles in the IoT:

- *Dioptase core* is the base version of Dioptase that enables developers to (i) manage embedded sensors and actuators of a device through services and (ii) deploy tasks onto the device at any time.
- *Dioptase task mapper* is a server that implements our research on task mapping and automated deployment. Given a task graph, this server computes where to deploy each task according to the characteristics of tasks and available devices, and then manages the execution of the deployed tasks over time.
- *Dioptase proxy* is a pub/sub broker that enables interactions between Things that can not communicate directly, because of non-compatible networking interfaces/protocols or the use of address translation techniques.
- *Dioptase exchange* is a privacy proxy that manages the data and the services provided by a network of Things. It authenticates outsider Things and users, enabling them to request data streams or services according to access control and data-accuracy policies.

As part of the design of Dioptase, we have been investigating how to open legacy sensors to the future IoT. Toward this end, we propose to take advantage of the multi-modal connectivity as well as the mobility of smartphones, using them as mobile proxies that opportunistically discover close-by static sensors and act as intermediaries between them and the IoT. Spinel is a prototype of such an opportunistic proxy for mobile phones, which monitors the smartphone's mobility and further infers when to discover and register the sensors to an IoT discovery infrastructure, for instance the MobIoT registry(§ 5.4). Spinel collects data from the close-by sensors and pushes the collected data to an IoT stream processing infrastructure, for instance the Dioptase middleware. We will shortly release both Dioptase and Spinel under open source licenses.

5.7. Yarta: Middleware for supporting Mobile Social Applications

Participant: Animesh Pathak [correspondent].

With the increased prevalence of advanced mobile devices (the so-called "smart" phones), interest has grown in *Mobile Social Ecosystems* (MSEs), where users not only access traditional Web-based social networks using their mobile devices, but are also able to use the context information provided by these devices to further enrich their interactions. We are developing a middleware framework for managing mobile social ecosystems, having a multi-layer middleware architecture consisting of modules, which will provide the needed functionalities, including:

- Extraction of social ties from context (both physical and virtual),
- Enforcement of access control to protect social data from arbitrary access,
- A rich set of MSE management functionalities, which can be used to develop mobile social applications.

Our middleware adopts a graph-based model for representing social data, where nodes and arcs describe socially relevant entities and their connections. In particular, we exploit the Resource Description Framework (RDF), a basic Semantic Web standard language that allows representing and reasoning about social vocabulary, and creating an interconnected graph of socially relevant information from different sources.

The current implementation of the Yarta middleware targets both desktop/laptop nodes running Java 2 SE, as well as Android smart phones.

The software is released under open source license at https://gforge.inria.fr/projects/yarta/.

MOAIS Project-Team

4. New Software and Platforms

4.1. XKaapi

Participants: Thierry Gautier [correspondant], François Broquedis, Vincent Danjean, Joao Ferreira Lima.

- ACM: D.1.3
- License: CeCILL
- OS/Middelware: Unix (Linux, MacOSX, ...)
- Programming language: C/C++, Fortran
- Characterization of Software : A-3 / SO-4 / SM-3 / EM-3 / SDL-4
- Own Contribution: DA-4 / CD-4 / MS-4 / TPM-4
- Additional information:

  XKaapi (http://kaapi.gforge.inria.fr, coordinator T. Gautier) is a library for high performance applications running on multi-cores/multi- processors with support for multi-GPUs. Publicly available at http://kaapi.gforge.inria.fr under CeCILL licence. XKaapi provides ABI compliant implementations of libGOMP (GCC runtime for OpenMP) and was one of the target runtime of the K'Star compiler (http://kstar.gforge.inria.fr). Direct competitors with : OMPSs (BSC), OpenMP, StarPU (Inria RUNTIME)

4.2. FlowVR

Participants: Bruno Raffin [correspondant MOAIS], Matthieu Dreher, Jérémy Jaussaud.

- ACM: D.1.3
- License: GPL and LGPL
- OS/Middelware: Unix (Linux, MacOSX, ...)
- Programming language: C/C++
- Characterization of Software : A-3 / SO-4 / SM-3 / EM-3 / SDL-4
- Own Contribution: DA-4 / CD-3 / MS-3 / TPM-4
- Additional information: FlowVR (http://flowvr.sf.net, coordinator B. Raffin) is an open source middelware to augment parallel simulations running on thousands of cores with in situ processing capabilities and live steering. FlowVR offers a very flexible environment while enabling high performance asynchronous in situ and in transit processing.

  FlowVR was initially used for large scale virtual reality applications like real-time multicamera 3D modeling or telepresence. We recently retargeted FlowVR at in situ processing with development efforts focused on optimizing FlowVR performance at large scale and easing its usage in supercomputer environments.

4.3. TakTuk - Adaptive large scale remote execution deployment

Participants: Guillaume Huard [correspondant], Pierre Neyron.

- Characterization of Software : A-2 / SO-3 / SM-5 / EM-3 / SDL-4
- Own Contribution: DA-4 / CD-4 / MS-4 / TPM-4
- Additional information:
  – web site: http://taktuk.gforge.inria.fr, Coordinator G. Huard
  – Objective of the software: TakTuk is a tool for deploying parallel remote executions of commands to a potentially large set of remote nodes. It spreads itself using an adaptive algorithm and sets up an interconnection network to transport commands and perform I/Os multiplexing/demultiplexing. The TakTuk mechanics dynamically adapt to environment (machine performance and current load, network contention) by using a reactive work-stealing algorithm that mixes local parallelization and work distribution.
  – Users community: TakTuk is a research open source project available in the Debian GNU/Linux distribution (package taktuk) used in lower levels of Grid5000 software architectures (nodes monitoring in OAR, environment diffusion in Kadeploy). The community is small : developers and administrators for large scale distributed platforms, but active.
  – Positioning: main competing tools are pdsh (but uses linear deployment) and gexec (not fault tolerant, requires installation), for more details : B. Claudel, G. Huard and O. Richard. TakTuk, Adaptive Deployment of Remote Executions. In Proceedings of the International Symposium on High Performance Distributed Computing (HPDC), 2009. TakTuk is the only tool to provide to deployed processes a communication layer (just like an MPIrun, but not tied to a specific environment) and synchronization capabilities.

4.4. Triva

Participant: Guillaume Huard [correspondant].

- Characterization of Software : A-2 / SO-4 / SM-5 / EM-3 / SDL-3
- Own Contribution: DA-4 / CD-3 / MS-3 / TPM-3
- Additional information:
  – web site: http://triva.gforge.inria.fr/, Coordinator, Lucas Schnorr
  – Objective of the software: Triva is an open-source tool used to analyze traces (in the pajé format) registered during the execution of parallel applications. The tool serves also as a sandbox to the development of new visualization techniques.
  – Users community: Research open source project, applications developers, especially parallel applications.
  – Positioning: Main competing tools are Vampir (classical 2D Gantt charts) and Tau (less advanced agregation techniques), more details in : A Hierarchical Aggregation Model to achieve Visualization Scalability in the analysis of Parallel Applications. Lucas Mello Schnorr, Guillaume Huard, Philippe Olivier Alexandre Navaux. Parallel Computing. Volume 38, Issue 3, March 2012.

4.5. OAR

Participants: Pierre Neyron [correspondant MOAIS], Grégory Mounié.

- Characterization of Software : A-5 / SO-3 / SM-4 / EM-4 / SDL-5
- Own Contribution: DA-3 / CD-2 / MS-1 / TPM-1
- Additional information: OAR (http://oar.imag.fr, Coordinator O. Richard, Inria MESCAL) is a batch scheduler. The MOAIS team develops the central automata and the scheduling module that includes successive evolutions and improvements of the policy.OAR is used to schedule jobs both on the CiGri (Grenoble region) and Grid50000 (France) grids. CiGri is a production grid that federates about 500 heterogeneous resources of various Grenoble laboratories to perform computations in physics. MOAIS has also developed the distributed authentication for access to Grid5000.

4.6. LinBox

Participants: Clément Pernet [correspondant], Thierry Gautier.

- Characterization of Software : A-3 / SO-4 / SM-2 / EM-3 / SDL-5
- Own Contribution: DA-4 / CD-3 / MS-3 / TPM-4
- Additional information:
  - web site: http://linalg.org
  - Objective of the software: LinBox is an open-source C++ template library for exact, high-performance linear algebra computations. It is considered as the reference library for numerous computations (such as linear system solving, rank, characteristic polynomial, Smith normal forms,...) over finite fields and integers with dense, sparse, and structured matrices.
  - The LinBox group is an international collaboration (USA: NCSU, UDel; Canada: U Waterloo, U Calgary; France: LIP, LIRMM, LJK and LIG). Articles related to the library have been published in the main Conferences of the area: ISSAC, ICMS. MOAIS contributes to its development and more specifically to its parallelization in the context of ANR HPAC project. It is currently experiencing a major change of design, to better integrate parallelism.
  - Users community: mostly researchers doing computational mathematics (number theory, cryptology, group theory, persistent homology. They use the library by either linking against it directly (the library is packaged in Debian, Fedora, etc) or withing the general purpose math software Sage (sagemath.org very broad diffusion) which includes LinBox as a kernel for exact linear algebra.

4.7. K'Star

Participants: Thierry Gautier [correspondant], François Broquedis, Pïerrick Brunet, Philippe Virouleau, Olivier Aumage [RUNTIME project-team, Inria Bordeaux - Sud-Ouest], Samuel Thibault [RUNTIME project-team, Inria Bordeaux - Sud-Ouest], Nathalie Furmento [RUNTIME project-team, Inria Bordeaux - Sud-Ouest], Samuel Pitoiset [RUNTIME project-team, Inria Bordeaux - Sud-Ouest].

- ACM: D.1.3
- License: CeCILL
- OS/Middelware: Unix (Linux, MacOSX, ...)
- Programming language: C/C++
- Characterization of Software : A-3 / SO-2 / SM-3 / EM-2 / SDL-4
- Own Contribution: DA-4 / CD-4 / MS-4 / TPM-4
- Additional information:

  The K'Star project (http://kstar.gforge.inria.fr) supports the development of Klang, a source-to-source compiler that turns C programs with OpenMP pragmas to C programs with calls to either the StarPU or the XKaapi runtime system. K'Star is a collaboration with the EPI RUNTIME/STORM.

4.8. Kastors

Participants: Thierry Gautier [correspondant], François Broquedis, Pïerrick Brunet, Philippe Virouleau, Olivier Aumage [RUNTIME project-team, Inria Bordeaux - Sud-Ouest], Samuel Thibault [RUNTIME project-team, Inria Bordeaux - Sud-Ouest], Nathalie Furmento [RUNTIME project-team, Inria Bordeaux - Sud-Ouest].

- ACM: D.1.3
- License: CeCILL
- OS/Middelware: Unix (Linux, MacOSX, ...)
- Programming language: C/C++
- Characterization of Software : A-3 / SO-2 / SM-3 / EM-2 / SDL-4
- Own Contribution: DA-4 / CD-4 / MS-4 / TPM-4
- Additional information:

The KaStORS benchmarks suite (http://kastors.gforge.inria.fr) has been designed to evaluate the implementation of the OpenMP dependent task paradigm, introduced as part of the OpenMP 4.0 specification. KaStORS is a collaboration with the EPI RUNTIME/STORM.

4.9. Platforms

4.9.1. Multi-camera Platforms Grimage and Kinovis

MOAIS has managed with the LJK-Inria Morpheo team the Grimage platfrom (http://grimage.inrialpes.fr) dedicated to off-line and on-line 3D modeling from multiple cameras and telepresence. In 2012, we received an Equipex funding, Kinovis (http://kinovis.inrialpes.fr), to renew this platform. Kinovis will be operational by early 2015 and will consist of 68 cameras, a compute cluster and a large acquisition space. FlowVR is the software backbone of both platforms for live processing. MOAIS is participating to the FP7 infrastructure project Visionair to enable European research teams to experiment on both platforms.

4.9.2. HPC Platforms Grid'5000 and Ciment

MOAIS is involved in the national platform Grid'5000, the regional mezzo center Ciment and obtained in 2014 with the Mescal and Erods team a grant (FAIRE from Grenoble-INP and LIG) to buy various large NUMA nodes and accelerators that will be integrated into the Grid'5000 infrastructure.

MUSE Team

5. New Software and Platforms

5.1. Fathom v2.0

Contributors: Anna-Kaisa Pietilainen, Stephane Archer

Available at: https://muse.inria.fr/fathom

Fathom [9] is a Firefox browser extension that explores the browser as a platform for network measurement and troubleshooting. It provides a wide range of networking primitives directly to in-page JavaScript including raw TCP/UDP sockets, higher-level protocol APIs such as DNS, HTTP, and UPnP, and ready-made functionality such as pings and traceroutes. Fathom v2.0 is a complete rewrite of the original Fathom using the new add-on SDK from Mozilla. In addition to javascript APIs, we have improved and added new built-in network measurement tools to Fathom such as 'Debug my Connection', 'Homenet Discovery' and 'Baseline Monitoring' that allow users to troubleshoot home network problems and share with us data for further research on home networks usage and diagnosis.

5.2. OpenWRT Packages for Network Measurements

Contributors: Anna-Kaisa Pietilainen, Sarthak Grover

Available at: https://github.com/apietila/browserlab

OpenWRT is a version of the Linux operating system to run on embedded devices, in particular, in home routers and access points. We have developed an OpenWRT package repository that provides fixes for and new ports of several existing network measurement tools (including iperf, iperf3, shaperprobe and pathload) for OpenWRT and an extended JSON RPC API for LuCl (OpenWRT control interface) to collaborate with the Fathom extension on home network diagnosis.

5.3. TagIt

Contributors: Sara El Aouad, Christophe Diot (Technicolor), Renata Teixeira

Available at: https://drive.google.com/file/d/0B-OcOkKOXok2b3lZYmt4QUw0cGM/view?usp=sharing
Video demo available at:
https://drive.google.com/file/d/0B-OcOkKOXok2VUxQR1NmRlg5VE0/view?usp=sharing

TagIt is an android app that makes it easy for users to enter movie reviews and to summarise the set of reviews of each movie. TagIt uses tags (or a short sequence of words) for entering user's opinions about a movie, so it is easy and quick for users to enter their feedback. TagIt also allows users to quickly see the opinion of other users about a movie with a tag cloud that summarises the set of reviews of a movie.

5.4. Where is the fault?

Contributors: Srikanth Sundaresan (Georgia Tech), Nick Feamster (Georgia Tech), Renata Teixeira

Where's The Fault? (WTF) is a system that localizes performance problems in home and access networks. We implement WTF as custom firmware that runs in an off-the-shelf home router. WTF uses timing and buffering information from passively monitored traffic at home routers to detect both access link and wireless network bottlenecks. We presented a demo of WTF at the ACM SIGCOMM conference in 2014. [4]

5.5. WeBrowse

Contributors: Giuseppe Scavo, Zied Ben Houidi (Alcatel-Lucent Bell Labs), Stefano Traverso (Politecnico di Torino), Marco Mellia (Politecnico di Torino), Renata Teixeira

Available at: http://tstat.polito.it/netcurator/

WeBrowse is the first passive crowdsource-based content curation system. Content curation is the act of assisting users to identify relevant and interesting content in the Internet. WeBrowse requires no active user engagement to promote content. Instead, it extracts the URLs users visit from traffic traversing an ISP network to identify popular content. WeBrowse contains a set of heuristics to identify the set of URLs users visit and to select the subset that are interesting to users [7]. The system proposes the interesting content in a web page available to all users.

5.6. UCN Data Collection

Contributors: Anna-Kaisa Pietilainen, Tom Logde (University of Nottingham), Richard Mortier (University of Nottingham), Peter Tolmie (University of Notthingham), Renata Teixeira

Available at: https://muse.inria.fr/ucn, code https://github.com/ucn-eu

The User-Centric Networking (UCN) project is seeking to understand how people consume various kinds of content when using computer networks. Within this project we are undertaking a detailed user study across a range of environments in order to understand the practices involved in consuming media and other content according to context. For the study, we have set up the following tools and software:

- **Registration and management website:** we have developed a website containing information about the experiment, and user and device registration interfaces.
- **VPN server and clients for network traffic data collection:** we are using OpenVPN open-source VPN server and available free clients on multiple platforms (OpenVPN for Linux, OpenVPN for Android, Tunnelblick for OS X, OpenVPN Connect for iOS) to collect network traffic traces from the participating devices. The VPN server is running on a secure Inria server, and we collect packet headers using tcpdump and http traffic logs with a Squid HTTP proxy. Collected data is stored on another server not directly accessible from the Internet.
- **Activity logging software:** we have developed a small Android application to log additional activity details such as list of running applications, foreground application, screen state, network connectivity details, and system resources (cpu, memory, network, battery) usage.
- **Data collection from Moves and Google Calendar:** we have written some code to import user data from Moves application and a Google Calendar based diary to add user location and daily activity logs to the data set.
- **Data visualisation:** the website contains a section to visualise all the collected data (network traffic as a function of location, time of day, activity) to support interviews with an ethnographer.

  We have obtained Ethics approval from Inria's COERLE for conducting the data collection and the user study and have done the CNIL declaration for this data collection. Our data collection and user study will start early 2015.

MYRIADS Project-Team

4. New Software and Platforms

4.1. ConPaaS

Contact: Guillaume Pierre, `Guillaume.Pierre@irisa.fr`

URL: http://www.conpaas.eu/

Status: Version 1.4.2

License: BSD

Presentation: ConPaaS [60] is a runtime environment for hosting applications in the cloud. It aims at offering the full power of the cloud to application developers while shielding them from the associated complexity of the cloud. ConPaaS is designed to host both high-performance scientific applications and online Web applications. It automates the entire life-cycle of an application, including collaborative development, deployment, performance monitoring, and automatic scaling. This allows developers to focus their attention on application-specific concerns rather than on cloud-specific details.

Active contributors (from the Myriads team): Eliya Buyukkaya, Ancuta Iordache, Morteza Neishaboori, Guillaume Pierre, Dzenan Softic, Genc Tato, Teodor Crivat.

Impact: ConPaaS is recognized as one of the major open-source PaaS environments. It is being developed by teams in Rennes, Amsterdan, Berlin and Ljubljana. Technology transfer of ConPaaS technology is ongoing in the context of the MC-DATA EIT ICT Labs project.

4.2. HOCL-tools

Contact: Cédric Tedeschi, `Cédric.Tedeschi@irisa.fr`

Status: Version 1.0 to be released in open source

License: TBD

Presentation: HOCL (Higher Order Chemical Language) is a chemical programming language based on the chemical metaphor presented before (see Section 3.5). It was developed for several years within the PARIS and Myriads teams. Within HOCL, following the chemical metaphor, computations can be regarded as chemical reactions, and data can be seen as molecules which participate in these reactions. If a certain condition is held, the reaction will be triggered, thus continuing until it gets inert: no more data can satisfy any computing conditions. To realize this program paradigm, a multiset is implemented to act as a chemical tank, containing necessary data and rules. An HOCL program is then composed of two parts: *chemical rule definitions* (reaction rules) and *multiset definition* (data). More specifically, HOCL provides the higher order: reaction rules are molecules that can be manipulated like any other molecules. In other words, HOCL programs can manipulate other HOCL programs.

An HOCL compiler was developed using Java to execute some chemical programs expressed with HOCL. This compiler is based on the translation of HOCL programs to Java code. As a support for service coordination and service adaptation, we recently extended the HOCL compiler so as to support decentralized workflow execution. Works around the implementation of a distributed multiset gave birth to an underlying layer for this compiler, making it able to deploy HOCL programs transparently over large scale platforms. This last part is currently considered to be interfaced with the current HOCL compiler. All these features are planned to be released under the common name of *HOCL-tools*.

Active contributors (from Myriads project-team): Matthieu Simonin, Cédric Tedeschi, Javier Rojas Balderrama.

Impact: The compiler is used as a tool within the team to develop HOCL programs. The decentralized workflow execution support has been extensively used to produce results published and presented at several conferences. It is also used in the framework of the DALHIS [0] associated team, as a workflow template executor, integrated with the TIGRES workflow manager developed at the Lawrence Berkeley National Lab. It is supported by the GinFlow ADT funded by Inria.

4.3. Merkat

Contact: Nikolaos Parlavantzas, `Nikolaos.Parlavantzas@irisa.fr`

URL: http://www.irisa.fr/myriads/software/Merkat/

Status: Version 1.0

License: TBD

Presentation: Merkat is a market-based private PaaS (Platform-as-a-Service) system, supporting dynamic, fine-grained resource allocation and automatic application management [49], [48] [3]. Merkat implements a proportional-share auction that ensures maximum resource utilization while providing incentives to applications to regulate their resource usage. Merkat includes generic mechanisms for application deployment and automatic scaling. These mechanisms can be adapted to support diverse performance goals and application types, such as master-worker, MPI, or MapReduce applications. Merkat is implemented in Python and uses OpenNebula for virtual machine management. Experimental results on the Grid'5000 testbed show that using Merkat increases resource utilization and improves application performance. Merkat is currently being evaluated by EDF R&D using EDF high-performance applications. The development was initiated in the framework of Stefania Costache PhD's thesis.

Active contributors (from the Myriads team): Stefania Costache, Christine Morin, Nikolaos Parlavantzas.

Impact: Merkat has been integrated in EDF R&D portal providing access to internal computing resources and is currently used on a testbed at EDF R&D.

4.4. Meryn

Contact: Nikolaos Parlavantzas, `Nikolaos.Parlavantzas@irisa.fr`

URL: http://www.irisa.fr/myriads/software/Meryn/

Status: Version 1.0

License: TBD

Presentation: Meryn is an open, SLA-driven PaaS architecture that supports cloud bursting and allows hosting an extensible set of application types. Meryn relies on a decentralized optimization policy that aims at maximizing the overall provider profit, taking into account the penalties incurred when quality guarantees are unsatisfied [51]. The current Meryn prototype was implemented using shell scripts, builds upon the Snooze VM manager software, and supports batch and MapReduce applications using respectively the Oracle Grid Engine OGE 6.2u7 and Hadoop 0.20.2 frameworks. Meryn was developed in the framework of Djawida Dib's PhD thesis [10].

Active contributors (from the Myriads team): Djawida Dib, Christine Morin, Nikolaos Parlavantzas.

Impact: Meryn is not yet distributed as open source.

4.5. Resilin

[0] http://project.inria.fr/dalhis

Contact: Christine Morin, `Christine.Morin@inria.fr`

URL: http://resilin.inria.fr

Status: Version 1.0

License: GNU Affero GPL

Presentation: Resilin [6] is an open-source system for creating and managing MapReduce execution platforms over clouds. Resilin is compatible with the Amazon Elastic MapReduce (EMR) API, but it goes beyond Amazon's proprietary EMR solution in allowing users (e.g. companies, scientists) to leverage resources from one or more public and/or private clouds. This enables performing MapReduce computations over a large number of geographically-distributed and diverse resources. Resilin can be deployed across most of the open-source and commercial IaaS cloud management systems (e.g., OpenStack, OpenNebula, Amazon EC2). Once deployed, Resilin takes care of provisioning Hadoop clusters and submitting MapReduce jobs, allowing users to focus on writing their MapReduce applications rather than managing cloud resources. Resilin is implemented in the Python language and uses the Apache Libcloud library to interact with IaaS clouds. Resilin has been evaluated on multiple clusters of the Grid'5000 experimentation testbed. The results show that Resilin enables the use of geographically distributed resources with a limited impact on MapReduce job execution time.

Active contributors (from the Myriads project-team): Ancuta Iordache, Céline Merlet, Christine Morin, Nikolaos Parlavantzas, Matthieu Simonin.

Impact: Resilin is being used in the MOAIS project-team at Inria Grenoble - Rhône Alpes.

4.6. Snooze

Contact: Christine Morin, `Christine.Morin@inria.fr`

URL: http://snooze.inria.fr

Status: Version 2.1.5

License: GPLv2

Presentation: Snooze [53], [52], [54] [4] is a novel Infrastructure-as-a-Service (IaaS) cloud-management system, which is designed to scale across many thousands of servers and virtual machines (VMs) while being easy to configure, highly available, and energy efficient. For scalability, Snooze performs distributed VM management based on a hierarchical architecture. To support ease of configuration and high availability Snooze implements self-configuring and self-healing features. Finally, for energy efficiency, Snooze integrates a holistic energy management approach via VM resource (i.e. CPU, memory, network) utilization monitoring, underload/overload detection and mitigation, VM consolidation (by implementing a modified version of the Sercon algorithm [59]), and power management to transition idle servers into a power saving mode. Snooze is a highly modular piece of software. It has been extensively evaluated on the Grid'5000 testbed using realistic applications.

Snooze is fully implemented from scratch in Java and currently comprises approximately 15.000 lines of maintainable abstractions-based code. In order to provide a uniform interface to the underlying hypervisors and support transparent VM monitoring and management, Snooze integrates the *libvirt* virtualization library. Cassandra (since 2.0.0) can be used as base backend, providing reliability and scalability to the database management system. At a higher level Snooze provides its own REST API as well as an EC2 compatible API (since 2.1.0). It can thus be controlled from the command line (using the legacy client or an EC2 compatible tool), or from different langage libraries (libcloud, jcloud ...). Snooze also provides a web interface to control the system.

Snooze was used as a building box for two internships projects during the summer of 2014. The EC2 interface was used to execute Hadoop jobs configured by Resilin software. As a result we show that (1) the EC2 interface was expressive enough to work with a higher level tool and (2) the control over Snooze allow a better placement of data chunks for Hadoop jobs which leads to a better reliability of the execution of the different jobs. The second internship topic took part in a collaboration with the Northeastern University of Boston. The goal was to build a *Checkpoint as a Service* system. The service allows users to execute their computations in a cloud environment in a reliable way. Periodic checkpoints are saved making it possible to restore the computation from a previous state in case of failures. This work is described in [31].

Active contributors (from Myriads team): Jiajun Cao, Gene Cooperman, Eugen Feller, Yvon Jégou, David Margery, Christine Morin, Matthieu Simonin.

Impact: Snooze has been used by students at LIFL, IRIT in France and LBNL in the US in the framework of internships. It has also been deployed and experimented at EDF R&D. Snooze entry won the 2nd prize of the scalability challenge at CCGrid2013. Finally, we know that it was experimented by external users from academia and industry as we received feed-back from them. Snooze development was supported by the Snooze ADT funded by Inria from October 2012 to September 2014.

4.7. Virtual Execution Platform (VEP)

Contact: Yvon Jégou, `Yvon.Jegou@inria.fr`

URL: http://project.inria.fr/vep/

Status: Version 2.2

License: BSD

Presentation: Virtual Execution Platform

(VEP) [57] is a Contrail (http://contrail-project.eu) service that sits just above IaaS layer at the service provider end of the Contrail cloud federation. The VEP service provides a uniform interface for managing the whole lifecycle of elastic applications on the cloud and hides the details of the IaaS layer to the user. VEP applications are described in OVF (Open Virtualization Format) standard format. Resource usage is controlled by CEE (Constrained Execution Environment) rules which can be derived from SLAs (Service Level Agreement). The VEP service integrates a monitoring system where the major events about the application, mainly resource usage, are made available to the user.

The VEP service provides a RESTful interface and can be exploited directly by users on top of the provider IaaS. OpenNebula and OpenStack IaaS frameworks were initially supported. During the VEP-S EIT ICT Labs activity in 2014, VEP was extended with a new OCCI IaaS driver which allows to control any IaaS framework providing a standard OCCI API. Support for the new OCCI SLA proposition from OGF has also been added and allows to represent the VEP CEEs in a standard format. Finally, during this activity, the Zabbix open source distributed monitoring system was integrated to VEP.

Active contributors (from Myriads project-team): Roberto-Gioacchino Cascella, Florian Dudouet, Filippo Gaudenzi, Yvon Jégou, Christine Morin, Arnab Sinha.

Impact: VEP is part of Contrail software stack. External users can experiment with it using the open testbed operated by Myriads team. Technology transfer of VEP technology is ongoing in the context of the VEP-S EIT ICT Labs activity.

PHOENIX Project-Team

5. New Software and Platforms

5.1. DiaSuite: a Development Environment for Sense/Compute/Control Applications

Participants: Charles Consel [correspondent], Milan Kabac, Paul Van Der Walt, Adrien Carteron, Alexandre Spriet.

Despite much progress, developing a pervasive computing application remains a challenge because of a lack of conceptual frameworks and supporting tools. This challenge involves coping with heterogeneous devices, overcoming the intricacies of distributed systems technologies, working out an architecture for the application, encoding it in a program, writing specific code to test the application, and finally deploying it.

DIASUITE is a suite of tools covering the development life-cycle of a pervasive computing application:

- *Defining an application area.* First, an expert defines a catalog of entities, whether hardware or software, that are specific to a target area. These entities serve as building blocks to develop applications in this area. They are gathered in a taxonomy definition, written in the taxonomy layer of the DIASPEC language.
- *Designing an application.* Given a taxonomy, the architect can design and structure applications. To do so, the DIASPEC language provides an application design layer [35]. This layer is dedicated to an architectural pattern commonly used in the pervasive computing domain [31]. Describing the architecture application allows to further model a pervasive computing system, making explicit its functional decomposition.
- *Implementing an application.* We leverage the taxonomy definition and the architecture description to provide dedicated support to both the entity and the application developers. This support takes the form of a Java programming framework, generated by the DIAGEN compiler. The generated programming framework precisely guides the developer with respect to the taxonomy definition and the architecture description. It consists of high-level operations to discover entities and interact with both entities and application components. In doing so, it abstracts away from the underlying distributed technologies, providing further separation of concerns.
- *Testing an application.*DIAGEN generates a simulation support to test pervasive computing applications before their actual deployment. An application is simulated in the DIASIM tool, without requiring any code modification. DIASIM provides an editor to define simulation scenarios and a 2D-renderer to monitor the simulated application. Furthermore, simulated and actual entities can be mixed. This hybrid simulation enables an application to migrate incrementally to an actual environment.
- *Deploying a system.* Finally, the system administrator deploys the pervasive computing system. To this end, a distributed systems technology is selected. We have developed a back-end that currently targets the following technologies: Web Services, RMI, SIP and OSGI. This targeting is transparent for the application code. The variety of these target technologies demonstrates that our development approach separates concerns into well-defined layers.

This development cycle is summarized in the Figure 2 .

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

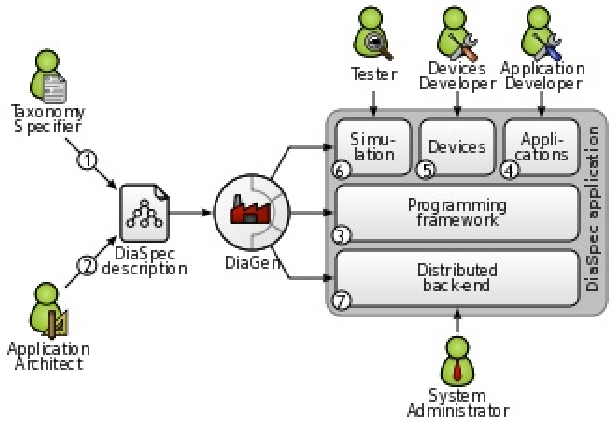


Figure 2. DiaSuite *Development Cycle*

5.1.1. DiaSpec: a Domain-Specific Language for Networked Entities

The core of the DiaSuite development environment is the domain specific language called DiaSpec and its compiler DiaGen:

- DiaSpec is composed of two layers:
  - The *Taxonomy Layer* allows the declaration of entities that are relevant to the target application area. An entity consists of sensing capabilities, producing data, and actuating capabilities, providing actions. Accordingly, an entity description declares a data source for each one of its sensing capabilities. As well, an actuating capability corresponds to a set of method declarations. An entity declaration also includes attributes, characterizing properties of entity instances. Entity declarations are organized hierarchically allowing entity classes to inherit attributes, sources and actions. A taxonomy allows separation of concerns in that the expert can focus on the concerns of cataloging area-specific entities. The entity developer is concerned about mapping a taxonomical description into an actual entity, and the application developer concentrates on the application logic.
  - The *Architecture Layer* is based on an architectural pattern commonly used in the pervasive computing domain [31]. It consists of context components fueled by sensing entities. These components process gathered data to make them amenable to the application needs. Context data are then passed to controller components that trigger actions on entities. Using an architecture description enables the key components of an application to be identified, allowing their implementation to evolve with the requirements (*e.g.,* varying light management implementations in a controller component to optimize energy consumption).
- DiaGen is the DiaSpec compiler that performs both static and runtime verifications over DiaSpec declarations and produces a dedicated programming framework that guides and eases the implementation of components. The generated framework is independent of the underlying distributed technology. As of today, DiaGen supports multiple targets: Local, RMI, SIP, Web Services and OSGI.

5.1.2. DiaSim: a Parametrized Simulator for Pervasive Computing Applications

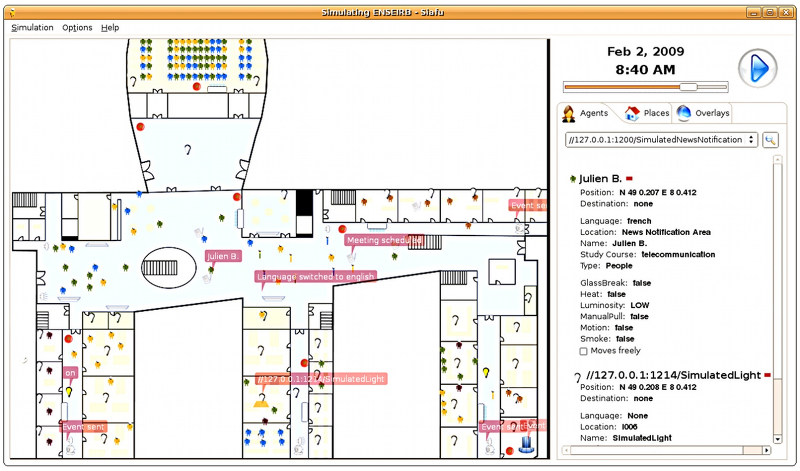


Figure 3. A screenshot of the DIASIM *simulator*

Pervasive computing applications involve both software and integration concerns. This situation is problematic for testing pervasive computing applications because it requires acquiring, testing and interfacing a variety of software and hardware entities. This process can rapidly become costly and time-consuming when the target environment involves many entities.

To ease the testing of pervasive applications, we are developing a simulator for pervasive computing applications: DIASIM. To cope with widely heterogeneous entities, DIASIM is parameterized with respect to a DIASPEC specification describing a target pervasive computing environment. This description is used to generate with DIAGEN both a programming framework to develop the simulation logic and an emulation layer to execute applications. Furthermore, a simulation renderer is coupled to DIASIM to allow a simulated pervasive system to be visually monitored and debugged. The simulation renderer is illustrated in Figure 3 .

5.2. DiaSuiteBox: an Open Orchestration Platform

Participants: Charles Consel, Adrien Carteron, Alexandre Spriet, Milan Kabac.

The DiaSuiteBox platform runs an open-ended set of applications leveraging a range of appliances and web services. Our solution consists of a dedicated development environment, a certifying application store, and a lightweight runtime platform. This solution is based on the DIASUITE project.

5.2.1. DiaSuiteBox platform architecture

The DiaSuiteBox platform can be embedded in a small plug-computer or deployed in the cloud. Thanks to the application store and the developer community, the platform is fed by a full offer of new innovative applications. During the submission process, an application is automatically analyzed and checked in order to be certified. The user is ensured the behavior of its applications are innocuous and correct with respect to the provided information. Finally, DiaSuiteBox provides an extensible software architecture. This allows the easily connect new device technologies to the platform. For example, the support for new wireless communication technologies such as Zigbee, Z-Wave or Sigfox can be easily added to the DiaSuiteBox platform.

More details can be found on the web page http://diasuitebox.inria.fr.

The iQSpot startup uses DiaSuiteBox as a software platform to ease the management of Smart Buildings. In this project, the DiaSuiteBox platform is first used to host building management functionalities such as lighting management, heating/ventilating/air conditioning management, energy efficiency monitoring. It is also used to host software drivers that allow the building management functionalities to interact with the connected devices deployed in buildings. These devices can use wired communication technologies such LonWorks, BACNet or KNX, as well as wireless communication technologies such as Z-Wave or Zigbee.

5.3. School+ Apps: Assistive tablet applications for school Inclusion

Participants: Charles Consel [correspondent], Hélène Sauzéon, Charles Fage, Cécile Magnier.

School+ is a package of 7 applications. Three applications are assistive applications, guiding the child doing specific tasks. Three others are training applications made as serious games, addressing specific skills. The last application is a meta-application, comprising a link to the three training applications, with an access to statistics of their usage. For each application, data are separated from the design, meaning that every element of each application (pictures, texts, settings, etc.) can be changed at any time. Each application records a log file containing all the interactions performed by the child.

5.3.1. Assistive applications:

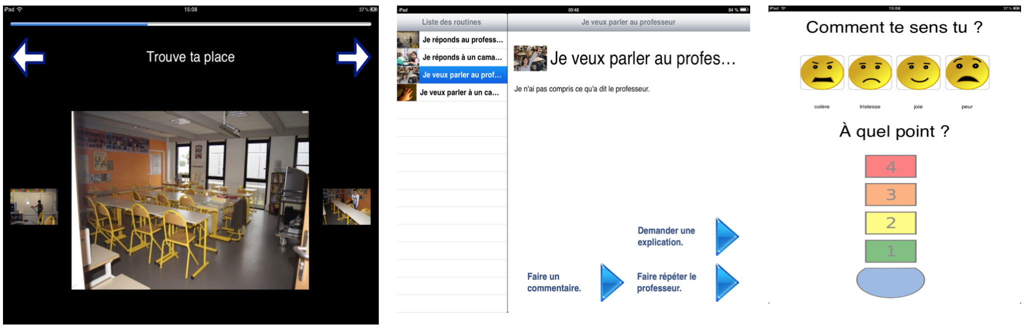


Figure 4. Assistive applications

- *Routines application*. This application shows a list of tasks, with a short description. After clicking the starting button, a specific slideshow is shown; it decomposes a task into steps. For each step, a text and a picture can be displayed. Thumbnail of previous and next steps are also displayed. This application guides the child through classroom situations: entering classroom, taking school materials out of a backpack, writing notes, handling agenda, leaving the classroom.
- *Communication application*. With the same design, the assistance provided by this application targets to communicating situations inside the classroom. The application covers four scenarios addressing two interaction situations (initiating and answering the interaction) and two types of interlocutors (professor and classmate). For each scenario, different slideshows guide the child, depending on the goal of the interaction.
- *Emotion Regulation application*. This application aims to assist the child to self-regulate his/her emotions. Four simplified emoticons are proposed to the child to choose from: anger, sadness, joy and fear. Then, (s)he selects a level of intensity via a thermometer with a scale from 1 to 4. In response, the application delivers different multimedia contents according to the level selected to help the child regulate his/her emotions. Typically, a text (breathing instructions) are shown at level 1, pictures at level 2, a video at level 3 and another text at level 4.

5.3.2. Training applications:

These three applications are serious games with increasing levels of difficulties, reachable after a ratio of good answers has been attained.

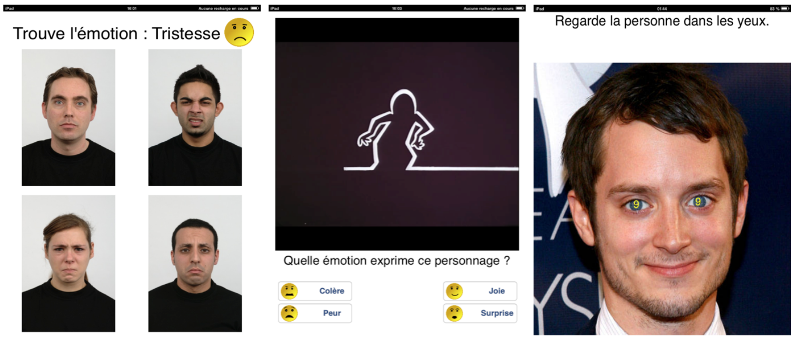


Figure 5. Training applications

- *Emotion Recognition application: pictures*. In this application, the child is instructed to identify a specific emotion among 4 pictures showing different people exhibiting an emotion. Seven emotions are involved in this application: joy, sadness, fear, anger, surprise, disgust and neutral. The emotion to be recognized is displayed together with its simplified emoticon. The type of pictures changes with the difficulty level: level 1 contains pictures of unfamiliar people and level 2 contains pictures of friends and relatives of the child.
- *Emotion Recognition application: videos*. In this application, the child is presented with a fragment of an animated cartoon. At some point, the video stops and the child is asked to identify the emotion of the character. Four emotions are involved in this application: joy, sadness, fear and anger. Videos are slowed down, with a speed percentage that can be changed at each level. Videos change with difficulty level: level 1 contains videos of a very basic cartoon (only one cartoon character drawn by basic form un-textured), level 2 contains a video of more sophisticated cartoons and level 3 contains movies with actors.
- *Attention Training*. In this application, the child is presented a picture of a face and asked to make eye contact with it. Second, a symbol appears briefly in the eyes of the character. Third, the child is asked to identify the symbol shown in the previously displayed picture, to make sure he kept eye contact. The speed at which the symbol appears and disappears is changed according to the difficulty level. Types of pictures also change with the level : level 1 contains pictures of faces and level 2 contains pictures of classroom situations.

5.4. HomeAssist: A Platform for Assistive Living

Participants: Charles Consel, Loïc Caroux [correspondent], Thomas Freslon, Adrien Carteron, David Daney, Lucile Dupuy, Geoffrey Escojido, Bernard N'Kaoua, Hélène Sauzéon, Alexandre Spriet.

The HomeAssist platform proposes a systemic approach to introducing an assistive technological platform for older people. To do so, we formed a trans-disciplinary team that allows (1) to identify the user needs from a gerontological and psychological viewpoint; (2) to propose assistive applications designed by human factors and HCI experts, in collaboration with caregivers and users; (3) to develop and test applications by software engineers; (4) to conduct a field study for assessing the benefits of the platform and assistive applications, in collaboration with caregivers, by deploying the system at the actual home of elders.

The HomeAssist platform is implemented on top of the DiaSuiteBox platform, using a suite of tools, namely DiaSuite, that have been designed, developed and tested by our research group at Inria. The DiaSuite tools include a dedicated integrated development environment that enables applications to be developed quickly and safely. This technology has been successfully applied to a variety of domains where environments consist of networked objects that need to be orchestrated.

5.4.1. Applications

HomeAssist offers an online catalog of applications. Using this catalog, the user and the caregiver determine what and how activities should be assisted by selecting the appropriate assistive applications and configuring them with respect to the user's requirements and preferences. The resulting set of applications forms a personalized assistive support. Additionally, to respond to evolving needs, our platform allows to stop/remove applications easily and to install new ones from the online catalog.

This platform proposes many applications in three domains of everyday life:

- Daily activities: including activity monitoring, light path, and a reminder.
- Home or personal safety: including entrance monitoring, stove monitoring, and warning if no movements are detected after a certain amount of time.
- Communications and social activities: including collaborative games, videoconference, information about local events, TV programming, *etc*.

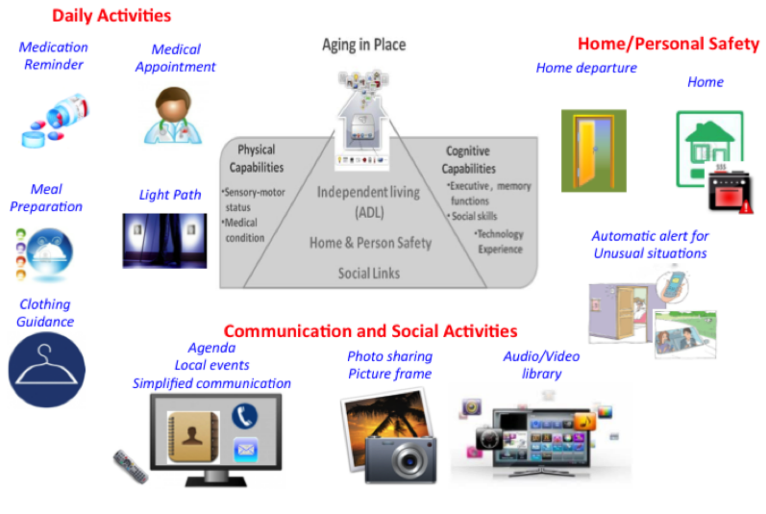


Figure 6. The HomeAssist platform and applications

For video presentations of HomeAssist, see the following:

- http://videotheque.inria.fr/videotheque/media/23705. Title: "Dia Suite Box". Produced in 2013.
- http://videotheque.inria.fr/videotheque/media/29998. Title: "DomAssist : L'assistance numérique à la personne". Produced in 2014.

5.4.2. Devices

Several entities have been identified to deliver an assistive support. These entities include (1) technological devices: wireless sensors (motion detectors, contact sensors and smart electric switches), and two tablets, and (2) software services (agenda, address book, mail agent, and photo agent) to monitor everyday activities and propose assistive applications. Sensors are placed in relevant rooms in the house: kitchen, bedroom, bathroom, and around the entrance.

Figure 7. HomeAssist devices

5.4.3. Experimental validation

A field study is currently being conducted with elderly people. The major purpose of this study is to identify the benefits of using HomeAssist for this population in an ecological framework. We selected 24 elderly people with different levels of autonomy (GIR scores). The HomeAssist technology has been installed in their house during 9 months. Twenty-four non-equipped elders were also selected to participate to the study, as control participants.

The expected impact of HomeAssist reflects the trans-disciplinary nature of the project. We aim to deliver results in the domain of (1) elderly care, (2) ergonomics and human factors, and (3) pervasive computing.

The major expected results are that HomeAssist (1) prolongs ageing in place, improves well-being of the users, and improves the efficiency of the caregiving environment; (2) is a cognitively low-cost assistive technology, and is well accepted and perceived as useful and usable by the users; (3) is technologically robust, and is a validated assistive platform

First results are expected in January 2015.

5.4.4. External Partners

The HomeAssist platform is being developed with support from the following partners:

- Équipe “Handicap et Système Nerveux” (EA 4136), Bordeaux University
- Chaire TSA, Université du Québec Trois-Rivières
- CRIUGM, Université de Montréal
- UDCCAS Gironde
- CARSAT
- Conseil Général 33
- Conseil Régional d’Aquitaine

5.5. DiaSwarm: a Development Environment for orchestrating smart objects at a large scale

Participants: Charles Consel [correspondent], Milan Kabac, Adrien Carteron, Eugène Volanschi.

The development of orchestrating applications which are responsible for large numbers of smart objects raises a number of challenges. We have addressed these by introducing a new design language called DiaSwarm, which is an extended version of the DiaSpec language.

5.5.1. Service discovery

Standard service discovery at the individual object level does not address the needs of applications orchestrating large numbers of smart objects. Instead, a high-level approach which provides constructs to specifying subsets of interest is needed. Our approach allows developers to introduce application-specific concepts (e.g., regrouping parking spaces into lots or districts) at the design time and then these can be used to express discovery operations. Following our design-driven development approach, these concepts are used to generate code to support and guide the programming phase.

5.5.2. Data gathering

Applications need to acquire data from a large number of objects through a variety of delivery models. For instance, air pollution sensors across a city may only push data to the relevant applications when pollution levels exceed tolerated levels. Tracking sensors, however, might determine the location of vehicles and send the acquired measurements to applications periodically (e.g., 10 min. intervals). Data delivery models need to be introduced at design time since they have a direct impact on the application's program structure. In doing so, the delivery models used by an application can be checked against sensor features early in the development process.

5.5.3. Data processing

Data that is generated from hundreds of thousands of objects and accumulated over a period of time calls for efficient processing strategies to ensure the required performance is attained. Our approach allows for an efficient implementation of the data processing stage by providing the developer with a framework based on the MapReduce [30] programming model which is intended for the processing of large data sets.

More details on this software platform can be found in the special issue on Smart Cities of the journal ERCIM News [18], 2014.

RAP Project-Team (section vide)

REGAL Project-Team

4. New Software and Platforms

4.1. NumaGiC

Participants: Lokesh Gidra, Marc Shapiro, Julien Sopena [correspondent], Gaël Thomas.

NumaGiC is a version of the HotSpot garbage collector (GC) adapted to many-core computers with very large main memories. In order to maximise GC throughput, it manages the trade-off between memory locality (local scans) and parallelism (work stealing) in a self-balancing manner. Furthemore, the collector features several memory placement heuristics that improve locality. NumaGiC is described in a paper accepted for publication at ASPLOS 2015 [29].

4.2. G-DUR

Participants: Masoud Saeida Ardekani, Dastagiri Reddy Malikireddy, Marc Shapiro [correspondent].

A large family of distributed transactional protocols have a common structure, called Deferred Update Replication (DUR). DUR provides dependability by replicating data, and performance by not re-executing transactions but only applying their updates. Protocols of the DUR family differ only in behaviors of few generic functions. Based on this insight, we offer a generic DUR middleware, called G-DUR, along with a library of finely-optimized plug-in implementations of the required behaviors. This paper presents the middleware, the plugins, and an extensive experimental evaluation in a geo-replicated environment. Our empirical study shows that:

1. G-DUR allows developers to implement various transactional protocols under 600 lines of code;
2. It provides a fair, apples-to-apples comparison between transactional protocols;
3. By replacing plugs-ins, developers can use G-DUR to understand bottlenecks in their protocols;
4. This in turn enables the improvement of existing protocols; and
5. Given a protocol, G-DUR helps evaluate the cost of ensuring various degrees of dependability.

G-DUR and the results of the comparison campaign are described in a paper to Middleware 2014 [33]. This research is supported in part by ConcoRDanT ANR project (Section 7.1.7) and by the FP7 grant SyncFree (Section 7.2.1.1).

Jessy is freely available on github under http://Github.com/msaeida/jessy under an Apache license.

4.3. SwiftCloud

Participants: Mahsa Najafzadeh, Marc Shapiro [correspondent], Serdar Tasiran, Marek Zawirski.

Client-side (e.g., mobile or in-browser) apps need local access to shared cloud data, but current technologies either do not provide fault-tolerant consistency guarantees, or do not scale to high numbers of unreliable and resource-poor clients, or both. Addressing this issue, the SwiftCloud distributed object database supports high numbers of client-side partial replicas. SwiftCloud offers fast reads and writes from a causally-consistent client-side cache. It is scalable, thanks to small and bounded metadata, and available, tolerating faults and intermittent connectivity by switching between data centres. The price to pay is a modest amount of staleness. A recent Inria Research Report (submitted for publication) presents the SwiftCloud algorithms, design, and experimental evaluation, which shows that client-side apps enjoy the same guarantees as a cloud data store, at a small cost.

SwiftCloud is supported by the ConcoRDanT ANR project (Section 7.1.7), by a Google Research Award, and by the FP7 grant SyncFree (Section 7.2.1.1).

The code is freely available on http://gforge.inria.fr/ under a BSD license.

4.4. Antidote

Participants: Tyler Crain, Marc Shapiro [correspondent], Serdar Tasiran, Alejandro Tomsic.

Antidote is the flexible cloud database platform currently under development in the SyncFree European project (Section 7.2.1.1). Antidote aims to be both a research platform for studying replication and consistency at the large scale, and an instrument for exploiting research results. The platform supports replication of CRDTs, in and between sharded (partitioned) data centres (DCs). The current stable version supports strong transactional consistency inside a DC, and causal transactional consistency between DCs. Ongoing research includes support for explicit consistency [23], for elastic version management, for adaptive replication, for partial replication, and for reconfigurable sharding.

RMOD Project-Team

5. New Software and Platforms

5.1. Moose5.0

Participants: Stéphane Ducasse [correspondant], Muhammad Bhatti, Andre Calvante Hora, Nicolas Anquetil, Anne Etien, Guillaume Larcheveque, Tudor Gîrba [University of Bern].

Web: http://www.moosetechnology.org/

The platform. Moose is a language-independent environment for reverse- and re-engineering complex software systems. Moose provides a set of services including a common meta-model, metrics evaluation and visualization, a model repository, and generic GUI support for querying, browsing and grouping. The development of Moose began at the Software Composition Group in 1997, and is currently contributed to and used by researchers in at least seven European universities. Moose offers an extensible meta-described metamodel, a query engine, a metric engine and several visualizations. Moose is currently in its fourth major release and comprises 55,000 lines of code in 700 classes.

The RMoD team is currently the main maintainer of the Moose platform. There are 200 publications (journal, international conferences, PhD theses) based on execution or use of the Moose environment.

The first version running on top of Pharo (Moose 4.0) was released in June 2010. The current focus is Moose 5.0, in late beta testing as of December 2014. A major development of 2014 is that tools and frameworks built for Moose are being integrated into Pharo4 as the default development tools.
Here is the self-assessment of the team effort following the grid given at http://www.inria.fr/institut/organisation/instances/commission-d-evaluation.

- **(A5)** Audience : 5 – Moose is used by several research groups, a consulting company, and some companies using it in ad-hoc ways.
- **(SO4)** Software originality : 4 – Moose aggregates the last results of several research groups.
- **(SM4)** Software Maturity : 4 – Moose is developed since 1996 and got two main redesign phases.
- **(EM4)** Evolution and Maintenance : 4 – Moose will be used as a foundation of our Synectique start up so its maintenance is planned.
- **(SDL4)** Software Distribution and Licensing : 4 – Moose is licensed under BSD
- **(OC)** Own Contribution : (Design/Architecture)DA-4, (Coding/Debugging)-4, (Maintenance/Support)-4, (Team/Project Management)-4

5.2. Pharo3.0

Participants: Marcus Denker [correspondant], Damien Cassou, Stéphane Ducasse, Esteban Lorenzano, Damien Pollet, Igor Stasenko, Camillo Bruni, Camille Teruel, Clément Bera.

Web: http://www.pharo.org/

The platform. Pharo is an open-source Smalltalk-inspired language and environment. It provides a platform for innovative development both in industry and research. By providing a stable and small core system, excellent developer tools, and maintained releases, Pharo's goal is to be a platform to build and deploy mission critical applications, while at the same time continue to evolve.

The first stable version, Pharo 1.0, was released in 2010. We are now releasing one new version of Pharo every year, with Pharo3 released in spring 2014. Pharo4 has seen already over 400 incremental updates and is scheduled for early 2015. It should be noted that Pharo, even though already used outside of research, still continues to improve radically.

In November 2012 RMoD launched the Pharo Consortium (http://consortium.pharo.org/) and the Pharo Association (http://association.pharo.org). The consortium has now 14 industrial members, 3 sponsors and 10 academic partners.

RMoD is the main maintainer and coordinator of Pharo.
Here is the self-assessment of the team effort following the grid given at http://www.inria.fr/institut/organisation/instances/commission-d-evaluation.

- **(A5)** Audience: 5 – Used in many universities for teaching, more than 25 companies.
- **(SO3)** Software originality : 3 – Pharo offers a classical basis for some aspects (UI). It includes new frameworks and concepts compared to other Smalltalk implementations.
- **(SM4)** Software Maturity: 4 – Bug tracker, continuous integration, large test suites are in place.
- **(EM4)** Evolution and Maintenance: 4 – Active user group, consortium and association has been set up.
- **(SDL4)** Software Distribution and Licensing: 4 – Pharo is licensed under MIT.
- **(OC5)** Own Contribution: (Design/Architecture) DA-5, (Coding/Debugging) CD-5, (Maintenance/Support) MS-5, (Team/Project Management) TPM-5

5.3. Pillar 0.17

Pillar is a markup syntax and associated tools to write and generate documentation and books. Pillar is currently used to write several books and other documentation. Two platforms have already been created on top of Pillar: PillarHub and Marina. http://www.smalltalkhub.com/#!/~Pier/Pillar

ROMA Team

5. New Software and Platforms

5.1. MUMPS

Participants: Patrick Amestoy, Alfredo Buttari, Jean-Yves L'Excellent [correspondent], Chiara Puglisi, Wissam M. Sid-Lakhdar, Bora Uçar.

MUMPS (for *MUltifrontal Massively Parallel Solver*) see http://mumps-solver.org is a software package for the solution of large sparse systems of linear equations. It implements a direct method, the so called multifrontal method; it is a parallel code capable of exploiting distributed-memory computers as well as multithreaded libraries; its main originalities are its numerical robustness (including partial threshold pivoting in distributed-memory environment) and its wide range of features.

The latest public release is MUMPS 4.10.0 (May 2011); the new release is scheduled for February 2015 and will be under the Cecill-C licence, following an agreement between CERFACS, CNRS, ENS Lyon, INPT, Inria and University of Bordeaux.

RUNTIME Team

5. New Software and Platforms

5.1. Common Communication Interface

Participant: Brice Goglin.

- The *Common Communication Interface* aims at offering a generic and portable programming interface for a wide range of networking technologies (Ethernet, InfiniBand, ...) and application needs (MPI, storage, low latency UDP, ...).
- CCI is developed in collaboration with the *Oak Ridge National Laboratory* and several other academics and industrial partners.
- CCI is in early development and currently composed of 19 000 lines of C.
- http://www.cci-forum.org

5.2. Hardware Locality

Participants: Brice Goglin, Samuel Thibault.

- *Hardware Locality* (HWLOC) is a library and set of tools aiming at discovering and exposing the topology of machines, including processors, cores, threads, shared caches, NUMA memory nodes and I/O devices.
- It builds a widely-portable abstraction of these resources and exposes it to the application so as to help them adapt their behavior to the hardware characteristics.
- HWLOC targets many types of high-performance computing applications [2] [20], from thread scheduling to placement of MPI processes. Most existing MPI implementations, several resource managers and task schedulers already use HWLOC.
- HWLOC is developed in collaboration with the OPEN MPI project. The core development is still mostly performed by Brice GOGLIN and Samuel THIBAULT from the RUNTIME team-project, but many outside contributors are joining the effort, especially from the OPEN MPI and MPICH2 communities.
- HWLOC is composed of 30 000 lines of C.
- http://www.open-mpi.org/projects/hwloc

5.3. Network Locality

Participant: Brice Goglin.

- *Netloc Locality* (NETLOC) is a library that extends hwloc to network topology information by assembling hwloc knowledge of server internals within graphs of inter-node fabrics such as Ethernet or Infiniband.
- NETLOC targets the same challenges as hwloc but focuses on a wider spectrum by enabling cluster-wide solutions such process placement [21].
- NETLOC is developed in collaboration with University of Wisconsin in LaCrosse and Cisco, within the OPEN MPI project.
- NETLOC is composed of 15 000 lines of C. It was recently merged in the HWLOC repository was better integration.
- http://netloc.org

5.4. KNem

Participant: Brice Goglin.

- KNEM (*Kernel Nemesis*) is a Linux kernel module that offers high-performance data transfer between user-space processes.
- KNEM offers a very simple message passing interface that may be used when transferring very large messages within point-to-point or collective MPI operations between processes on the same node.
- Thanks to its kernel-based design, KNEM is able to transfer messages through a single memory copy, much faster than the usual user-space two-copy model.
- KNEM also offers the optional ability to offload memory copies on INTEL I/O AT hardware which improves throughput and reduces CPU consumption and cache pollution.
- KNEM is developed in collaboration with the MPICH2 team at the Argonne National Laboratory and the OPEN MPI project. These partners already released KNEM support as part of their MPI implementations.
- KNEM is composed of 8 000 lines of C. Its main contributor is Brice GOGLIN.
- http://runtime.bordeaux.inria.fr/knem/

5.5. Open-MX

Participant: Brice Goglin.

- The OPEN-MX software stack is a high-performance message passing implementation for any generic ETHERNET interface.
- It was developed within our collaboration with Myricom, Inc. as a part of the move towards the convergence between high-speed interconnects and generic networks.
- OPEN-MX exposes the raw ETHERNET performance at the application level through a pure message passing protocol.
- While the goal is similar to the old GAMMA stack [45] or the recent iWarp [44] implementations, OPEN-MX relies on generic hardware and drivers and has been designed for message passing.
- OPEN-MX is also wire-compatible with Myricom MX protocol and interface so that any application built for MX may run on any machine without Myricom hardware and talk other nodes running with or without the native MX stack.
- OPEN-MX is also an interesting framework for studying next-generation hardware features that could help ETHERNET hardware become legacy in the context of high-performance computing. Some innovative message-passing-aware stateless abilities, such as multiqueue binding and interrupt coalescing, were designed and evaluated thanks to OPEN-MX [5].
- Brice GOGLIN is the main contributor to OPEN-MX. The software is already composed of more than 45 000 lines of code in the Linux kernel and in user-space.
- http://open-mx.gforge.inria.fr/

5.6. StarPU

Participants: Olivier Aumage, Andra Hugo, Nathalie Furmento, Raymond Namyst, Marc Sergent, Samuel Thibault, Pierre-André Wacrenier.

- STARPU permits high performance libraries or compiler environments to exploit heterogeneous multicore machines possibly equipped with GPGPUs or Xeon Phi processors.
- STARPU offers a unified offloadable task abstraction named codelet.In case a codelet may run on heterogeneous architectures, it is possible to specify one function for each architectures (e.g. one function for CUDA and one function for CPUs).

- STARPU takes care to schedule and execute those codelets as efficiently as possible over the entire machine. A high-level data management library enforces memory coherency over the machine: before a codelet starts (e.g. on an accelerator), all its data are transparently made available on the compute resource.
- STARPU obtains portable performances by efficiently (and easily) using all computing resources at the same time.
- STARPU also takes advantage of the heterogeneous nature of a machine, for instance by using scheduling strategies based on auto-tuned performance models.
- STARPU can also leverage existing parallel implementations, by supporting *parallel tasks*, which can be run concurrently over the machine.
- STARPU provides *scheduling contexts* which can be used to partition computing resources. Scheduling contexts can be dynamically resized to optimize the allocation of computing resources among concurrently running libraries.
- STARPU provides integration in MPI clusters through a lightweight DSM over MPI.
- STARPU provides a scheduling platform, which makes it easy to implement and experiment with scheduling heuristics
- STARPU comes with a plug-in for the GNU Compiler Collection (GCC), which extends languages of the C family with syntactic devices to describe STARPU's main programming concepts in a concise, high-level way.
- STARPU has support for simulating the execution, by using the simgrid simulator, which allows to reproduce experiments on a remote system, or to even virtually modify the platform used to run the application
- STARPU has been extended to provide runtime support for the KLANG-OMP OpenMP compiler. StarPU's OpenMP runtime support is compliant with most OpenMP 3.0 constructs, and also supports new dependent tasks and accelerated targets constructs introduced by OpenMP 4.0.
- http://runtime.bordeaux.inria.fr/StarPU/

5.7. Klang-OMP

Participants: Olivier Aumage, Nathalie Furmento, Samuel Pitoiset, Samuel Thibault.

- The KLANG-OMP software is a source-to-source OpenMP compiler for languages C and C++. It is developed as part of the Inria development action "ADT K'STAR " jointly managed by Inria teams MOAIS (Inria Montbonnot) and RUNTIME (Inria Bordeaux - Sud-Ouest). The KLANG-OMP compiler translates OpenMP directives and constructs into API calls from the StarPU runtime system or the XKaapi runtime system (XKaapi is developed by the MOAIS team).
- The KLANG-OMP compiler is virtually fully compliant with OpenMP 3.0 constructs.
- The KLANG-OMP compiler supports OpenMP 4.0 dependent tasks and accelerated targets.

5.8. Kastors

Participants: Olivier Aumage, Nathalie Furmento, Samuel Pitoiset, Samuel Thibault.

- The KASTORS software is a suite of benchmarks for testing the performance of OpenMP compilers on codes making use of the new dependent tasks OpenMP 4.0 constructs. It is ported and maintained as part of the Inria development action "ADT K'STAR " jointly managed by Inria teams MOAIS (Inria Montbonnot) and RUNTIME (Inria Bordeaux - Sud-Ouest). It is constituted of well known computing kernels that have been ported on the OpenMP 4.0 dependent tasks model. The KASTORS suite has been introduced to the OpenMP community during the IWOMP 2014 conference [32].

5.9. NewMadeleine and PIOMan

Participant: Alexandre Denis.

- NEWMADELEINE is a communication library for high performance networks, based on a modular architecture using software components.
- The NEWMADELEINE optimizing scheduler aims at enabling the use of a much wider range of communication flow optimization techniques such as packet reordering or cross-flow packet aggregation.
- NEWMADELEINE targets applications with irregular, multiflow communication schemes such as found in the increasingly common application conglomerates made of multiple programming environments and coupled pieces of code, for instance.
- It is designed to be programmable through the concepts of optimization *strategies*, allowing experimentations with multiple approaches or on multiple issues with regard to processing communication flows, based on basic communication flows operations such as packet merging or reordering.
- PIOMAN is a generic framework to be used by communication libraries, that brings seamless asynchronous progression of communication by opportunistically using available cores. It uses system threads and thus is composable with any runtime system used for multithreading.
- PIOMAN is closely integrated with the NEWMADELEINE communication library and PadicoTM.
- The reference software development branch of the NEWMADELEINE software consists in 60 000 lines of code. NEWMADELEINE is available on various networking technologies: Myrinet, Infiniband, Quadrics and ETHERNET. It is developed and maintained by Alexandre DENIS.
- http://pm2.gforge.inria.fr/newmadeleine/

5.10. PadicoTM

Participant: Alexandre Denis.

- PadicoTM is a high-performance communication framework for grids. It is designed to enable various middleware systems (such as CORBA, MPI, SOAP, JVM, DSM, etc.) to utilize the networking technologies found on grids.
- PadicoTM aims at decoupling middleware systems from the various networking resources to reach transparent portability and flexibility.
- PadicoTM architecture is based on software components. Puk (the PadicoTM micro-kernel) implements a light-weight high-performance component model that is used to build communication stacks.
- PadicoTM component model is now used in NEWMADELEINE. It is the cornerstone for networking integration in the projects “LEGO” and “COOP” from the ANR.
- PadicoTM is composed of roughly 60 000 lines of C.
- PadicoTM is registered at the APP under number IDDN.FR.001.260013.000.S.P.2002.000.10000.
- http://pm2.gforge.inria.fr/PadicoTM/

5.11. MAQAO

Participants: Denis Barthou, Olivier Aumage, Christopher Haine, James Tombi A Mba.

- MAQAO is a performance tuning tool for OpenMP parallel applications. It relies on the static analysis of binary codes and the collection of dynamic information (such as memory traces). It provides hints to the user about performance bottlenecks and possible workarounds.
- MAQAO relies on binary codes for Intel x86 and ARM architectures. For x86 architecture, it can insert probes for instrumention directly inside the binary. There is no need to recompile. The static/dynamic approach of MAQAO analysis is the main originality of the tool, combining performance model with values collected through instrumentation.

- MAQAO has a static performance model for x86 and ARM architectures. This model analyzes performance of the codes on the architectures and provides some feed-back hints on how to improve these codes, in particular for vector instructions.
- The dynamic collection of data in MAQAO enables the analysis of thread interactions, such as false sharing, amount of data reuse, runtime scheduling policy, ...
- MAQAO is in the European FP7 project ”MontBlanc”.

5.12. QIRAL

Participants: Denis Barthou, Olivier Aumage.

- QIRAL is a high level language (expressed through LaTeX) that is used to described Lattice QCD problems. It describes matrix formulations, domain specific properties on preconditionings, and algorithms.
- The compiler chain for QIRAL can combine algorithms and preconditionings, checking validity of the composition automatically. It generates OpenMP parallel code, using libraries, such as BLAS.
- This code is developped in collaboration with other teams participating to the ANR PetaQCD project.

5.13. TreeMatch

Participants: Emmanuel Jeannot, Guillaume Mercier, François Tessier.

- TREEMATCH is a library for performing process placement based on the topology of the machine and the communication pattern of the application.
- TREEMATCH provides a permutation of the processes to the processors/cores in order to minimize the communication cost of the application.
- Important features are : the number of processors can be greater than the number of applications processes ; it assumes that the topology is a tree and does not require valuation of the topology (e.g. communication speeds) ; it implements different placement algorithms that are switched according to the input size.
- Some core algorithms are parallel to speed-up the execution.
- TREEMATCH is integrated into various software such as the Charm++ programming environment as well as in both major open-source MPI implementations: Open MPI and MPICH2.
- TREEMATCH is available at: http://treematch.gforge.inria.fr.

SCALE Team

5. New Software and Platforms

5.1. Platforms

5.1.1. EventCloud

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

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

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

5.1.2. BtrPlace

Participants : Fabien Hermenier, Vincent Kherbache, Ludovic Henrio.

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

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

5.1.3. OSA

Participants : Olivier Dalle.

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

5.1.4. VerCors

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

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

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

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

SOCRATE Project-Team

5. New Software and Platforms

5.1. WSnet

Socrate is an active contributor to WSnet (http://wsnet.gforge.inria.fr/) a multi-hop wireless network discrete event simulator. WSnet was created in the ARES team and it is now supported by the D-NET team of Inria Rhône-Alpes.

5.2. Wiplan

Wiplan is a software including an Indoor propagation engine and a wireless LAN optimization suite, which has been registered by INSA-Lyon. The heart of this software is the propagation simulation core relying on an original method, MR-FDPF (multi-resolution frequency domain ParFlow), proposed by JM Gorce in 2001 and further extended. The discrete ParFlow equations are translated in the Fourier domain providing a large linear system, solved in two steps taking advantage of a multi- resolution approach. The first step computes a cell-based tree structure referred to as the pyramid. In the second phase, a radiating source is simulated, taking advantage of the pre-processed pyramidal structure. Using of a full-space discrete simulator instead of classical ray-tracing techniques is a challenge due to the inherent high computation requests. However, we have shown that the use of a multi-resolution approach allows the main computational load to be restricted to a pre-processing phase. Extensive works have been done to make predictions more realistic. The development of the wiplan software has been a part of the european project iPlan (IAPP-FP7 project) and has been integrated in NS-3 simulator.

5.3. FloPoCo

The purpose of the open-source FloPoCo project is to explore the many ways in which the flexibility of the FPGA target can be exploited in the arithmetic realm. FloPoCo is a generator of operators written in C++ and outputting synthesizable VHDL automatically pipelined to an arbitrary frequency. Among the known users of FloPoCo are U. Bristol,U. Cape Town, U.T. Cluj-Napoca, Imperial College, U. Essex, U. Madrid, U. P. Milano, T.U. Muenchen, T. U. Kaiserslautern, U. Paderborn, CalTech, U. Pernambuco, U. Perpignan, U. Tohoku, U. Tokyo, Virginia Tech U. and several companies.

In 2014, FloPoCo was enhanced with a generator of FIR filters accurate to the last bit [19] and several variants of the Atan2 function [46].

Web page: http://flopoco.gforge.inria.fr/

5.4. FIT/CorteXlab software

During the setting up of the FIT/CorteXlab platform, important software tools have been developped for the platform. The main tools is Minus which is used to deploy software programs on SDR hardware, it is developped in Python and is able to deploy complete configuration of NI USRP or Nutaq PicoSDR platforms. A seconf tools is DAS (*Automatic deployement system*) which is used to create the complete software environment of the servers of FIT/CorteXlab. This software could be use to create another testbed based on the same principle: hardware SDR nodes programmed from internet. These softwarea are currently used on the deployement testbed and on the production testbed.

SPIRALS Team

5. New Software and Platforms

5.1. APISENSE®

Participants: Clive Ferret-Canape, Julien Duribreux, María Gómez Lacruz, Nicolas Haderer, Christophe Ribeiro, Romain Rouvoy [correspondant], Antoine Veuiller.

In 2014, new developments have been made on our APISENSE® distributed crowdsensing platform. APISENSE® now builds on a distributed infrastructure hosted in the Cloud that can better cope with scalability issues in the number of experiments, users, and volume of data to be collected in the wild. Data collected by participants can be exposed to applications and stakeholders via an Open Data API, which provides the ability to build realtime web applications from crowdsourced datasets. The APISENSE® mobile app, named BEE, can be freely downloaded from the Google Play Store. APISENSE® is part of the results of the PhD thesis of Nicolas Haderer [12] that was defended in November 2014. In 2014, APISENSE® has also been at the core of an industrial transfer action that aims at creating a spin-off company. The project is managed by Christophe Ribeiro and Romain Rouvoy. The project has been accepted (so-called qualification) in 2014 by the Inria investment fund IT-Translation. The project in supported by Direction Transfert & Innovation which will fund in 2015 the 1-year engineer contract of Christophe Ribeiro for maturing the project.

APISENSE® is a distributed platform dedicated to crowdsensing activities. Crowdsensing intends to leverage mobile devices to seamlessly collect valuable dataset for different categories of stakeholders. APISENSE® intends to be used in a wide variety of scientific and industrial domains, including network quality monitoring, social behavior analysis, epidemy predictions, emergency crisis support, open maps initiatives, debugging of applications in the wide. APISENSE® is composed of BEE.HIVE delivered as a *Platform-as-a-Service* (PaaS) to the stakeholders who can pilot and customize their own crowdsensing environment [108], and *Bee.mob* supporting participants with a mobile application to control the sensors to be shared with the rest of the world [96], [97]. The platform is used by the Metroscope consortium, an Internet scientific observatory initiative supported by Inria.

APISENSE® is at the core of the Inria ADT Focus CrowdLab project (see Section 8.2).

Web site: http://www.apisense.fr. Registered with the APP (*Agence pour la Protection des Programmes*) under reference IDDN.FR.001.080006.000.S.P.2013.000.10000 is pending. License: Proprietary.

5.2. FraSCAti

Participants: Philippe Merle [correspondant], Fawaz Paraiso, Romain Rouvoy, Lionel Seinturier.

The novelty of 2014 consists in the development of the SOCLOUD platform for distributed multi-cloud systems. This platform has been defined in the context of the PhD thesis of Fawaz Paraiso [15] that was defended in June 2014. SOCLOUD is built on top of our existing FRASCATI platform. SOCLOUD enables to deploy, execute and manage an application that spans on several different cloud systems.

FRASCATI is a service-oriented component-based middleware platform implementing OASIS *Service Component Architecture* (SCA) specifications. The main originality of FRASCATI is to bring FRACTAL-based reflectivity to SCA, *i.e.*, any FRASCATI software component is equipped with both the SOA capabilities brought by SCA and the reflective capabilities (*i.e.*, introspection and reconfiguration) brought by FRACTAL. Various micro-benchmarks have shown that FRASCATI reflectivity is achieved without hindering its performance relative to the de facto reference SCA implementation, *i.e.*, Apache Tuscany. Non-functional concerns (logging, transaction, security, etc.), so-called intents in SCA terms, are also programmed as FRASCATI components and are (un)woven on business components dynamically at runtime, this is based on aspect-oriented concepts defined in FAC [110]. FRASCATI supports various implementation technologies (SCA Composite, Java, WS-BPEL, Spring Framework, OSGi, Fractal ADL, native C library, Apache Velocity templates, and seven scripting languages as BeanShell, FScript, Groovy, JavaScript, JRuby, Jython, XQuery) for programming services or integrating legacy code, various binding protocols (SOAP, REST, JSON-RPC, UPnP, HTTP servlets, Java RMI, JMS, JGroups) and interface definition languages (WSDL, Java, WADL) for interoperating with existing services. FRASCATI provides management tools like standalone, Web-based, and JMX-based graphical consoles and a dedicated scripting language for reconfiguring SCA applications. The whole FRASCATI platform is itself built as a set of reflective SCA components.

Inria Evaluation Committee Criteria for Software Self-Assessment: A-4-up, SO-4, SM-4-up, EM-3-up, SDL-4-up, DA-4, CD-4, MS-4, TPM-4. FRASCATI is a project of the OW2 consortium for open-source middleware. Web site: http://frascati.ow2.org. 292 Kloc (mainly Java). Registered with the APP (Agence pour la Protection des Programmes) under reference FR.001.050017.000.S.P.2010.000.10000. License: LGPL. Embedded into several industrial software systems: EasySOA, Petals Link EasyViper, EasyBPEL, EasyESB, OW2 PEtALS, OW2 Scarbo. Various demonstrators built during funded projects: ANR SCOrWare, FP7 SOA4All, ANR ITEmIS, ANR SALTY, ANR SocEDA, FUI Macchiato, FUI EasySOA, ADT Galaxy and ADT Adapt. Main publications: [117], [116], [103], [104], [93], [92].

5.3. PowerAPI

Participants: Maxime Colmant, Loïc Huertas, Adel Noureddine, Romain Rouvoy [correspondant].

In 2014, new developments have been made on our POWERAPI library for monitoring energy in software systems. POWERAPI now includes an accurate power model, which supports both DFVS, hyper threads and turbo boost features of modern processors. This model has been assessed on acknowledged benchmarks (PARSEC, SPEC CPU, SPECjbb) and is used as a basis to estimate the power consumption of applications running in virtualised environments. Finally, POWERAPI has evolved towards a modular toolkit that can be used to build software-defined power meters supporting a wide range of input sources (*e.g.*, hardware performance counters, RAPL, PowerSpy). POWERAPI is part of the results of the PhD thesis of Adel Noureddine [14] that was defended in March 2014.

POWERAPI is a Scala-based library for monitoring energy in software systems. It is based on a modular and asynchronous event-driven architecture using the Akka library. POWERAPI differs from existing energy process-level monitoring tool in its pure software, fully customizable and modular aspect which let users precisely define what they want to monitor, without plugging any external device. POWERAPI offers an API which can be used to express requests about energy spent by a process, following its hardware resource utilization (in terms of CPU, memory, disk, network, etc.). Its applications cover energy-driven benchmarking [105], [88], [86], [87], energy hotspots and bugs detection [106], [107], and real-time distributed system monitoring.

POWERAPI is at the core of the Inria ADT eSurgeon project (see Section 8.2).

Web site: http://www.powerapi.org. Registered with the APP (Agence pour la Protection des Programmes) under reference IDDN.FR.001.400015.000.S.P.2012.000.10000. License: AGPL.

5.4. Saloon

Participants: Laurence Duchien, Clément Quinton, Daniel Romero Acero, Lionel Seinturier [correspondant].

SALOON is a framework for the selection and configuration of Cloud providers according to application requirements. The framework enables the specification of such requirements by defining ontologies. Each ontology provides a unified vision of provider offers in terms of frameworks, databases, languages, application servers and computational resources (i.e., memory, storage and CPU frequency). Furthermore, each provider is related to a Feature Model (FM) with attributes and cardinalities, which captures its capabilities. By combining the ontology and FMs, the framework is able to match application requirements with provider capabilities and select a suitable one. Specific scripts to the selected provider are generated in order to enable its configuration.

SALOON is the result of the PhD thesis of Clément Quinton [16] that was defended in October 2014. SALOON is partially developed in the context of the FP7 PaaSage project (see Section 8.3).

Registered with the APP (Agence pour la Protection des Programmes) under reference IDDN.FR.001.300002.000.S.P.2014.000.10800.

5.5. Spoon

Participants: Martin Monperrus [correspondant], Gérard Paligot, Nicolas Petitprez.

In 2014, SPOON has been at the core of an industrial transfer action that aims at creating a spin-off company. The project is managed by Nicolas Petitprez and Martin Monperrus. The project has been accepted (so-called qualification) in 2014 by the Inria investment fund IT-Translation. The project in supported by Direction Transfert & Innovation which will fund in 2015 the 1-year engineer contract of Nicolas Petitprez for maturing the project. As an open source project Spoon has attracted new contributors in 2014. The Spoon development team is now composed of 8 active members, including 4 that are not at all related to Inria. Second, Spoon now supports analyzing and transforming Java 7 code, which is the now the dominant version of Java. Third, Spoon is the technical foundation of five important papers published in 2014. To sum up, year 2014 was a major year for warming up the Spoon project. Thanks to the support of Inria through the ADT, year 2015 is expected to be as vibrant and rich.

SPOON is a library for analyzing and transforming Java source code [76] [109]. SPOON provides a core API and associated tools for static analysis and generative programming within the Java 5+ environment. SPOON must be seen as a basis to ensure Software Quality through code validation and generation. It can be used in the software development process during the validation phase, as well as for engineering and re-engineering software. The first key point of SPOON is to provide a well-typed and comprehensive AST API which is designed to facilitate analysis and transformation work for programmers. Scanners and processors allow the programmer to implement various program traversal strategies on the Java program. Also, the program representation is built with a well-known and well-tested open source Java compiler: the Eclipse JDT compiler, which ensures the support of the latest Java features. The second key point of SPOON is to provide a pure Java API to specify program transformations using a well-typed generative programming technique (called Spoon Templates). By using well-typed templates, SPOON makes programming of transformations easier and safer for the end-user programmers.

SPOON is at the core of the Inria ADT Spoon3R project (see Section 8.1).

Web site: http://spoon.gforge.inria.fr. Registered with the APP (Agence pour la Protection des Programmes) under reference IDDN.FR.001.070037.000.S.P.2007.000.10600. License: CeCILL-C.

TACOMA Team

5. New Software and Platforms

5.1. THEGAME

Context-aware applications have to sense the environment in order to adapt themselves and provide with contextual services. This is the case of Smart Homes equipped with sensors and augmented appliances. However, sensors can be numerous, heterogeneous and unreliable. Thus the data fusion is complex and requires a solid theory to handle those problems. The aim of the data fusion, in our case, is to compute small pieces of context we call context attributes. Those context attributes are diverse and could be for example the presence in a room, the number of people in a room or even that someone may be sleeping in a room. For this purpose, we developed an implementation of the belief functions theory (BFT). THE GAME (THeory of Evidence in a lanGuage Adapted for Many Embedded systems) is made of a set of C-Libraries. It provides the basics of belief functions theory, computations are optimized for an embedded environment (binary representation of sets, conditional compilation and diverse algorithmic optimizations).

THE GAME is published under apache licence (https://github.com/bpietropaoli/THEGAME/). It is maintained and experimented by Aurélien Richez within a sensor network platform developed by TACOMA since June 2013.

TYREX Project-Team

5. New Software and Platforms

5.1. XML Reasoning Solver

Participants: Pierre Genevès, Nabil Layaïda, Nils Gesbert, Louis Jachiet, Nicola Guido.

The XML Reasoning Solver is a tool for the static analysis of queries and schemas based on our theoretical advances [9]. It allows automated verification of properties that are expressed as logical formulas over trees. A logical formula may for instance express structural constraints or navigation properties (like e.g. path existence and node selection) in finite trees.

The reasoner is built on top of a finite tree logic solver for a new modal logic equipped with recursion and backward axes. The solver is very fast in practice and uses symbolic techniques (Binary Decision Diagrams). The solver has been recently extended to support functions, parametric functions and polymorphic subtyping. One notable difficulty was to elaborate many advanced optimizations with symbolic implementation techniques. The logical solver significantly advances the state of the art. In particular, it is the first implementation that effectively solves the query containment problem for a large fragment of the XPath query language. It supports all navigation axes and regular tree constraints. Although researchers had studied XPath satisfiability before, such prior works were either unimplementable or deemed to explode even for tiny examples. As of 2014, it is still the only implementation actually capable of solving this problem in practice for real world instances.

The reasoner includes compilers and various static analyzers for web query and schema languages. This includes compilers for XPath, for XML schemas (DTDs, XML Schemas, Relax NGs) into logical formulas, parsers, benchmarks, and libraries for automated testing. Various difficulties reside in the compilation of real-world queries, including compiling XPath queries into fixed-point logics, developing specific implementation techniques in order to avoid worst case blow-ups as much as possible when e.g. supporting unordered XML attributes among (ordered) XML elements, etc. The reasoner also generates counter-examples that allow program defects to be reproduced independently from the analyzer.

The off-line version of the solver (with a native library) is fast and up-to-date with the latest advances. We developed and deployed an interactive web interface to make the solver available to the international scientific community. For this purpose, we redesigned the libraries used for the manipulation of binary decision diagrams (BDDs) so that they could used in a fully concurrent and multithreaded manner. This is in order to allow several instances of the logical solver to run concurrently for several users on a web server (GWT-based), while decreasing performance as less as possible.

The reasoner helps us to guide and validate our approach. We continue to develop, maintain and use it on an almost-daily basis.

5.2. XQuery type-checker

Participants: Pierre Genevès, Nabil Layaïda, Nils Gesbert.

This prototype implements a sound static type-system for XQuery, which, as of december 2014, is the most precise type system known for XQuery. It supports the static typing of backward axes that no other does nor is supported in the XQuery recommendation. It also includes precise typing for conditional statements which is challenging as such statement are usually sensitive to the program context. Our type checker successfully verifies complex programs for which existing type-checkers (either known from the literature or those developed in commercial software) fail by reporting false alarms. One major benefit is to allow the cost of validation to be deferred from runtime to compile-time (once only). This prototype is implemented in Scala and interacts with the solver by issuing externals calls for deciding complex subtyping relations. This prototype is described in preprint [20]

5.3. CSS Analyzer

Participants: Pierre Genevès, Nabil Layaïda, Marti Bosch Padros.

This software now consists in two distinct prototypes: two static analyzers (with a different purpose) that share a common compiler for CSS. The first prototype is used for bug detection and verification of a cascading style sheet (CSS) file. It involves a compiler for CSS rules (and in particular selectors) into logical formulas, adapted for the semantics of CSS (see the initial WWW'12 paper). The second prototype performs automated refactoring for size reduction of CSS style sheets. It reuses the first compiler and the logical solver for detecting which rules can be refactored and how. It implements various optimisation techniques (like early pruning), for the purpose of dealing with large-size real CSS files. This prototype reduces the size of CSS files found in the most popular websites (such as CNN, facebook, Google Sites, Apple, etc.) by up to 30% while preserving their semantics [13].

5.4. ClaireCourseMaker Library

Participants: Nicolas Hairon, Cécile Roisin, Nabil Layaïda.

The goal of the ClaireCourseMaker is to provide direct and visual editing tools for structuring, annotating and timeline-based authoring of continuous content such as audio or video. I it is mainly devoted to the synchronisation and layout of pedagogical material (video, slides, chaptering, etc.) and enables the incorporation of rich media content in MOOCs. The underlying technology is based on Web standards and relies on the open source JavaScript Popcorn library and Popcorn Maker web application developed by the Mozilla Foundation. The tool is a wysiwyg web-based authoring tool which benefits from the generic features of Popcorn and offers structuring methods such chaptering and container-based synchronisation.

ClaireCourseMaker is the direct follow-up tool of the Timesheet library developed in the project. Timesheet library is a cross-browser JavaScript implementation for scheduling the dynamic behaviour of HTML5 content. It uses and provides a reference implementation for declarative synchronisation markup such as SMIL Timing and Synchronization and SMIL Timesheets.

ClaireCourseMaker is developed in collaboration with the OpenClassrooms company in the context of the Claire project (see section 7.1.1).

5.5. Interactive eXtensible Engine (IXE)

Participants: Nabil Layaïda, Pierre Genevès, Thibaud Michel, Mathieu Razafimahazo.

PDRTrack is a localization utility running on iOS or Android smartphones used for recording and playing data sets (accelerometer, gyroscope, barometer and magnetometer values) to study the effect of different pedometer and map matching parameters on indoor and outdoor localization accuracy. This application uses the PDR library, written in C++, which provides the user's location in real time based on the interpretation of mobile phone sensors. Three main modules have been designed to build this localization system:

- a pedometer that estimates the distance the user has walked and his speed
- a motion manager that enables data set recording and simulation but also the creation of virtual sensors or filters (e.g gyroscope drift compensation, linear acceleration, altimeter)
- a map-matching algorithm that provides location estimates on a given OpenStreetMap description and the current user's trajectory

The PDR library is a central component of the VENTURI project. It has been used for applications such guiding a visually impaired people. Others partners have used this localisation system for retrieving a scale factor needed for the computer vision part (i.e SLAM).

GPS navigation systems, when used in an urban environment, are limited in precision and can only give instructions at the level of the street and not of the pavement or corridor. GPS is also limited to outdoor navigation and requires some transitioning system when switching to indoor navigation.

PDRTrack is embedded in IXE. IXE is an urban pedestrian navigation system based on Inertial Measurement Units (IMU) and running on mobile phones with onboard geographic data and a routing engine. IXE allows augmented reality queries on customised embedded geographical data. Queries on route nodes or POIs, on ways and relations are predefined for efficiency and quality of information. Following a web paradigm, IXE can be seen as web browser for XML documents describing navigation networks. by using the micro-format concept, one can define inside OpenStreetMap a complex format for pedestrian navigation networks allowing navigation at the level of pavements or corridors.

The big advantage of IXE is that it relies on a standard OpenStreetMap editor called JOSM to create navigation networks and augmented reality content. IXE browser reads OSM documents and produces from them visible or audible navigation information. IXE is composed of three engines, one for dead-reckoning navigation, one for interactive audio and the last one for Augmented Reality visual information.

URBANET Team

5. New Software and Platforms

5.1. WSNet

Participants: Rodrigue Domga Komguem, Quentin Lampin, Trista Lin, Alexandre Mouradian, Fabrice Valois (contact).

UrbaNet is an active contributor to WSnet (http://wsnet.gforge.inria.fr/), a discrete event simulator dedicated to large scale wireless networks developed and maintained by members of Inria and CITI lab. A major part of this contribution is represented by the implementation of state of the art protocols for medium access control and routing.

The WSNet simulation results obtained following this process are sometimes used as an input for another part of our development effort, which consists in prototype software based on the combination of CPLEX and AMPL for solving mixed integer linear programming problems with column generation.

5.2. TAPASCologne vehicular mobility dataset

Participants:Marco Fiore (contact), Diala Naboulsi, Razvan Stanica.

Based on the data made available by the Institute of Transportation Systems at the German Aerospace Center (ITS-DLR), the dataset aims at reproducing, with a high level of realism, car traffic in the greater urban area of the city of Cologne, Germany. To that end, different state-of-art data sources and simulation tools are brought together, so to cover all of the specific aspects required for a proper characterization of vehicular traffic:

- The street layout of the Cologne urban area is obtained from the OpenStreetMap (OSM) database;
- The microscopic mobility of vehicles is simulated with the Simulation of Urban Mobility (SUMO) software;
- The traffic demand information on the macroscopic traffic flows across the Cologne urban area (i.e., the O/D matrix) is derived through the Travel and Activity PAtterns Simulation (TAPAS) methodology;
- The traffic assignment of the vehicular flows described by the TAPASCologne O/D matrix over the road topology is performed by means of Gawron’s dynamic user assignment algorithm.

The resulting synthetic trace of the car traffic in a the city of Cologne covers a region of 400 square kilometers for a period of 24 hours, comprising more than 700.000 individual car trips. More information is available on the project website at http://kolntrace.project.citi-lab.fr/ .

5.3. PrivaMovApp

Participants:Djamel Benferhat, Patrice Raveneau, Hervé Rivano, Razvan Stanica (contact).

UrbaNet is leading the development of an Android application for user data collection purposes. The application is based on the Funf (http://www.funf.org/) framework, and is currently available on Google Play. A first deployment of the application, on 25 users, took place in December, at the ACM Middleware 2014 conference, in Bordeaux.

5.4. Sense in the City

Participants:Khaled Boussetta (contact), Hervé Rivano, Hamadoun Tall.

We are developing a lightweight experimentation platform for wireless sensor networks. The main objective of this platform is to be easily transferable and deployable on the field. It allows a simplified deployment of the code running on the sensors and the collection of logs generated by the instrumentation of the code on a centralized database. In the early stage of the platform, the sensors are powered by small PCs, e.g. Raspberry Pis, but we are investigating the integration of energy harvesting capabilities such as solar panels. First practical deployments of the platform will be used to showcase some protocols developed in the team in 2015.

WHISPER Team

5. New Software and Platforms

5.1. Platforms

5.1.1. Coccinelle

Our recent research is in the area of code manipulation tools for C code, particularly targeting Linux kernel code. This work has led to the Coccinelle tool that we are continuing to develop. Coccinelle serves both as a basis for our future research and the foundation of our interaction with the Linux developer community.

The need to find patterns of code, and potentially to transform them, is pervasive in software development. Examples abound. When a bug is found, it is often fruitful to see whether the same pattern occurs elsewhere in the code. For example, the recent Heartbleed bug in OpenSSL partly involves the same fragment of code in two separate files. [0] Likewise, when the interface of an API function changes, all of the users of that function have to be updated to reflect the new usage requirements. This generalizes to the case of code modernization, in which a code base needs to be adapted to a new compiler, new libraries, or a new coding standards. Finding patterns of code is also useful in code understanding, *e.g.*, to find out whether a particular function is ever called with a particular lock held, and in software engineering research, *e.g.*, to understand the prevalence of various kinds of code structures, which may then be correlated with other properties of the software. For all of these tasks, there is a need for an easy to use tool that will allow developers to express patterns and transformations that are relevant to their source code, and to apply these patterns and transformations to the code efficiently and without disrupting the overall structure of the code base.

To meet these needs, we have developed the Coccinelle program matching and transformation tool for C code. Coccinelle has been under development for over 7 years, and is mature software, available in a number of Linux distributions (Ubuntu, Debian, Fedora, etc.). Coccinelle allows matching and transformation rules to be expressed in terms of fragments of C code, more precisely in the form of a *patch*, in which code to add and remove is highlighted by using + and -, respectively, in the leftmost column, and other, unannotated, code fragments may be provided to describe properties of the context. The C language is extended with a few operators, such as metavariables, for abstracting over subterms, and a notion of positions, which are useful for reporting bugs. The pattern matching rules can interspersed with rules written in Python or OCaml, for further expressiveness. The process of matching patterns against the source code furthermore takes into account some semantic information, such as the types of expressions and reachability in terms of a function's (intraprocedural) control-flow graph, and thus we refer to Coccinelle matching and transformation specifications as *semantic patches*.

Coccinelle was originally motivated by the goal of modernizing Linux 2.4 drivers for use with Linux 2.6, and was originally validated on a collection of 60 transformations that had been used in modernizing Linux 2.4 drivers [8]. Subsequent research involving Coccinelle included a formalization of the logic underlying its implementation [1] and a novel mechanism for identifying API usage protocols [50]. More recently, Coccinelle has served as a practical and flexible tool in a number of research projects that somehow involve code understanding or transformation. These include identifying misuses of named constants in Linux code [52], extracting critical sections into procedures to allow the implementation of a centralized locking service [58], generating a debugging interface for Linux driver developers [31], detecting resource release omission faults in Linux and other infrastructure software [68], and understanding the structure of device driver code in our current DrGene project [70].

[0] http://git.openssl.org/gitweb/?p=openssl.git;a=commitdiff;h=96db902

Throughout the development of Coccinelle, we have also emphasized contact with the developer community, particularly the developers of the Linux kernel. We submitted the first patches to the Linux kernel based on Coccinelle in 2007. Since then, over 2000 patches have been accepted into the Linux kernel based on the use of Coccinelle, including around 700 by around 90 developers from outside our research group. Over 40 semantic patches are available in the Linux kernel source code itself, with appropriate infrastructure for developers to apply these semantic patches to their code within the normal `make` process. Many of these semantic are also included in a 0-day build-testing system for Linux patches maintained by Intel. [0] Julia Lawall was invited to the Linux Kernel Summit as a core attendee (invitation only) in 2010 and 2014, and has been invited to the internal 2014 SUSE Labs Conference. She has also presented Coccinelle at developer events such as LinuxCon Europe, Kernel Recipes (Paris), FOSDEM (Brussels), and RTWLS, and has supervised a summer intern financed by the Linux Foundation, as part of the GNOME Foundation's Outreach Program for Women.

Finally, we are aware of several companies that use Coccinelle for modernizing code bases. These include Metaware in Paris, with whom we have had a 5-month contract in 2013-2014 for the customization and maintenance of Coccinelle. We hope to be able to organize other such contracts in the future.

5.1.2. Better Linux

Over the past few years, Julia Lawall and Gilles Muller have designed and developed of a number of tools such as Coccinelle, Diagnosys [31] [30] and Hector [68], to improve the process of developing and maintaining systems code. The BtrLinux action aims to increase the visibility of these tools, and to highlight Inria's potential contributions to the open source community. We will develop a web site `https://BtrLinux.inria.fr`, to centralize the dissemination of the tools, collect documentation, and collect results. This action is supported by Inria by the means of a young engineer (ADT), Quentin Lambert. In the case of Coccinelle, we will focus on enhancing its visibility and its dissemination, by using it to find and fix faults in Linux kernel code, and by submitting the resulting patches to the Linux maintainers. We now present the other tools considered in the BtrLinux action in more detail.

Diagnosys is a hybrid static and dynamic analysis tool that first collects information about Linux kernel APIs that may be misused, and then uses this information to generate wrapper functions that systematically log at runtime any API invocations or return values that may reflect such misuse. A developer can then use a specific make-like command to build an executable driver that transparently uses these wrapper functions. At runtime, the wrappers write log messages into a crash resilient region of memory that the developer can inspect after any crash. Diagnosys is complementary to Coccinelle in the kind of information that it provides to developers. While Coccinelle directly returns a report for every rule match across the code base, often including false positives that have to be manually isolated by the developer, Diagnosys only reports on conditions that occur in the actual execution of the code. Diagnosys thus produces less information, but the information produced is more relevant to the particular problem currently confronting the developer. As such, it is well suited to the case of initial code development, where the code is changing frequently, and the developer wants to debug a specific problem, rather than ensuring that the complete code base is fault free. Diagnosys is a complete functioning system, but it needs to be kept up to date with changes in the kernel API functions. As part of the BtrLinux action, we will regularly run the scripts that collect information about how to create the wrappers, and then validate and make public the results.

Hector addresses the problem of leaking resources in error-handling code. Releasing resources when they are no longer needed is critical, so that adequate resources remain available over the long execution periods characteristic of systems software. Indeed, when resource leaks accumulate, they can cause unexpected resource unavailability, and even single leaks can put the system into an inconsistent state that can cause crashes and open the door to possible attacks. Nevertheless, developers often forget to release resources, because doing so often does not make any direct contribution to a program's functionality. A major challenge in detecting resource-release omission faults is to know when resource release is required. Indeed, the C language does not provide any built-in support for resource management, and thus resource acquisition and release are typically implemented using ad hoc operations that are, at best, only known to core developers.

[0] E.g., http://comments.gmane.org/gmane.linux.kernel.kbuild/269

Previous work has focused on mining sequences of such functions that are used frequently across a code base, [43], [56] but these approaches have very high rates of false negatives and false positives. [53] We have proposed Hector, a static analysis tool that finds resource-release omission faults based on inconsistencies in the operations performed within a single function, rather than on usage frequency. This strategy allows Hector to have a low false positive rate, of 23% in our experiments, while still being able to find hundreds of faults in Linux and other systems.

Hector was developed as part of the PhD thesis of Suman Saha and was presented at DSN 2013, where it received the William C. Carter award for the best student paper. Hector is complementary to Coccinelle, in that it has a more restricted scope, focusing on only one type of fault, but it uses a more precise static analysis, tailored for this type of fault, to ensure a low false positive rate. Hector, like Coccinelle, is also complementary to Diagnosys, in that it exhaustively reports on faults in a code base, rather than only those relevant to a particular execution, and is thus better suited for use by experienced developers of relatively stable software. Over 70 patches have been accepted into Linux based on the results of Hector. The current implementation, however, is somewhat in a state of disarray. As part of the BtrLinux action, we will first return the code to working condition and then actively use it to find faults in Linux. Based on these results, we will either submit appropriate patches to the Linux developers or notify the relevant developer when the corresponding fix is not clear.

ALICE Project-Team

5. New Software and Platforms

5.1. Vorpaline

Participants: Dobrina Boltcheva, Bruno Lévy, Thierry Valentin.

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

5.2. IceSL

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

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

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

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

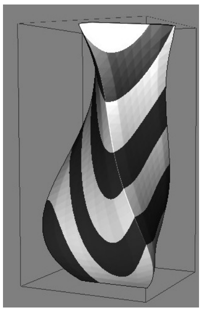


Figure 1. Left. A two-colored vase is modeled in IceSL. Right. An early printed result.

[0] Constructive Solid Geometry

5.3. Graphite

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

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

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

5.4. GraphiteLifeExplorer

Participant: Samuel Hornus.

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

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

5.5. OpenNL - Open Numerical Library

Participants: Bruno Lévy, Nicolas Ray, Rhaleb Zayer.

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

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

5.6. GEOGRAM

Participant: Bruno Lévy.

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

5.7. LibSL

Participant: Sylvain Lefebvre.

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

ALPAGE Project-Team

5. New Software and Platforms

5.1. Syntax

Participants: Pierre Boullier [correspondant], Benoît Sagot.

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

The (currently beta) version 6.0 of the SYNTAX system (freely available on Inria GForge) includes various deterministic and non-deterministic CFG parser generators. It includes in particular an efficient implementation of the Earley algorithm, with many original optimizations, that is used in several of Alpage's NLP tools, including the pre-processing chain SXPipe and the LFG deep parser SXLFG. This implementation of the Earley algorithm has been recently extended to handle probabilistic CFG (PCFG), by taking into account probabilities both during parsing (beam) and after parsing (n-best computation). SYNTAX 6.0 also includes parsers for various contextual formalisms, including a parser for Range Concatenation Grammars (RCG) that can be used among others for TAG and MC-TAG parsing.

In 2014, an in-depth rewriting of the RCG parser has started, in order for RCG parsers produced by SYNTAX to handle input DAGs while remaining efficient [60], although parsing time complexity might, on such inputs, become exponential w.r.t. their length, whereas RCGs exactly cover the set of languages that are parsable in polynomial time (if the input is a string).

Direct NLP users of SYNTAX for NLP, outside Alpage, include Alexis Nasr (Marseilles) and other members of the (now closed) SEQUOIA ANR project, Owen Rambow and co-workers at Columbia University (New York), as well as (indirectly) all SXPipe and/or SXLFG users. The project-team VASY (Inria Rhône-Alpes) is one of SYNTAX' user for non-NLP applications.

5.2. DyALog

Participant: Éric Villemonte de La Clergerie [maintainer].

DYALOG *on Inria GForge: http://dyalog.gforge.inria.fr/*

DYALOG provides an environment to compile and execute grammars and logic programs. It is essentially based on the notion of tabulation, i.e. of sharing computations by tabulating traces of them. DYALOG is mainly used to build parsers for Natural Language Processing (NLP). It may nevertheless be used as a replacement for traditional PROLOG systems in the context of highly ambiguous applications where sub-computations can be shared.

The current release of DYALOG (version 1.14.0) is freely available by FTP under an open source license and runs on Linux platforms for x86 and architectures and on Mac OS intel (both 32 and 64bits architectures). In particular, it has been ported for the CLANG/LLVM compiler used in recent Mac OS systems (Mavericks).

The current release handles logic programs, DCGs (Definite Clause Grammars), FTAGs (Feature Tree Adjoining Grammars), FTIGs (Feature Tree Insertion Grammars) and XRCGs (Range Concatenation Grammars with logic arguments). Several extensions have been added to most of these formalisms such as intersection, Kleene star, and interleave operators. Typed Feature Structures (TFS) as well as finite domains may be used for writing more compact and declarative grammars [135]. Version 1.14.0 now includes an efficient handler for feature-based statistical models, derived from the work on DYALOG-SR and now used in FRMG parser.

C libraries can be used from within DYALOG to import APIs (mysql, libxml, SQLite, ...).

DYALOG is largely used within ALPAGE to build parsers but also derivative softwares, such as a compiler of Meta-Grammars (cf. 5.3). It has also been used for building FRMG, a parser from a large coverage French TIG/TAG grammar derived from a Meta-Grammar. This parser has been used for the Parsing Evaluation campaign EASy, the two Passage campaigns (Dec. 2007 and Nov. 2009) [130], [134], and very large amount of data (700 millions of words) in the SCRIBO project. New results concerning FRMG are described in 6.5 .

DYALOG is also used to run DYALOG-SR, a transition-based dependency parser (see new results in 6.5)

DYALOG and other companion modules (including DYALOG-SR) are available on Inria GForge.

5.3. Tools and resources for Meta-Grammars

Participant: Éric Villemonte de La Clergerie [maintainer].

`mgcomp`, MGTOOLS, and FRMG on Inria GForge: http://mgkit.gforge.inria.fr/

DYALOG (cf. 5.2) has been used to implement `mgcomp`, Meta-Grammar compiler. Starting from an XML representation of a MG, `mgcomp` produces an XML representation of its TAG expansion.

The current version **1.5.0** is freely available by FTP under an open source license. It is used within ALPAGE and (occasionally) at LORIA (Nancy) and at University of Pennsylvania.

The current version adds the notion of namespace, to get more compact and less error-prone meta-grammars. It also provides other extensions of the standard notion of Meta-Grammar in order to generate very compact TAG grammars. These extensions include the notion of *guarded nodes*, i.e. nodes whose existence and non-existence depend on the truth value of a guard, and the use of the regular operators provided by DYALOG on nodes, namely disjunction, interleaving and Kleene star. The current release provides a dump/restore mechanism for faster compilations on incremental changes of a meta-grammars.

The current version of `mgcomp` has been used to compile a wide coverage Meta-Grammar FRMG (version 2.0.1) to get a grammar of around 200 TAG trees [132]. Without the use of guarded nodes and regular operators, this grammar would have more than several thousand trees and would be almost intractable. FRMG has been packaged and is freely available.

To ease the design of meta-grammars, a set of tools have been implemented, mostly by Éric Villemonte de La Clergerie, and collected in MGTOOLS (version **2.2.2**). This package includes a converter from a compact format to a XML pivot format, an Emacs mode for the compact and XML formats, a graphical viewer interacting with Emacs and XSLT stylesheets to derive HTML views.

The various tools on Metagrammars are available on Inria GForge. FRMG is used directly or indirectly (through a Web service or by requiring parsed corpora) by several people and actions (ANR Rhapsodie, ANR Chronoline, ...)

5.4. The Bonsai PCFG-LA parser

Participants: Marie-Hélène Candito [correspondant], Djamé Seddah, Benoit Crabbé.

Web page:
http://alpage.inria.fr/statgram/frdep/fr_stat_dep_parsing.html

Alpage has developed as support of the research papers [75], [67], [68], [122] a statistical parser for French, named Bonsai, trained on the French Treebank. This parser provides both a phrase structure and a projective dependency structure specified in [66] as output. This parser operates sequentially: (1) it first outputs a phrase structure analysis of sentences reusing the Berkeley implementation of a PCFG-LA trained on French by Alpage (2) it applies on the resulting phrase structure trees a process of conversion to dependency parses using a combination of heuristics and classifiers trained on the French treebank. The parser currently outputs several well known formats such as Penn treebank phrase structure trees, Xerox like triples and CONLL-like format for dependencies. The parsers also comes with basic preprocessing facilities allowing to perform elementary sentence segmentation and word tokenisation, allowing in theory to process unrestricted text. However it is believed to perform better on newspaper-like text.

The parser is available under a GPL license.

5.5. Alpage's linguistic workbench, including SxPipe and MElt

Participants: Benoît Sagot [correspondant], Kata Gábor, Marion Baranes, Pierre Magistry, Pierre Boullier, Éric Villemonte de La Clergerie, Djamé Seddah.

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

Alpage's linguistic workbench is a set of packages for corpus processing and parsing. Among these packages, two packages are of particular importance: the SxPipe pre-processing chain, and the MElt part-of-speech tagger.

SxPipe [109] is a modular and customizable chain aimed to apply to raw corpora a cascade of surface processing steps. It is used

- as a preliminary step before Alpage's parsers (e.g., FRMG);
- for surface processing (named entities recognition, text normalization, unknown word extraction and processing...).

Developed for French and for other languages, SxPipe includes, among others, various named entities recognition modules in raw text, a sentence segmenter and tokenizer, a spelling corrector and compound words recognizer, and an original context-free patterns recognizer, used by several specialized grammars (numbers, impersonal constructions, quotations...). It can now be augmented with modules developed during the former ANR EDyLex project for analysing unknown words; this involves in particular (i) new tools for the automatic pre-classification of unknown words (acronyms, loan words...) (ii) new morphological analysis tools, most notably automatic tools for constructional morphology (both derivational and compositional), following the results of dedicated corpus-based studies. New local grammars for detecting new types of entities and improvement of existing ones, developed in the context of the PACTE project, will soon be integrated within the standard configuration.

MElt is a part-of-speech tagger, initially developed in collaboration with Pascal Denis (Magnet, Inria — then at Alpage), which was trained for French (on the French TreeBank and coupled with the Le*fff*), also trained on English [79], Spanish [88], Italian [124], German [38], Dutch, Polish, Kurmanji Kurdish [138] and Persian [119], [120]. It is state-of-the-art for French. It is now able to handle noisy corpora (French and English only; see below). MElt also includes a lemmatization post-processing step. A preliminary version of MElt which accepts input DAGs has been developed in 2013, and is currently under heavy rewriting and improvement in the context of the PACTE project (see 6.3).

MElt is distributed freely as a part of the Alpage linguistic workbench.

In 2014, additional efforts have been achieved for a better pre-processing of noisy input text. This covers two different scenarios:

- user-generated content (see 6.2); two sets of tools are available for processing user-generated content: (i) very noisy computer-mediated content, such as found on social media, forums or blogs, are adressed within the MElt part-of-speech tagger via a three-step procedure (normalisation, tagging, de-normalisation with tag redistribution); this work is performed in relation with the CoMeRe project, funded by the Institut de Linguistique Française [14]; (ii) less noisy customer data, for preparing shallow semantic analysis; this work is performed in collaboration with the viavoo company [17].
- output of OCR systems, in the context of the PACTE project (see 6.3).

5.6. The Alexina framework: the Lefff syntactic lexicon, the Aleda entity database and other Alexina resources

Participants: Benoît Sagot [correspondant], Laurence Danlos.

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

Alexina is Alpage's Alexina framework for the acquisition and modeling of morphological and syntactic lexical information. The first and most advanced lexical resource developed in this framework is the Le*fff*, a morphological and syntactic lexicon for French.

Historically, the Le*fff* 1 was a freely available French morphological lexicon for verbs that has been automatically extracted from a very large corpus. Since version 2, the Le*fff* covers all grammatical categories (not just verbs) and includes syntactic information (such as subcategorization frames); Alpage's tools, including Alpage's parsers, rely on the Le*fff*. The version 3 of the Le*fff*, which has been released in 2008, improves the linguistic relevance and the interoperability with other lexical models.

Other Alexina lexicons exist, at various stages of development, in particular for Spanish (the Le*ff*e), Polish, Slovak, English, Galician, Persian, Kurdish, Italian, German, as well as for Latin verbs and a subset of Maltese and Khaling verbs. These lexicons are used in various tools, including instances of the MElt POS-tagger, and for studies in quantitative morphology.

Alexina also hosts *Aleda* [128], [118] a large-scale entity database currently developed for French but under development for English, Spanish and German, extracted automatically from Wikipedia and Geonames. It is used among others in the SxPipe processing chain and its NP named entity recognition, as well as in the Nomos named entity linking system.

5.7. The free French wordnet WOLF

Participants: Benoît Sagot [correspondant], Valérie Hanoka.

The WOLF (Wordnet Libre du Français) is a wordnet for French, i.e., a lexical semantic database. The development of WOLF started in 2008 [112], [113]. At this time, we focused on benefiting from available resources of three different types: general and domain-specific bilingual dictionaries, multilingual parallel corpora and Wiki resources (Wikipedia and Wiktionaries). This work was achieved in a large part in collaboration with Darja Fišer (University of Ljubljana, Slovenia), in parallel with the development of a free Slovene wordnet, sloWNet. However, it was also impacted by specific collaborations, e.g., on adverbial synsets [114].

In 2014, updated betas of the new version of the WOLF have been published (now version 1.0b4), which integrates and extends the various efforts performed and published somewhat independently in 2012, together with the result of additional filtering, both manual and semi-automatic.

The WOLF is freely available under the Cecill-C license. It has already been used in various experiments, within and outside Alpage.

5.8. OGRE (Optimized Graph Rewriting Engine)

Participants: Corentin Ribeyre [correspondant], Djamé Seddah, Éric Villemonte de La Clergerie, Marie-Hélène Candito.

OGRE (Optimized Graph Rewriting Engine) is a graph rewriting system specifically designed for manipulating linguistic trees and graphs [105]. It relies on a rule specification language for expressing graph rewriting patterns. The transformation is performed in two steps:

1. First, the system performs simple transformations following the rewriting patterns;
2. Second, constraints can be applied on edges, which applies transformations depending on their environment that are propagated while all constraints are satisfied.

The system has been designd for the analysis and manipulation of attributed oriented and multi-relational graphs.

Web site: http://www.corentinribeyre.fr/projects/view/OGRE

5.9. LexViz

Participants: Mikaël Morardo [maintainer], Éric Villemonte de La Clergerie.

In the context of the industrial collaboration of ALPAGE with the company Lingua & Machina, we have extended their WEB plateform Libellex with a new component used to visualize and collaboratively validate lexical resources. In particular, this extension is used to manage terminological lists and lexical networks. The implemented graph-based representation has proved to be intuitive and quite useful for navigating in such large lexical resources (on the order to 10K to 100K entries).

5.10. Mgwiki

Participants: Paul Bui Quang [maintainer], Éric Villemonte de La Clergerie.

In the context of Inria ADT Mgwiki, Paul Bui Quang has developed a linguistic wiki that may used to discuss linguistic phenomena with the possibility to add annotated illustrative sentences. The work is essentially devoted to the construction of an instance for documenting and discussing FRMG, with the annotations of the sentences automatically provided by parsing them with FRMG. This instance also offers the possibility to parse small corpora with FRMG and an interface of visualization of the results. Large parsed corpora (like French Wikipedia or Wikisource) are also available. The parsed corpora can also be queried through the use of the DPath language. The resulting wiki has been officially opened in 2014 on http://alpage.inria.fr/frmgwiki.

Another instance was deployed for managing the annotation guide for the Deep version of the Sequoia treebank, confirming the potential of the notion of linguistic wiki

The source code of the wiki is available on the GForge.

AVIZ Project-Team

5. New Software and Platforms

5.1. MakerVis

Participants: Sai Ganesh Swaminathan, Shi Conglei, Yvonne Jansen, Pierre Dragicevic [correspondant], Lora Oehlberg, Jean-Daniel Fekete.

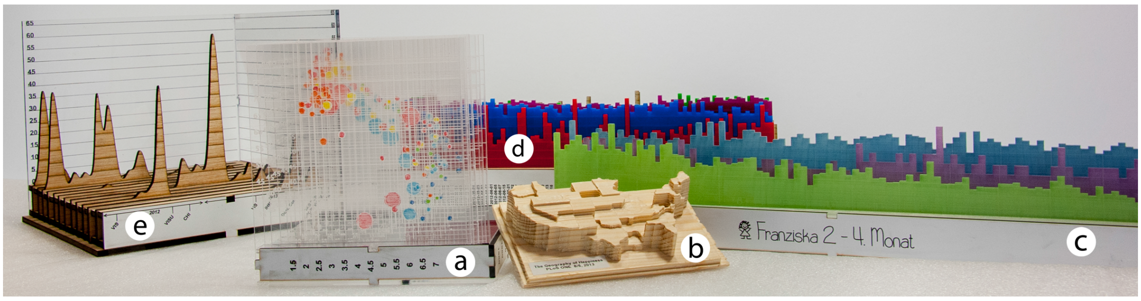


Figure 1. Physical visualizations created with our fabrication tool MakerVis: a) a scatterplot created after Hans Rosling's TED talk, b) a prism map showing happiness across the US computed from Twitter sentiments, c),d),e) visualizations created by users during design sessions.

An increasing variety of physical visualizations are being built, for purposes ranging from art and entertainment to business analytics and scientific research. However, crafting them remains a laborious process and demands expertise in both data visualization and digital fabrication. We created the MakerVis prototype [34], the first tool that integrates the whole workflow, from data filtering to physical fabrication. The design of MakerVis tries to overcome the limitations of current workflows, that we initially analyzed through three real case studies. Design sessions with three end users shows that tools such as MakerVis can dramatically lower the barriers behind producing physical visualizations. Observations and interviews also revealed important directions for future research. These include rich support for customization, and extensive software support for materials that accounts for their unique physical properties as well as their limited supply.

More details on the Web page: www.aviz.fr/makervis

5.2. Bertifier

Participants: Charles Perin, Pierre Dragicevic, Jean-Daniel Fekete.

Bertifier [20] is a web application (available at www.bertifier.com) for rapidly creating tabular visualizations from spreadsheets. Bertifier draws from Jacques Bertin's matrix analysis method, whose goal was to "simplify without destroying" by encoding cell values visually and grouping similar rows and columns. Although there were several attempts to bring this method to computers, no implementation exists today that is both exhaustive and accessible to a large audience. Bertifier remains faithful to Bertin's method while leveraging the power of today's interactive computers. Tables are formatted and manipulated through *crossets* [36], a new interaction technique for rapidly applying operations on rows and columns. Vertifier also introduces *visual reordering*, a semi-interactive reordering approach that lets users apply and tune automatic reordering algorithms in a WYSIWYG manner. We showed in an evaluation that Bertifier has the potential to bring Bertin's method to a wider audience of both technical and non-technical users, and empower them with data analysis and communication tools that were so far only accessible to a handful of specialists.

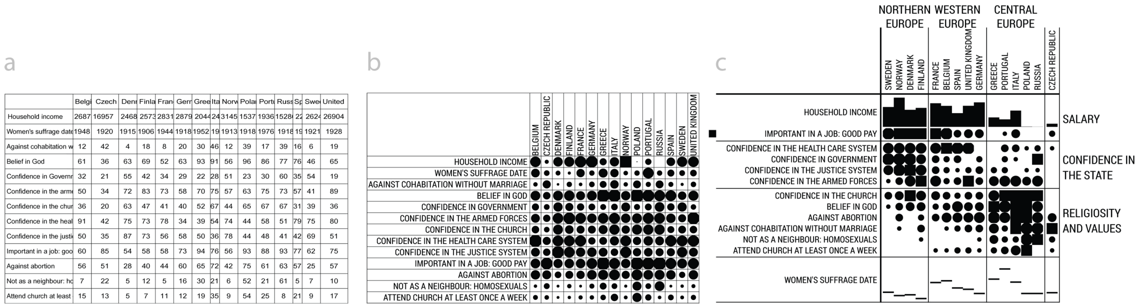


Figure 2. A spreadsheet formatted and reordered with BERTIFIER: a) the original numerical table; b) the corresponding tabular visualization; c) the final result, reordered, formatted and annotated. The final result is ready to be exported and inserted as a figure.

More details about the software are available at www.aviz.fr/bertifier

5.3. Sparklificator

Participants: Pascal Goffin, Wesley Willett, Jean-Daniel Fekete, Petra Isenberg.

Figure 3. Four examples of the integration of word-scale visualizations into HTML documents

Sparklificator [17] is a general open-source jQuery library that eases the process of integrating word-scale visualizations into HTML documents. It provides a range of options for adjusting the position (on top, to the right, as an overlay), size, and spacing of visualizations within the text. The library includes default visualizations, including small line and bar charts, and can also be used to integrate custom word-scale visualizations created using web-based visualization toolkits such as D3.

More on the project Web page: www.aviz.fr/sparklificator

5.4. GraphDiaries

Participants: Benjamin Bach [correspondant], Emmanuel Pietriga, Jean-Daniel Fekete.

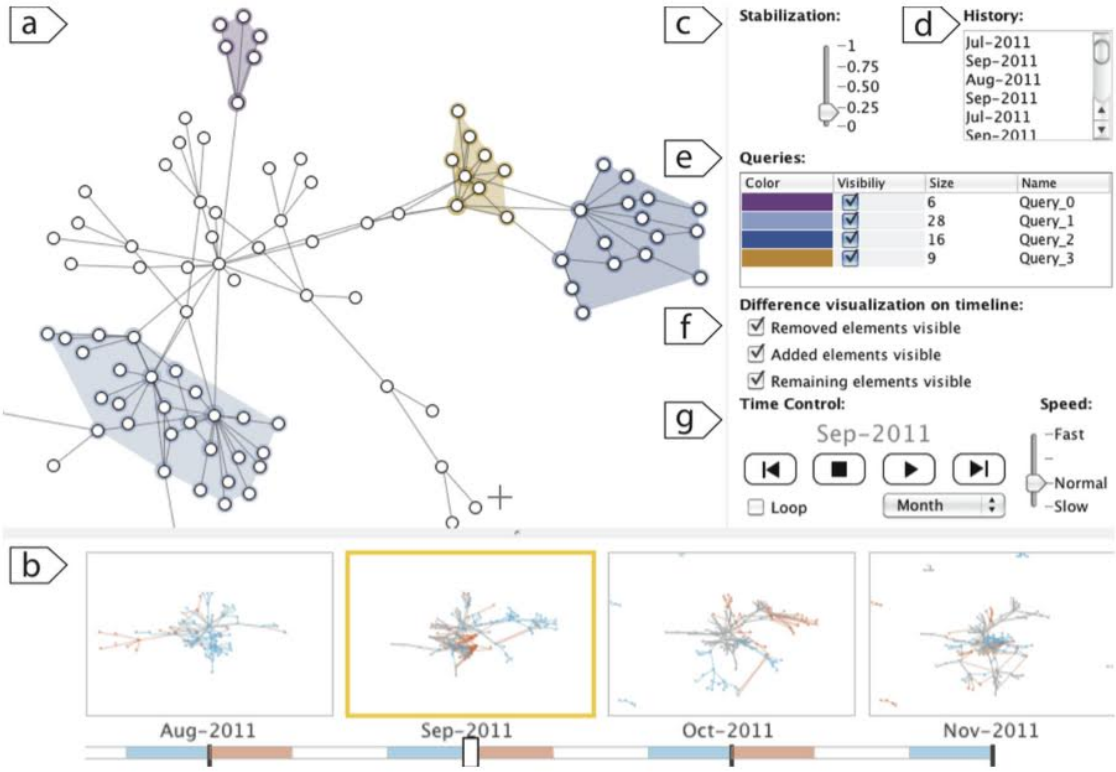


Figure 4. GraphDiaries interface: a) Network view, b) Time- line, c) Layout stabilization slider, d) Navigation history, e) Node queries, f) Panel to change visibility of red, blue or gray elements in the Timeline, g) Animation playback panel.

Identifying, tracking and understanding changes in networks that change over time, such as social networks, brain connectivy or migration flows, are complex and cognitively demanding tasks. To better understand the tasks related to the exploration of these networks, we introduced a task taxonomy which informed the design of GraphDiaries, [13], a new visual interface (Figure 4) designed to improve support for these tasks. GraphDiaries relies on animated transitions that highlight changes in the network between time steps, thus helping users identify and understand changes. GraphDiaries features interaction techniques to quickly navigate between individual time steps of the network. We conducted on a user study, based on representative tasks identified through the taxonomy, that compares GraphDiaries to existing techniques for temporal navigation in dynamic networks, showing that it outperforms them both in terms of task time and errors for several of these tasks.

5.5. Cubix

Participants: Benjamin Bach [correspondant], Emmanuel Pietriga, Jean-Daniel Fekete.

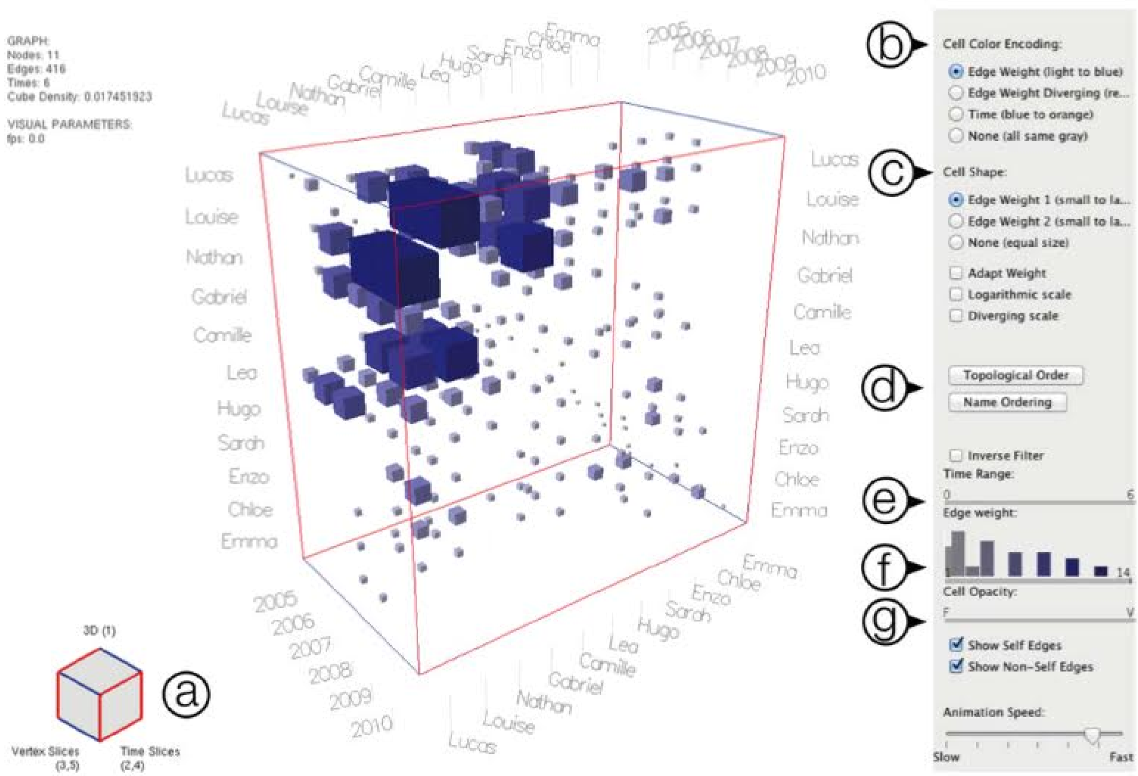


Figure 5. Cubix UI screenshot. a) Cubelet Widget, b) Cell color encod ing, c) Cell shape encoding, d) Vertex ordering, e) Time range slider, f) Cell weight filter with histogram indicating edge weight distribution, g) Cell opacity.

Designing visualizations of dynamic networks is challenging, both because the data sets tend to be complex and because the tasks associated with them are often cognitively demanding. Different tasks may require different visualizations and visual mappings, but combined in a simple interface. We developed Cubix [23] (Figure 5), a software featuring a novel visual representation and navigation model for dynamic networks, inspired by the way people comprehend and manipulate physical cubes. Users can change their perspective on the data by rotating or decomposing the 3D cube. These manipulations can produce a range of different 2D visualizations that emphasize specific aspects of the dynamic network suited to particular analysis tasks. A range of interactions can be performed on dynamic networks using the Cubix system. We showed how two domain experts, an astronomer and a neurologist, successfully used Cubix to explore and report on their own network data.

More on the project Web page: www.aviz.fr/cubix

5.6. EditorsNotes

Participants: Jean-Daniel Fekete [correspondant], Nadia Boukhelifa, Evanthia Dimara.

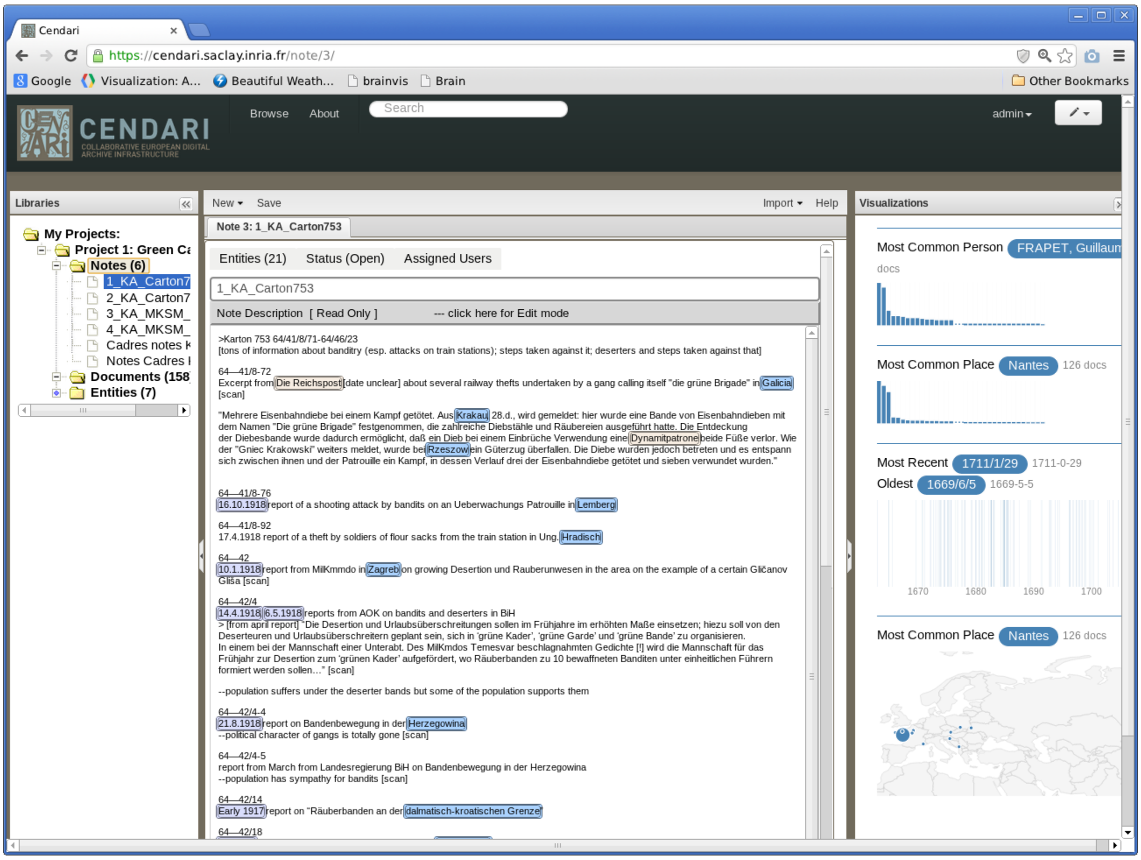


Figure 6. EditorsNotes environment with its three main panes: on the left, the list of projects, in the middle the editor and related documents, on the right the visualizations showing entities appearing in the current project.

CENDARI Is a European Infrastructure project funded by the EU for 4 years: 2012-2016. Aviz is in charge of the Human-Computer Interface for the project, and develops a tool to allow historians and archivists to take notes, enter them online, manage their images in relations with the notes and documents, and visualize the entities they find in the documents and notes. This system is an extension of the original EditorsNotes project, integrating several innovative components asked by the historians: visualizations, relations with the Semantic Web, and a management of access rights respecting the researchers' desire of privacy for their notes, as well as desire of sharing entities and relations gathered throught the notes and documents.

More on the project Web page: www.aviz.fr/Research/CENDARI

AYIN Team (section vide)

DAHU Project-Team

5. New Software and Platforms

5.1. New Software

5.1.1. The Webdamlog system

The Webdamlog system is a distributed knowledge management system. A new version of the system has been developed in collaboration with Drexel University (Prof. Julia Stoyanovich). The new version includes access control.

DREAM Project-Team

5. New Software and Platforms

5.1. Introduction

The pieces of software described in this section are prototypes implemented by members of the project. Any interested person should contact relevant members of the project.

5.2. Platforms

The Dream project-team, in collaboration with their applicative partners, has proposed and maintains several important software platforms for its main research topics.

5.2.1. Platform: Environmental decision-support systems

Participants: Marie-Odile Cordier, Christine Largouët, Véronique Masson, Yulong Zhao.

SACADEAU: the SACADEAU system is an environmental decision software (cf. 4.2) that implements the SACADEAU transfer model. The SACADEAU simulation model couples two qualitative models, a transfer model describing the pesticide transfer through the catchment and a management model describing the farmer decisions. Giving as inputs a climate file, a topological description of a catchment, and a cadastral repartition of the plots, the SACADEAU model simulates the application of herbicides by the farmers on the maize plots, and the transfer of these pollutants through the catchment until the river. The two main simulated processes are the runoff and the leaching. The output of the model simulation is the quantity of herbicides arriving daily to the stream and its concentration at the outlets. The originality of the model is the representation of water and pesticide runoffs with tree structures where leaves and roots are respectively up-streams and down-streams of the catchment.

The software allows the user to see the relationships between these tree structures and the rules learnt from simulations. A more elaborated version allows to launch simulations, to learn rules on-line and to access to two recommendation action algorithms. This year, we have developed a new visualization tool designed to compare two sets of rules learnt from simulations. The user can choose one (or more) rule(s) to compare from one set of rules, and a distance to apply from several multidimensional distances. The most similar rules in the second set of rules are found and the comparison can be easily visualized. The examples covered by "similar" rules can also be presented to the user by highlighting shared positive and negative covered examples. The software is mainly in Java.

The following website is devoted to the presentation of the SACADEAU: http://www.irisa.fr/dream/SACADEAU/. See also [57] for a presentation.

ECOMATA: The ECOMATA tool-box provides means for qualitative modeling and exploring ecosystems and for aiding to design environmental guidelines.We have proposed a new qualitative approach for ecosystem modeling (cf. 4.2) based on timed automata (TA) formalism combined to a high-level query language for exploring scenarios.

To date, ECOMATA is dedicated to ecosystems that can be modeled as a collection of species (prey-predator systems) under various human pressures and submitted to environmental disturbances. It has two main parts: the Network Editor and the Query Launcher. The Network Editor let a stakeholder describe the trophic food web in a graphical way (the species icons and interactions between them). Only few ecological parameters are required and the user can save species in a library. The number of qualitative biomass levels is set as desired. An efficient algorithm generates automatically the network of timed automata. EcoMata provides also a dedicated window to help the user define different fishing pressures, a nice way being by using chronograms. In the Query Launcher, the user selects the kind of query and the needed parameters (for example the species biomass levels to define a situation). Results are provided in a control panel or in files that can be exploited later.

Several additional features are proposed in EcoMata: building a species library, import/export of ecosystem model, batch processing for long queries, etc. EcoMata is developed in Java (Swing for the GUI) and the model-checker called for the timed properties verification is UPPAAL.

The following website is devoted to the presentation of ECOMATA: http://oban.agrocampus-ouest.fr:8080/ecomata.

PATURMATA: The Paturmata tool-box provides means for qualitative modeling and exploring agrosystems, specifically management of herd based on pasture [5]. The system is modelled using a hierarchical hybrid model described in timed automata formalism.

In PaturMata software, users can create a pasture system description by entering herds and plots information. For each herd, the only parameter is the number of animals. For each plot, users should enter the surface, the density, the herb height, the distance to the milking shed, a herb growth profile and an accessibility degree.

Users then specify pasturing and fertilization strategies. Finally, users can launch a pasture execution. PaturMata displays the results and a detailed trace of pasture. Users can launch a batch of different strategies and compare the results in order to find the best pasture strategy.

PaturMata is developed in Java (Swing for the GUI) and the model-checker that is called for the timed properties verification is UPPAAL.

Another feature which will be soon added to PaturMata is strategy synthesis. Users choose a pasture configuration or a type of pasture configuration and PaturMata proposes the best pasture and fertilization strategy in order to minimize the pasture procedure cost and use of nitrogen fertilizer.

5.2.2. Platform: Pattern Mining

Participants: Thomas Guyet, René Quiniou.

QTempIntMiner: the QTEMPINTMINER (Quantitative Temporal Interval Miner) data mining software implements several algorithms presented in [46] and [3] (QTIAPRIORI and QTIPREFIXSPAN). The software is mainly implemented in Matlab. It uses the Mixmod toolbox [33] to compute multi-dimensional Gaussian distributions. The main features of QTEMPINTMINER are:

- a tool for generating synthetic noisy sequences of temporal events,
- an implementation of the QTEMPINTMINER, QTIAPRIORI and QTIPREFIXSPAN algorithms,
- a graphical interface that enables the user to generate or import data set and to define the parameters of the algorithm and that displays the extracted temporal patterns.
- a sequence transformer to process long sequences of temporal events. Long sequences are transformed into a database of short temporal sequences that are used as input instances for the available algorithms.

The software includes one new algorithm based on the separation of the set of interval to extract more efficiently but less accurately the time interval in temporal patterns. This new algorithm version is still under evaluation on simulated and real datasets (care pathways).

The following website gives many details about the algorithms and provides the latest stable implementation of QTEMPINTMINER: http://www.irisa.fr/dream/QTempIntMiner/.

5.2.3. Platform: Diagnostic and Monitoring Systems

Participants: Marie-Odile Cordier, René Quiniou, Sophie Robin.

Odisseptale: the Odisseptale software implements disease detectors using monitoring of data provided by sensors placed on calves or cows. Sensors record streams of data such as body temperature, physical activity, feeding behavior, etc. These data are transmitted regularly to a monitoring software that aims to detect if a noticeable change has occurred on the data streams. Several detectors can be simultaneously active and each contribute to the final decision (detection of a disease). Two kinds of detectors have been implemented: a generic detector based on adaptive CUSUM and a symbolic pattern-based detector. Odisseptale provides also facilities for parameter setting and performance evaluation.

ManageYourself: the ManageYourself software comes from a collaborative project between Dream and the Telelogos company aiming at monitoring smartphones from a stream of observations made on the smartphone state.

Today's smartphones are able to perform calls, as well as to realize much more complex activities. They are small computers. But as in computers, the set of applications embedded on the smartphone can lead to problems. The aim of the project ManageYourself is to monitor smartphones in order to avoid problems or to detect problems and to repair them.

The ManageYourself application includes three parts :

- A monitoring part which triggers preventive rules at regular time to insure that the system is working correctly, e.g. *if the memory is full then delete the tmp directory*. This part is always running on the smartphone.
- A reporting part which records regularly the state of the smartphone (the memory state - free vs allocated -, the connection state, which applications are running, etc.). This part also is always running on the smartphone. The current state is stored in a report at regular period and is labeled *normal*. When an application or the system bugs, the current buggy state is stored in a report and is labeled *abnormal*. At regular timestamps, all the reports are sent to a server where the learning process is executed.
- A learning part which learns new bug rules from the report dataset. This part is executed offline on the server. Once the bug rules are learnt, human experts translates them into preventive rules which are downloaded and integrated in the monitoring part of the smartphones.

5.3. TraceSquiz: reduction of captured trace volume

Participants: Serge Vladimir Emteu Tchagou, Alexandre Termier.

TraceSquiz is a software developped in collaboration with STMicroelectronics. Its goal is to reduce the volume of execution trace captured during endurance tests of multimedia applications. It uses anomaly detection techniques to "learn" regular parts of the trace and only capture the irregular ones. The software is written in C++.

E-MOTION Project-Team

4. New Software and Platforms

4.1. PROTEUS Software

Participants: Amaury Nègre, Juan Lahera-Perez.

This toolkit offers a automatic mobile robot driver, some sensors drivers sensors as Sick laser, GPS, motion tracker, mono or stereo camera), and a 3D Simulator.

The latest developments have been focuses on the robotics simulator. This simulator is based on the simulation and 3D rendering engine "mgEngine" (http://mgengine.sourceforge.net/) embedded with the physics engine "bullets physics" (http://bulletphysics.org) for realistic robot dynamic simulation.

We also worked on the interface with the robotics middleware "ROS" (http://www.ros.org) in order to offer interoperability with many robotics applications.

The simulator is now fully integrated with the robotics middleware "ROS" (http://www.ros.org) which allow interoperability with a large set of robotics applications and visualization tools. This software is developed in C++ and the simulator operates with the Lua scripting language. The simulation software is used in the ANR Proteus (http://www.anr-proteus.fr), as a simulation engine for the PROTEUS Toolkit.

- Version: 2.0
- APP:IDDN.FR.001.510040.000.S.P.2005.000.10000
- Programming language: C/C++, Lua

4.2. AROSDYN

Participants: Igor Paromtchik, Mathias Perrollaz, Alexandros Makris, Amaury Nègre, Christian Laugier.

ArosDyn (http://arosdyn.gforge.inria.fr/) is a system which integrates our recently developped techniques to provide a real-time collision risk estimation in a dynamic environment. The main features of this software are:

1. The design provides high maintainability, scalability and reuseness of the models and algorithms.
2. The software has a user interface (UI) which is user-friendly.
3. The software facilitates the parameter tuning of the models.
4. It uses the GPU to accelerate the computation.
5. Working together with the Hugr middleware (http://gforge.inria.fr/projects/cycabtk), it can run on our experimental vehicle in real-time.

Another important property of this software is a large part of the computation task executed on GPU. As the processing of stereo image and the computaion in the BOF can be highly parallelized, we run these tasks on the GPU to improve the time performance. The GPU calculation is based on CUDA library and is carried out in an independent thread.

Furthermore, thanks to the design of the software, we can easily add new models to it and let them work together. The fast detection and tracking algorithm (FCTA) and the Gaussian process based collision assessment algorithm are added into this framework. The sofware is implemented on the Lexus car. In 2012, a demand for deposing the GPU BOF software to the APP is in progress.

4.3. Embedded Perception

Participants: Mathias Perrollaz, Amaury Nègre, Christian Laugier.

The method for computing occupancy grids from a stereoscopic sensor, developped in the e-motion team, has been implemented on GPU, using NVIDIA CUDA. This allows a real time implementation and an online processing within the Lexus experimental platform.

The program has been deposed to the APP in 2012, under the reference: IDDN.FR.001.270004.000.S.P.2012.000.10800

4.4. Bayesian Occupancy Filter

People involved: Kamel Mekhnacha, Tay Meng Keat Christopher, C. Laugier, M. Yguel, Pierre Bessière.
The BOF toolbox is a C++ library that implements the Bayesian Occupancy Filter. It is often used for modelling dynamic environments. It contains the relevant functions for performing bayesian filtering in grid spaces. The output from the BOF toolbox are the estimated probability distributions of each cell's occupancy and velocity. Some basic sensor models such as the laser scanner sensor model or Gaussian sensor model for gridded spaces are also included in the BOF toolbox. The sensor models and BOF mechanism in the BOF toolbox provides the necessary tools for modelling dynamic environments in most robotic applications. This toolbox is patented under two patents : "Procédé d'assistance à la conduite d'un véhicule et dispositif associé" n. 0552735 (9 september 2005) and "Procédé d'assistance à la conduite d'un véhicule et dispositif associé amélioré" n. 0552736 (9 september 2005) and commercialized by ProBayes.

- Version: 1
- Patent: 0552736 (2005), 0552735 (2005)
- Programming language: C/C++

4.5. PROBT

People involved: Juan-Manuel Ahuactzin, Kamel Mekhnacha, Pierre Bessière, Emmanuel Mazer, Manuel Yguel, Christian Laugier.
ProBT is both available as a commercial product (ProBAYES.com) and as a free library for public research and academic purposes (http://emotion.inrialpes.fr/BP/spip.php?rubrique6). Formerly known as *OPL*, *ProBT* is a C++ library for developing efficient Bayesian software. It is available for Linux, Unix, PC Windows (Visual C++), MacOS9, MacOSX and Irix systems. The ProBT library (http://www.probayes.com/) has two main components: (i) a friendly Application Program Interface (API) for building Bayesian models, and (ii) a high-performance Bayesian Inference Engine (BIE) allowing to execute all the probability calculus in exact or approximate way. *ProBT* is now commercialized by our start-up *Probayes*; it represents the main Bayesian programming tool of the *e-Motion* project-team, and it is currently used in a variety of external projects both in the academic and industrial field (e.g., for the European project BACS and for some industrial applications such as Toyota or Denso future driving assistance systems).

EXMO Project-Team

5. New Software and Platforms

5.1. Alignment API

Participants: Jérôme Euzenat [Correspondent], Jérôme David, Nicolas Guillouet, Armen Inants, Luz Maria Priego-Roche.

We have designed a format for expressing alignments in a uniform way [1]. The goal of this format is to share available alignments on the web. It should help systems using alignments, e.g., mediators, translators, to take advantage of any matching algorithm and it will help matching algorithms to be used in many different tasks. This format is expressed in RDF, so it is freely extensible.

The API itself [1] is a JAVA description of tools for accessing the common format. It defines five main interfaces (OntologyNetwork, Alignment, Cell, Relation and Evaluator).

We provide an implementation for this API which can be used for producing transformations, rules or bridge axioms independently from the algorithm which produced the alignment. The proposed implementation features:

- a base implementation of the interfaces with all useful facilities;
- a library of sample matchers;
- a library of renderers (XSLT, RDF, SKOS, SWRL, OWL, C-OWL, SPARQL);
- a library of evaluators (various generalisation of precision/recall, precision/recall graphs);
- a flexible test generation framework which allows for generating evaluation datasets;
- a library of wrappers for several ontology API;
- a parser for the format.

To instanciate the API, it is sufficient to refine the base implementation by implementing the `align()` method. Doing so, the new implementation will benefit from all the services already implemented in the base implementation.

In 2014, the Alignment API integrated an implementation of link keys (§6.3.4) and transformations of these into SPARQL queries.

We have developed, on top of the Alignment API, an Alignment server that can be used by remote clients for matching ontologies and for storing and sharing alignments. It is developed as an extensible platform which allows to plug-in new interfaces. The Alignment server can be accessed through HTML, web service (SOAP and REST) and agent communication interfaces. It has been used this year in the Ready4SmartCities project (§7.2.1.1).

The Alignment API is used in the Ontology Alignment Evaluation Initiative data and result processing (§6.2.1). It is also used by more than 50 other teams worldwide.

The Alignment API is freely available since december 2003, under the LGPL licence, at http://alignapi.gforge.inria.fr.

5.2. The OntoSim library

Participants: Jérôme David [Correspondent], Jérôme Euzenat.

OntoSim is a library offering similarity and distance measures between ontology entities as well as between ontologies themselves. It materialises our work towards better ontology proximity measures.

There are many reasons for measuring a distance between ontologies. For example, in semantic social networks, when a peer looks for particular information, it could be more appropriate to send queries to peers having closer ontologies because it will be easier to translate them and it is more likely that such a peer has the information of interest. OntoSim provides a framework for designing various kinds of similarities. In particular, we distinguish similarities in the ontology space from those in the alignment space. The latter ones use available alignments in an ontology network while the former only rely on ontology data. OntoSim is provided with 4 entity measures which can be combined using various aggregation schemes (average linkage, Hausdorff, maximum weight coupling, etc.), 2 kinds of vector space measures (boolean and TFIDF), and 4 alignment space measures. It also features original comparison methods such as agreement/disagreement measures. In addition, the framework embeds external similarity libraries which can be combined to our own.

OntoSim is based on an ontology interface allowing for using ontology parsed with different APIs. It is written in JAVA and is available, under the LGPL licence, at http://ontosim.gforge.inria.fr.

FLOWERS Project-Team

5. New Software and Platforms

5.1. Perception Tools

Participants: David Filliat [correspondant], Louis-Charles Caron, Alexander Gepperth.

5.1.1. Of 3-D point cloud

Participants: Louis-Charles Caron [correspondant], Alexander Gepperth, David Filliat.

This software scans the 3-D point cloud of a scene to find objects and match them against a database of known objects. The process consists in 3 stages. The segmentation step finds the objects in the point cloud, the feature extraction computes discriminating properties to be used in the classification stage for object recognition.

The segmentation is based on simple assumptions about the geometry of an indoor scene and the movement of a wheeled mobile robot. The floor plane coefficients are known a priori and are used to eliminate from the point cloud all points that are close to this plane and have a normal perpendicular to it. The floor plane coefficients also allow the detection of walls. Successive RANSACs are run to find planes that are perpendicular to the floor plane, and contain a large number of points. With these large structural regions removed, the only points remaining in the point cloud are the objects in the scene. These objects are separated by clustering the points based on a distance criteria. Close-by points are considered to form a single object.

Objects are characterized by their shape, texture. The texture information is encoded as a histogram that apprioximates the form of the distribution of color values in the object. A separate histogram is built for the red, green and blue channels. The shape of an object is encoded by computing thousands of randomly chosen Surflet-pair relation features and compiing them into a histrogram of occurrence.

The classification is done by a 3-layer feed-forward neural network. The network is trained on a dataset of point clouds of 53 objects. After training, the neural network is run on the features computed from each object detected in the segmentation stage [86].

5.1.2. PEDDETECT: GPU-accelerated person detection demo

Participant: Alexander Gepperth [correspondant].

PEDDETECT implements real-time person detection in indoor or outdoor environments. It can grab image data directly from one or several USB cameras, as well as from pre-recorded video streams. It detects mulitple persons in 800x600 color images at frame rates of >15Hz, depending on available GPU power. In addition, it also classifies the pose of detected persons in one of the four categories "seen from the front", "seen from the back", "facing left" and "facing right". The software makes use of advanced feature computation and nonlinear SVM techniques which are accelerated using the CUDA interface to GPU programming to achieve high frame rates. It was developed in the context of an ongoing collaboration with Honda Research Institute USA, Inc.

5.1.3. A Python OptiTrack client

Participant: Pierre Rouanet [correspondant].

This python library allows you to connect to an OptiTrack from NaturalPoint (http://www.naturalpoint.com/optitrack/). This camera permits the tracking of 3D markers efficiently and robustly. With this library, you can connect to the Motive software used by the OptiTrack and retrieve the 3D position and orientation of all your tracked markers directly from python.

5.2. Datasets

5.2.1. Assemblies of objects for the 3rd hand project

Participants: Yoan Mollard [correspondant], Thibaut Munzer, Manuel Lopes.

The 3rd hand project aims to develop a semi-autonomous robot assistant that acts as a third hand of a human worker. Especially, both should be able to undertake assembly tasks together, in a cooperative way. In order to analyse assembly tasks we recorded 6 datasets of two objects being assembled by a human. The experiment setup has the form of a single user assembling simple furnitures (a chair and a bench) composed by several distinct parts (seating, back, legs). Each part is tracked thanks to an Optitrack system and a set of reflective markers during the whole assembly. The experimental setup records the absolute poses of each part (position and orientation) and relative poses of each couple of objects.

5.3. Learning algorithms

5.3.1. KidLearn

Participants: Manuel Lopes [correspondant], Benjamin Clement, Pierre-Yves Oudeyer, Didier Roy.

The KidLearn software provides an Intelligent Tutoring System that optimizes teaching sequences based on the estimated level of each particular student. Two algorithms, RiARiT and ZPDES have been developpped and are described in [37], [39] and [38]. We updated the Game of Money that we developped last year wich allows students between 7-8 years to learn how to use money. It still includes 3 main components: i) a webserver that handles the requests and stores the experiments in a databased; ii) a GUI that provides the interface for the game; and iii) the optimization software.

Graphical interfaces in ITS can have unwanted side effects. For this reason, the interface was entirely designed with the help of a didactician, with several specific design choices motivated by pedagogical, motivational and attention requirements. For example, the interface, shown in Figure 1 . is such that:

- display is as clear and simple as possible;
- there is no chronometer, so that students are not put under time pressure;
- coins and banknotes have realistic visual appearance, and their relative sizes are respected;
- costumer and merchant are represented to indicate clearly the role of the student;
- text quantity is kept to minimum;

Four principal regions are defined in the graphical interface, as shown in Figure 1 , on the left picture. The first is the wallet location where users can pick and drag the money items and drop them on the repository location to compose the correct price. The object and the price are present in the object location.

Figure 1. Interface with two exemple of type of exercises, Left: customer/one object, Right : merchant/two objects

We performed a more developed and complete user study than last year, considering 5 different schools in the Bordeaux metropolitan area. We had a total of 400 students between 7 and 8 years old. We divided them into 4 groups, with one control group where student does not use the software and 3 groups where exercises are proposed using : a) a predefined sequence; b) ZPDES; c) RiARiT. To measure student learning, students pass pre-test few days before using the interface, and a post test fews days after using the interface. The control group pass the pre and post test at the same time that others but without using the interface between. The results of this study have been presented in [69].

5.3.2. DMP-BBO Matlab library

Participant: Freek Stulp [correspondant].

The dmp_bbo (Black-Box Optimization for Dynamic Movement Primitives) Matlab library is a direct consequence of the insight that black-box optimization outperforms reinforcement learning when using policies represented as Dynamic Movement Primitives. It implements several variants of the PI^{BB} algorithm for direct policy search. It is currently being used and extended by several FLOWERS members (Manuel Lopes, Clément Moulin-Frier) and external collaborators (Jonas Buchli, Hwangbo Jemin of ETH Zurich). In the context of the DIGITEO-funded project “PrActIx” , CEA LIST has now started using this library. In 2014, parts have been made real-time safe for use on the Meka Humanoid robot. This has been fundamental in achieving the results for [65], [64].

5.3.3. Self-calibration BCI - Matlab library

Participants: Jonathan Grizou [correspondant], Iñaki Iturrate, Luis Montesano, Manuel Lopes, Pierre-Yves Oudeyer.

The Matlab software implements the algorithms described in [45]. Downloadable from https://github.com/jgrizou/lfui.

It allows a robot to be instructed a new task by a human using communicative signals initially totally unknown to the robot. It is was extended and improved in the context of EEG-based brain-machine interfaces (BMIs) [44].

It results in a BCI based control of sequential tasks with feedback signals that do not require any calibration process. As a by-product, the method provides an unsupervised way to train a decoder with the same performance than state-of-the-art supervised classifiers, while keeping the system operational and solving, with a lower performance during the first steps, the unknown task. The algorithm has been tested with online experiments (fig. 2), showing that the users were able to guide from scratch an agent to a desired position.

To improve the efficiency of the algorithm, we introduced a new planning method that uses the uncertainty in the signal-target estimation. This planner is inspired by exploration methods with exploration bonuses that allow guiding to reduce the uncertainty in an efficient way. We showed that trying to follow the best hypothesis does not explore the space significantly to reduce uncertainty and thus identify the correct task. Only through an approach that plans how to reduce the uncertainty multiple steps ahead are we sure that the agent will reach states that can only be explained by the correct hypothesis.

5.3.4. DyNAMoS: parallel multi-process simulation of distributed neural architectures

Participants: Alexander Gepperth [correspondant], Mathieu Lefort.

This simulation software comes in the form of a PYTHON module and allows a user to define and simulate complex neural architectures while making use of the parallelism inherent to modern multi-core processors. A special focus lies on on-line learning, processing inputs one by one, in contrast to batch processing of whole databases at a time.

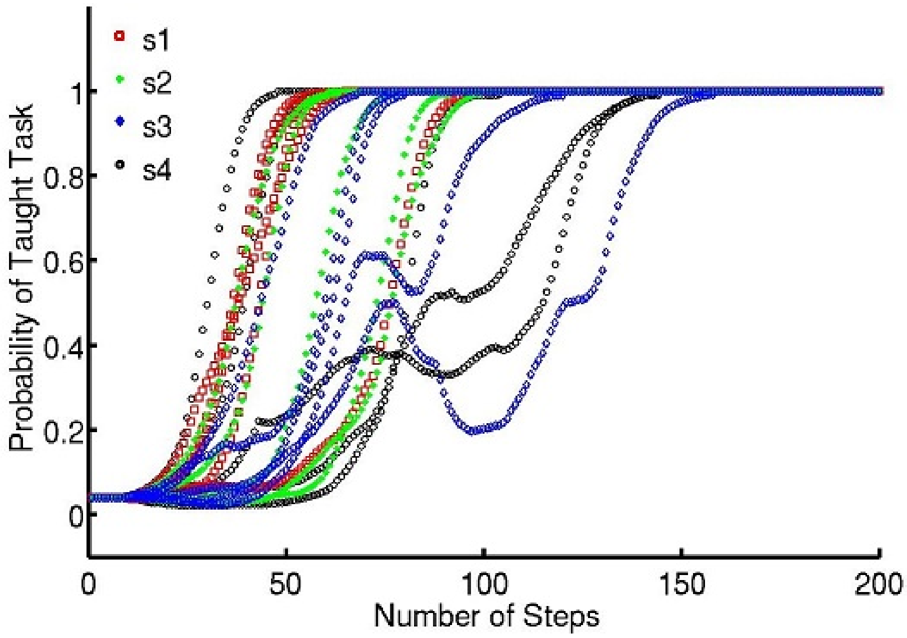


Figure 2. Results from the online BCI experiment for identifying the task. Evolution of the probability of the taught task for each subject and run

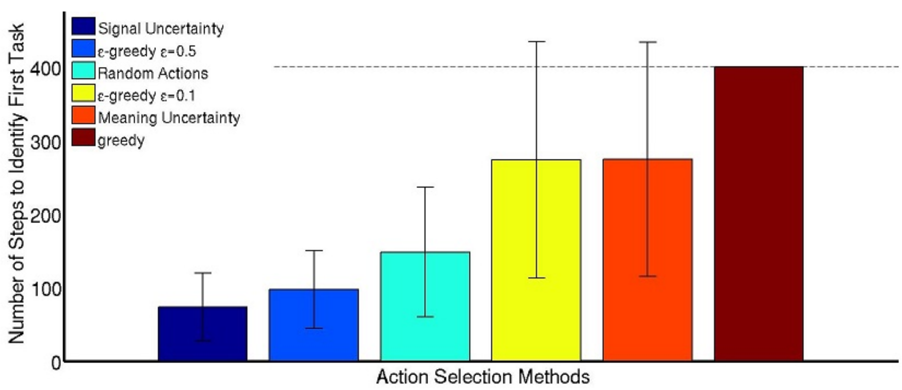


Figure 3. Comparison between different exploration methods. Planning wrt. uncertainty in noth task and signal space is the most efficient method

The connectivity of an architecture, as well as neural dynamics and learning rules, are defined by editing simple text-based configuration files. A simple instantiation of a pre-defined simulator class together with the name of the configuration file launches the simulation. Users can provide continuous input to the architecture, as well as inspect and visualize all elements of the simulation, by subclassing the simulator class and redefining the appropriate methods in a clean and Pythonic way. DyNAMoS can be, and is in fact meant to be, extended by user-defined learning methods and dynamics models, which is possible through a well-documented interface all such functions must respect. DyNAMoS distributes computation across multiple processes that are spawned dynamically, possibly on multiple computers, which communicate by TCP/IP or Linux interprocess communication depending on whether they are on the same computer. All aspects of multi-process handling and communication are completely hidden from the user who may merely specify which neural map is executed on which physical process if he wishes to.

This software has been used to speed up computations and provides a common platform for implementing online and incremental learning algorithms. Up to now, we have included linear and logistic regression, various versions of self-organizing maps, MLP and LWPR. It will be made available on GitHub in 2015 after final tests have been concluded.

5.3.5. *pyStreamPlayer: synchronized replay of multiple sensor recordings and supplementary data*

Participant: Alexander Gepperth [correspondant].

This Python software is intended to facilitate the application of machine learning algorithms by avoiding to work directly with an embodied agent but instead with data recorded in such an agent. Assuming that non-synchronous data from multiple sensors (e.g., camera, Kinect, laser etc.) have been recorded according to a flexible format defined by the pyStreamPlayer architecture, pyStreamPlayer can replay these data while retaining the exact temporal relations between different sensor measurements. As long as the current task does not involve the generation of actions, this software allows to process sensor data as if it was coming from an agent which is usually considerably easier. At the same time, pyStreamPlayer allows to replay arbitrary supplementary information such as, e.g., object information, as if it was coming from a sensor. In this way, supervision information can be stored and accessed together with sensory measurements using an unified interface. pyStreamPlayer has been used to facilitate real-world object recognition tasks, and several of the major databases in this field (CalTech Pedestrian database, HRI RoadTraffic traffic objects database, CVC person database, KITTI traffic objects database) have been converted to the pyStreamPlaer format and now serve as a source of training and test data for learning algorithms.

pyStreamPlayer has been integrated into a ROS node as well, allowing th replay and transmission across networks of distributed processes.

5.3.6. *Multimodal: framework around the NMF algorithm for multimodal learning*

Participant: Olivier Mangin [correspondant].

The python code provides a minimum set of tools and associated libraries to reproduce the experiments in [98], together with the choreography datasets. The code, publicly available at https://github.com/omangin/multimodal, under the new BSD license, is primarily intended for reproduction of the mulimodal learning experiment mentioned above. It has already been reused in several experimentations by other member of the team and is expected to play an important role in further collaborations. It is also expected that the public availability of the code encourages further experimentation by other scientists with data coming from other domains, thus increasing both the impact of the aforementioned publication and the knowledge on the algorithm behaviors. The nonnegative matrix factorization algorithm used in the experiments is also available as a third party extension to http://scikit-learn.org.

5.3.7. *Explauto: an autonomous exploration library*

Participants: Clément Moulin-Frier [correspondant], Pierre Rouanet.

Explauto is a framework developed to study, model and simulate curiosity-driven learning and exploration in virtual and robotic agents. The code repository is available at: https://github.com/flowersteam/explauto.

This library provides high-level API for an easy definition of:

- Virtual and robotics setups (Environment level)
- Sensorimotor learning iterative models (Sensorimotor level)
- Active choice of sensorimotor experiments (Interest level)

It is crossed-platform and has been tested on Linux, Windows and Mac OS. It has been released under the GPLv3 license.

Explauto's scientific roots trace back from Intelligent Adaptive Curiosity algorithmic architecture [15], which has been extended to a more general family of autonomous exploration architecture by [3] and recently expressed as a compact and unified formalism [102]. The library is detailed in [60].

This library has been used in many experiments including:

- the control of a 2D simulated arm
- the exploration of the inverse kinematics of a poppy humanoid (both on the real robot and on the simulated version)
- accoustic model of a vocal tract

5.3.8. *Explorers Framework*

Participants: Benureau Fabien [correspondant], Pierre-Yves Oudeyer.

The Explorers framework is aimed at creating, testing and comparing autonomous exploration strategies for sensorimotor spaces in robots. The framework is largely strategy-agnostic, and is aimed as expressing motor babbling, goal babbling and intrinsically motivated exploration algorithms, among other. It is also able to express strategies that feature transfer learning, such as the reuse algorithm we introduce in [34].

At the center of the framework, an explorer receives observations and provides motor commands for the environment to execute.

We can then easily express a typical goal babbling architecture (the feedback update is not pictured).

Here, the explorer interacts with the environment, rather than the inverse model. Such an architecture allows to filter motor commands that are proposed by the inverse model, and eventually to select another goal if the motor command is not satisfactory or possible to execute. The framework is organized in a modular way. This allows to create flexible hierarchical architectures made of several, atomic or themselves composite, exploration strategies.

The framework has been released this year under the *OpenScience* license (http://fabien.benureau.com/openscience.html), and made available on github (https://github.com/humm/explorers). Using provided examples, users can easily modify the exploration parameters and investigate for instance the differences between motor and goal babbling exploration strategies.

5.3.9. *PyQMC: Python library for Quasi-Metric Control*

Participant: Steve Nguyen [correspondant].

PyQMC (https://github.com/SteveNguyen/pyqmc) is a python library implementing the control method described in http://dx.doi.org/10.1371/journal.pone.0083411 It allows to solve discrete markovian decision processes by computing a Quasi-Metric on the state space. This model based method has the advantage to be goal independant and thus can produce a policy for any goal with relatively few recomputation. New addition to this method is the possibility of online learning of the transition model and the Quasi-Metric.

5.4. Software Platforms

5.4.1. *Meka robot plateform enhancement and maintenance*

Participants: Antoine Hoarau, Freek Stulp, David Filliat.

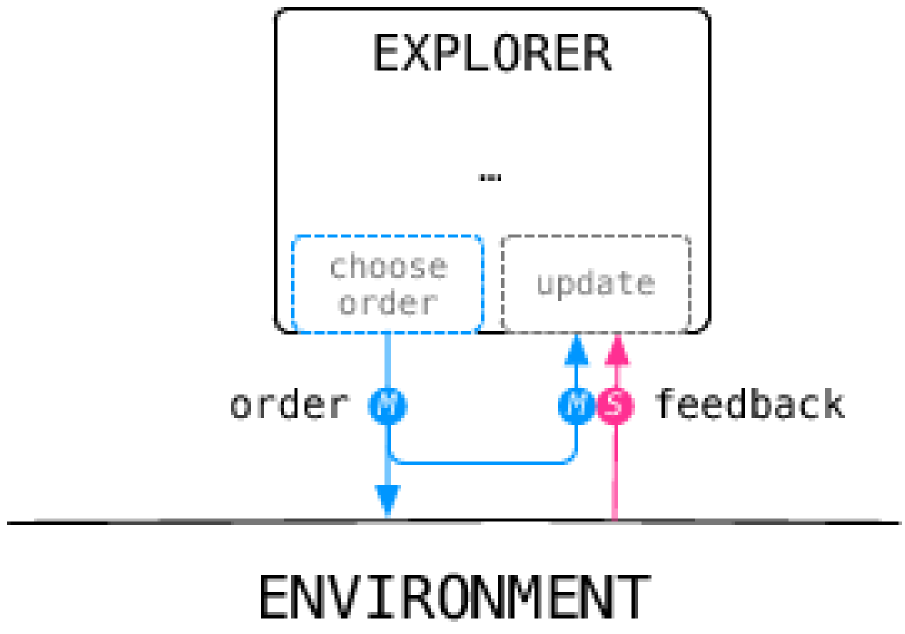


Figure 4.

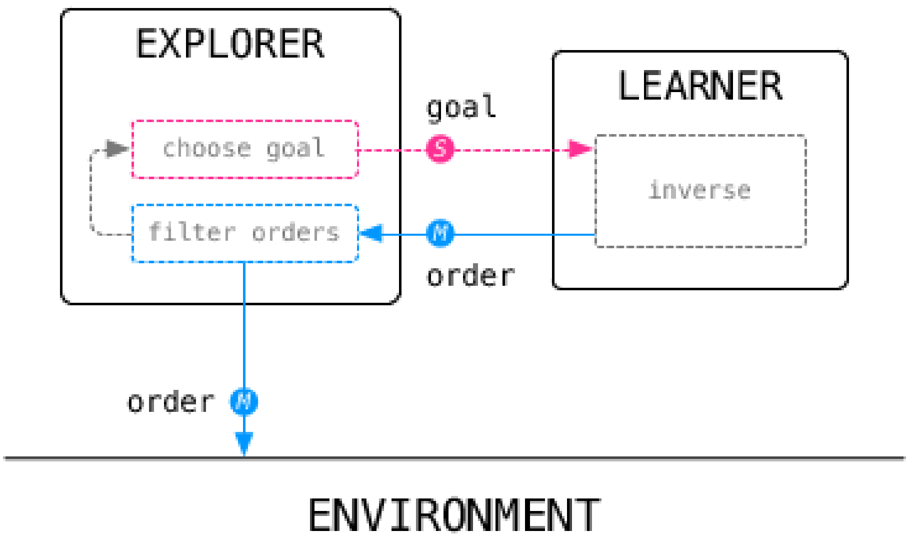


Figure 5.

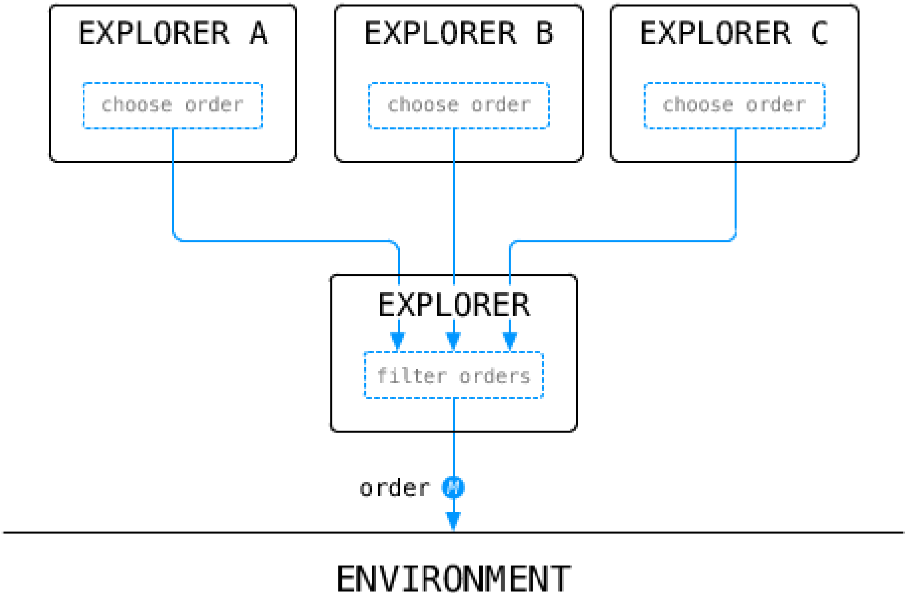


Figure 6.

Autonomous human-centered robots, for instance robots that assist people with disabilities, must be able to physically manipulate their environment. There is therefore a strong interest within the FLOWERS team to apply the developmental approach to robotics in particular to the acquisition of sophisticated skills for manipulation and perception. ENSTA-ParisTech has recently acquired a Meka (cf. 7) humanoid robot dedicated to human-robot interaction, and which is perfectly fitted to this research. The goal of this project is to install state-of-the-art software architecture and libraries for perception and control on the Meka robot, so that this robot can be jointly used by FLOWERS and ENSTA. In particular, we want to provide the robot with an initial set of manipulation skills.
The goal is to develop a set of demos, which demonstrate the capabilities of the Meka, and provide a basis on which researchers can start their experiments.

The platform is evolving as the sofware (Ubuntu, ROS, our code) is constantly updated and requires some maintenance so less is needed for later. A few demos were added, as the hand shaking demo, in which the robot detects people via kinect and initiates a hand shake with facial expressions. This demo has been used to setup a bigger human robot interaction experiment, currently tested on subjects at Ensta (cf. 8). Finally, we've seen that the robot itself also needs some maintenance; some components broke (a finger tendon), a welding got cold (in the arm) and a few cables experienced fatigue (led matrix and cameras) (cf. 9).

5.4.2. Teaching concepts to the Meka robot

Participants: Fabio Pardo [Correspondant], Olivier Mangin, Anna-Lisa Vollmer, Yuxin Chen, David Filliat.

This plateform was developped during Fabio Pardo's internship, in the dual context of the study of Anna-Lisa Vollmer's research on human robot interaction protocole during a learning task, and Olivier Mangin's research on mechanism for word learning and multimodal concept acquisition. The plateform is centered around an interaction zone where objects are presented to a Meka robot augmented with a kinect camera placed on top of the interaction zone. Several colorful objects are available to be presented and described to the robot. Several

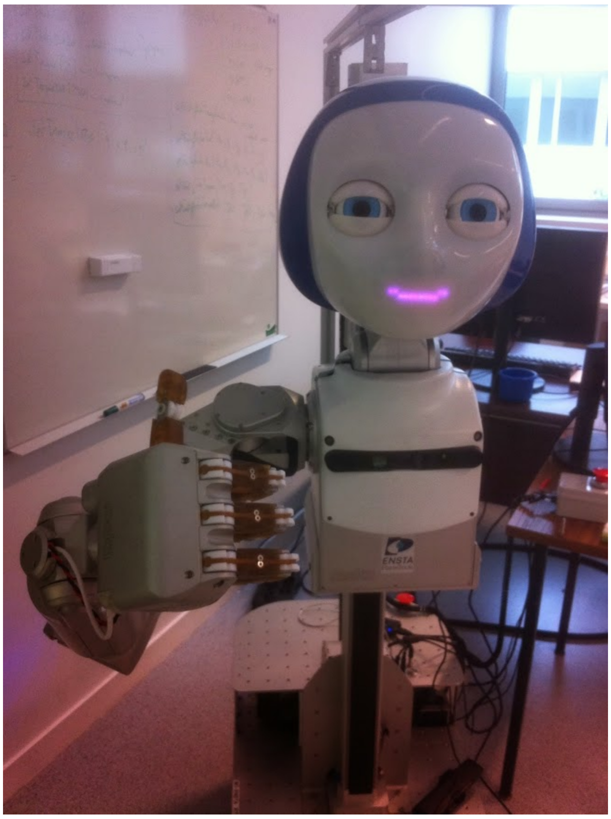


Figure 7. The Meka robot plateform acquired by ENSTA ParisTech

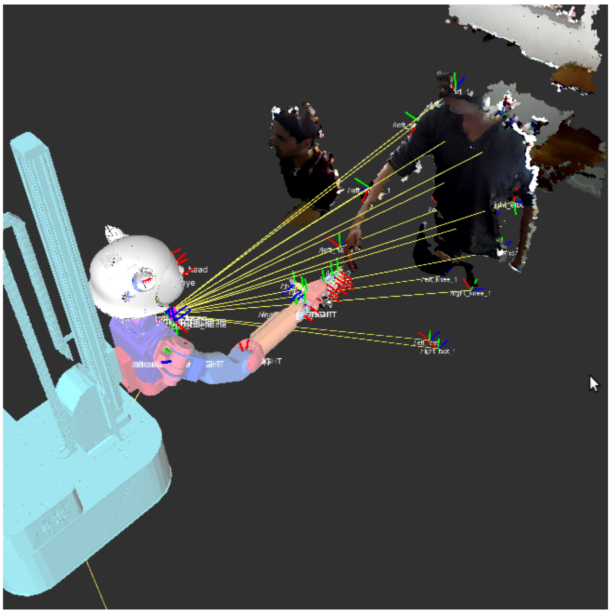


Figure 8. Hand shake demo visualized on Rviz (ROS)

object may be present at the same time on the table. Typical object are easily caracterized by their colors and shapes, such as the *red ball*, the *yellow cup*, or the *blue wagon with red wheels*.

The robot software is capable of abstracting the visual and acoustic perception in the following way. The camera image is segmented into objects; from each object, a set of descriptors is extracted, typically SIFT or shape descriptors and color histograms. An incremental clustering algorithm transforms the continuous descriptors into a histogram of descrete visual descriptors that is provided to the learning algorithm. The acoustic stream is segmented into sentences by a silence detection process and each sentence is fed to Google's text to speach API. Finally each sentence is represented as a histogram of the words recognized in the sentence.

The robot is capable of learning multimodal concepts, spanning words and visual concepts, through the nonnegative matrix factorization framework introduced by Olivier Mangin (see). In addition several behaviors are programmed in the robot such as gaze following objects or understanding a few interaction questions.

The framework is illustrated on the following video https://www.youtube.com/watch?v=Ym5aYfzoQX8. It enables to modify the interaction as well as the learning mechanisms in order to study the interaction between the teacher and the learning robot.

5.4.3. Experiment platform for multiparameters simulations

Participants: Fabien Benureau, Paul Fudal.

Simulations in robotics have many shortcommings. At the same time, they offer high customizability, rapidity of deployment, abscence of failure, consistency across time and scalability. In the context of the PhD work of Fabien Benureau, it was decided to investigate hypothesis first in simulation before moving to real hardware. In order to be able to test a high number of different hypothesis, we developped a software platform that would scale to the computing ressource available.

We designed simple continuous simulations around a of-the-shelf 2D physic engine and wrote a highly modular platform that would automatically deploy experiments on cluster environments, with proper handling

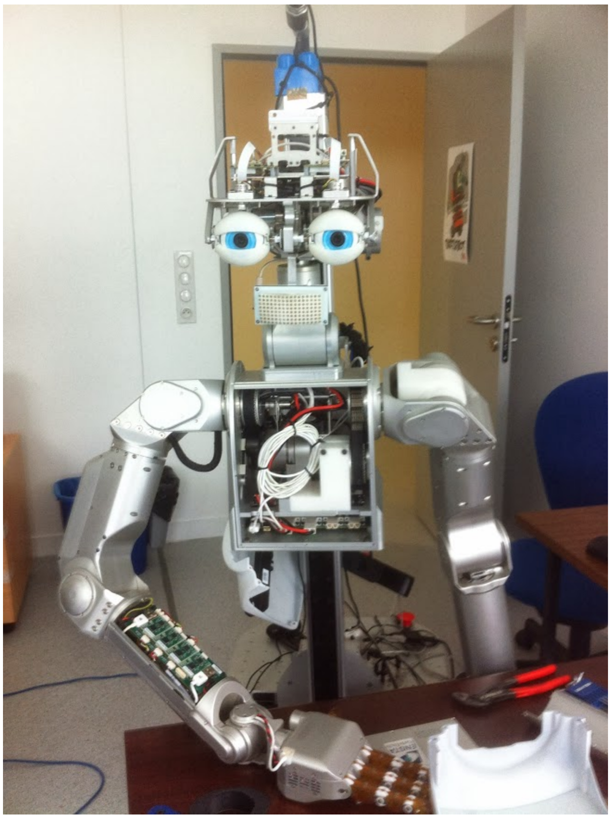


Figure 9. Maintenance is required on the robot

of dependencies; our work investigate transfer learning, and some experiments's input data is dependent of the results of another.

So far, this platform and the university cluster has allowed to conduct thousands of simulations in parallel, totaling more than 10 years of simulation time. It has led us to present many diverse experiments in our published work [34], each repeated numerous times. It has allowed us to conduct a multi-parameter analysis on the setup, which led to new insights, which are being presented in a journal article to be submitted in the beginning of this year.

Because of its high modularity, this platform is proving to be highly flexible. We are currently adaptating it to a modified, cluster-ready, version of the V-REP simulator. Those simulations will serve to back ones on similar real-world hardware that are currently setup.

5.4.4. pypot

Participants: Pierre Rouanet [correspondant], Steve N'Guyen, Matthieu Lapeyre.

Pypot is a framework developed to make it easy and fast to control custom robots based on dynamixel motors. This framework provides different levels of abstraction corresponding to different types of use. More precisely, you can use pypot to:

1. directly control robotis motors through a USB2serial device,
2. define the structure of your particular robot and control it through high-level commands,
3. define primitives and easily combine them to create complex behavior.

Pypot has been entirely written in Python to allow for fast development, easy deployment and quick scripting by non-necessary expert developers. It can also benefits from the scientific and machine learning libraries existing in Python. The serial communication is handled through the standard library and thus allows for rather high performance (10ms sensorimotor loop). It is crossed-platform and has been tested on Linux, Windows and Mac OS.

Pypot is also compatible with the V-REP simulator (http://www.coppeliarobotics.com). This allows the transparent switch from a real robot to its simulated equivalent without having to modify the code.

Pypot also defined a REST API permitting the development of web apps such as a web control interface faciliting the use of a robotic platform.

Pypot is part of the Poppy project (http://www.poppy-project.org) and has been released under an open source license GPL V3. More details are available on pypot website: https://github.com/poppy-project/pypot

5.5. Experimental Setups

5.5.1. Experimental Platform for User Study of Curiosity-driven Exploration

Participants: Pierre Rouanet [correspondant], Jonathan Grizou, Brice Miard, Julie Golliot.

This platform has been developed to investigate curiosity-driven behaviors and more precisely how humans explore new sensori-motor spaces. It consists in several simple games where users control a 2D/3D shape with the movements of their body. They have to discover the mapping between their movements and a shape displayed on the screen and learn how to make the controlled shape match the target one (fig 10).

The software is entirelly written in Python. It includes a Kinect wrapper allowing the access of 3D position of tracked skeleton joints. It provides a framework for creating new games based on the 2D drawing library (pygame). It also includes a web server used to display game instructions, cut-scene videos and questionnaire.

The presentation of the platform and the preliminary results of a user's study have been rapported in [58].

5.5.2. Learning and representing object assembly tasks

Participants: Yoan Mollard [correspondant], Thibaut Munzer, Pierre Rouanet, Manuel Lopes.

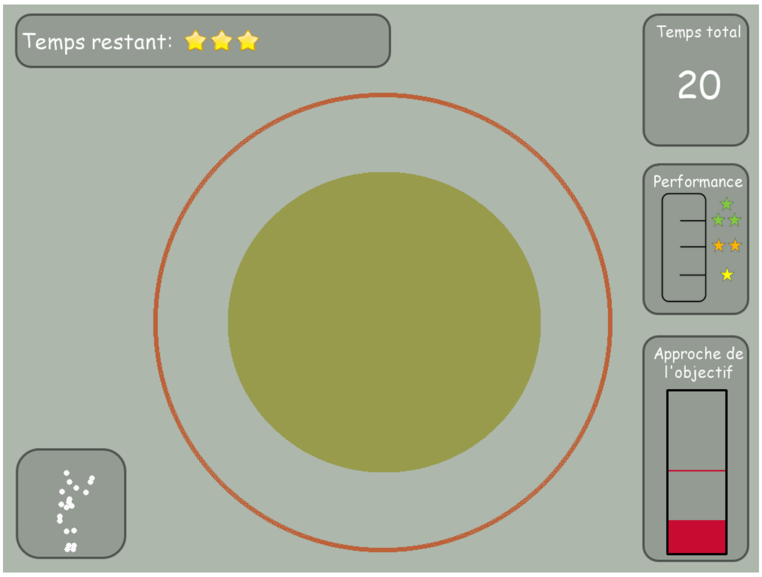


Figure 10. A screenshot representing the game interface as seen by the user.

In the context of the 3rd hand project 8.3.1.1 we created a framework for learning assembly tasks from demonstration. In this work we showed how a complex assembly task could be automatically decomposed in components allowing to learn constraints between different objects and their assembly plan. We created also a Graphical User Interface (GUI) allowing to present the learned data in a intuitive way, so that the user can be aware of what the computer has learned. This awareness is crucial for Human-Robot cooperation since the robot will base its decisions on the learned data. Making them clear to the user also allow to rely on him to find potential errors and correct the noise. Thus, the user can program the robot by combining demonstrations and manual corrections minimizing the overall programming phase. Our experimental setup consists in several sequencial phases:

1. **Demonstrations**: User provides several demonstrations of an assembly. All parts of the objects are individually tracked by an Optitrack tracking system
2. **Constraint extraction**: Trajectories are analysed to extract rigid constraints
3. **Segmentation**: Constraints on all demonstrations are segmented to find one constraint per object
4. **Plan computation**: We deduce relational MDP trajectories from raw data, creating one assembly step per constraint
5. **Presentation and correction**: The raw constraints and assembly plan are presented to the user in a friendly way through a 3D GUI so that he is able to visualize and correct them 12
6. **Execution**: The corrected informations are then sent to the robot for actual execution. The execution system only receives constraints and the plan, all motions are computed by a motion planner to reach the goals, but motions could also been extracted from the demonstrations using dynamic motor primitives (DMP). We used a simulated Baxter robot that we acquired during the year.

The framework is written in C++ (GUI) and Python (tracking system, data analysis and execution system), and is completely integrated into ROS. The main steps of the workflow are shown on figure 11 .

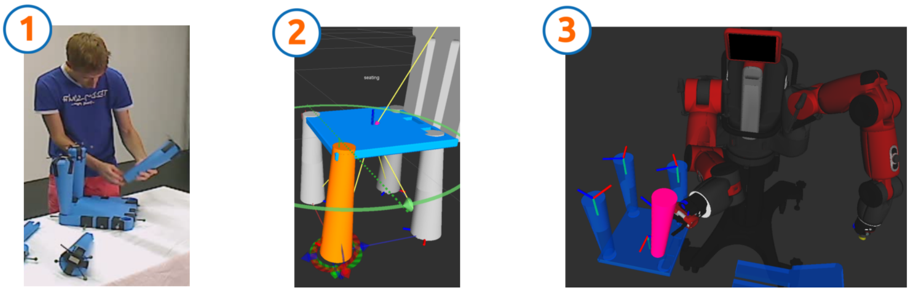


Figure 11. Demonstration, correction and execution of an assembly

The GUI itself 12 represents rigid constraints visually, and provides all the controls necessary to correct them using a graphical procedure. It shows the learned assembly plan as a list of sequential steps that the user can browse like any assembly manual. Also the GUI introduces degrees of freedom in the form of standard mechanical joints (rotational, prismatic, cylindrical joints ...) that the robot can use during execution to simplify the motions and decrease failures during motion planning. The GUI draw graphical cues to represent them and is also able to animate them to make them even clearer.

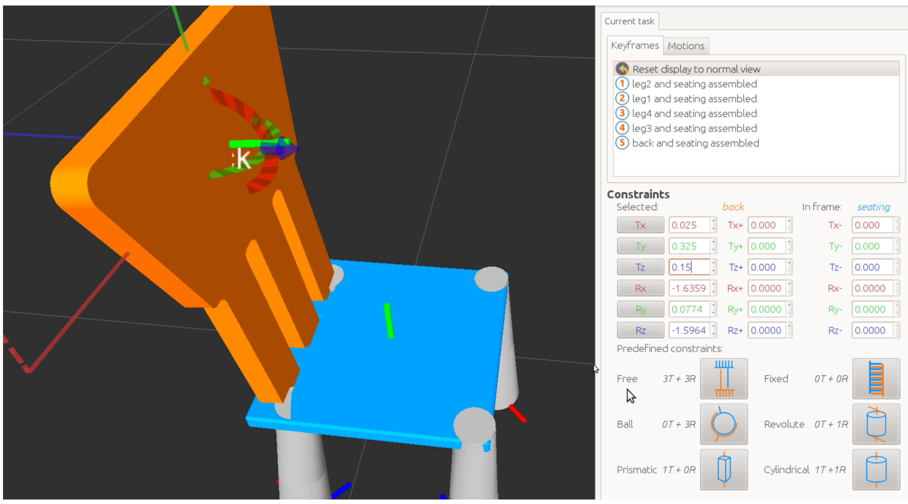


Figure 12. Detailed view of the GUI showing the learned constraints and assembly plan

5.6. Hardware

5.6.1. Poppy Platform

Participants: Matthieu Lapeyre [correspondant], Pierre Rouanet, Jonathan Grizou, Pierre-Yves Oudeyer [supervisor].

The Poppy Project [54], [53], [23] (see Figure 13 , http://www.matthieu-lapeyre.com/thesis.pdf) develops an open-source 3D printed humanoid platform based on robust, flexible, easy-to-use and reproduce hardware and software. In particular, the use of 3D printing and rapid prototyping technologies is a central aspect of this project, and makes it easy and fast not only to reproduce the platform, but also to explore morphological variants. Poppy targets three domains of use: science, education and art (see http://www.poppy-project.org).

Poppy was initially designed with a scientific objective, aiming to be a new experimental platform opening the possibility to systematically study the role of morphology in sensorimotor control, in human-robot interaction and in cognitive development. Indeed, a suitable design of a robot morphology can greatly simplify control problems, increase robustness, and open new modes of interaction with the physical and social world. Thus, being able to study the body as an experimental variable, something which can be systematically changed and experimented, is of paramount importance. Yet, until recently it was complicated because building a robot relied on heavy and costly manufacturing techniques. 3D printing has changed the landscape of what is possible: Poppy Project transposed it to humanoid robotics, and it is now possible to explore new body shapes in just a few days. It enables and simplifies the experimentation, the reproduction and the modification of the morphology in research laboratories. It also allows collaborative working, sharing and replication of the results on these issues between laboratories. The ambition is to become a reference platform for benchmarking and dissemination of scientific results.

Thanks to the fact that it integrates advanced and yet easily accessible techniques in an embodiment that motivates students and the wider public, this platform also meets a growing societal need: education and training in technologies combining computer science, electronics and mechanics, as well as a training tool to the emergent revolutionary 3D printing process. With its openness, its design and its rather low-cost, Poppy provides a unique context for experimentation and learning of these technologies in a Do-It-Yourself (DIY) approach. Several experiences with Poppy in secondary, high schools, science museums and Fablabs in France and abroad are underway and will be discussed in the incoming sections. Finally, the possibility to easily modify both the hardware and the software also makes Poppy a useful tool for artistic projects working with interactive computerized installations.

5.6.1.1. Open-Source Robotic Platform

Poppy is the first complete 3D printed open-source and open-hardware humanoid robot. Its 3D printed skeleton and its Arduino-based electronics are open-hardware (Creative Commons). Its software is open-source (GPL V3), and allows programming beginners as well as advanced roboticists to control the robot in Python thanks to the PyPot library (https://github.com/poppy-project/pypot). Its motors are common off-the-shell Robotis actuators (http://www.robotis.com/xe/dynamixel_en), and allow for compliant control and soft physical human-robot interaction. Poppy presents an original mechanical structure which permits to obtain a light structure with 3.5kg for 84cm height. Before the arrival of 3D printing techniques, this kind of complex structure was either impossible to produce or extremely expensive. Now, anyone can produce and modify such robot in their home using affordable personal 3D printers.

Several web tools support collaboration and sharing among members of the Poppy community: a portal web site (www.poppy-project.org), GitHub repositories for the hardware and software with associated wikis for documentation (www.github.com/poppy-project/), and a forum based on Discourse [0] technology (forum.poppy-project.org).

[0] www.discourse.org

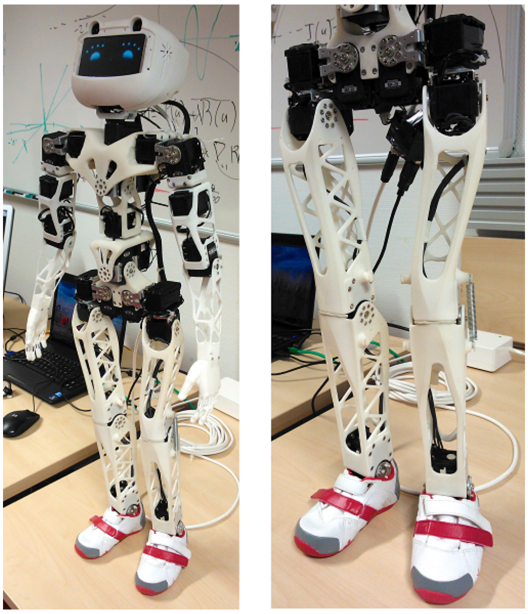


Figure 13. a. Global view of the Poppy platform. b. Zoom on legs design

GRAPHIK Project-Team

5. New Software and Platforms

5.1. Cogui

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

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

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

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

5.2. Graal

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

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

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

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

HEPHAISTOS Team

5. New Software and Platforms

5.1. Introduction

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

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

5.2. Interval analysis libraries

5.2.1. ALIAS

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

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

`ALIAS` is made of two parts:

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

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

5.3. Platforms

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

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

[0] see http://www-sop.inria.fr/hephaistos/prototypes/main.html

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

HYBRID Project-Team

5. New Software and Platforms

5.1. OpenViBE

Participants: Anatole Lécuyer [contact], Marsel Mano, Jussi Lindgren.

OpenViBE is a free and open-source software platform devoted to the design, test and use of Brain-Computer Interfaces (BCI). The platform consists of a set of software modules that can be integrated easily and efficiently to design BCI applications. The key features of OpenViBE software are its modularity, its high-performance, its portability, its multiple-users facilities and its connection with high-end/VR displays. The "designer" of the platform enables to build complete scenarios based on existing software modules using a dedicated graphical language and a simple Graphical User Interface (GUI). This software is available on the Inria Forge under the terms of the AGPL licence, and it was officially released in June 2009. Since then, the OpenViBE software has already been downloaded more than 30000 times, and it is used by numerous laboratories, projects, or individuals worldwide. The OpenViBE software is supported and improved in the frame of OpenViBE-NT project (section 8.2.7). More information, downloads, tutorials, videos, documentations are available on the OpenViBE website.

5.2. Collaviz

Participants: Thierry Duval, Thi Thuong Huyen Nguyen [contact].

The aim of Collaviz software (collaborative interactive visualization) is to allow to design, deploy and share collaborative virtual environments (CVE). Collaviz allows VR developpers to concentrate on the behavior of virtual objects that can be shared between users in a CVE. Indeed, Collaviz provides a software architecture that hides the network programmation details of the distribution and the synchronization of the content of the CVE, and that facilitates the coupling with the 3D graphics API used for rendering. Collaviz is written mainly in Java and is runnable on multiple hardware configurations: laptop or desktop computer, immersive room, mobile devices. The PAC-C3D software architecture of Collaviz makes it possible to use various 3D APIs for graphic rendering: Java3D, jReality, jMonkeyEngine, OpenSG, Unity3D (work in progress) and Havok Anarchy (work in progress), and also to use various physical engines such as jBullet and SOFA. The distribution over the network can be achieved using TCP or HTTP. A collaboration with DIVERSE team intended to extend Collaviz using a Model Driven Engineering approach in order to provide high-level tools to generate a large part of java code of virtual objects.

IMAGINE Project-Team

5. New Software and Platforms

5.1. MyCorporisFabrica

Participants: Ali-Hamadi Dicko, François Faure, Olivier Palombi, Federico Ulliana.

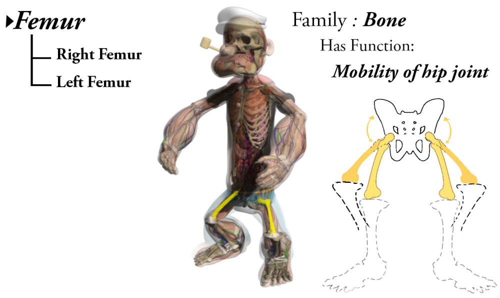


Figure 1. My Corporis Fabrica is an anatomical knowledge database developed in our team.

My Corporis Fabrica (MyCF) is an anatomical knowledge ontology developed in our group. It relies on FMA (Foundational Model of Anatomy), developed under Creative Commons license (CC-by). MyCF browser is available on line, and is already in use for education and research in anatomy: http://www.mycorporisfabrica.org/. Moreover, the MyCf's generic programming framework can be used for other domains, since the link it provides between semantic and 3D models matches several other research applications at IMAGINE.

5.2. SOFA

Participants: François Faure, Armelle Bauer, Olivier Carré, Aurélie Dégletagne, Ali Hamadi Dicko, Matthieu Nesme, Romain Testylier.

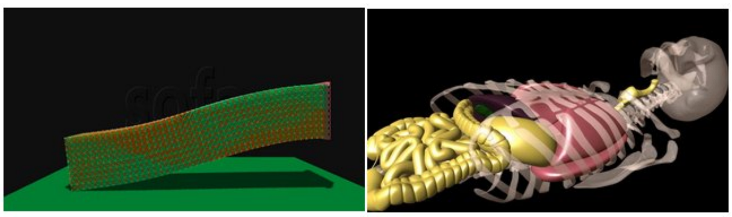


Figure 2. SOFA is an open source simulator for physically based modeling.

SOFA is a is a real-time physically based simulation library developed for more 8 years with other Inria research groups (Shacra and Asclepios). It primarily targeted medical simulation research, but we are using it as well for many other applications, from the entertainment industry (films and games) to earth science projects. Based on an advanced software architecture, it allows to (1) create complex and evolving simulations by combining new algorithms with algorithms already included in SOFA; (2) modify most features of the simulation (deformable behavior, surface representation, solver, constraints, collision algorithm, etc.) by simply editing an XML file; (3) build complex models from simpler ones using a scenegraph description; (4) efficiently simulate the dynamics of interacting objects using abstract equation solvers; and (5) reuse and easily compare a variety of available methods.

SOFA is gaining momentum. A start-up based on SOFA, InSimo, has been created in Strasbourg by Inria people, and one of our former engineers, François Jourdes, has been hired.

5.3. Expressive

Participants: Marie-Paule Cani, Antoine Begault, Rémi Brouet, Even Entem, Thomas Delame, Ulysse Vimont, Cédric Zanni.

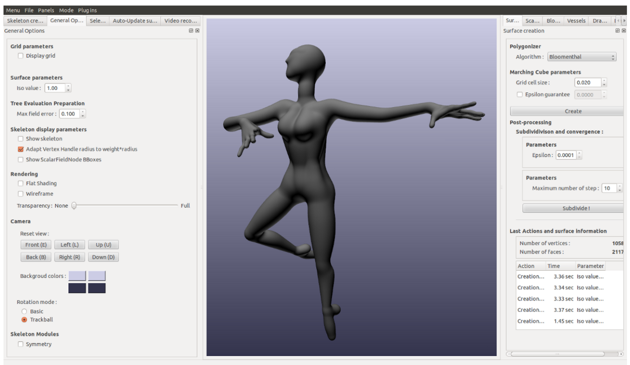


Figure 3. GUI and Example of implicit surface and modeled with the Expressive platform.

Expressive is a new C++ library created in 2013 for gathering and sharing the models and algorithms developed within the ERC Expressive project. It enables us to make our latest research results on new creative tools - such as high level models with intuitive, sketching or sculpting interfaces - soon available to the rest of the group and easily usable for our collaborators, such as Evelyne Hubert (Inria, Galaad) or Loic Barthe (IRIT, Toulouse). The most advanced part is a new version of Convol, a library dedicated to implicit modeling, with a main focus on integral surfaces along skeletons. Convol incorporates all the necessary material for constructive implicit modeling, a variety of blending operators and several methods for tessellating an implicit surface into a mesh, and for refining it in highly curved regions. The creation of new solid geometry can be performed by direct manipulation of skeletal primitives or through sketch-based modeling and multi-touch deformations.

IN-SITU Project-Team

5. New Software and Platforms

5.1. WILDER Platform

Participants: Michel Beaudouin-Lafon [correspondant], Olivier Chapuis, Cédric Fleury, Olivier Gladin, Rémi Hellequin, Stéphane Huot, Amani Kooli, Monireh Sanaei, Gabriel Tezier, Jonathan Thorpe.

WILDER is InSitu's second experimental ultra-high-resolution interactive environment, following up on the WILD platform developed since 2009 [2] (Figure 1). It features a wall-sized display with seventy-five 20" LCD screens, i.e. a 5m50 x 1m80 (18' x 6') wall displaying 14 400 x 4 800 = 69 million pixels, powered by a 10-computer cluster and two front-end computers. The platform also features a camera-based motion tracking system supporting interaction with the wall as well as within the surrounding space, a multitouch frame making the entire wall-sized display touch sensitive and various mobile devices. WILDER is part of the DIGISCOPE Equipment of Excellence and, in combination with WILD and the other DIGISCOPE rooms, provides a unique experimental environment for collaborative interaction. In addition to using WILD and WILDER for our research, we have also developed software architectures and toolkits that enable developers to run applications on such multi-device, cluster-based systems.

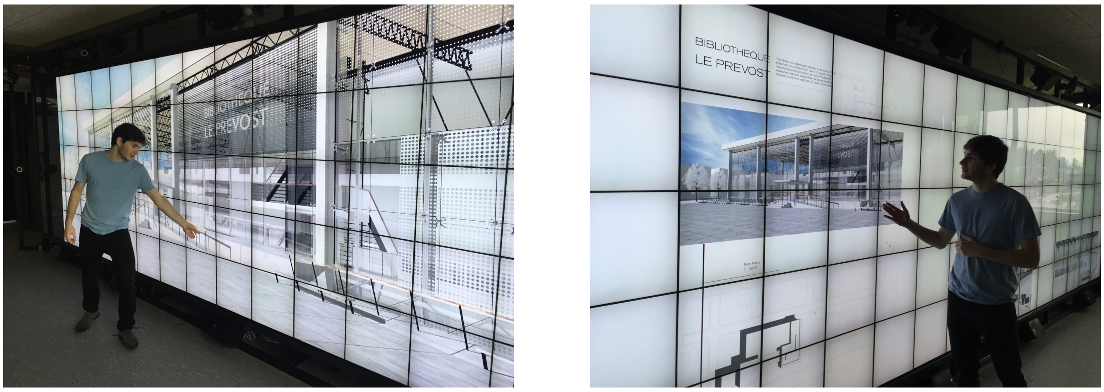


Figure 1. The WILDER platform.

5.2. Smarties

Participants: Olivier Chapuis [correspondant], Anastasia Bezerianos, Bruno Fruchard.

The Smarties system [16] provides an easy way to add mobile interactive support to collaborative applications for wall displays.

It consists of (i) a mobile interface that runs on mobile devices for input, (ii) a communication protocol between the mobiles and the wall application, and (iii) libraries that implement the protocol and handle synchronization, locking and input conflicts. The library presents the input as an event loop with callback functions and handles all communication between mobiles and wall application. Developers can customize the mobile interface from the wall application without modifying the mobile interface code.

On each mobile we find a set of cursor controllers associated with keyboards, widgets and clipboards. These controllers (pucks) can be shared by multiple collaborating users. They can control simple cursors on the wall application, or specific content (objects or groups of them). The developer can decide the types of widgets associated to pucks from the wall application side.

Smarties mobile clients currently run on Android, while server libraries have been developed in C++ and Java.

Figure 2. Left: Multiple Lenses, starting from the left a magnification lens, a DragMag and a fisheye. Right: two synchronized Smarties clients running on tablets. The four colored pucks are attached respectively to a magnification lens (left of wall), the anchor and lens of a DragMag (middle) and a fisheye (right). The active puck is the blue for the device on top, and green for the bottom. The described widgets added by the application are seen on the widget area.

Smarties is available at the http://smarties.lri.fr/ under a GNU GPL licence.

- ACM: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)
- Software benefit: See O. Chapuis, A. Bezerianos, S. Franzeskakis (2014) Smarties: An Input System for Wall Display Development. In CHI '14: Proceedings of the 32nd international conference on Human factors in computing systems. ACM, pages 225-234.
- OS/Middleware: Crossplatform
- Required library or software: none
- Programming language: C++, Java

5.3. WildOS

Participant: Michel Beaudouin-Lafon [correspondant].

WildOS is middleware to support applications running in an interactive room featuring various interaction resources, such as our WILD and WILDER rooms: a tiled wall display, a motion tracking system, tablets and smartphones, etc. The conceptual model of WildOS is a *platform*, such as the WILD or WILDER room, described as a set of devices and on which one or more applications can be run.

WildOS consists of a server running on a machine that has network access to all the machines involved in the platform, and a set of clients running on the various interaction resources, such as a display cluster or a tablet. Once *WildOS* is running, applications can be started and stopped and devices can be added to / removed from the platform.

WildOS relies on Web technologies, most notably Javascript and node.js, as well as node-webkit and HTML5. This makes it inherently portable (it is currently tested on Mac OS X and Linux). While applications can be developed only with these Web technologies, it is also possible to bridge to existing applications developed in other environments if they provide sufficient access to be remote controlled.

WildOS is used in several InSitu projects, and is also deployed on several of Google's interactive rooms in Mountain View, Dublin and Paris. It is available under on Open Source licence at https://bitbucket.org/mblinsitu/wildos.

- ACM: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)
- Software benefit: helps the development of multisurface applications.
- OS/Middleware: Crossplatform
- Required library or software: node.js, node-webkit
- Programming language: Javascript

5.4. GlideCursor

Participants: Michel Beaudouin-Lafon [correspondant], Stéphane Huot.

GlideCursor is a Mac OS X application that implements the inertial cursor described in [15]. The current version only works when moving the cursor with a trackpad. The application lets users configure gliding, and can also log cursor activity for later analyses.

GlideCursor is available under on Open Source licence at https://bitbucket.org.

- ACM: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)
- Software benefit: can improve cursor pointing on large displays.
- OS/Middleware: Mac OS X
- Required library or software: none
- Programming language: Objective-C

LAGADIC Project-Team

5. New Software and Platforms

5.1. ViSP: a visual servoing and tracking software library

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

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

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

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

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

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

5.2. DESlam software

Participant: Patrick Rives [correspondant].

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

5.3. HandiViz software

Participants: Marie Babel [correspondant], François Pasteau.

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

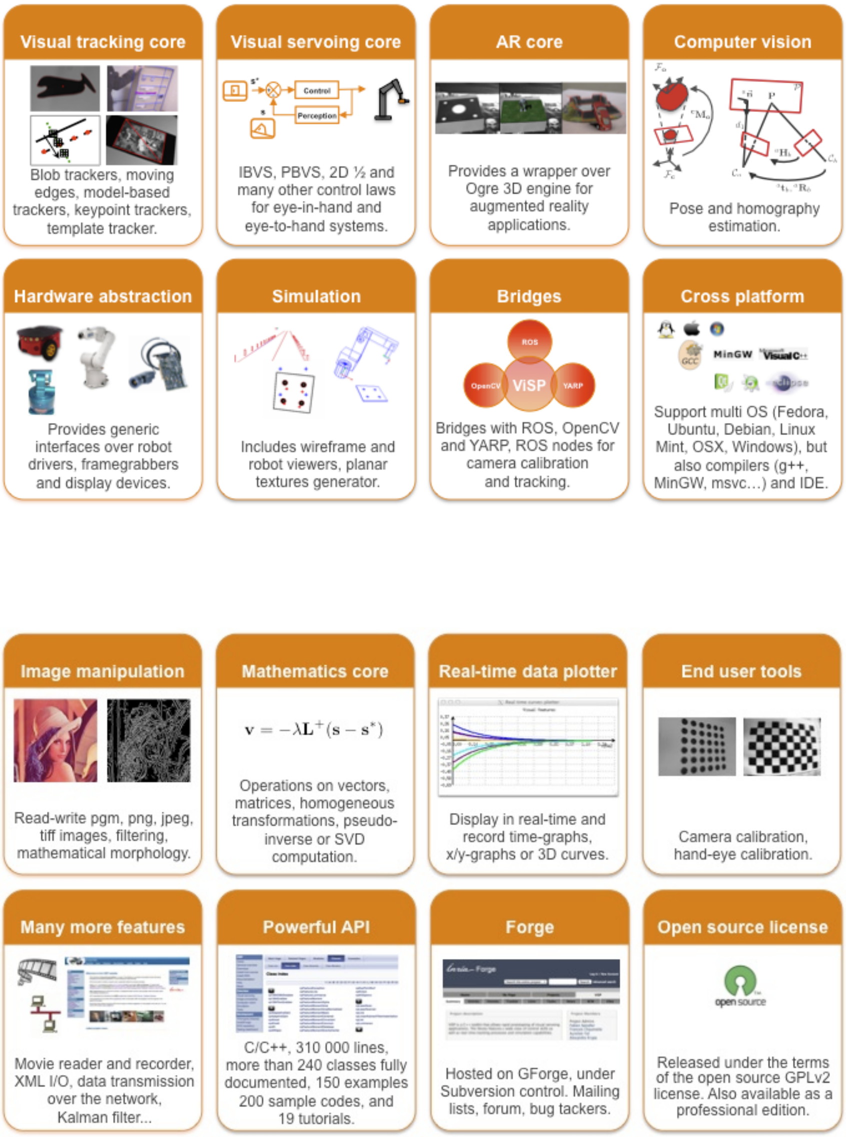


Figure 1. This figure highlights ViSP main capabilities for visual tracking, visual servoing, and augmented reality that may benefit from computer vision algorithms. ViSP allows controlling specific platforms through hardware abstraction or in simulation. ViSP provides also bridges over other frameworks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

5.4. Platforms

5.4.1. Robot vision platforms

Participant: Fabien Spindler [correspondant].

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

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

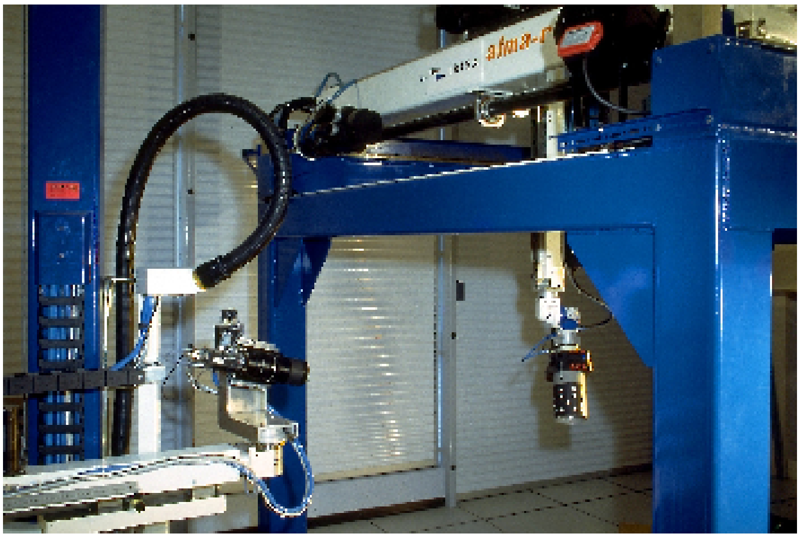


Figure 2. Lagadic robotics platforms for vision-based manipulation

5.4.2. Mobile robotics platforms

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

5.4.2.1. Indoor mobile robots

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

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

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

5.4.2.2. *Outdoor mobile robots*

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

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

5.4.2.3. *Technological Development Action (ADT) P2N*

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

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

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

5.4.3. *Medical robotics platforms*

Participants: Fabien Spindler [correspondant], Alexandre Krupa.

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

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

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

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

5.4.4. *Humanoid robot*

Participants: Giovanni Claudio, Fabien Spindler [correspondant].

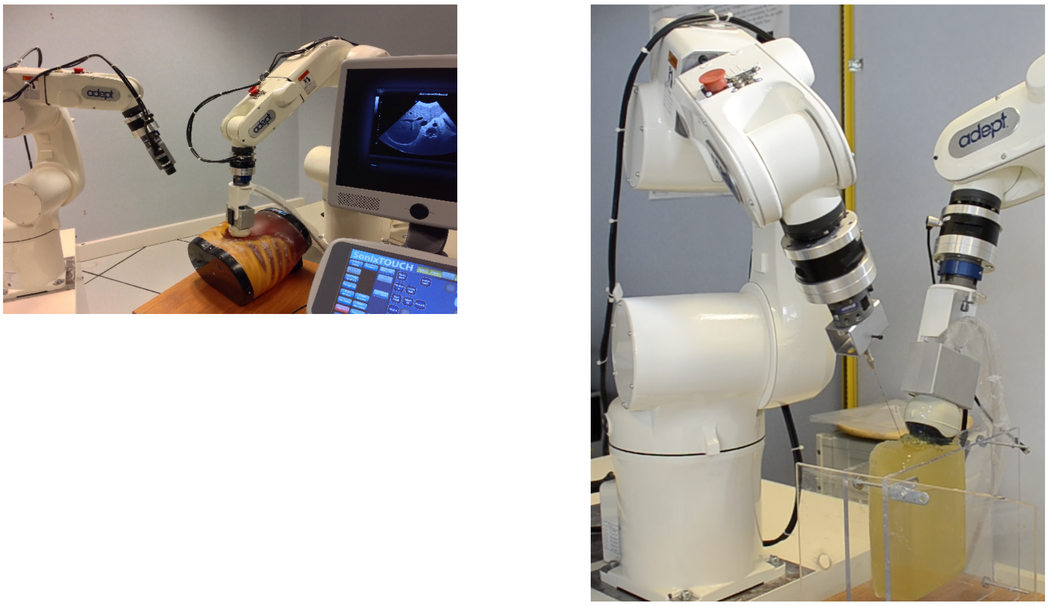
Romeo is a humanoid robot from Aldebaran Robotics which is intended to be a genuine personal assistant and companion. In September, we were the first of the four European research laboratories that acquire a Romeo. For the moment only the upper part of the body (arms, head) is working. This research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

5.4.5. *Unmanned Aerial Vehicles (UAVs)*

Participants: Fabrizio Schiano, Paolo Robuffo Giordano.

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

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



(a) (b)

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

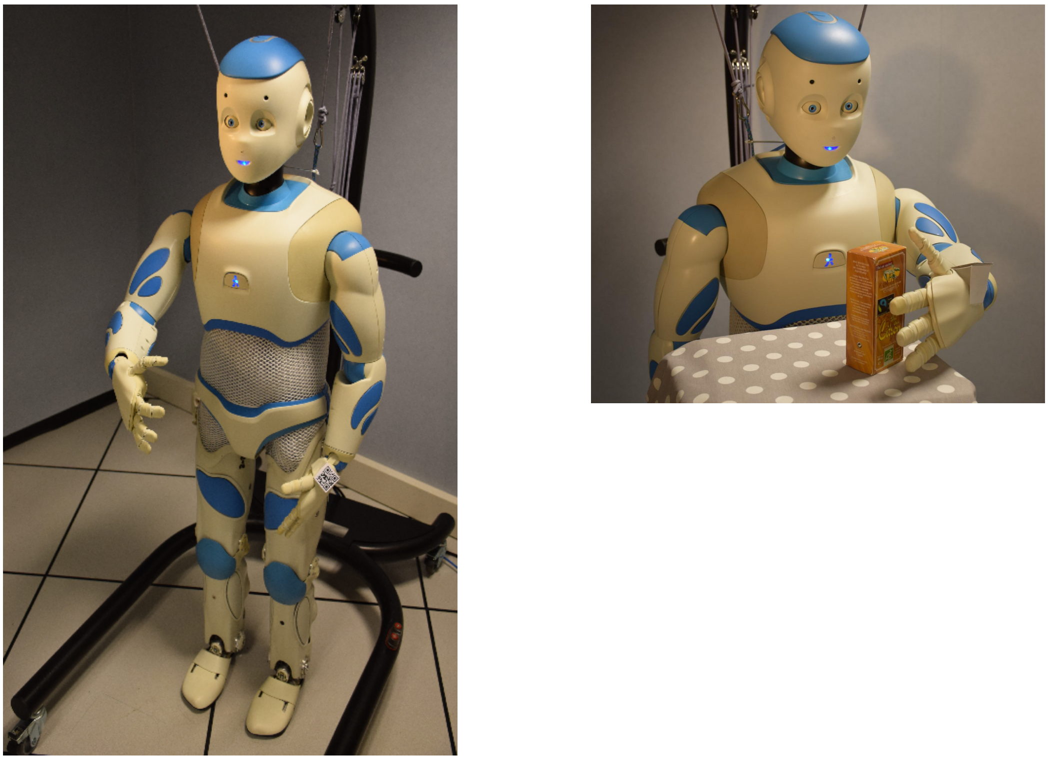


Figure 5. Romeo experimental platform.

(a) (b)

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

LEAR Project-Team

5. New Software and Platforms

5.1. Yael library

Participants: Matthijs Douze [correspondant], Herve Jegou [TEXMEX Team Inria Rennes].

Yael [14] is a library with Matlab and Python bindings providing optimized (multi-threaded, Blas/Lapack, low level optimization) implementations of functions useful in vision and machine learning such as k-means, GMM, exact nearest neighbor search and Fisher vector computation.

In 2014, it was extended to include a generic inverted file implementation, that can accomodate any type of signature that refines the similarity computation between documents. The Fisher vector computation code was also optimized.

5.2. SPArse Modeling Software (SPAMS)

Participants: Julien Mairal [correspondant], Yuansi Chen, Zaid Harchaoui.

SPAMS v2.5 was released as open-source software in May 2014 (v1.0 was released in September 2009). It is an optimization toolbox implementing algorithms to address various machine learning and signal processing problems involving

- Dictionary learning and matrix factorization (NMF, sparse PCA, ...);
- Solving medium-scale sparse decomposition problems with LARS, coordinate descent, OMP, SOMP, proximal methods;
- Solving large-scale sparse estimation problems with stochastic optimization;
- Solving structured sparse decomposition problems (sparse group lasso, tree-structured regularization, structured sparsity with overlapping groups,...).

The software and its documentation are available at http://spams-devel.gforge.inria.fr/.

This year, we added new functionalities to the toolbox. The implementation of archetypal analysis corresponding to the paper [9] was added.

5.3. FlipFlop: Fast Lasso-based Isoform Prediction as a Flow Problem

Participants: Elsa Bernard [Institut Curie, Ecoles des Mines-ParisTech], Laurent Jacob [CNRS, LBBE Laboratory], Julien Mairal [correspondant], Jean-Philippe Vert [Institut Curie, Ecoles des Mines-ParisTech].

FlipFlop is an open-source software, implementing a fast method for de novo transcript discovery and abundance estimation from RNA-Seq data [4]. It differs from classical approaches such as Cufflinks by simultaneously performing the identification and quantitation tasks using a penalized maximum likelihood approach, which leads to improved precision/recall. Other software taking this approach have an exponential complexity in the number of exons of a gene. We use a novel algorithm based on network flow formalism, which gives us a polynomial runtime. In practice, FlipFlop was shown to outperform penalized maximum likelihood based softwares in terms of speed and to perform transcript discovery in less than 1/2 second for large genes.

FlipFlop 1.4.1 is a user friendly bioconductor R package, which was released in October 2014. It is freely available on the Bioconductor website under a GPL licence: http://bioconductor.org/packages/release/bioc/html/flipflop.html.

5.4. DeepFlow

Participants: Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, Cordelia Schmid.

We developed a package for the “deep flow” algorithm. “Deep flow” combines a standard variational framework with a our new matching algorithm “deep matching”. The code for “deep matching” is in python and the code for “deep flow” in C. Both of them are available on-line at http://lear.inrialpes.fr/src/deepmatching. Note that the run time is a few seconds per images pair, which is less than for most other methods. The latest release was published in March 2014.

5.5. Mixing Body-Part Sequences for Human Pose Estimation

Participants: Cherian Anoop, Mairal Julien, Alahari Karteek, Schmid Cordelia.

The code corresponding to the publication [11] has been released as an open-source MATLAB package along with a dataset for human pose estimation in videos called “Poses in the Wild”. It is available at http://lear.inrialpes.fr/research/posesinthewild/#dataset. This dataset has 30 video sequences generated from three Hollywood movies, namely “Forrest Gump”, “The Terminal”, and “Cast Away”. Each sequence has approximately 30 frames and is manually annotated for human upper-body keypoints, namely (i) neck, (ii) left and right shoulders, (iii) left and right elbows, (iv) left and right wrists, and (v) mid-torso. In comparison to earlier evaluation datasets publicly available for this problem, Poses in the Wild is significantly more representative of real-world scenarios with background clutter, body-part occlusions, and severe camera motion.

5.6. Image Transformation Pursuit

Participants: Mattis Paulin, Jerome Revaud, Zaid Harchaoui, Florent Perronnin [XRCE], Cordelia Schmid.

This is an open-source software package corresponding to the papers [19], [23], available here http://lear.inrialpes.fr/people/paulin/projects/ITP/. The code has three main purposes. Starting from input images, it can be used to generate transformed versions to use as "virtual examples". It implements the main algorithm of the article (ITP), performing an automatic selection of a small set of transformations in order to improve classification performance. Lastly, it provides a complete classification framework, allowing to train and test a classifier on an image dataset.

5.7. Convolutional Kernel Networks

Participants: Julien Mairal, Piotr Koniusz, Zaid Harchaoui, Cordelia Schmid.

This is an open-source software package corresponding to the paper [16], available at http://ckn.gforge.inria.fr/. In this software package, convolutional neural networks are learned in an unsupervised manner. We control what the non-linearities of the network are really doing: the network tries to approximate the kernel map of a reproducing kernel.

5.8. EpicFlow

Participants: Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, Cordelia Schmid.

We developed a package for the EpicFlow method [29]. EpicFlow computes a dense correspondence field by performing a sparse-to-dense interpolation from an initial sparse set of matches, leveraging contour cues using an edge-aware geodesic distance. The resulting dense correspondence field is fed as an initial optical flow estimate to a one-level variational energy minimization. The code is written in C/C++ and is available at http://lear.inrialpes.fr/src/epicflow.

LINKMEDIA Project-Team

5. New Software and Platforms

5.1. Software

5.1.1. News and updates

5.1.1.1. Peyote

Participants: Sébastien Campion, Hervé Jégou [correspondent].

Peyote is a framework for Video and Image description, indexation and nearest neighbor search. It can be used as-is by a video-search or image-search front-end with the implemented descriptors and search modules. It can also be used via scripting for large-scale experimentation. Finally, it is modular and as such can be used for scientific experimentation on new descriptors or indexation methods. Peyote is used in the AABOT software and was used for the Mediaeval Placing task and the Trecvid Instance Search task.

Peyote is used by Lamark, a start-up that is currently being incubating at Inria Rennes.

Last APP deposit: IDDN.FR.001.420008.001.S.A.2012.000.21000.

5.1.1.2. Yael

Participant: Hervé Jégou [correspondent].

This software is jointly maintained by Matthijs Douze, from Inria Grenoble.

Yael [30] is a C/python/Matlab library providing implementations of computationally demanding functions. In particular, the library provides very optimized functions for k-means clustering and exact nearest neighbor search. It is maintained and continuously improved. This year, we have in particular added a few tutorials implementing two simple image search systems, see http://yael.gforge.inria.fr/tutorial/tuto_imgindexing.html.

The current release (v401) was registered at APP under no IDDN.FR.001.220014.002.S.P.2010.000.10000 on July 2014.

5.2. The A||GO web services

Participants: Sébastien Campion [correspondent], Guillaume Gravier.

Available at http://allgo.irisa.fr, the A||GO platform allows for the easy deployment of the technology developed in the team as web services. The engineer hired by SED in October 2013 developed several new features that enable software providers to deploy autonomously their algorithm. Dedicated hardware equipment was also purchased in 2014, composed by a main server with 1.3 TB of storage and 3 nodes for computing task and setup, and will be available in 2015. In addition to a strong involvement in the development of the platform, LINKMEDIA contributed several services.

5.3. Experimental platform

Participant: Sébastien Campion [correspondent].

Our experimental platform, consisting of dedicated equipments to experiment on very large collections of multimedia data, was upgraded in 2014. In order to replace old hardware, we acquired 380 TB of additional disk space. Divided in two categories, the first part (140 TB) must replace in 2015 our current network area storage where datasets are stored. The second part (240 TB) is dedicated to a distributed storage filesystem (CEPH), used to store our experimental results with high IO performances for use with the mesocluster IGRIDA.

LINKS Team

4. New Software and Platforms

4.1. QuiX-Tool Suite

Participants: Joachim Niehren [correspondant], Denis Debarbieux, Tom Sebastian.

The QuiX-Tool Suite provides tools to process XML streams and documents. The QuiX-Tool Suite is based on early algorithms: query answers are delivered as soon as possible and in all practical cases at the earliest time point. The QuiX-Tool Suite provides an implementation of the main XML standart over streams. XPath, XSLT, XQuery and XProc are W3C standarts while Schematron is an ISO one. The QuiX-Tool suite is developed in the Inria transfer project QuiXProc in cooperation with Innovimax. It includes among the others existing tools such as FXP and QuixPath, along with new tools, namely X-Fun. Both, a free and a professional version are available. The ownership of QuiX-Tool Suite is shared between Inria and Innovimax. The main application of QuiX-Tool Suite is its usage in QuiXProc, an professional implementation of the W3C pipeline language XProc owned by Innovimax.

The QuiXPath language is a large fragment of XPath with full support for the XML data model. The QuiXPath library provides a compiler from QuiXPath to FXP, which is a library for querying XML streams with a fragment of temporal logic.

The X-Fun language is a functional language for defining transformations between XML data trees, while providing shredding instructions. X-Fun can be understood as an extension of Frisch's XStream language with output shredding, while pattern matching is replaced by tree navigation with XPath expressions. The QuiX-Tool suite includes QuiXSLT, which is a compiler from XSLT into a fragment of X-Fun, which can be considered as the core of XSLT. It also provides QuiXSchematron, which is a compiler from Schematron to X-Fun, and QuiXQuery, which is a compiler from XQuery to X-Fun.

QuixPath now covers 100 per cent of the XPathMark, a W3C benchmark for the language Xpath (querying XML trees). In particular, it includes aggregation operators, joins and arithmetics operations.

See also the web page https://project.inria.fr/quix-tool-suite/.

- Version: QuixPath v2.0.3
- Version: X-Fun v0.5.0
- Version: QuiXSLT v0.5.0
- Version: QuiXSchematron v1.0.2

4.2. SmartHal

Participants: Joachim Niehren [correspondant], Antoine Mbaye Ndione, Guillaume Bagan.

SmartHal is a better tool for querying the HAL bibliography database, while is based on Haltool queries. The idea is that a Haltool query returns an XML document that can be queried further. In order to do so, SmartHal provides a new query language. Its queries are conjunctions of Haltool queries (for a list of laboratories or authors) with expressive Boolean queries by which answers of Haltool queries can be refined. These Boolean refinement queries are automatically translated to XQuery and executed by Saxon. A java application for extraction from the command line is available. On top of this, we have build a tool for producing the citation lists for the evaluation report of the LIFL, which can be easily adapter to other Labs.

See also the web page http://smarthal.lille.inria.fr/.

This year, Smathal has been adapted for querying the version 3.0 of Hal. Moreover, maintenance and optimization has been proceded all over the year.

- Version: SmartHal v1.0.0

MAGNET Team

5. New Software and Platforms

5.1. CoRTex

Participants: Pascal Denis [correspondent], David Chatel.

CoRTex is a LGPL-licensed Python library for Noun Phrase coreference resolution in natural language texts. This library contains implementations of various state-of-the-art coreference resolution algorithms, including those developed in our research. In addition, it provides a set of APIs and utilities for text pre-processing, reading the main annotation formats (ACE, CoNLL and MUC), and performing evaluation based on the main evaluation metrics (MUC, B-CUBED, and CEAF). As such, CoRTex provides benchmarks for researchers working on coreference resolution, but it is also of interest for developers who want to integrate a coreference resolution within a larger platform. This project is hosted on Inria gforge: https://gforge.inria.fr/projects/cortex/.

MAGRIT Project-Team

5. New Software and Platforms

5.1. Ralib

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

5.2. PoLAR

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

5.3. Ltrack

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

MAIA Project-Team

5. New Software and Platforms

5.1. AA4MM Suite

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

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

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

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

Currently, two new software are submitted to the APP:

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

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

5.2. MASDYNE

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

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

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

5.3. FiatLux

Participant: Nazim Fatès.

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

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

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

5.4. Platforms

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

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

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

MANAO Project-Team

4. New Software and Platforms

4.1. Software

4.1.1. ALTA Library

Participants: X. Granier & R. Pacanowski & L. Belcour & P. Barla

Keywords: BRDF fitting and analysis

ALTA is a multi-platform software library to analyze, fit and understand BRDFs. It provides a set of command line software to fit measured data to analytical forms, and tools to understand models and data. The targeted audience is composed of all the researchers and professionals who are working on BRDFs, and who want to benchmark new BRDF models and easily compare them with state-of-the-art BRDF models and data. It is also suitable for researchers and professionals who are working on optical measurements, and who want to experiment different fitting procedures and models, or just to perform statistical analysis on their data. The major features in the ALTA library are:

- Open common BRDF data formats (MERL, ASTM)
- Non-linear fitting of BRDF (using third party packages)
- Rational interpolation of BRDF
- Analytic BRDF models
- Scripting mechanism to automatize fitting

ALTA has been supported by the ANR ALTA (ANR-11-BS02-006).

Facts:

- Web: http://alta.gforge.inria.fr/
- License: MPLv2

4.1.2. Eigen

Participants: G. Guennebaud

Keywords: Linear algebra

Efficient numerical computation is central to many computer science domains. In particular, in computer graphics, space transformations and local regressions involve dense linear algebra, data interpolation and differential equations require sparse linear algebra, while more advanced problems involve non-linear optimization or spectral analysis. On the one hand, solutions such as MatLab are limited to prototyping. On the other hand, optimized libraries coming from the HPC (high performance computing) world are often tedious to use and more adapted for very large problems running on clusters. Moreover, all these solutions are very slow at handling very small but numerous problems which often arise in computer graphics, vision, or robotics. As a result, researchers of these domains used to waste a lot of time at either implementing their own half cooked solution, or dealing with dozens of complex to use libraries.

The objective of Eigen is to fill this gap by proposing an easy to use, efficient, and versatile C++ mathematical template library for linear algebra and related algorithms. In particular it provides fixed and dynamic size matrices and vectors, matrix decompositions (LU, LLT, LDLT, QR, eigenvalues, etc.), sparse matrices with iterative and direct solvers, some basic geometry features (transformations, quaternions, axis-angles, Euler angles, hyperplanes, lines, etc.), some non-linear solvers, automatic differentiations, etc. Thanks to expression templates, Eigen provides a very powerful and easy to use API. Explicit vectorization is performed for the SSE, AltiVec and ARM NEON instruction sets, with graceful fallback to non-vectorized code. Expression templates allow to perform global expression optimizations, and to remove unnecessary temporary objects.

Eigen is already a well established library with about 30k unique visitors of the website per month. Eigen is co-developed and maintained with a couple of other researchers and occasional contributors spread over the world. Its development started in 2008, and the last release is the 3.2 version in July 2013. Eigen has been supported by Inria through an ADT started in January 2012, and that ended in September 2013. Eigen received the **"high-quality software in geometry processing award"** from the Symposium on Geometry Processing 2013. Eigen is continuously and actively developed with this year an important refactoring of the expression evaluation mechanism, a divide & conquer SVD algorithm, support for AVX in collaboration with Google, and many other features.

Facts:

- Web: http://eigen.tuxfamily.org/
- License: MPLv2

4.1.3. PatateLib

Participants: N. Mellado, G. Ciaudo, S. Boyé, G. Guennebaud, P. Barla

Keywords: multi-scale analysis, material appearance, vector graphics, expressive rendering, 2D animation

Patate is a header only C++/CUDA library for graphics applications released under the MPL license.

It provides a collection of Computer Graphics techniques that incorporate the latest innovations from Inria research teams working in the field. It strives for efficiency and ease-of-use by focusing on low-level core operators and key algorithms, organized in modules, each tackling a specific set of issues. The central goal of the library is to drastically reduce the time and efforts required to turn a research paper into a ready-to-use solution, for both commercial and academic purposes.

Each module is initially developed by a few persons, usually those who have authored the corresponding research papers. An engineer, Gautier Ciaudo, has been recruited via the ADT program to perform unit tests, bug tracking, and make examples. Our first module provides efficient methods for the fitting and analysis of point-clouds in arbitrary dimensions. It may be used for varied purposes such as curvature computation, surface reconstruction, scale-space analysis, image processing, and sketch vectorization. More modules will be developed in 2015 by Simon Boyé.

Facts:

- Web: http://patate.gforge.inria.fr
- License: MPLv2

4.1.4. PFSTools

Participant: I. Ihrke

Keywords: high dynamic range image processing, merging, calibration and tone-mapping

The `pfstools` package is a set of command line programs for reading, writing, manipulating and viewing high-dynamic range (HDR) images and video frames. All programs in the package exchange data using a simple generic high dynamic range image format, `pfs`, and they use unix pipes to pass data between programs and to construct complex image processing operations.

`pfstools` come with a library for reading and writing pfs files. The library can be used for writing custom applications that can integrate with the existing `pfstools` programs. It also offers a good integration with high-level mathematical programming languages, such as MATLAB or GNU Octave. `pfstools` can be used as an extension for MATLAB or Octave for reading and writing HDR images or simply to effectively store large matrices. The `pfstools` package integrates existing high dynamic range image formats by providing a simple data format that can be used to exchange data between applications. It is accompanied by the `pfscalibration` and `pfstmo` packages.

The `pfscalibration` package provides algorithms for the photometric calibration of cameras and for the recovery of high dynamic range (HDR) images from a set of low dynamic range (LDR) exposures. Maintenance of the pfscalibration package is performed by Ivo Ihrke since January 2011. A major update to make the software compatible with current digital SLR cameras and their raw file formats, especially for measurement purposes, has been performed. A new set of MATLAB scripts has been developed for improved calibration performance. It is intended to merge these new procedures into the existing software.

The `pfstmo` package contains the implementation of seven state-of-the-art tone mapping operators suitable for convenient processing of both static images and animations.

The software received wider interest in the Open Source community and third party contributors prepared installation packages which are included in several Linux distributions including Debian, Fedora and Suse.
Facts:

- Web: http://pfstools.sourceforge.net/
- License: GPL

4.2. Platforms

4.2.1. COEL - Computational Optics Experimentation Laboratory

We are setting a dedicated experimentation facility up to validate our theoretical tools to design hybrid (optics & computer sciences) systems by creating real setups. Such a facility is unique thanks to the close collaboration between optics and computer science in Bordeaux. Now located in the LP2N, this laboratory consists in a set of on-the-shell elements to design optical systems combined with controllable large-band lighting systems (from pure white sources, to tunable lasers and video-projectors), with a fabrication laboratory to build non-conventional components, with large-scale mechanical elements, with display technologies, and high-performance processing resources.

After initial delays, the lab has now found its final location in LP2N. The basic equipment is in place and first experiments are being performed. We still have to work on the illumination conditions in the room, as well as on the construction of a light-sealed control area inside the experimentation room for independent experiments.

The construction and equipment is financed by a special regional grant of the "Conseil Régional d'Acquitaine" (Carer xD) in conjunction with project-specific funds.

MAVERICK Project-Team

4. New Software and Platforms

4.1. Introduction

Maverick insists on sharing the software that is developed for internal use. These are all listed in a dedicated section on the web site http://artis.imag.fr/Software.

4.2. Gratin

Participant: Romain Vergne [contact].

Gratin is a node-based compositing software for creating, manipulating and animating 2D and 3D data. It uses an internal direct acyclic multi-graph and provides an intuitive user interface that allows to quickly design complex prototypes. Gratin has several properties that make it useful for researchers and students. (1) it works in real-time: everything is executed on the GPU, using OpenGL, GLSL and/or Cuda. (2) it is easily programmable: users can directly write GLSL scripts inside the interface, or create new C++ plugins that will be loaded as new nodes in the software. (3) all the parameters can be animated using keyframe curves to generate videos and demos. (4) the system allows to easily exchange nodes, group of nodes or full pipelines between people. In a research context, Gratin aims at facilitating the creation of prototypes, testing ideas and exchanging data. For students, Gratin can be used to show real-time demos/videos, or help learning how to program with the GPU. Gratin has already been used for creating new computer graphics tools but also for designing perceptual experiments. Most of the work published by R. Vergne was done with Gratin.

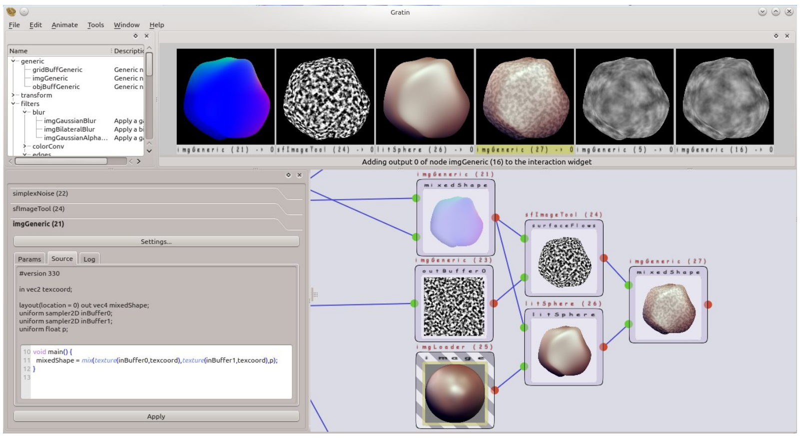


Figure 2. Gratin interface.

4.3. PlantRad

Participant: Cyril Soler [contact].

PlantRad is a software program for computing solutions to the equation of light equilibrium in a complex scene including vegetation. The technology used is hierarchical radiosity with clustering and instantiation. Thanks to the latter, PlantRad is capable of treating scenes with a very high geometric complexity (up to millions of polygons) such as plants or any kind of vegetation scene where a high degree of approximate self-similarity permits a significant gain in memory requirements. Its main domains of applications are urban simulation, remote sensing simulation (See the collaboration with Noveltis, Toulouse) and plant growth simulation, as previously demonstrated during our collaboration with the LIAMA, Beijing.

4.4. High Quality Renderer

Participant: Cyril Soler [contact].

In the context of the European project RealReflect, the Maverick team has developed the HQR software based on the photon mapping method which is capable of solving the light balance equation and of giving a high quality solution. Through a graphical user interface, it reads X3D scenes using the X3DToolKit package developed at Maverick, it allows the user to tune several parameters, computes photon maps, and reconstructs information to obtain a high quality solution. HQR also accepts plugins which considerably eases the developpement of new algorithms for global illumination, those benefiting from the existing algorithms for handling materials, geometry and light sources. HQR is freely available for download at http://artis.imag.fr/~Cyril.Soler/HQR.

4.5. MobiNet

Participants: Fabrice Neyret [contact], Joëlle Thollot.

The MobiNet software allows for the creation of simple applications such as video games, virtual physics experiments or pedagogical math illustrations. It relies on an intuitive graphical interface and language which allows the user to program a set of mobile objects (possibly through a network). It is available in public domain at http://mobinet.inrialpes.fr for Linux,Windows and MacOS, and originated in a collaboration with the EVASION project-team.

The main aim of MobiNet is to allow young students at high school level with no programming skills to experiment, with the notions they learn in math and physics, by modeling and simulating simple practical problems, and even simple video games. This platform has been massively used during the Grenoble INP "engineer weeks" since 2002: 150 senior high school pupils per year, doing a 3 hour practice. This work is partly funded by Grenoble INP. Various contacts are currently developed in the educational world. Besides "engineer weeks", several groups of "monitors" PhD students conducts experimentations based on MobiNet with a high scool class in the frame of the courses. Moreover, presentation in workshops and institutes are done, and a web site repository is maintained. A web version is currently under preliminary developpement.

4.6. Freestyle

Participant: Joëlle Thollot [contact].

Freestyle is a software for Non-Photorealistic Line Drawing rendering from 3D scenes (Figure 3). It is designed as a programmable interface to allow maximum control over the style of the final drawing: the user "programs" how the silhouettes and other feature lines from the 3D model should be turned into stylized strokes using a set of programmable operators dedicated to style description. This programmable approach, inspired by the shading languages available in photorealistic renderers such as Pixar's RenderMan, overcomes the limitations of integrated software with access to a limited number of parameters and permits the design of an infinite variety of rich and complex styles. The system currently focuses on pure line drawing as a first step. The style description language is Python augmented with our set of operators. Freestyle was developed in the framework of a research project dedicated to the study of stylized line drawing rendering from 3D scenes. This research has lead to two publications [18], [19].

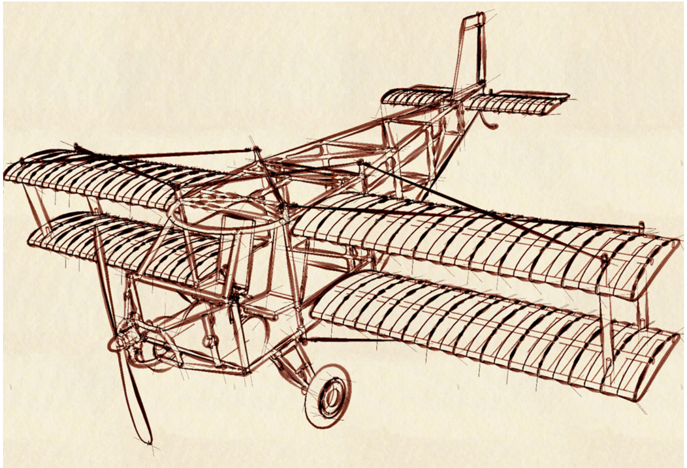


Figure 3. Stylized plane using Freestyle.

In 2008, Freestyle get a new life, completely outside Maverick or Inria: it was the basis of one of the 6 *Google Summer of Code* projects awarded to the *Blender Foundation* [0]! The goal of the project was to integrate Freestyle to the well known free 3D modeler *Blender*, as its standard NPR line-drawing renderer. Maxime Curioni (under the mentoring of Jean-Luc Peurière from the *Blender Foundation*), is currently making the integration. First beta versions are publicly available, and tested by enthusiasts around the web.

4.7. Diffusion Curves

Participant: Joëlle Thollot [contact].

We provide an implementation of the vector drawing tool described in our Diffusion Curves Siggraph paper [2] (Figure 4). This prototype is composed of the Windows binary, along with the required shader programs (ie. in source code). The software is available for download at http://artis.imag.fr/Publications/2008/OBWBTS08 for free, for non-commercial research purposes.

4.8. VRender: vector figures

Participant: Cyril Soler [contact].

The VRender library is a simple tool to render the content of an OpenGL window to a vectorial device such as Postscript, XFig, and soon SVG. The main usage of such a library is to make clean vectorial drawings for publications, books, etc.

In practice, VRender replaces the z-buffer based hidden surface removal of OpenGL by sorting the geometric primitives so that they can be rendered in a back-to-front order, possibly cutting them into pieces to solve cycles.

VRender is also responsible for the vectorial snapshot feature of the QGLViewer library. VRender is released under the LGPL licence and is freely available for download at http://artis.imag.fr/Software/VRender.

4.9. ProLand

Participants: Fabrice Neyret [contact], Eric Bruneton.

[0] http://www.blender.org/

Figure 4. Diffusion curves freely downloadable demo.

Now available at http://proland.inrialpes.fr/ in double licencing GPL/commercial.

Proland (for procedural landscape) is a software platform originally developed at the Evasion team-project by Eric Bruneton, and currently funded by the ANR-JCJC SimOne. The goal of this platform is the real-time quality rendering and editing of large landscapes. All features can work with planet-sized terrains, for all viewpoints from ground to space. Most of the work published by Eric Bruneton and Fabrice Neyret has been done within Proland, and a large part has been integrated in the main branch. Several licences have been transfered to companies. Eric Bruneton was hired by Google-Zürich in september 2011, but will be able to keep some participation in the project.

4.10. GigaVoxels

Participants: Fabrice Neyret [contact], Prashant Goswami, Jérémy Sinoir, Cyril Crassin, Pascal Guehl, Paul Gannay, Eric Heitz.

Soon available at http://gigavoxels.inrialpes.fr/index.htm in double licencing GPL/commercial.

Gigavoxel is a software platform initiated from the PhD work of Cyril Crassin, and currently funded by the ANR CONTINT RTIGE (Figure 5). The goal of this platform is the real-time quality rendering of very large and very detailed scenes which couldn't fit memory. Performances permit showing details over deep zooms and walk through very crowdy scenes (which are rigid, for the moment). The principle is to represent data on the GPU as a Sparse Voxel Octree which multiscale voxels bricks are produced on demand only when necessary and only at the required resolution, and kept in a LRU cache. User defined producer lays accross CPU and GPU and can load, transform, or procedurally create the data. Another user defined function is called to shade each voxel according to the user-defined voxel content, so that it is user choice to distribute the appearance-making at creation (for faster rendering) or on the fly (for storageless thin procedural details). The efficient rendering is done using a GPU differential cone-tracing using the scale corresponding to the 3D-MIPmapping LOD, allowing quality rendering with one single ray per pixel. Data is produced in case of cache miss, and thus only whenever visible (accounting for view frustum and occlusion). Soft-shadows and depth-of-field is easily obtained using larger cones, and are indeed cheaper than unblurred rendering. Beside the representation, data management and base rendering algorithm themself, we also worked on realtime light transport, and on quality prefiltering of complex data. Ongoing researches are addressing animation. GigaVoxels is currently used for the quality real-time exploration of the detailed galaxy in ANR RTIGE. This work led to several publications and several licences have been sold to companies.

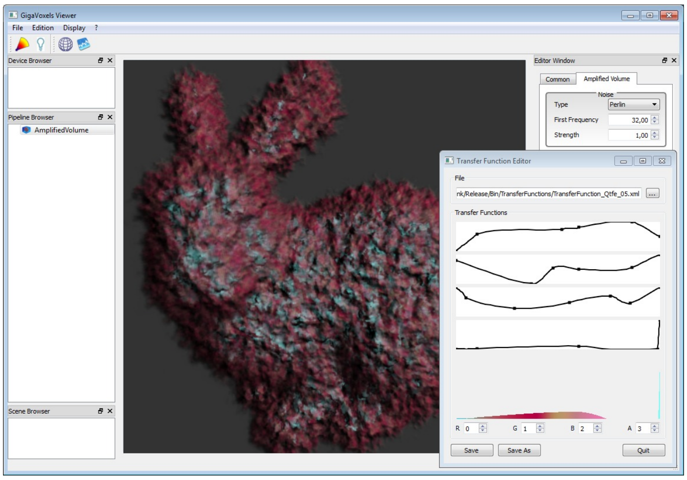


Figure 5. GigaVoxels freely downloadable demo.

MIMETIC Project-Team

5. New Software and Platforms

5.1. Populate

Participants: Carl Jorgensen, Fabrice Lamarche [contact].

Populate is a toolkit dedicated to task scheduling under time and space constraints in the field of behavioral animation. It is currently used to populate virtual cities with pedestrian performing different kind of activities implying travels between different locations. However the generic aspect of the algorithm and underlying representations enable its use in a wide range of applications that need to link activity, time and space. The main scheduling algorithm relies on the following inputs: an informed environment description, an activity an agent needs to perform and individual characteristics of this agent. The algorithm produces a valid task schedule compatible with time and spatial constraints imposed by the activity description and the environment. In this task schedule, time intervals relating to travel and task fulfillment are identified and locations where tasks should be performed are automatically selected.

The software provides the following functionalities:

- A high level XML dialect that is dedicated to the description of agents activities in terms of tasks and sub activities that can be combined with different kind of operators : sequential, without order, interlacedŠThis dialect also enables the description of time and location constraints associated to tasks.
- An XML dialect that enables the description of agent's personal characteristics.
- An informed graph describes the topology of the environment as well as the locations where tasks can be performed. A bridge between TopoPlan and Populate has also been designed. It provides an automatic analysis of an informed 3D environment that is used to generate an informed graph compatible with Populate.
- The generation of a valid task schedule based on the previously mentioned descriptions.

With a good configuration of agents characteristics (based on statistics), we demonstrated that tasks schedules produced by Populate are representative of human ones. In conjunction with TopoPlan, it has been used to populate a district of Paris as well as imaginary cities with several thousands of pedestrians navigating in real time.

MINT Project-Team

5. New Software and Platforms

5.1. LibGINA

Participant: Laurent Grisoni [correspondant].

This year we used it with Ankama SME for a 3D videogame installation (La mine), done in collaboration with Idées-3com and LightUp. The library architecture has been rethought in order to provide ease of use and genericity.

Current version: version 1.1

Software characterization: A-2 SO-3 SM-2-up EM-3 SDL-3 OC-DA4-CD4-MS2-TPM4

5.2. 3D interaction using mobile phone

Participants: Samuel Degrande [correspondant], Laurent Grisoni.

This work has been achieved in the context of the Idées-3com I-lab. In this context a module, that allows to use any android based smartphone to control an Explorer module for navigation and interaction with VRML-based content. This module was used as a basis by Idées-3com in their commercial product this year.

Current version: version 1.0

Software characterization: A-2 SO-3 SM-2-up EM-2-up SDL-3 OC-DA4-CD4-MS2-TPM4

5.3. tIO (tactile input & output)

Participants: Marc-Antoine Dupre, Matthieu Falce, Nicolas Roussel [correspondant], Takashi Miyaki.

tIO is a library designed to facilitate the implementation of doubly tactile interaction techniques (tactile input coupled with tactile feedback) based on the STIMTAC technology. Supporting all current STIMTAC prototypes, it makes it easy to move the system pointer of the host computer according to motions detected on them and adapt their vibration amplitude based on the color of the pointed pixel or the nature of the pointed object. The library includes a set of demo applications that illustrate these two different approaches and makes it easy to “augment” existing `Qt` applications with tactile feedback. It also makes it possible to supplement or substitute tactile feedback with basic auditory feedback synthesized using `portaudio` (friction level is linearly mapped to the frequency of a sine wave). This not only facilitates the development and documentation of tactile-enhanced applications but also makes it easier to demonstrate them to a large audience.

Software characterization: A2, SO3-up, SM-2, EM2, SDL1.

5.4. libpointing

Participants: Géry Casiez [correspondant], Damien Marchal, Nicolas Roussel, Izzatbek Mukhanov.

Libpointing is a software toolkit that provides direct access to HID pointing devices and supports the design and evaluation of pointing transfer functions [2]. The toolkit provides resolution and frequency information for the available pointing and display devices and makes it easy to choose between them at run-time through the use of URIs. It allows to bypass the system’s transfer functions to receive raw asynchronous events from one or more pointing devices. It replicates as faithfully as possible the transfer functions used by Microsoft Windows, Apple OS X and Xorg (the X.Org Foundation server). Running on these three platforms, it makes it possible to compare the replicated functions to the genuine ones as well as custom ones. The toolkit is written in C++ with Python and Java bindings available. It is publicly available under the GPLv2 license.

Izzatbek Mukhanov was recruited in October 2014 for two years as an engineer (IJD) to support the development and deployment of the library.

Web site: http://libpointing.org/

Software characterization: A3, SO3, SM-2, EM2, SDL4

5.5. PIRVI platform

Participants: Fabrice Aubert [correspondant], Damien Marchal.

MINT participates to the PIRVI platform (Framework for Computer Human Animation, Virtual Reality and Images, which aims at promoting research achieved by participant research teams (6 research teams, among which MINT), as well as encouraging collaborations with regional economical tissue on the knowledge fields covered within the associated research teams. The PIRVI allows these research teams to share a Virtual-Reality Room and various mid-size research equipments : multitouch tables, cameras (depth, infrared, ...), interactive devices (force-feedback, multitouch, smartphones...), a configurable multitouch wall.

MORPHEO Project-Team

5. New Software and Platforms

5.1. Software packages

5.1.1. Shape Tracking

We are developing a software suite to track shapes over temporal sequences. The motivation is to provide temporally coherent 4D Models, i.e. 3D models and their evolutions over time , as required by motion related applications such as motion analysis. This software takes as input a temporal sequence of 3D models in addition to a template and estimate the template deformations over the sequence that fit the observed 3D models. This software is particularly developed in the context of the FUI project Creamove.

5.1.2. LucyViewer

Lucy Viewer http://4drepository.inrialpes.fr/lucy_viewer/ is an interactive viewing software for 4D models, i.e, dynamic three-dimensional scenes that evolve over time. Each 4D model is a sequence of meshes with associated texture information, in terms of images captured from multiple cameras at each frame. Such data is available from various websites over the world including the 4D repository website hosted by Inria Grenoble http://4drepository.inrialpes.fr/. The software was developed in the context of the European project iGlance, it is available as an open source software under the GNU LGP Licence.

5.1.3. Ethomice

Ethomice http://morpheo.inrialpes.fr/people/reveret/ethomice/ is a motion analysis software to characterize motor behavior of small vertebrates such as mice or rats. From a multiple views video input, a biomechanical model of the skeleton is registered. Study on animal model is the first important step in Biology and Clinical research. In this context, the analysis of the neuro-motor behaviour is a frequent cue to test the effect of a gene or a drug. Ethomice is a platform for simulation and analysis of the small laboratory animal, such as rat or mouse. This platform links the internal skeletal structure with 3D measurements of the external appearance of the animal under study. From a stream of multiple views video, the platform aims at delivering a three dimensional analysis of the body posture and the behaviour of the animal. The software was developed by Lionel Reveret and Estelle Duveau. An official APP repository has been issued this year.

5.2. Databases

5.2.1. 4D repository (http://4drepository.inrialpes.fr/)

This website hosts dynamic mesh sequences reconstructed from images captured using a multi-camera set up. Such mesh-sequences offer a new promising vision of virtual reality, by capturing real actors and their interactions. The texture information is trivially mapped to the reconstructed geometry, by back-projecting from the images. These sequences can be seen from arbitrary viewing angles as the user navigates in 4D (3D geometry + time) . Different sequences of human / non-human interaction can be browsed and downloaded from the data section. A software to visualize and navigate these sequences is also available for download.

5.3. Platforms

5.3.1. Platform Grimage

The Grimage platform is an experimental multi-camera platform dedicated to spatio-temporal modeling including immersive and interactive applications. It hosts a multiple-camera system connected to a PC cluster, as well as visualization facilities including head mounted displays. This platform is shared by several research groups, most proeminently Moais, Morpheo and Perception. In particular, Grimage allows challenging real-time immersive applications based on computer vision and interactions between real and virtual objects, Figure 1 . Note that the Grimage platform, while still active in 2014, is now replaced by the Kinovis platform that exhibit a larger acquisition space and better acquisition facilities.

Figure 1. Platform: the Grimage acquisition.

5.3.2. Platform Kinovis

Kinovis (http://kinovis.inrialpes.fr/) is a new multi-camera acquisition project that was was selected within the call for proposals ”Equipements d’Excellence” of the program “Investissement d’Avenir˝ funded by the French government. The project involves 2 institutes: the Inria Grenoble Rhône-Alpes, the université Joseph Fourier and 4 laboratories: the LJK(laboratoire Jean Kuntzmann - applied mathematics), the LIG (laboratoire d’informatique de Grenoble - Computer Science), the Gipsa lab (Signal, Speech and Image processing) and the LADAF (Grenoble Hospitals - Anatomy). The Kinovis environment will be composed of 2 complementary platforms. A first platform located at the Inria Grenoble will have a 10mx10m acquisition surface and will be equipped with 60 cameras. It is the evolution of the Grimage platform previously described towards the production of better models of more complex dynamic scenes. A second platforms located at Grenoble Hospitals, within the LADAF anatomy laboratory, will be equipped with both color and X-ray cameras to enable combined analysis of internal and external shape structures, typically skeleton and bodies of animals. Installation works of both platforms started in 2013 and are now finished. Members of Morpheo are highly involved in this project. Edmond Boyer is coordinating this project and Lionel Reveret is in charge of the LADAF platform. Thomas Pasquier and Julien Pansiot are managing the technical resources of both platforms.

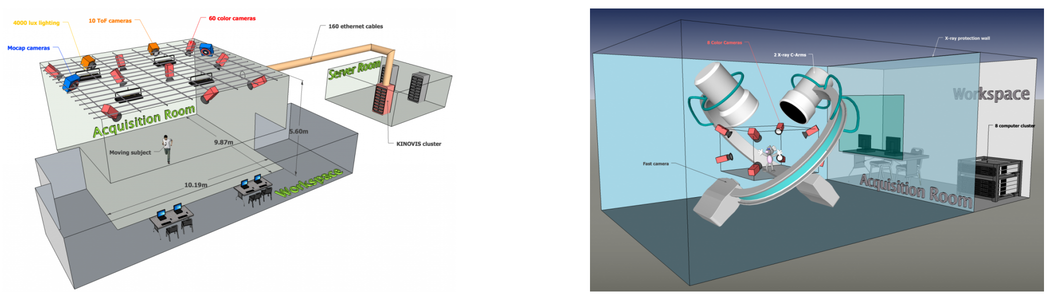


Figure 2. Kinovis platforms: on the left the Inria platform; on the right Grenoble Hospital platform.

5.3.3. Multicamera platform for video analysis of mice behavior

This project is a follow-up of the experimental set-up developed for a CNES project with Mathieu Beraneck from the CESeM laboratory (centre for the study of sensorimotor control, CNRS UMR 8194) at the Paris-

Descartes University. The goal of this project was to analyze the 3D body postures of mice with various vestibular deficiencies in low gravity condition (3D posturography) during a parabolic flight campaign. The set-up has been now adapted for new experiments on motor-control disorders for other mice models. This experimental platform is currently under development for a broader deployment for high throughput phenotyping with the technology transfer project ETHOMICE. This project involves a close relationship with the CESeM laboratory and the European Mouse Clinical Institute in Strasbourg (Institut Clinique de la Souris, ICS).

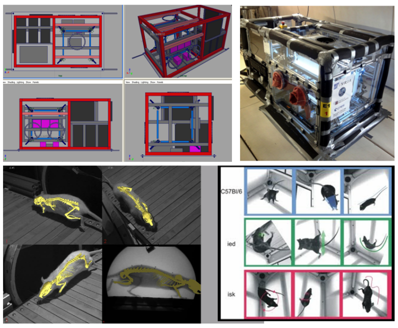


Figure 3. Ethomice: Experimental platform for video analysis of mice behavior.

MULTISPEECH Team

5. New Software and Platforms

5.1. Introduction

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

5.2. Speech processing tools

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

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

5.2.1. ANTS (Automatic News Transcription System)

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

5.2.2. FASST (Flexible Audio Source Separation Toolbox)

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

5.2.3. KAM (Kernel Additive Modelling)

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

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

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

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

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

5.2.5. CoALT (Comparing Automatic Labeling Tools)

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

5.2.6. SoJA (Speech synthesis platform in Java)

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

5.3. Speech visualization tools

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

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

5.3.1. SNOORI: speech analysis and visualization software

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

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

5.3.2. VisArtico: Visualization of EMA Articulatory data

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

[0] http://astali.loria.fr
[0] http://soja-tts.loria.fr
[0] http://www.loria.fr/~laprie/WinSnoori/
[0] http://visartico.loria.fr/

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

5.3.3. Xarticulators: delineation of speech articulators in medical images

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

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

5.4. Data acquisition

Participants: Vincent Colotte, Slim Ouni, Yves Laprie.

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

5.4.1. JCorpusRecorder

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

5.4.2. EMA acquisition platform

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

5.4.3. MRI acquisition platform

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

OAK Project-Team

5. New Software and Platforms

5.1. Amada

Name: Amada (https://team.inria.fr/oak/amada/)

Contact: Jesús Camacho-Rodríguez (jcamachor[at]gmail.com))

Other contacts: Ioana Manolescu (ioana.manolescu[at]inria.fr), Dario Colazzo (dario.colazzo[at]dauphine.fr), François Goasdoué (fg[at]irisa.fr)

Presentation: A platform for Web data management in the Amazon cloud.

5.2. PAXQuery

Name: PAXQuery (https://team.inria.fr/oak/projects/paxquery/)

Contact: Jesús Camacho-Rodríguez (jcamachor[at]gmail.com))

Other contacts: Ioana Manolescu (ioana.manolescu[at]inria.fr), Dario Colazzo (dario.colazzo[at]dauphine.fr), Juan Alvaro M. Naranjo (juan-alvaro.munoz-naranjo[at]inria.fr)

Presentation: A system for the massively parallel processing of XQuery queries, developed as an extension of the Apache Flink system (http://flink.apache.org/)

5.3. CliqueSquare

Name: CliqueSquare (https://team.inria.fr/oak/projects/cliquesquare/)

Contact: Stamatis Zampetakis (stamatis.zampetakis[at]inria.fr))

Other contacts: Ioana Manolescu (ioana.manolescu[at]inria.fr), François Goasdoué (fg[at]irisa.fr), Benjamin Djahandideh (benjamin.djahandideh[at]inria.fr)

Presentation: A system for the massively parallel evaluation of conjunctive SPARQL queries, built on top of Hadoop. The system has been released in open-source: https://sourceforge.net/projects/cliquesquare/.

5.4. FactMinder

Name: FactMinder (http://tripleo.saclay.inria.fr/xr/demo/)

Contact: Ioana Manolescu (ioana.manolescu[at]inria.fr)

Other contacts: Stamatis Zampetakis (stamatis.zampetakis[at]inria.fr), François Goasdoué (fg[at]irisa.fr).

Presentation: A system for archiving, annotating, and querying semantic-rich Web content.

5.5. Nautilus Analyzer

Name: Nautilus Analyzer (http://nautilus.saclay.inria.fr/)

Contact: Melanie Herschel (melanie.herschel[at]lri.fr)

Other contacts: n.a.

Presentation: A tool for analyzing and debugging SQL queries using why-provenance and why-not provenance.

5.6. PigReuse

Name: PigReuse

Contact: Jesús Camacho-Rodríguez (jcamachor[at]gmail.com)

Other contacts: Ioana Manolescu (ioana.manolescu[at]inria.fr), Dario Colazzo (dario.colazzo[at]dauphine.fr)

Presentation: A PigLatin optimization tool based on identifying and sharing repeated subexpressions.

5.7. WARG

Name: WARG (https://team.inria.fr/oak/warg/)

Contact: Alexandra Roatiş (alexandra.roatis[at]gmail.com)

Other contacts: Ioana Manolescu (ioana.manolescu[at]inria.fr), Sejla Cebiric (sejla.cebiric[at]inria.fr), François Goasdoué (fg[at]irisa.fr)

Presentation: A platform for specifying and exploiting warehouses of RDF data.

ORPAILLEUR Project-Team

5. New Software and Platforms

5.1. Generic Symbolic KDD Systems

5.1.1. The Coron Platform

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

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

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

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

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

5.1.2. Orion: Skycube Computation Software

Participant: Chedy Raïssi [contact person].

Keywords: skyline, skycube

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

5.2. Stochastic systems for knowledge discovery and simulation

5.2.1. The CarottAge System

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

Keywords: Hidden Markov Models, stochastic process

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

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

5.2.2. The ARPEnTAge System

Participant: Jean-François Mari [contact person].

Keywords: Hidden Markov Models, stochastic process

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

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

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

5.3. KDD in Systems Biology

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

5.3.1. IntelliGO Online

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

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

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

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

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

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

5.4. Knowledge-Based Systems and Semantic Web Systems

5.4.1. The Kasimir System for Decision Knowledge Management

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Participant: Emmanuel Nauer [contact person].

Keywords: tests, non-regression, knowledge evolution

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

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

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

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

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

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

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

PANAMA Project-Team

5. New Software and Platforms

5.1. FASST: a Flexible Audio Source Separation Toolbox

Participants: Nancy Bertin, Frédéric Bimbot.

Emmanuel Vincent [contact person]

FASST is a Flexible Audio Source Separation Toolbox, designed to speed up the conception and automate the implementation of new model-based audio source separation algorithms.

FASST development was jointly achieved by the PAROLE team in Nancy and the TEXMEX team in Rennes through an Inria funded ADT (Action de Développement Technologique). PANAMA contributed to the development by coordinating and performing user tests, and to the dissemination in a Show-and-Tell ICASSP poster [58].

While the first implementation was in Matlab, the new implementation is in C++ (for core functions), with Matlab and Python user scripts. Version 2, including speedup and new features was released in 2014 and can be downloaded from http://bass-db.gforge.inria.fr/fasst/.

PERCEPTION Project-Team

4. New Software and Platforms

4.1. The MIXCAM Hardware/Software Platform

We developed a multiple camera platform composed of both high-definition color cameras and low-resolution depth cameras. This platform combines the advantages of the two camera types. On one side, depth (time-of-flight) cameras provide coarse low-resolution 3D scene information. On the other side, depth and color cameras can be combined such as to provide high-resolution 3D scene reconstruction and high-quality rendering of textured surfaces. The software package developed during the period 2011-2014 contains the calibration of TOF cameras, alignment between TOF and color cameras, TOF-stereo fusion, and image-based rendering. These software developments were performed in collaboration with the Samsung Advanced Institute of Technology, Seoul, Korea. The multi-camera platform and the basic software modules are products of 4D Views Solutions SAS, a start-up company issued from the PERCEPTION group.

Website: https://team.inria.fr/perception/mixcam-lab/

Figure 2. The MIXCAM laboratory is a multiple-camera multiple-PC hardware/software platform that combines high-resolution color (RGB) cameras with low-resolution time-of-flight (TOF) cameras. The cameras are arranged in "units", where each unit is composed of two RGB cameras and one TOF camera (left image). Currently the system is composed of four such units (right image), or a total of eight RGB and four TOF cameras. Over years, in collaboration with 4D View Solutions, we have developed and maintained software packages for camera, multiple-camera, and cross-modal calibration, 3D reconstruction, multiple-camera stereo, TOF-stereo fusion, and image-based rendering.

4.2. Audiovisual Robots and Heads

We have developed two audiovisual (AV) robot heads: the POPEYE head and the NAO stereo head. Both are equipped with a binocular vision system and with four microphones. The software modules comprise stereo matching and reconstruction, sound-source localization and audio-visual fusion. POPEYE has been developed within the European project POP (https://team.inria.fr/perception/pop/) in collaboration with the project-team MISTIS and with two other POP partners: the Speech and Hearing group of the University of Sheffield and the Institute for Systems and Robotics of the University of Coimbra. The NAO stereo head was developed under the European project HUMAVIPS (http://humavips.inrialpes.fr) in collaboration with Aldebaran Robotics (which manufactures the humanoid robot NAO) and with the University of Bielefeld, the Czech Technical Institute, and IDIAP. The software modules that we develop are compatible with both these robot heads [33].

For more information on POPEYE and on NAO please visit https://team.inria.fr/perception/popeye/ and https://team.inria.fr/perception/nao/.

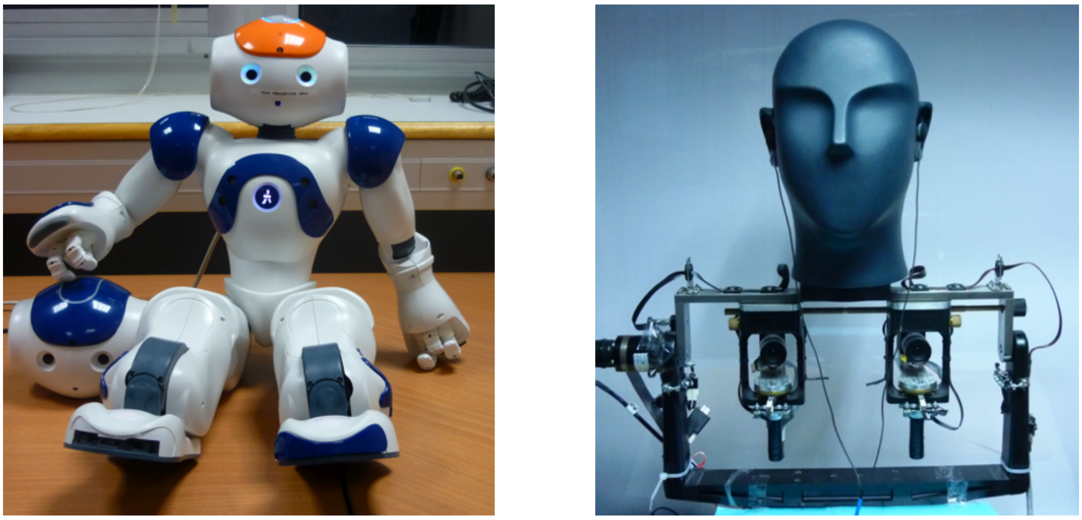


Figure 3. Left: The consumer humanoid robot NAO is equipped with a binocular-binaural head specially designed for human-humanoid interaction; Right: The binocular-binaural robot head POPEYE equipped with a four degrees of freedom stereo camera pair and with an acoustic dummy head.

POTIOC Project-Team

5. New Software and Platforms

5.1. PapARt

Participant: Jérémy Laviole [Main developer].

As part of his thesis work, Jérémy Laviole has developed a software suite for PapARt : Paper Augmented Reality Toolkit. This work is being extended to become a state-of-the-art library for projection mapping (spatial augmented reality) and tangible interfaces.

Papart is a Processing [0] library, and follows the Processing philosophy of openness and ease of use. The main features are :

1. Augmented reality rendering that enables rendering for cameras and projectors.
2. Tracking for Augmented Reality : tracking from ARToolkitPlus [0], and OpenCV SURF [0]. Extensions with other libraries are planned.
3. Camera support : in addition to the Processing Video library, PapARt support for video from OpenCV, OpenKinect, FFMPEG, FlyCapture and more is available from JavaCV [0]. It provides a wide support on Mac, Windows, Linux and possibly Android.
4. "Tactile" input on planar surface : Touch and hovering can be detected by a depth camera such as Kinect [0], the current support is Kinect XBOX 360 with openKinect drivers. Extensions are planned for Kinect 2, Kinect for Windows, Microsoft SDK (on Windows), DepthSense and most consumer market depth cameras. It scales elegantly from touch input from finger on small surfaces (such as A3 size) to large surfaces (2m x 3m size).
5. Software infrastructure to create "paper touch screens", following Processing's methods to create drawings and interactive experiences.

Technical challenges for the next few years :

- Color camera, depth camera and projector calibration made easy and more automated.
- Software and hardware installation of such cameras documented with tutorials and technical advices.

Research questions and challenges :

- Creation of tangible interfaces, tangibles elements can be tracked from cameras and depth cameras.
- Capture of part of pieces of paper for image analysis. E.g. to analyse and monitor drawings.
- Interactive projection mapping is an active research field, and such tools could power new research projects.

website: http://papart.gforge.inria.fr

5.2. OpenViBE

Participants: Fabien Lotte [local correspondant], Alison Cellard [engineer].

[0] http://www.processing.org
[0] https://launchpad.net/artoolkitplus
[0] http://opencv.org/
[0] http://bytedeco.org/
[0] http://www.microsoft.com/en-us/kinectforwindows/

Figure 4. Example of a 3D user interface where a child manipulates a 3D scene projected on a sheet of paper to prepare a drawing. Palais de la découverte, Paris, Nov. 2011.

As part of our research work on BCI, we contribute to the development of the OpenViBE [0] software, which is an open source platform dedicated to the design, evaluation and use of BCI for real and virtual applications. OpenViBE development is led by Inria, and Potioc is one of the Inria team contributing to its evolution. Moreover, Potioc is involved in the Inria ADT (Technological Development Action) OpenViBE-NT and OpenViBE-X that is dedicated to the development of OpenViBE together with 3 other Inria teams (Hybrid, Athena, Neurosys). In 2014, we developped new EEG signal processing modules for the OpenViBE software (connectivity measured, wavelets, signal denoising, etc.) and new EEG visualization tools. We also organized demonstrations and workshops about OpenViBE at international conferences (PhyCS 2014, International BCI conference 2014).

[0] http://openvibe.inria.fr

PRIMA Project-Team

4. New Software and Platforms

4.1. OMiSCID Middleware for Distributed Multimodal Perception

Participants: Amaury Negre, Patrick Reignier, Dominique Vaufreydaz [correspondant].

Middleware, Distributed perceptual systems

OMiSCID is lightweight middleware for dynamic integration of perceptual services in interactive environments. This middleware abstracts network communications and provides service introspection and discovery using DNS-SD (DNS-based Service Discovery). Services can declare simplex or duplex communication channels and variables. The middleware supports the low-latency, high-bandwidth communications required in interactive perceptual applications. It is designed to allow independently developed perceptual components to be integrated to construct user services. Thus our system has been designed to be cross-language, cross-platform, and easy to learn. It provides low latency communications suitable for audio and visual perception for interactive services.

OMiSCID has been designed to be easy to learn in order to stimulate software reuse in research teams and is revealing to have a high adoption rate. To maximize this adoption and have it usable in projects involving external partners, the OMiSCID middleware has been released under an open source licence. To maximize its target audience, OMiSCID is available from a wide variety of programming languages: C++, Java, Python and Matlab. A website containing information and documentations about OMiSCID has been set up to improve the visibility and promote the use of this middleware.

4.2. Pal Middleware

Participants: Amaury Negre, Dominique Vaufreydaz [correspondant].

Middleware, Distributed perceptual systems, Robotic Operating System (ROS), IPL PAL

A part of our efforts in the PAL project has been put toward developing a solution that would ease the integration of our multi-partners' software components.

The design of PAL Middleware responds to a requirement that within the PAL project, each partner is responsible for maintaining 1) its software heritage 2) its resources 3) its competences and fields of research and expertise; 4) current practices in terms of programming language, (c/c++, Java, Python), computing platforms (OSx, Linux, Windows, Android, etc.) and interconnect software components (OSGi, OMiSCID, MPI, PVM, etc.); and 5) its particular needs and constraints.

For it to be widely accepted, the PAL middleware must be designed to be ecologic and pragmatic. Ecologic in the sense that the solution does not perturb the ecology of each ecosystem, pragmatic in the sense that setting up this solution did not require an heavy development effort, also because PAL and is required to reuse existing software solutions.

For developing PALGate we introduced a novel concept: software gate. Unlike software components/services which can be instantiated, a software gate is only a concept, it is defined as an ecologic and hermetic interface between different ecosystems. A software gate is characterized by the subset of functionalities it exposes to other gates, where the functionalities it exposes are provided by the software components/services of its belonging ecosystem. A software gate is hermetic in the sense that only a selected subset of functionalities of an ecosystem are exposed but also because it propagates only filtered information exposed by other gates into its ecosystem. The last characteristic of a software gate is that it makes explicit to other gates the communication mechanisms it uses.

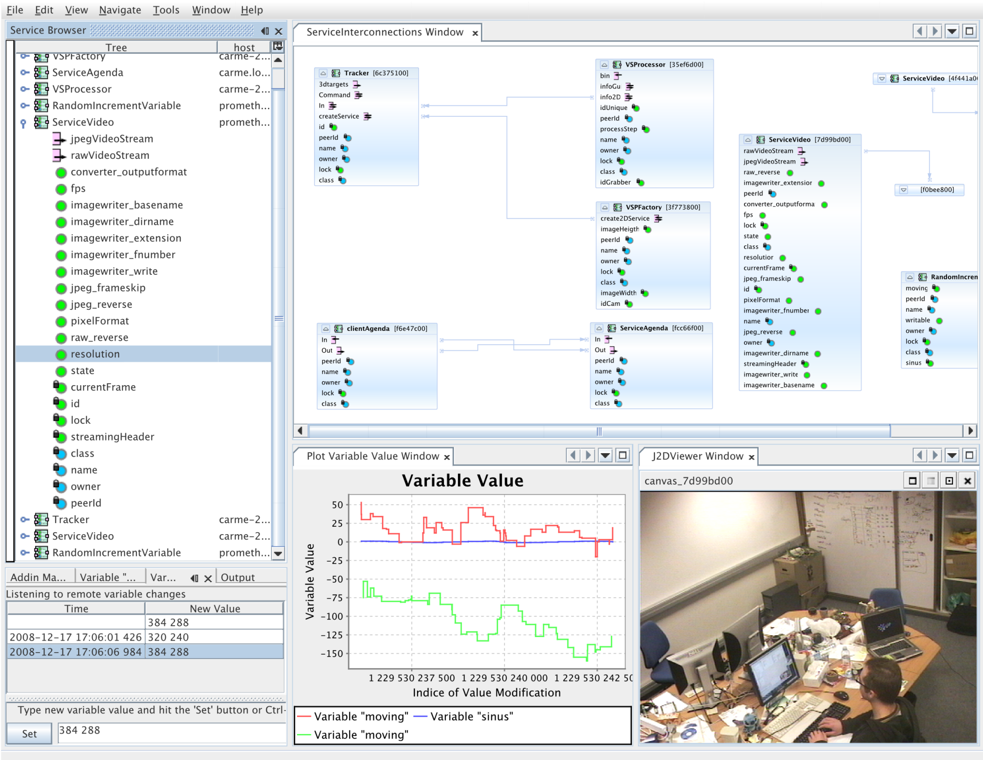


Figure 2. OMiSCID GUI showing a list of running services and some modules for service interconnections, variable plotting, live video stream display and variable control

While a software gate is only conceptual, the PAL middleware is an implementation of a gate oriented middleware. The PAL Middleware uses ROS to support the basic communication between gates. Within PALGate, each ecosystem is associated to only one software gate. Practically, PAL middleware 1) is a ROS stack containing gates definition 2) is a set of conventions (e.g. stack organization, package/node/topic/service names, namespaces, etc.) 3) it provides dedicated tools to ease the integration and its usage by partners. A software gate in PAL is a ROS package containing definition of ROS types (i.e. msgs and srvs types), but also exposed ROS communication channels (i.e. topics and RPCs).

With this architecture each partner has to provide the PAL middleware with a package containing the definition of its gate. Then in order a) to expose functionalities out of their ecosystem and b) to propagate information into their ecosystem, each partner must create ROS nodes. These ROS nodes let each partner interface their ecosystem through ROS topics and ROS services without having to change anything about their architecture. For instance if a partner is using Java and OSGi, it can create nodes in ROS Java that will expose/register functionalities through ROS services, publish/subscribe information using ROS topics.

4.3. EmoPRAMAD

Participant: Dominique Vaufreydaz [correspondant].

Affective computing,

Within the Pramad project, we want to offer a full affective loop between the companion robot and the elderly people at home. This affective loop is necessary within the context of everyday interaction of elderly and the companion robot. A part of this loop is to make the robot express emotions in response to the emotional state of the user. To do that, we need to test our working hypothesis about the visual representation of emotions with the 3D face of robot. EmoPRAMAD is an evaluation tool designed to conduct comparative studies between human faces and the 3D faces expressing a defined set of emotions.

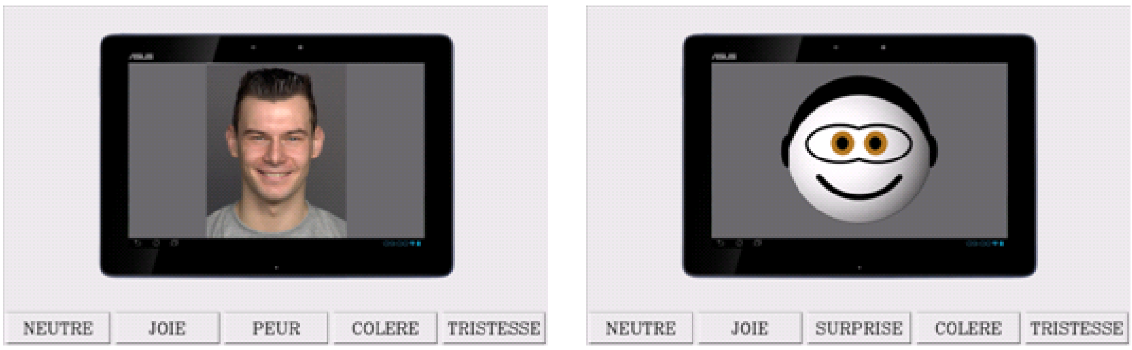


Figure 3. EmoPRAMAD interfaces with a human face and a 3D face from our virtual agent.

The evaluation conducted though EmoPRAMAD concerns both unimodal (facial only) and bimodal conditions (facial/sound). The emotions set is composed of 4 basic emotions (joy, fear, anger, sadness) and a neutral state. While experimenting, the software collects several parameters in order to evaluate more than correctness of the answers: time to respond, length of mouse moves, etc. Experimentation is still in progress at Inria in Grenoble, University Pierre and Marie Currie and Broca Hospital in Paris. A set of 235 participants from 14 to 88 years old was already recorded.

4.4. Detection and Tracking of Pedestrians in INRETS Intelligent Urban Spaces Platform

Participants: Claudine Combe, James Crowley [correspondant], Lukas Rummelhard.

Visual detection and tracking of pedestrians, Intelligent Urban Space

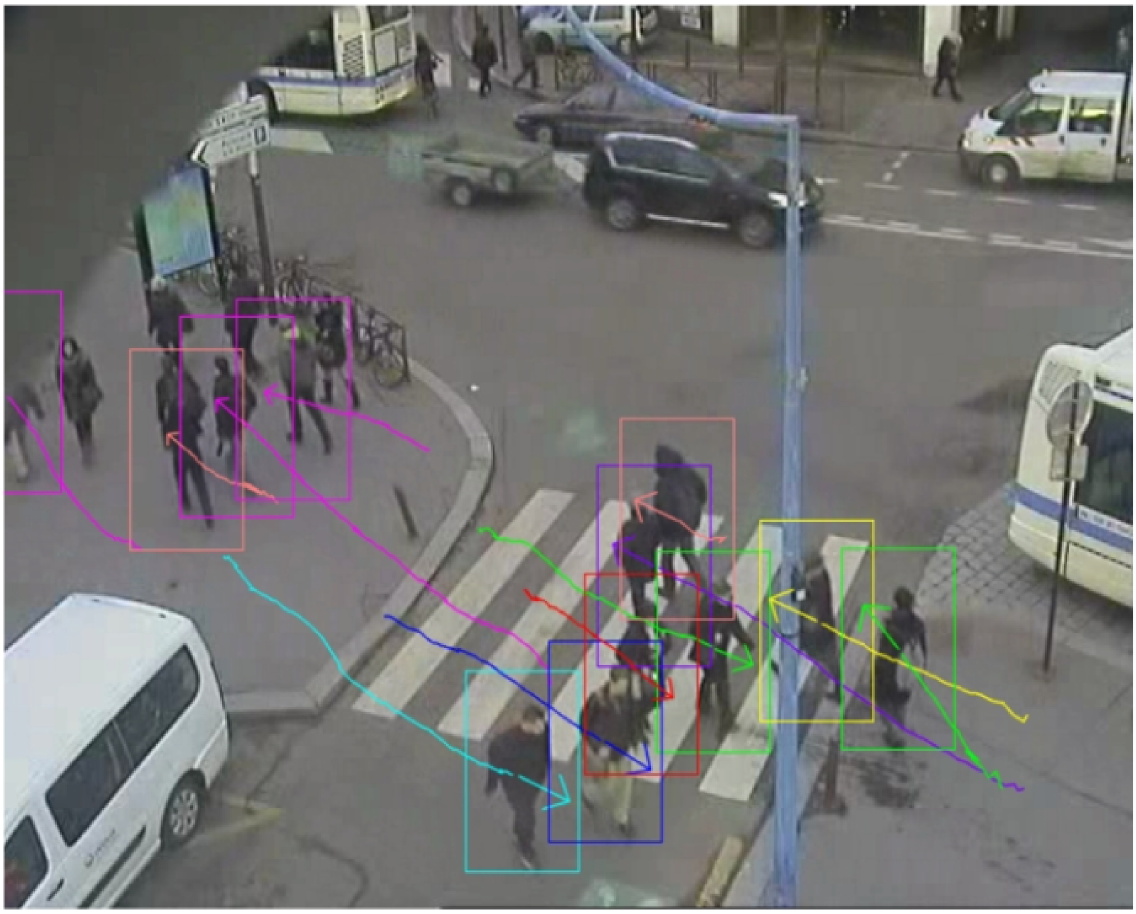


Figure 4. Cipebus: pedestrian tracking system.

The project ANR-07-TSFA-009-01 CIPEBUS ("Carrefour Intelligent - Pole d'Echange - Bus) has been proposed by INRETS-IFSTTAR, in collaboration with Inria, Citilog, Fareco, and the city of Versaille. The Objective of the CIPEBUS project is to develop an experimental platform for observing activity in a network of urban streets in order to experiment with techniques for optimizing circulation by context aware control of traffic lights.

Within CipeBus, Inria has developed a real time multi-camera computer vision system to detect and track people using a network of surveillance cameras. The CipeBus combines real time pedestrian detection with 2D and 3D Bayesian tracking to record the current position and trajectory of pedestrians in an urban environment under natural view conditions. The system extends the sliding window approach to use a half-octave Gaussian Pyramid to explore hypotheses of pedestrians at different positions and scales. A cascade classifier is used to determine the probability that a pedestrian can be found at a particular position and scale. Detected pedestrians are then tracked using a particle filter.

The resulting software system has been installed and tested at the INRETS CipeBus platform and is currently used for experiments in controlling the traffic lights to optimize the flow of pedertrians and public transportation while minimizing the delay imposed on private automobiles.

4.5. Multisensor observation of human activity for integrated energy and comfort management

Participants: Claudine Combe, James Crowley [correspondant], Lucas Nacsa, Amaury Negre, Lukas Rummelhard.

multimodal tracking of human activity

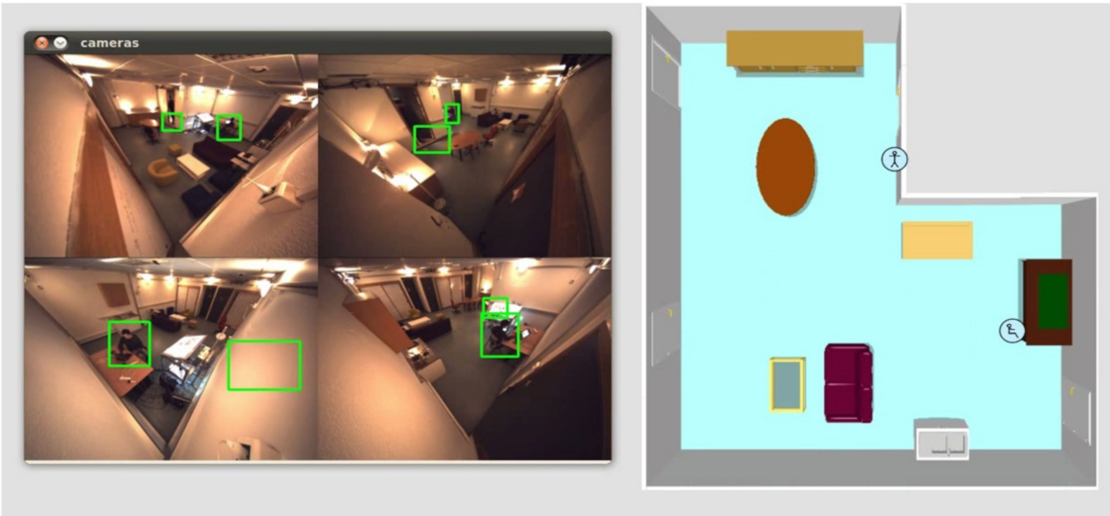


Figure 5. The 3D tracker integrates observations from multiple sensors

As part of Inria’s contribution of ICTLabs Action TSES - Smart Energy Systems, we have constructed a system that integrates information from multiple environmental sensor to detect and track people in indoor environments. This system, constructed as part of activity 11831 Open SES Experience Labs for Prosumers and New Services, has been released to ICTLabs partners in June 2012. It has also been used for construction of a smart spaces testbed at Schneider Electric.

This software, named MultiSensor activity tracker, integrates information from multiple environmental sensors to keep track of the location and activity of people in a smart environment. This model is designed to be used by a home energy broker that would work in conjunction with a smart grid to manage the energy consumption of home appliances, balancing the needs of inhabitants with opportunities for savings offered by electricity rates. This database will also be used for by advisor services that will offer advice to inhabitants on the consequences to energy consumption and energy cost that could potentially result from changes to lifestyle or home energy use.

Work in this task draws from earlier result from a number of development projects at Inria. In the ANR Casper project Inria created Bayesian tracking system for human activity using a voxel based occupancy grid. Within the INRA ADT PAL project, Inria is creating methods for plug and play installation of visual and acoustic sensors for tracking human activity within indoor environments.

While a voxel based Bayesian tracker has served well for a number of applications, a number of limitations have been observed. For example, under certain circumstances, the sensor data can provide contradictory or ambiguous data about the location and activities of people. Resolving such cases required the Bayesian tracker to choose between a numbers of competing hypotheses, potentially resulting in errors. Several members of our group have argued that an alternative integration approach based on the use of a Particle filter would solve these problems and provide a more reliable tracking system. This task has been undertaken to evaluate this hypothesis. The system configured and optimized for detecting and tracking people within rooms using multiple calibrated cameras. The system currently uses corner mounted cartesian cameras, ceiling mounted cameras with wide angle lenses and panoramic cameras placed on tables. Cameras may be connected and disconnected while the component is running, but they must be pre-calibrated to a common room reference frame. We are currently experimenting with techniques for Bayesian estimation of camera parameters for auto-calibration. Cameras may be connected dynamically.

The original system 3DBT has been declared with the APP "Agence pour la Protection des Programmes" under the Interdeposit Digital number IDDN.FR.001.490023.000.S.P.2006.000.10000. A revised declaration for the latest version of the system is currently being prepared.

4.6. Stereo Viewfinder

Participants: Frédéric Devernay [correspondant], Loic Lefort, Elise Mansilla, Sergi Pujades-Rocamora.

Stereoscopy, Auto-calibration, Real-time video processing, Feature matching

This software has been filed with the APP "Agence pour la Protection des Programmes" under the Interdeposit Digital number IDDN.FR.001.370083.000.S.P.2007.000.10000

4.7. Visual Emotion Recognition for Health and Well Being.

Participants: James Crowley [correspondant], Varun Jain, Sergi Pujades-Rocamora.

Visual Emotion Recognition

People express and feel emotions with their face. Because the face is the both externally visible and the seat of emotional expression, facial expression of emotion plays a central role in social interaction between humans. Thus visual recognition of emotions from facial expressions is a core enabling technology for any effort to adapt ICT to improve Health and Wellbeing.

Constructing a technology for automatic visual recognition of emotions requires solutions to a number of hard challenges. Emotions are expressed by coordinated temporal activations of 21 different facial muscles assisted by a number of additional muscles. Activations of these muscles are visible through subtle deformations in the surface structure of the face. Unfortunately, this facial structure can be masked by facial markings, makeup, facial hair, glasses and other obstructions. The exact facial geometry, as well as the coordinated expression of muscles is unique to each individual. In additions, these deformations must be observed and measured under a large variety of illumination conditions as well as a variety of observation angles. Thus the visual recognition of emotions from facial expression remains a challenging open problem in computer vision.

Despite the difficulty of this challenge, important progress has been made in the area of automatic recognition of emotions from face expressions. The systematic cataloging of facial muscle groups as facial action units by Ekman [38] has let a number of research groups to develop libraries of techniques for recognizing the elements of the FACS coding system [30]. Unfortunately, experiments with that system have revealed that the system is very sensitive to both illumination and viewing conditions, as well as the difficulty in interpreting the resulting activation levels as emotions. In particular, this approach requires a high-resolution image with a high signal-to-noise ratio obtained under strong ambient illumination. Such restrictions are not compatible with the mobile imaging system used on tablet computers and mobile phones that are the target of this effort.

As an alternative to detecting activation of facial action units by tracking individual face muscles, we propose to measure physiological parameters that underlie emotions with a global approach. Most human emotions can be expressed as trajectories in a three dimensional space whose features are the physiological parameters of Pleasure-Displeasure, Arousal-Passivity and Dominance-Submission. These three physiological parameters can be measured in a variety of manners including on-body accelerometers, prosody, heart-rate, head movement and global face expression.

The PRIMA Group at Inria has developed robust fast algorithms for detection and recognition of human faces suitable for use in embedded visual systems for mobile devices and telephones. The objective of the work described in this report is to employ these techniques to construct a software system for measuring the physiological parameters commonly associated with emotions that can be embedded in mobile computing devices such as cell phones and tablets.

A revised software package has recently been released to our ICTlab partners for face detection, face tracking, gender and age estimation, and orientation estimation, as part of ICTlabs Smart Spaces action line. This software has been declared with the APP "Agence pour la Protection des Programmes" under the Interdeposit Digital number IDDN.FR.001.370003.000.S.P.2007.000.21000.

A software library, named PrimaCV has been designed, debugged and tested, and released to ICTLabs partners for real time image acquisition, robust invariant multi-scale image description, highly optimized face detection, and face tracking. This software has been substantially modified so as to run on an mobile computing device using the Tegra 3 GPU.

4.8. AppsGate - Smart Home Application Gateway

Participants: Alexandre Demeure, James Crowley [correspondant], Emeric Grange, Cedric Gerard, Camille Lenoir, Kouzma Petoukhov.

Smart Home Applications Gateway

PRIMA has participated in the development of the AppsGate Home Application Gateway Architecture. The AppsGate architecture is based on the HMI Middleware developed in cooperation with the IIHM and Adele groups of the UMR Laboratoire Informatique de Grenoble (LIG). The HMI Middleware is designed to facilitate the development of end-user applications on top of the core software components described in the sections above, while ensuring service continuity and usability. The key features of the HMI Middleware include:

- Integration of sensors and actuators managed by a variety of protocols, and provision of a uniform abstraction for these devices as component-oriented-services,
- Integration of Web services made available on the cloud by a variety of web service providers, and provision of a uniform abstraction for these services as component-oriented-services,
- Communication between the HMI middleware and client applications - typically, user interfaces for controlling and programming the smart home, that run on high-end devices such as smartphones, tablets, and TVs.

As part of the Appsgate middleware, we have developed SPOK, an End User Development Environment, that enables inhabitants to control and program their smart Homes via a web interface. The current version of SPOK includes an editor for editing programs using a pseudo-natural language and an interpreter. A multi-syntax editor as well as additional services such as a debugger and a simulator are currently under development.

4.9. a SmartEnergy Serious Game

Participant: Patrick Reignier.

This ongoing serious game is the result of a collaboration with Ayesha Kashif (LIG), Stephane Ploix (G-Scop) and Julie Dugdale (LIG). It has been developped as part of the Grenoble INP SmartEnergy project.

Inhabitants play a key role in buildings global energy consumption but it is difficult to involve them in energy management. Our objective is to make energy consumption visible by simulating inside a serious game the energy impact of inhabitants behaviours. A serious game is curently under development, coupling a 3D virtual environment and a building energy simulator. The 3D virtual environment is based on the JMonkey 3D engine. New houses can be easily imported using SweetHome 3D and Blender. The building energy simulator is EnergyPlus. The 3D engine and the energy engine are coupled using the Functional Mock-up Interface (FMI) standard. Using this standard will allow to easily switch between existing building energy simulators

REVES Project-Team

5. New Software and Platforms

5.1. Multi-View Image-Based Relighting Suite

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

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

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

5.2. IBR-Common

Participants: Jerome Esnault, Gaurav Chaurasia, George Drettakis.

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

5.3. IBR in Unity

Participants: Jerome Esnault, Gaurav Chaurasia, George Drettakis.

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

Our implementation offers important features to game developers:

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

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

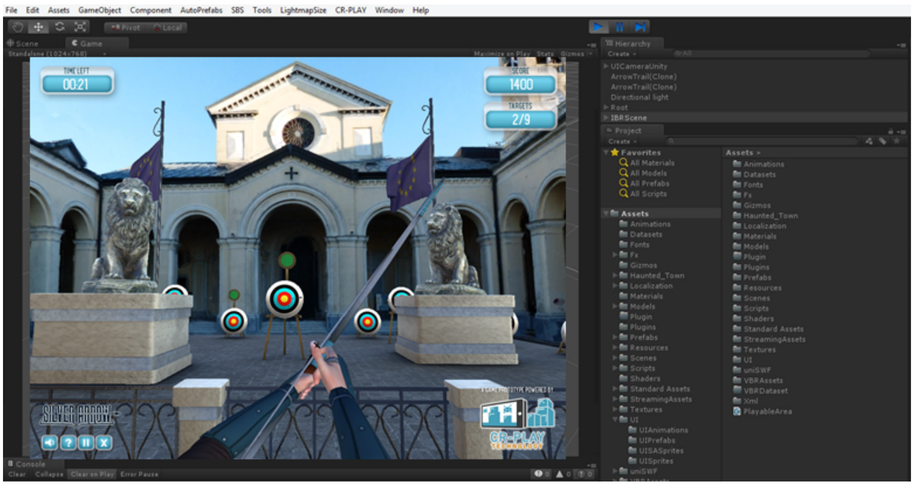


Figure 3. Screen capture of the Unity game development tool. The background buildings are rendered with our image-based rendering algorithm.

RITS Team

5. New Software and Platforms

5.1. New Software

The following software items have been submitted very recently to the APP; some already have IDDN references. Some of these are under improvement and other have already been transferred to industrial partners.

- **SODA**: This software has been developed in the context of the French ABV [0] project. This package contains the functions that are necessary to automate the vehicle navigation in its secured lane. This software has been purchased by a private partner (Valeo Group) aiming at developing its own automated vehicle.
- **MELOSYM**: this is the latest laser based Hierarchical ML-SLAM algorithm developed by RITS. It contains all the functions needed to perform the vehicle localization and the mapping of the environment. Windows compatible, it was initially developed under the ^{RT}MAPS platform but the version includes a standalone version. This software has been evaluated by a private partner aiming at developing its own automated vehicle for indoor applications. It is currently evaluated by another private partner aiming at implementing our solution on its outdoor automated shuttles.
- **STEREOLOC**: this is the package performing stereovision based localization and mapping. It performs semi-dense mapping of outdoor large environments and provides real-time estimates of the vehicle position. The software was tested and validated using ^{RT}MAPS like databases as well as the KITTI benchmark.

5.1.1. DOLAR

This software performs real-time obstacle detection and tracking using laser data scanned with one or several laser sensors with different geometric configurations. Obstacle detection is based on laser data segmentation while obstacle tracking uses PHD-based filtering techniques. The software is currently evaluated by a private partner aiming at implementing our solution on its outdoor automated unmanned vehicles.

5.1.2. FEMOT

FEMOT (Fuzzy Embedded MOTor) is an experimental motor for implementing fuzzy logic controllers, including all the fuzzy stages (fuzzification, inference, and defuzzification). This library has been compiled in Microsoft Visual (MVS) Studio and RTMaps. The proposed library is modular and adaptable to different situations and scenarios, especially for autonomous driving applications. FEMOT allows the development of the fuzzy rules to be written as sentences in an almost natural language. It allows the user to define variables and their fuzzy rules and to join them with other variables in rules to yield crisp signals for the controllers. The APP deposit was delivered May 2014. The Properties defined in FEMOT shows the number of inputs, outputs and fuzzy rules that the controller needs.

This software is used for the arbitration and control for fully automated functions. The behaviour of a human driver can be emulated with this technique. First simulations are showing promising results, and the library allows an easy adaptation in decision marking situations.

5.1.3. V2Provue

It is a software developed for the Vehicle-to-Pedestrian (V2P) communications, risk calculation, and alarming pedestrians of collision risk [34]. This software is made of an Android application dedicated to pedestrians and RtMaps modules for the vehicles.

[0] Automatisation Basse Vitesse

On the pedestrian side, the application is relying on GPS data to localize the user and Wi-Fi communications are used to receive messages about close vehicles and send information about the pedestrian positioning. Besides, a service has been developed to evaluate the collision risk with the vehicles near the pedestrian and an HMI based on OpenStreetMap displays all the useful information such as pedestrian and vehicles localization and, collision risk.

On the vehicle side, RtMaps modules allowing V2X communications have been developed. These modules contain features such as TCP/UDP socket transmissions, broadcast, multicast, unicast communications, routing, forwarding algorithms, and application specific modules. In the V2ProVu software, a particular application module has been implemented to create data packets containing information about the vehicle state (position, speed, yaw rate,...) and the V2X communication stack is used to broadcast these packets towards pedestrians. Moreover, the V2proVu application can also receive data from pedestrians and create objects structures that can be shared with the vehicle perception tools.

SEMAGRAMME Project-Team

5. New Software and Platforms

5.1. ACG Development Toolkit

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

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

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

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

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

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

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

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

5.2. Grew

Participants: Bruno Guillaume [correspondent], Guy Perrier.

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

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

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

[0] Available at http://acg.gforge.inria.fr with a CeCILL license.
[0] https://opam.ocaml.org/packages/acgtk/acgtk.1.1/

5.3. Leopar

Participants: Bruno Guillaume [correspondent], Guy Perrier.

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

The main features of current version of the software are:

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

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

5.4. ZombiLingo

Participants: Bruno Guillaume [correspondent], Karën Fort.

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

5.5. Other developments

Participants: Maxime Amblard [correspondent], Bruno Guillaume.

Main topics: data managment, disfluencies and dependency

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

SIROCCO Project-Team

5. New Software and Platforms

5.1. Visual Fixation Analysis

Participant: Olivier Le Meur [contact person].

From a set of fixation data and a picture, the software called Visual Fixation Analysis extracts from the input data a number of features (fixation duration, saccade length, orientation of saccade...) and computes a human saliency map. The software can also be used to assess the degree of similarity between a ground truth (eye fixation data) and a predicted saliency map. This software is dedicated to people working in cognitive science and computer vision. This software has been registered at the APP (Agence de Protection des Programmes).

5.2. Hierarchical super-resolution based inpainting

Participant: Olivier Le Meur [contact person].

From an input binary mask and a source picture, the software performs an examplar-based inpainting. The method is based on the combination of multiple inpainting applied on a low resolution of the input picture. Once the combination has been done, a single-image super-resolution method is applied to recover the details and the high frequency in the inpainted areas. The developments have been pursued in 2014, in particular by introducing a Poisson blending step in order to improve the visual quality of the inpainted video. This software is dedicated to people working in image processing and post production. This software is being registered at the APP (Agence de Protection des Programmes).

5.3. Salient object extraction

Participants: Zhi Liu, Olivier Le Meur [contact person].

This software detects salient object in an input picture in an automatic manner. The detection is based on super-pixel segmentation and contrast of histogram. This software is dedicated to people working in image processing and post production. This software is being registered at the APP (Agence de Protection des Programmes).

SMIS Project-Team

5. New Software and Platforms

5.1. Introduction

In our research domain, developing software prototypes is mandatory to validate research solutions and is an important vector for publications, demonstrations at conferences as well as for cooperations with industry. Our software strategy is also driven by our ambition to see our research results produce a real societal impact. To reach this goal, we integrate our prototypes in experiments in the field - notably in the healthcare domain and with scientists of other disciplines - and we recently set up educational platforms to raise students awareness of privacy protection problems and embedded programming.

This prototyping task is however difficult because it requires specialized hardware platforms, themselves sometimes at an early stage of development. For a decade, we have developed successive prototypes relying on different hardware platforms provided by Schlumberger then Gemalto, e.g., PicoDBMS a full-fledged DBMS embedded in a smart card [37] [26], Chip-Secured Data Access (C-SDA) a tamper-resistant mediator between a client and an untrusted server hosting encrypted data [32], Chip-Secured XML Access (C-SXA) an XML-based access rights controller, recipient of the e-gate open 2004 Silver Award and SIMagine 2005 Gold award [33]. Today, most of our software development efforts are organized around a unified platform named PlugDB and we are designing our own hardware platforms, that are produced by electronic SMEs. This opens up new research and experiment opportunities and we are engaged in an open-source/open hardware initiative to disseminate our results at a larger scale, both for scientific, educational and business purposes.

The next subsections detail the two prototypes we are focusing on today.

5.2. PlugDB engine

Participants: Nicolas Anciaux [correspondent], Luc Bouganim, Aydogan Ersoz, Quentin Lefebvre, Philippe Pucheral.

More than a stand-alone prototype, PlugDB is part of a complete architecture dedicated to a secure and ubiquitous management of personal data. PlugDB aims at providing an alternative to a systematic centralization of personal data. To meet this objective, the PlugDB architecture lies on a new kind of hardware device called Secure Portable Token (SPT). Roughly speaking, a SPT combines a smart-card and a micro-controller with a large external Flash memory (Gigabyte sized). The SPT can host data on Flash (e.g., a personal folder) and safely run code embedded in the micro-controller. PlugDB engine is the cornerstone of this embedded code. PlugDB engine manages the database on Flash (tackling the peculiarities of NAND Flash storage), enforces the access control policy defined on this database, protects the data at rest against piracy and tampering, executes queries (tackling low RAM constraint) and ensures transaction atomicity. Part of the on-board data can be replicated on a server (then synchronized) and shared among a restricted circle of trusted parties through crypto-protected interactions. PlugDB engine has been registered at APP (Agence de Protection des Programmes) in 2009 [27] and a new version is registered each year. PlugDB has been experimented in the field in the Yvelines District to implement a secure and portable medical-social folder helping the coordination of medical care and social services provided at home to dependent people. This field experiment is being audited by ARS-Ile de France (the Regional Healthcare Agency) and CG78 (General Council of Yvelines District), in order to envision the opportunity of a larger deployment. In parallel, we are improving the PlugDB prototype to overcome the limitations identified during the experiment. Notably, we have integrated a Bluetooth module to communicate in wireless with the token, a fingerprint module to authenticate users and a microphone to record voice messages. These are key elements in the perspective of a generalization. Link: https://project.inria.fr/plugdb/.

5.3. Eagle Tree

Participants: Matias Bjørling, Philippe Bonnet, Luc Bouganim, Niv Dayan [correspondent].

Solid State Drives (SSDs) are a moving target for system designers: they are black boxes, their internals are undocumented, and their performance characteristics vary across models. There is no appropriate analytical model and experimenting with commercial SSDs is cumbersome, as it requires a careful experimental methodology to ensure repeatability. Worse, performance results obtained on a given SSD cannot be generalized. Overall, it is impossible to explore how a given algorithm, say a hash join or LSM-tree insertions, leverages the intrinsic parallelism of a modern SSD, or how a slight change in the internals of an SSD would impact its overall performance. In this paper, we propose a new SSD simulation framework, named EagleTree, which addresses these problems, and enables a principled study of SSD-Based algorithms. EagleTree is an extensible, customizable SSD simulator designed to enable deep analyses of the interplay between the FTL, block management scheme, IO scheduling policy and application workload. It is able to generate visual illustrations of a host of performance metrics. EagleTree is available for Linux, and is licensed under GPL. EagleTree's git repository is : https://github.com/nivdayan/EagleTree.

STARS Project-Team

5. New Software and Platforms

5.1. SUP

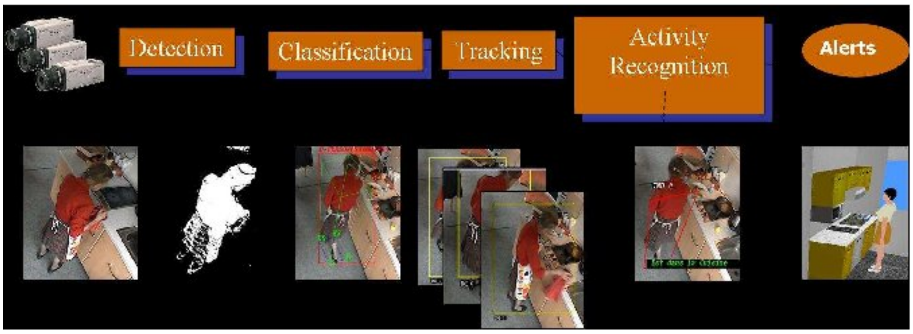


Figure 5. SUP workflow

5.1.1. Presentation

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

This generic architecture is designed to facilitate:

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

Goals of SUP are twofold:

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

The plugins cover the following research topics:

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

5.1.2. Improvements

Currently, the OpenCV library is fully integrated with SUP. OpenCV provides standardized data types, a lot of video analysis algorithms and an easy access to OpenNI sensors such as the Kinect or the ASUS Xtion PRO LIVE.
In order to supervise the GIT update progress of SUP, an evaluation script is launched automatically everyday. This script updates the latest version of SUP then compiles SUP core and SUP plugins. It executes the full processing chain (from image acquisition to activity recognition) on selected data-set samples. The obtained performance is compared with the one corresponding to the last version (i.e. day before). This script has the following objectives:

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

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

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

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

5.2. ViSEvAl

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

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

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

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

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

The GUI is composed of 3 different parts:

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

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

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

5.3. Clem

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

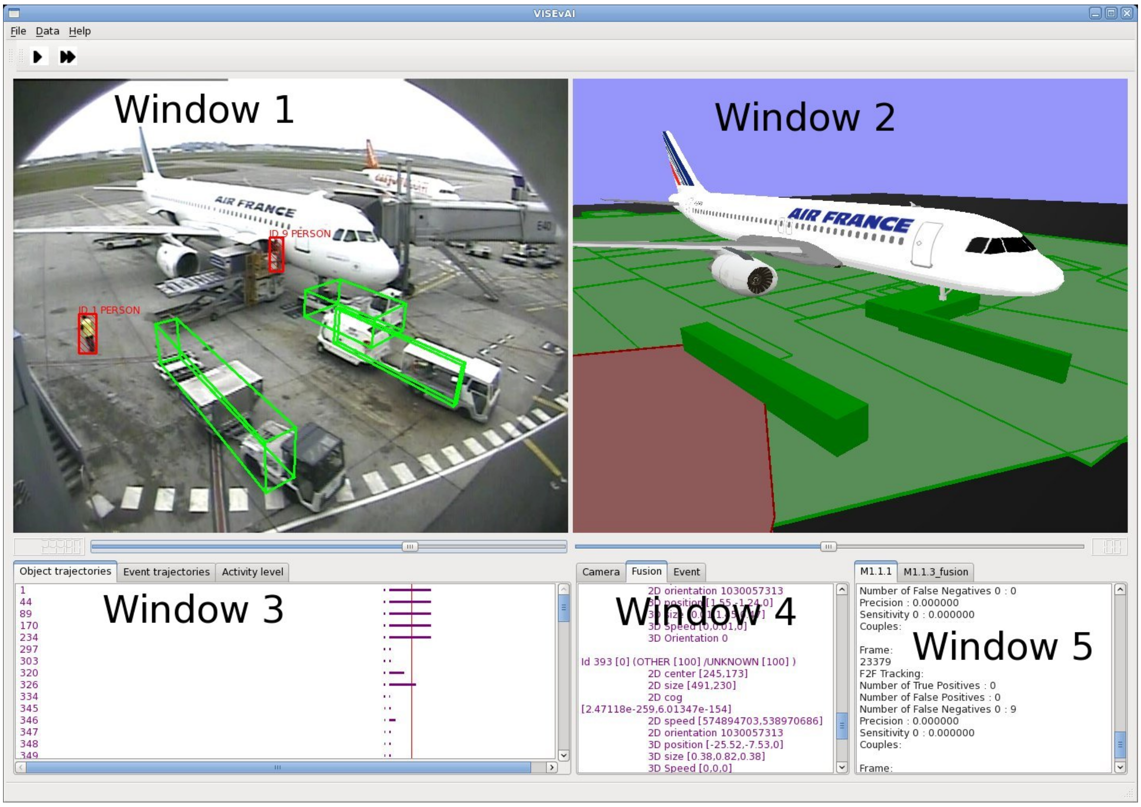


Figure 6. GUI of the ViSEvAl software

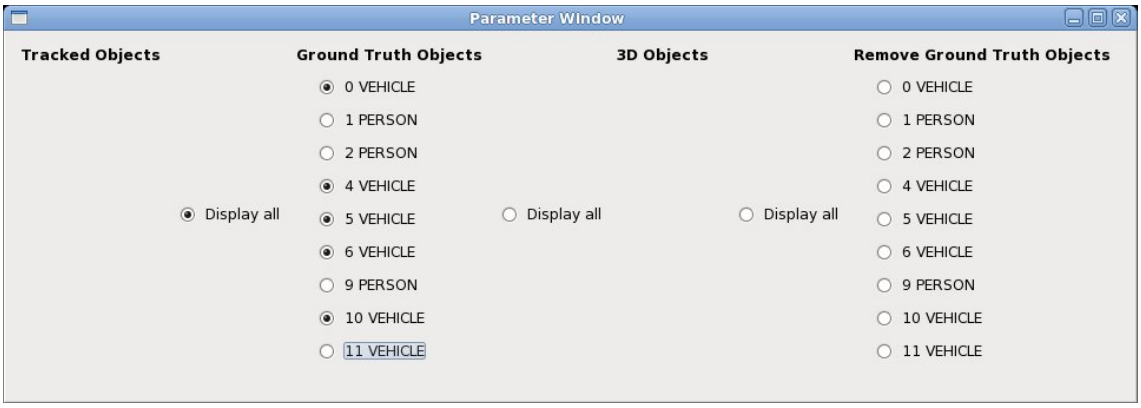


Figure 7. The object window enables users to choose the object to display

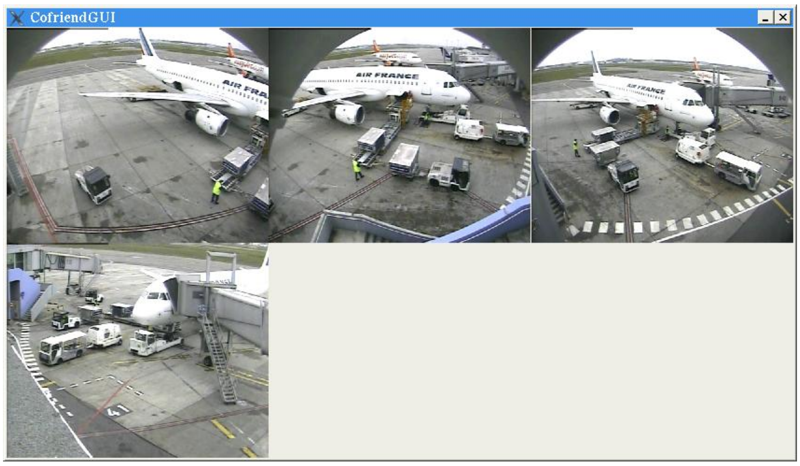


Figure 8. The multi-view window

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

The work on CLEM was published in [68], [69], [18], [19].
Web page: http://www-sop.inria.fr/teams/pulsar/projects/Clem/

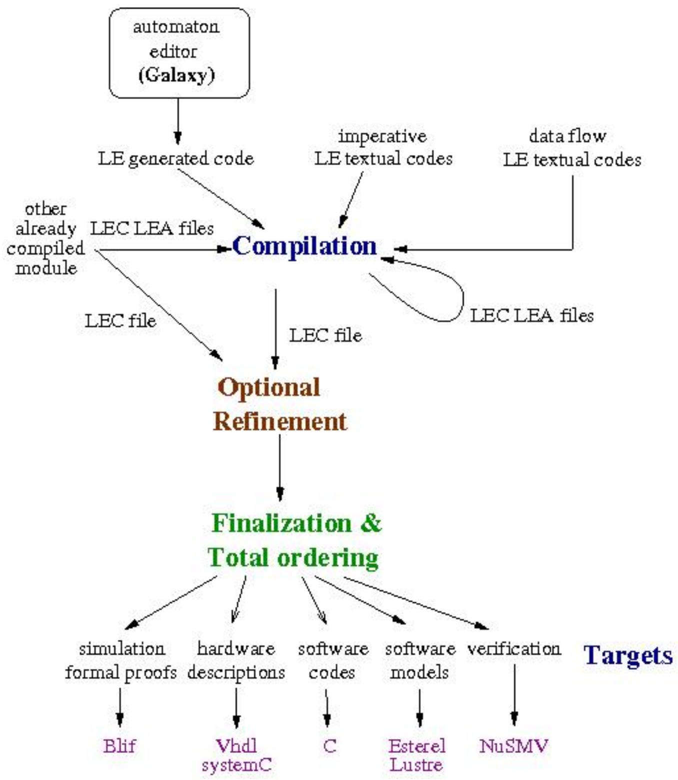


Figure 9. The Clem Toolkit

TITANE Project-Team

5. New Software and Platforms

5.1. CGAL, the Computational Geometry Algorithms Library

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

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

WILLOW Project-Team

5. New Software and Platforms

5.1. SPArse Modeling Software (SPAMS)

SPAMS v2.5 was released as open-source software in May 2014 (v1.0 was released in September 2009, v2.0 in November 2010). It is an optimization toolbox implementing algorithms to address various machine learning and signal processing problems involving

- Dictionary learning and matrix factorization (NMF, sparse PCA, ...)
- Solving sparse decomposition problems with LARS, coordinate descent, OMP, SOMP, proximal methods
- Solving structured sparse decomposition problems (ℓ_1/ℓ_2, ℓ_1/ℓ_∞, sparse group lasso, tree-structured regularization, structured sparsity with overlapping groups,...).

The software and its documentation are available at http://www.di.ens.fr/willow/SPAMS/.

5.2. Efficient video descriptors for action recognition

This package contains source code for highly-efficient extraction of local space-time video descriptors for action recognition. The accuracy of descriptors measured at standard benchmarks for action recognition is comparable to the state-of-the-art dense trajectory features, while being more than 100 times faster on standard CPU. The previous version of this code was evaluated in our recent work [12]. The package is available from http://www.di.ens.fr/~laptev/download/fastvideofeat-1.0.zip. Earlier version of our space-time video features is available at http://www.di.ens.fr/~laptev/download/stip-2.0-linux.zip.

5.3. Weakly Supervised Action Labeling in Videos Under Ordering Constraints

This is a package of Matlab code implementing weakly-supervised learning of actions from input videos and corresponding sequences of action labels. The code finds optimal alignment of action labels and video intervals during training. Along the optimization, the method trains corresponding action model. The package is available at http://www.di.ens.fr/willow/research/actionordering/. The method corresponding to this code package has been described and evaluated in Bojanowski *et al.* ECCV 2014 [10].

5.4. Visual Place Recognition with Repetitive Structures

Open-source release of the software package for visual localization in urban environments has been made publicly available in May 2014. The software package implements the method [A. Torii et al., CVPR 2013] for representing visual data containing repetitive structures (such as building facades or fences), which often occur in urban environments and present significant challenge for current image matching methods. This is an extended version that includes geometric verification. The original version was released in 2013. The software is available at http://www.di.ens.fr/willow/research/repttile/download/repttile_demo_ver03.zip.

5.5. Seeing 3D chairs: exemplar part-based 2D-3D alignment using a large dataset of CAD models

Open-source release of the software package for 2D-3D category-level alignment has been made publicly available. The software package implements newly developed method [9] for category-level recognition that not only outputs the bounding box of the object but predicts an approximate 3D model aligned with the input image. The software is available at http://www.di.ens.fr/willow/research/seeing3Dchairs/.

5.6. Painting-to-3D Model Alignment Via Discriminative Visual Elements

Open-source release of the software package for alignment of 3D models to historical and non-photographic depictions has been made publicly available. The software package implements the method of [9] for alignment of 3D models to input historical and non-photographic depictions such as paintings, drawings or engravings, where standard local feature-based method fail. The software is available at http://www.di.ens.fr/willow/research/painting_to_3d/.

5.7. Painting recognition from wearable cameras

Open-source release of the software package for painting recognition from wearable cameras has been made publicly available. The software implements a method described in [20] that recognizes 2D paintings on a wearable Google Glass device, for example, for a virtual museum guide application. The software runs directly on Google Glass without sending images to external servers for processing and recognizes a query painting in a database of 100 paintings in one second. The report and software are publicly available at http://www.di.ens.fr/willow/research/glasspainting/.

5.8. Learning and transferring mid-level image representations using convolutional neural networks

The first version of the open source software package for convolutional neural networks [13] has been released online. The software package is based on the cuda-convnet implementation of convolutional neural networks and includes a pre-trained convolutional neural network that can be applied to visual object classification as in the Pascal VOC 2012 set-up, where it achieves state-of-the-art single network results. The package also includes functions for visualization of object localization. The software is publicly available at http://www.di.ens.fr/willow/research/cnn/code/voc12-cvpr-reproduce.tar.

WIMMICS Project-Team

5. New Software and Platforms

5.1. Corese

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

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

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

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

Corese is used and has been used in more than 60 applications, 24 PhD Thesis and is used for education by several institutions. It has been used in European projects such as Ontorule, Palette, SevenPro, SeaLife and in ANR projects such as Kolflow, Ginseng, Neurolog, VIP, ISICIL, e-WOK Hub. Corese is the Semantic Web engine of Discovery Hub [0] and of the Semantic Web Import Plugin for Gephi visualization [0].

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

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

5.2. Question Answering Wikiframework-based System

Participant: Elena Cabrio.

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

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

5.3. French Chapter of DBpedia

Participants: Raphaël Boyer, Fabien Gandon.

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

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

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

[0] http://www.discoveryhub.co
[0] https://marketplace.gephi.org/plugin/semanticwebimport/

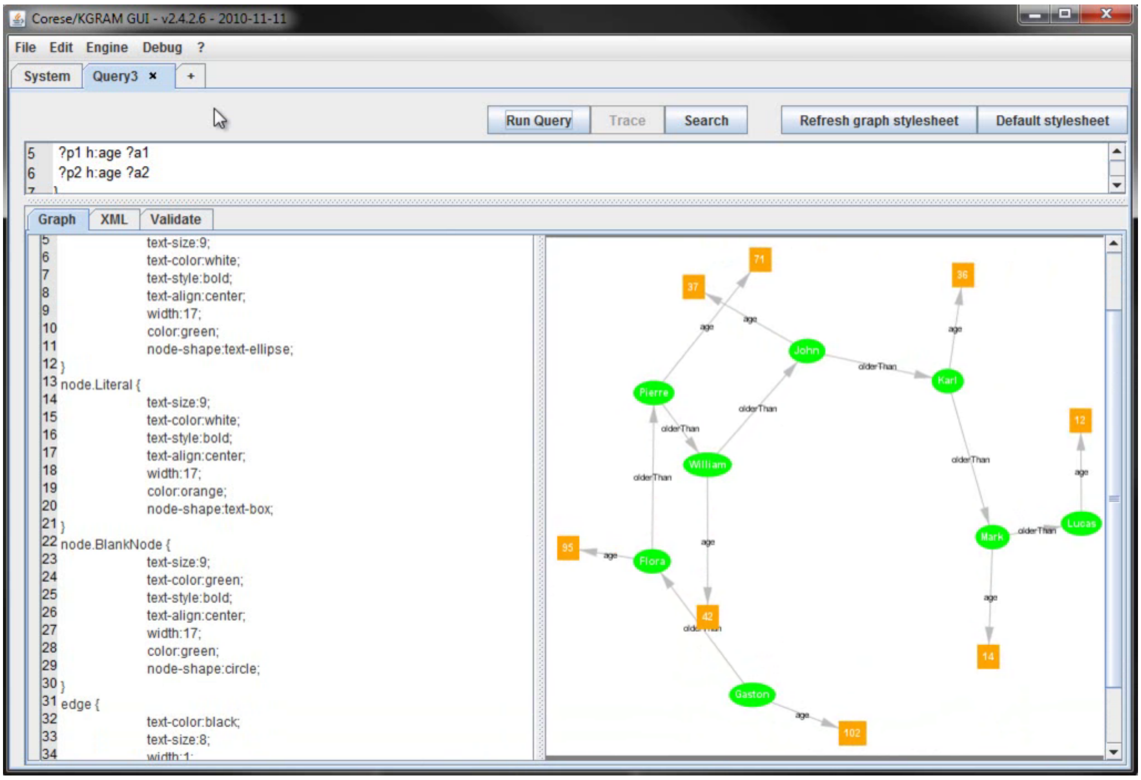


Figure 1. Corese

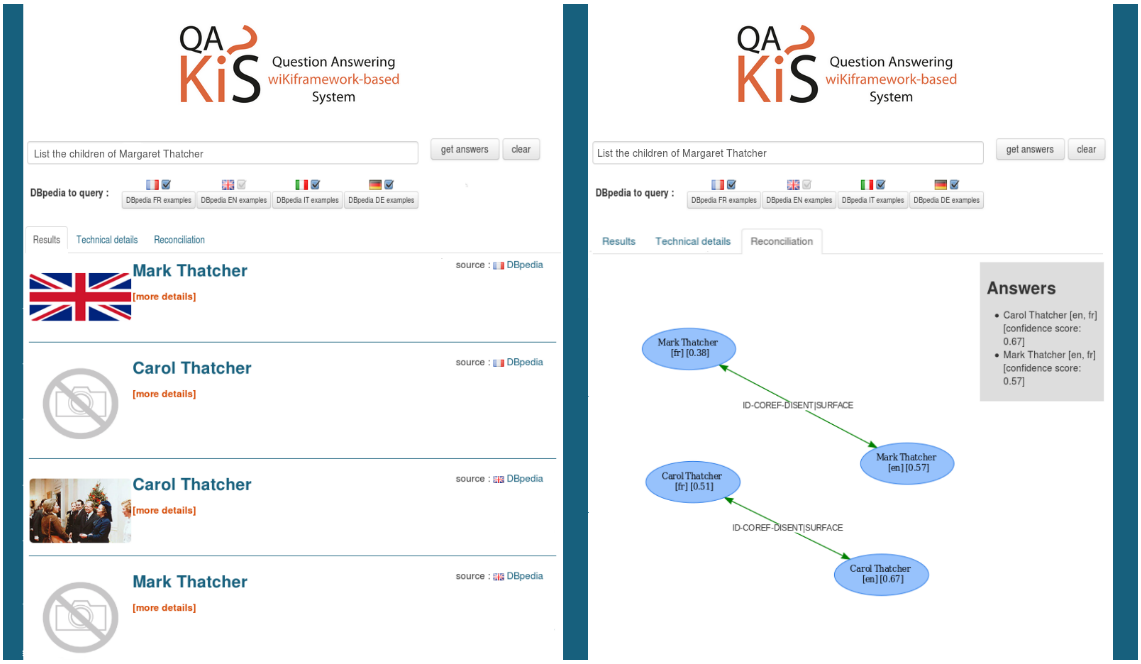


Figure 2. QAKIS

ZENITH Project-Team

5. New Software and Platforms

5.1. Hadoop_g5k

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

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

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

5.2. LogMagnet

Participants: Julien Diener, Florent Masseglia.

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

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

5.3. MultiSite-Rec

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

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

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

5.4. PlantRT: Gossip-Based Recommendation

Participants: Alexis Joly, Julien Champ, Miguel Liroz-Gistau, Esther Pacitti, Maximilien Servajean [contact].

URL: http://www2.lirmm.fr/~servajean/prototypes/plant-sharing/plant-rt.html

PlantRT is a distributed gossip-based platform for content sharing enabling plants observation keywords search and GPS position based recommendation. It combines advantages from centralized and P2P systems.

5.5. Pl@ntNet

Participants: Julien Champ, Hervé Goëau, Alexis Joly [contact].

URL: http://goo.gl/CpSrr3

Pl@ntNet is an image sharing and retrieval application for the identification of plants. It is developed in the context of the Pl@ntNet project that involves four French research organisations (Inria, Cirad, INRA, IRD) and the members of Tela Botanica social network. The key feature of the iOS and Android front ends is to help identifying plant species from photographs, through a server-side visual search engine based on several results of ZENITH team on content-based information retrieval. Since its first release in March 2013 on the apple store, the application was downloaded by around 300K users in more than 150 countries (between 500 and 5000 active users daily with peaks occurring during the week-ends). The collaborative training set that allows the content-based identification is continuously enriched by the users of the application and the members of Tela Botanica social network. At the time of writing, it includes about 100K images covering more than 5000 French plant species about 4/5 of the whole French flora (this is actually the widest identification tool built anytime).

5.6. SON (Shared-data Overlay Network)

Participants: Esther Pacitti, Didier Parigot [contact], Patrick Valduriez.

URL: http://www-sop.inria.fr/teams/zenith/SON

SON is an open source development platform for P2P networks using web services, JXTA and OSGi. SON combines three powerful paradigms: components, SOA and P2P. Components communicate by asynchronous message passing to provide weak coupling between system entities. To scale up and ease deployment, we rely on a decentralized organization based on a DHT for publishing and discovering services or data. In terms of communication, the infrastructure is based on JXTA virtual communication pipes, a technology that has been extensively used within the Grid community. Using SON, the development of a P2P application is done through the design and implementation of a set of components. Each component includes a technical code that provides the component services and a code component that provides the component logic (in Java). The complex aspects of asynchronous distributed programming (technical code) are separated from code components and automatically generated from an abstract description of services (provided or required) for each component by the component generator.

5.7. Snoop & SnoopIm

Participants: Alexis Joly [contact], Julien Champ, Jean-Christophe Lombardo.

URL: http://otmedia.lirmm.fr/

Snoop is a generalist C++ library dedicated to high-dimensional data management and efficient similarity search. Its main features are dimension reduction, high-dimensional feature vectors hashing, approximate k-nearest neighbors search and Hamming embedding. Snoop is a refactoring of a previous library called PMH developed jointly with the French National Institute of Audiovisual. It is based on the joined research work of Alexis Joly and Olivier Buisson. SnoopIm is a content-based image search engine built on top of Snoop and allowing to retrieve small visual patterns or objects in large collections of pictures. The software is being experimented in several contexts including a logo retrieval application set up in collaboration with the French Press Agency, an experimental plant identification tool mixing textual and visual information retrieval (in the context of the Pl@ntNet project) and a research project on high-throughput analysis of root architecture images.

5.8. SciFloware

Participants: Dimitri Dupuis, Didier Parigot [contact].

URL: http://www-sop.inria.fr/members/Didier.Parigot/pmwiki/Scifloware

SciFloware is an action of technology development (ADT Inria) with the goal of developing a middleware for the execution of scientific workflows in a distributed and parallel way. It capitalizes on our experience with SON and an innovative algebraic approach to the management of scientific workflows. SciFloware provides a development environment and a runtime environment for scientific workflows, interoperable with existing systems. We validate SciFloware with workflows for analyzing biological data provided by our partners CIRAD, INRA and IRD.

5.9. WebSmatch (Web Schema Matching)

Participants: Emmanuel Castanier, Patrick Valduriez [contact].

URL: http://websmatch.gforge.inria.fr/

In the context of an action of technology development (ADT) started in october 2010, WebSmatch is a flexible, open environment for discovering and matching complex schemas from many heterogeneous data sources over the Web. It provides three basic functions: (1) metadata extraction from data sources; (2) schema matching (both 2-way and n-way schema matching), (3) schema clustering to group similar schemas together. WebSmatch is being delivered through Web services, to be used directly by data integrators or other tools, with RIA clients. Implemented in Java, delivered as Open Source Software (under LGPL) and protected by a deposit at APP (Agence de Protection des Programmes). WebSmatch is being used by Datapublica and CIRAD to integrate public data sources.