



RESEARCH CENTER  
**Paris**

FIELD

Activity Report 2018

**Section Software**

Edition: 2019-03-07



ALGORITHMICS, PROGRAMMING, SOFTWARE AND ARCHITECTURE

|                                        |    |
|----------------------------------------|----|
| 1. ANTIQUE Project-Team                | 4  |
| 2. AOSTE2 Team                         | 8  |
| 3. CASCADE Project-Team (section vide) | 11 |
| 4. GALLIUM Project-Team                | 12 |
| 5. OURAGAN Team                        | 15 |
| 6. PARKAS Project-Team                 | 17 |
| 7. PIR2 Project-Team                   | 22 |
| 8. POLSYS Project-Team                 | 26 |
| 9. PROSECCO Project-Team               | 28 |
| 10. SECRET Project-Team                | 31 |

APPLIED MATHEMATICS, COMPUTATION AND SIMULATION

|                                         |    |
|-----------------------------------------|----|
| 11. CAGE Project-Team (section vide)    | 32 |
| 12. MATHERIALS Project-Team             | 33 |
| 13. MATHRISK Project-Team               | 34 |
| 14. MOKAPLAN Project-Team               | 36 |
| 15. QUANTIC Project-Team (section vide) | 37 |
| 16. SIERRA Project-Team                 | 38 |

DIGITAL HEALTH, BIOLOGY AND EARTH

|                         |    |
|-------------------------|----|
| 17. ANGE Project-Team   | 39 |
| 18. ARAMIS Project-Team | 41 |
| 19. MAMBA Project-Team  | 43 |
| 20. REO Project-Team    | 45 |
| 21. SERENA Project-Team | 47 |

NETWORKS, SYSTEMS AND SERVICES, DISTRIBUTED COMPUTING

|                                         |    |
|-----------------------------------------|----|
| 22. ALPINES Project-Team                | 50 |
| 23. DELYS Team (section vide)           | 52 |
| 24. DYOGENE Project-Team (section vide) | 53 |
| 25. EVA Project-Team                    | 54 |
| 26. GANG Project-Team                   | 56 |
| 27. MIMOVE Project-Team                 | 57 |
| 28. WHISPER Project-Team                | 60 |

PERCEPTION, COGNITION AND INTERACTION

|                         |    |
|-------------------------|----|
| 29. ALMAnaCH Team       | 61 |
| 30. COML Team           | 65 |
| 31. RITS Project-Team   | 66 |
| 32. VALDA Project-Team  | 67 |
| 33. WILLOW Project-Team | 69 |

## ANTIQUÉ Project-Team

# 5. New Software and Platforms

## 5.1. APRON

**SCIENTIFIC DESCRIPTION:** The APRON library is intended to be a common interface to various underlying libraries/abstract domains and to provide additional services that can be implemented independently from the underlying library/abstract domain, as shown by the poster on the right (presented at the SAS 2007 conference). You may also look at:

**FUNCTIONAL DESCRIPTION:** 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.

- Participants: Antoine Miné and Bertrand Jeannot
- Contact: Antoine Miné
- URL: <http://apron.cri.ensmp.fr/library/>

## 5.2. Astrée

*The AstréeA Static Analyzer of Asynchronous Software*

**KEYWORDS:** Static analysis - Static program analysis - Program verification - Software Verification - Abstraction

**SCIENTIFIC DESCRIPTION:** 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.

**FUNCTIONAL DESCRIPTION:** 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.

Astrée is a static analyzer for sequential programs based on abstract interpretation. The Astrée static analyzer aims at proving the absence of runtime errors in programs written in the C programming language.

- Participants: Antoine Miné, Jérôme Feret, Laurent Mauborgne, Patrick Cousot, Radhia Cousot and Xavier Rival
- Partners: CNRS - ENS Paris - AbsInt Angewandte Informatik GmbH
- Contact: Patrick Cousot
- URL: <http://www.astree.ens.fr/>

### 5.3. AstréeA

*The AstréeA Static Analyzer of Asynchronous Software*

KEYWORDS: Static analysis - Static program analysis

SCIENTIFIC DESCRIPTION: 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 AR-INC 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.

FUNCTIONAL DESCRIPTION: AstréeA is a static analyzer prototype for parallel software based on abstract interpretation. The AstréeA prototype is a fork of the Astrée static analyzer that adds support for analyzing parallel embedded C software.

- Participants: Antoine Miné, Jérôme Feret, Patrick Cousot, Radhia Cousot and Xavier Rival
- Partners: CNRS - ENS Paris - AbsInt Angewandte Informatik GmbH
- Contact: Patrick Cousot
- URL: <http://www.astreea.ens.fr/>

### 5.4. ClangML

KEYWORD: Compilation

FUNCTIONAL DESCRIPTION: 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.

- Participants: Devin Mccoughlin, François Berenger and Pippijn Van Steenhoven
- Contact: Xavier Rival
- URL: <https://github.com/Antique-team/clangml/tree/master/clang>

### 5.5. FuncTion

SCIENTIFIC DESCRIPTION: 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. 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).

**FUNCTIONAL DESCRIPTION:** 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.

- Participants: Antoine Miné and Caterina Urban
- Contact: Caterina Urban
- URL: <http://www.di.ens.fr/~urban/FuncTion.html>

## 5.6. HOO

*Heap Abstraction for Open Objects*

**FUNCTIONAL DESCRIPTION:** 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.

- Participant: Arlen Cox
- Contact: Arlen Cox

## 5.7. MemCAD

*The MemCAD static analyzer*

**KEYWORDS:** Static analysis - Abstraction

**FUNCTIONAL DESCRIPTION:** MemCAD is 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.

- Participants: Antoine Toubhans, François Berenger, Huisong Li and Xavier Rival
- Contact: Xavier Rival
- URL: <http://www.di.ens.fr/~rival/memcad.html>

## 5.8. KAPPA

*A rule-based language for modeling interaction networks*

**KEYWORDS:** Systems Biology - Modeling - Static analysis - Simulation - Model reduction

**SCIENTIFIC DESCRIPTION:** OpenKappa is a collection of tools to build, debug and run models of biological pathways. It contains a compiler for the Kappa Language, a static analyzer (for debugging models), a simulator, a compression tool for causal traces, and a model reduction tool.

FUNCTIONAL DESCRIPTION: Kappa is provided with the following tools: - a compiler - a stochastic simulator - a static analyzer - a trace compression algorithm - an ODE generator.

RELEASE FUNCTIONAL DESCRIPTION: On line UI, Simulation is based on a new data-structure (see ESOP 2017 ), New abstract domains are available in the static analyzer (see SASB 2016), Local traces (see TCBB 2018), Reasoning on polymers (see SASB 2018).

- Participants: Jean Krivine, Jérôme Feret, Kim Quyen Ly, Pierre Boutillier, Russ Harmer, Vincent Danos and Walter Fontana
- Partners: ENS Lyon - Université Paris-Diderot - HARVARD Medical School
- Contact: Jérôme Feret
- URL: <http://www.kappalanguage.org/>

## 5.9. QUICr

FUNCTIONAL DESCRIPTION: 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.

- Participant: Arlen Cox
- Contact: Arlen Cox

## 5.10. LCertify

KEYWORD: Compilation

SCIENTIFIC DESCRIPTION: 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.

FUNCTIONAL DESCRIPTION: 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 semantic equivalence. 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. ntees that no compiler bug did cause incorrect code to be generated.

- Participant: Xavier Rival
- Partners: CNRS - ENS Paris
- Contact: Xavier Rival
- URL: <http://www.di.ens.fr/~rival/lcertify.html>

## 5.11. Zarith

FUNCTIONAL DESCRIPTION: 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 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.

- Participants: Antoine Miné, Pascal Cuoq and Xavier Leroy
- Contact: Antoine Miné
- URL: <http://forge.ocamlcore.org/projects/zarith>

## AOSTE2 Team

# 6. New Software and Platforms

## 6.1. SynDEX

**KEYWORDS:** Distributed - Optimization - Real time - Embedded systems - Scheduling analyses

**SCIENTIFIC DESCRIPTION:** SynDEX is a system level CAD software implementing the AAA methodology for rapid prototyping and for optimizing distributed real-time embedded applications. It is developed in OCaml.

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

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

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

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

**FUNCTIONAL DESCRIPTION:** Software for optimising the implementation of embedded distributed real-time applications and generating efficient and correct by construction code

**NEWS OF THE YEAR:** We improved the distribution and scheduling heuristics to take into account the needs of co-simulation.

- Participant: Yves Sorel
- Contact: Yves Sorel
- URL: <http://www.syndex.org>

## 6.2. EVT Kopernic

**KEYWORDS:** Embedded systems - Worst Case Execution Time - Real-time application - Statistics

**SCIENTIFIC DESCRIPTION:** The EVT-Kopernic tool is an implementation of the Extreme Value Theory (EVT) for the problem of the statistical estimation of worst-case bounds for the execution time of a program on a processor. Our implementation uses the two versions of EVT - GEV and GPD - to propose two independent methods of estimation. Their results are compared and only results that are sufficiently close allow to validate an estimation. Our tool is proved predictable by its unique choice of block (GEV) and threshold (GPD) while proposing reproducible estimations.



**FUNCTIONAL DESCRIPTION:** EVT-Kopernic is tool proposing a statistical estimation for bounds on worst-case execution time of a program on a processor. The estimator takes into account dependences between execution times by learning from the history of execution, while dealing also with cases of small variability of the execution times.

**NEWS OF THE YEAR:** Any statistical estimator should come with an representative measurement protocole based on the processus of composition, proved correct. We propose the first such principle of composition while using a Bayesien modeling taking into account iteratively different measurement models. The composition model has been described in a patent submitted this year with a scientific publication under preparation.

- Participants: Adriana Gogonel and Liliana Cucu
- Contact: Adriana Gogonel
- URL: <http://inria-rscript.serveftp.com/>

### 6.3. LoPhT-manycore

*Logical to Physical Time compiler for many cores*

**KEYWORDS:** Real time - Compilation - Task scheduling - Automatic parallelization

**SCIENTIFIC DESCRIPTION:** Lopht is a system-level compiler for embedded systems, whose objective is to fully automate the implementation process for certain classes of embedded systems. Like in a classical compiler (e.g. gcc), its input is formed of two objects. The first is a program providing a platform-indepedent description of the functionality to implement and of the non-functional requirements it must satisfy (e.g. real-time, partitioning). This is provided under the form of a data-flow synchronous program annotated with non-functional requirements. The second is a description of the implementation platform, defining the topology of the platform, the capacity of its elements, and possibly platform-dependent requirements (e.g. allocation).

From these inputs, Lopht produces all the C code and configuration information needed to allow compilation and execution on the physical target platform. Implementations are correct by construction Resulting implementations are functionally correct and satisfy the non-functional requirements. Lopht-manycore is a version of Lopht targeting shared-memory many-core architectures.

The algorithmic core of Lopht-manycore is formed of timing analysis, allocation, scheduling, and code generation heuristics which rely on four fundamental choices. 1) A static (off-line) real-time scheduling approach where allocation and scheduling are represented using time tables (also known as scheduling or reservation tables). 2) Scalability, attained through the use of low-complexity heuristics for all synthesis and associated analysis steps. 3) Efficiency (of generated implementations) is attained through the use of precise representations of both functionality and the platform, which allow for fine-grain allocation of resources such as CPU, memory, and communication devices such as network-on-chip multiplexers. 4) Full automation, including that of the timing analysis phase.

The last point is characteristic to Lopht-manycore. Existing methods for schedulability analysis and real-time software synthesis assume the existence of a high-level timing characterization that hides much of the hardware complexity. For instance, a common hypothesis is that synchronization and interference costs are accounted for in the duration of computations. However, the high-level timing characterization is seldom (if ever) soundly derived from the properties of the platform and the program. In practice, large margins (e.g. 100%) with little formal justification are added to computation durations to account for hidden hardware complexity. Lopht-manycore overcomes this limitation. Starting from the worst-case execution time (WCET) estimations of computation operations and from a precise and safe timing model of the platform, it maintains a precise timing accounting throughout the mapping process. To do this, timing accounting must take into account all details of allocation, scheduling, and code generation, which in turn must satisfy specific hypotheses.

FUNCTIONAL DESCRIPTION: Accepted input languages for functional specifications include dialects of Lustre such as Heptagon and Scade v4. To ensure the respect of real-time requirements, Lopht-manycore pilots the use of the worst-case execution time (WCET) analysis tool (ait from AbsInt). By doing this, and by using a precise timing model for the platform, Lopht-manycore eliminates the need to adjust the WCET values through the addition of margins to the WCET values that are usually both large and without formal safety guarantees. The output of Lopht-manycore is formed of all the multi-threaded C code and configuration information needed to allow compilation, linking/loading, and real-time execution on the target platform.

NEWS OF THE YEAR: In the framework of the ITEA3 ASSUME project we have extended the Lopht-manycore to allow multiple cores to access the same memory bank at the same time. To do this, the timing accounting of Lopht has been extended to take into account memory access interferences during the allocation and scheduling process. Lopht now also pilots the aiT static WCET analysis tool from AbsInt by generating the analysis scripts, thus ensuring the consistency between the hypotheses made by Lopht and the way timing analysis is performed by aiT. As a result, we are now able to synthesize code for the computing clusters of the Kalray MPPA256 platform. Lopht-manycore is evaluated on avionics case studies in the perspective of increasing its technology readiness level for this application class.

- Participants: Dumitru Potop-Butucaru and Keryan Didier
- Contact: Dumitru Potop-Butucaru

**CASCADE Project-Team (section vide)**

## GALLIUM Project-Team

# 6. New Software and Platforms

## 6.1. CompCert

*The CompCert formally-verified C compiler*

KEYWORDS: Compilers - Formal methods - Deductive program verification - C - Coq

FUNCTIONAL DESCRIPTION: CompCert is a compiler for the C programming language. Its intended use is the compilation of life-critical and mission-critical software written in C and meeting high levels of assurance. It accepts most of the ISO C 99 language, with some exceptions and a few extensions. It produces machine code for the ARM, PowerPC, RISC-V, and x86 architectures. What sets CompCert C apart from any other production compiler, is that it is formally verified to be exempt from miscompilation issues, using machine-assisted mathematical proofs (the Coq proof assistant). In other words, the executable code it produces is proved to behave exactly as specified by the semantics of the source C program. This level of confidence in the correctness of the compilation process is unprecedented and contributes to meeting the highest levels of software assurance. In particular, using the CompCert C compiler is a natural complement to applying formal verification techniques (static analysis, program proof, model checking) at the source code level: the correctness proof of CompCert C guarantees that all safety properties verified on the source code automatically hold as well for the generated executable.

RELEASE FUNCTIONAL DESCRIPTION: Novelties include a formally-verified type checker for CompCert C, a more careful modeling of pointer comparisons against the null pointer, algorithmic improvements in the handling of deeply nested struct and union types, much better ABI compatibility for passing composite values, support for GCC-style extended inline asm, and more complete generation of DWARF debugging information (contributed by AbsInt).

- Participants: Xavier Leroy, Sandrine Blazy, Jacques-Henri Jourdan, Sylvie Boldo and Guillaume Melquiond
- Partner: AbsInt Angewandte Informatik GmbH
- Contact: Xavier Leroy
- URL: <http://compcert.inria.fr/>

## 6.2. Diy

*Do It Yourself*

KEYWORD: Parallelism

FUNCTIONAL DESCRIPTION: 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.

- Participants: Jade Alglave and Luc Maranget
- Partner: University College London UK
- Contact: Luc Maranget
- URL: <http://diy.inria.fr/>

## 6.3. Menhir

KEYWORDS: Compilation - Context-free grammars - Parsing

**FUNCTIONAL DESCRIPTION:** Menhir is a LR(1) parser generator for the OCaml programming language. That is, Menhir compiles LR(1) grammar specifications down to OCaml code. Menhir was designed and implemented by François Pottier and Yann Régis-Gianas.

- Contact: François Pottier
- Publications: [A Simple, Possibly Correct LR Parser for C11 - Reachability and Error Diagnosis in LR\(1\) Parsers](#)

## 6.4. OCaml

**KEYWORDS:** Functional programming - Static typing - Compilation

**FUNCTIONAL DESCRIPTION:** The OCaml language is a functional programming language that combines safety with expressiveness through the use of a precise and flexible type system with automatic type inference. The OCaml system is a comprehensive implementation of this language, featuring two compilers (a bytecode compiler, for fast prototyping and interactive use, and a native-code compiler producing efficient machine code for x86, ARM, PowerPC and System Z), a debugger, a documentation generator, a compilation manager, a package manager, and many libraries contributed by the user community.

- Participants: Damien Doligez, Xavier Leroy, Fabrice Le Fessant, Luc Maranget, Gabriel Scherer, Alain Frisch, Jacques Garrigue, Marc Shinwell, Jeremy Yallop and Leo White
- Contact: Damien Doligez
- URL: <https://ocaml.org/>

## 6.5. PASL

**KEYWORD:** Parallel computing

**FUNCTIONAL DESCRIPTION:** PASL is a C++ library for writing parallel programs targeting the broadly available multicore computers. The library provides a high level interface and can still guarantee very good efficiency and performance, primarily due to its scheduling and automatic granularity control mechanisms.

- Participants: Arthur Charguéraud, Michael Rainey and Umut Acar
- Contact: Michael Rainey
- URL: <http://deepsea.inria.fr/pasl/>

## 6.6. ZENON

**FUNCTIONAL DESCRIPTION:** 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 retargeted to output scripts for different frameworks (for example, Isabelle and Dedukti).

- Author: Damien Doligez
- Contact: Damien Doligez
- URL: <http://zenon-prover.org/>

## 6.7. OPAM Builder

**KEYWORDS:** Ocaml - Continuous integration - Opam

FUNCTIONAL DESCRIPTION: OPAM Builder checks in real-time the installability on a computer of all packages after any modification of the repository. To achieve this result, it uses smart mechanisms to compute incremental differences between package updates, to be able to reuse cached compilations, and switch from a quadratic complexity to a linear complexity.

- Partner: OCamlPro
- Contact: Fabrice Le Fessant
- URL: <http://github.com/OCamlPro/opam-builder>

## 6.8. TLAPS

*TLA+ proof system*

KEYWORD: Proof assistant

FUNCTIONAL DESCRIPTION: TLAPS 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 proof steps that can be checked independently. 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+ set theory 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. NEWS OF THE YEAR: Ioannis Filippidis joined the development team in November 2018 and started designing and implementing support for reasoning about TLA+'s ENABLED construct.

- Participants: Damien Doligez, Stephan Merz and IOANNIS FILIPPIDIS
- Contact: Stephan Merz
- URL: <https://tla.msr-inria.inria.fr/tlaps/content/Home.html>

## 6.9. CFML

*Interactive program verification using characteristic formulae*

KEYWORDS: Coq - Software Verification - Deductive program verification - Separation Logic

FUNCTIONAL DESCRIPTION: 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 notations and tactics for manipulating characteristic formulae interactively in Coq.

- Participants: Arthur Charguéraud, Armaël Guéneau and François Pottier
- Contact: Arthur Charguéraud
- URL: <http://www.chargueraud.org/softs/cfml/>

## 6.10. ldrngen

*Liveness-driven random C code generator*

KEYWORDS: Code generation - Randomized algorithms - Static program analysis

FUNCTIONAL DESCRIPTION: The ldrngen program is a generator of C code: On every call it generates a new random C function and prints it to the standard output. The generator is "liveness-driven", which means that it tries to avoid generating dead code: All the computations it generates are (in a certain, limited sense) actually used to compute the function's return value. This is achieved by generating the program backwards, in combination with a simultaneous liveness analysis that guides the random generator's choices.

- Participant: Gergő Barany
- Contact: Gergő Barany
- Publication: [Liveness-Driven Random Program Generation](#)
- URL: <https://github.com/gergo-/ldrgen>

## OURAGAN Team

# 6. New Software and Platforms

## 6.1. ISOTOP

*Topology and geometry of planar algebraic curves*

KEYWORDS: Topology - Curve plotting - Geometric computing

FUNCTIONAL DESCRIPTION: 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.

NEWS OF THE YEAR: In 2018, an engineer from Inria Nancy (Benjamin Dexheimer) finished the implementation of the web server to improve the diffusion of our software.

- Participants: Luis Penaranda, Marc Pouget and Sylvain Lazard
- Contact: Marc Pouget
- Publications: [Rational Univariate Representations of Bivariate Systems and Applications - Separating Linear Forms for Bivariate Systems - On The Topology of Planar Algebraic Curves - New bivariate system solver and topology of algebraic curves - Improved algorithm for computing separating linear forms for bivariate systems - Solving bivariate systems using Rational Univariate Representations - On the topology of planar algebraic curves - On the topology of real algebraic plane curves - Bivariate triangular decompositions in the presence of asymptotes - Separating linear forms and Rational Univariate Representations of bivariate systems](#)
- URL: <https://isotop.gamble.loria.fr/>

## 6.2. RS

FUNCTIONAL DESCRIPTION: Real Roots isolation for algebraic systems with rational coefficients with a finite number of Complex Roots

- Participant: Fabrice Rouillier
- Contact: Fabrice Rouillier
- URL: <https://team.inria.fr/ouragan/software/>

## 6.3. A NewDsc

*A New Descartes*

KEYWORD: Scientific computing

FUNCTIONAL DESCRIPTION: Computations of the real roots of univariate polynomials with rational coefficients.

- Authors: Fabrice Rouillier, Alexander Kobel and Michael Sagraloff
- Partner: Max Planck Institute for Software Systems
- Contact: Fabrice Rouillier
- URL: <https://anewdsc.mpi-inf.mpg.de>

## 6.4. SIROPA

KEYWORDS: Robotics - Kinematics

FUNCTIONAL DESCRIPTION: Library of functions for certified computations of the properties of articulated mechanisms, particularly the study of their singularities

- Authors: Damien Chablat, Fabrice Rouillier, Guillaume Moroz and Philippe Wenger
- Partner: LS2N
- Contact: Guillaume Moroz
- URL: <http://siropa.gforge.inria.fr/>

## **6.5. MPFI**

KEYWORD: Arithmetic

FUNCTIONAL DESCRIPTION: MPFI is a C library based on MPFR and GMP for multi precision floating point arithmetic.

- Contact: Fabrice Rouillier
- URL: <http://mpfi.gforge.inria.fr>



## PARKAS Project-Team

# 5. New Software and Platforms

## 5.1. Cmmtest

**FUNCTIONAL DESCRIPTION:** Cmmtest is a tool for hunting concurrency 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:

compiles the program using the compiler and compiler optimisations that are being tested,

runs the compiled program in an instrumented execution environment that logs all memory accesses to global variables and synchronisations,

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.

- Participants: Anirudh Kumar, Francesco Zappa Nardelli, Pankaj More, Pankaj Pawan, Pankaj Prateek Kewalramani and Robin Morisset
- Contact: Francesco Zappa Nardelli
- URL: <http://www.di.ens.fr/~zappa/projects/cmmtest/>

## 5.2. GCC

**KEYWORDS:** Compilation - Polyhedral compilation

**FUNCTIONAL DESCRIPTION:** 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++, libgccj,...). GCC was originally written as the compiler for the GNU operating system. The GNU system was developed to be 100

- Participants: Albert Cohen, Feng Li, Nhat Minh Le, Riyadh Baghdadi and Tobias Grosser
- Contact: Albert Cohen
- URL: <http://gcc.gnu.org/>

## 5.3. Heptagon

**KEYWORDS:** Compilers - Synchronous Language - Controller synthesis

**FUNCTIONAL DESCRIPTION:** 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 hierarchical 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.

The current version of the compiler includes the following features: - Inclusion of discrete controller synthesis within the compilation: the language is equipped with a behavioral contract mechanisms, where assumptions can be described, as well as an "enforce" property part. The semantics of this latter is that the property should be enforced by controlling the behaviour of the node equipped with the contract. This property will be enforced by an automatically built controller, which will act on free controllable variables given by the programmer. This extension has been named BZR in previous works. - Expression and compilation of array values with modular memory optimization. The language allows the expression and operations on arrays (access, modification, iterators). With the use of location annotations, the programmer can avoid unnecessary array copies.

- Participants: Adrien Guatto, Brice Gelineau, Cédric Pasteur, Eric Rutten, Gwenaël Delaval, Léonard Gérard and Marc Pouzet
- Partners: UGA - ENS Paris - Inria - LIG
- Contact: Gwenaël Delaval
- URL: <http://heptagon.gforge.inria.fr>

## 5.4. isl

FUNCTIONAL DESCRIPTION: 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.

- Participants: Albert Cohen, Sven Verdoolaege and Tobias Grosser
- Contact: Sven Verdoolaege
- URL: <http://freshmeat.net/projects/isl>

## 5.5. Lem

*lightweight executable mathematics*

FUNCTIONAL DESCRIPTION: 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.

- Participants: Francesco Zappa Nardelli, Peter Sewell and Scott Owens
- Contact: Francesco Zappa Nardelli
- URL: <http://www.cl.cam.ac.uk/~pes20/lem/>

## 5.6. Lucid Sychrone

FUNCTIONAL DESCRIPTION: Lucid Sychrone 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.

RELEASE FUNCTIONAL DESCRIPTION: 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 Sychrone 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 originally introduced in Lucid Sychrone.

- Contact: Marc Pouzet
- URL: <http://www.di.ens.fr/~pouzet/lucid-sychrone/>

## 5.7. Lucy-n

*Lucy-n: an n-synchronous data-flow programming language*

FUNCTIONAL DESCRIPTION: 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.

- Participants: Adrien Guatto, Albert Cohen, Louis Mandel and Marc Pouzet
- Contact: Albert Cohen
- URL: <https://www.lri.fr/~mandel/lucy-n/>

## 5.8. Ott

FUNCTIONAL DESCRIPTION: 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:

a LaTeX source file that defines commands to build a typeset version of the definition,

a Coq version of the definition,

an Isabelle version of the definition, and

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.

- Participants: Francesco Zappa Nardelli, Peter Sewell and Scott Owens
- Contact: Francesco Zappa Nardelli
- URL: <http://www.cl.cam.ac.uk/~pes20/ott/>

## 5.9. PPCG

FUNCTIONAL DESCRIPTION: 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).

- Participants: Albert Cohen, Riyadh Baghdadi, Sven Verdoolaege and Tobias Grosser
- Contact: Sven Verdoolaege
- URL: <http://freshmeat.net/projects/ppcg>

## 5.10. ReactiveML

**FUNCTIONAL DESCRIPTION:** 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.

- Participants: Cédric Pasteur, Guillaume Baudart and Louis Mandel
- Contact: Guillaume Baudart

## 5.11. SundialsML

*Sundials/ML*

**KEYWORDS:** Simulation - Mathematics - Numerical simulations

**SCIENTIFIC DESCRIPTION:** 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.

**FUNCTIONAL DESCRIPTION:** Sundials/ML is a comprehensive OCaml interface to the Sundials suite of numerical solvers (CVODE, CVODES, IDA, IDAS, KINSOL, ARKODE).

**RELEASE FUNCTIONAL DESCRIPTION:** Adds support for v3.1.x of the Sundials Suite of numerical solvers.

Notably this release adds support for the new generic matrix and linear solver interfaces. The OCaml interface changes but the library is backward compatible with Sundials 2.7.0.

OCaml 4.02.3 or greater is now required and optionally OCamlMPI 1.03.

\* New Sundials.Matrix and Sundials.LinearSolver modules. \* Better treatment of integer type used for matrix indexing. \* Refactor DIs and SIs modules into Sundials.Matrix. \* Add confidence intervals to performance graph. \* Miscellaneous improvements to configure script. \* Potential incompatibility: changes to some label names: com\_fn -> comm, ite\_type -> iter. \* Untangle the ARKODE mass-solver interface from the Jacobian interface.

- Participants: Jun Inoue, Marc Pouzet and Timothy Bourke
- Partner: UPMC
- Contact: Marc Pouzet
- URL: <http://inria-parkas.github.io/sundialsml/>

## 5.12. Zelus

**SCIENTIFIC DESCRIPTION:** 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.

FUNCTIONAL DESCRIPTION: 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.

- Participants: Marc Pouzet and Timothy Bourke
- Contact: Marc Pouzet

## PI.R2 Project-Team

# 5. New Software and Platforms

## 5.1. Coq

*The Coq Proof Assistant*

KEYWORDS: Proof - Certification - Formalisation

SCIENTIFIC DESCRIPTION: Coq is an interactive proof assistant based on the Calculus of (Co-)Inductive Constructions, extended with universe polymorphism. This type theory features inductive and co-inductive families, an impredicative sort and a hierarchy of predicative universes, making it a very expressive logic. The calculus allows to formalize both general mathematics and computer programs, ranging from theories of finite structures to abstract algebra and categories to programming language metatheory and compiler verification. Coq is organised as a (relatively small) kernel including efficient conversion tests on which are built a set of higher-level layers: a powerful proof engine and unification algorithm, various tactics/decision procedures, a transactional document model and, at the very top an IDE.

FUNCTIONAL DESCRIPTION: Coq provides both a dependently-typed functional programming language and a logical formalism, which, altogether, support the formalisation of mathematical theories and the specification and certification of properties of programs. Coq also provides a large and extensible set of automatic or semi-automatic proof methods. Coq's programs are extractible to OCaml, Haskell, Scheme, ...

RELEASE FUNCTIONAL DESCRIPTION: Coq version 8.8.2 contains the result of refinements and stabilization of features and deprecations, cleanups of the internals of the system along with a few new features.

Summary of changes:

Kernel: fix a subject reduction failure due to allowing fixpoints on non-recursive values (#407), by Matthieu Sozeau. Handling of evars in the VM (#935) by Pierre-Marie Pédro.

Notations: many improvements on recursive notations and support for destructuring patterns in the syntax of notations by Hugo Herbelin.

Proof language: tacticals for profiling, timing and checking success or failure of tactics by Jason Gross. The focusing bracket { supports single-numbered goal selectors, e.g. 2:{, (#6551) by Théo Zimmermann.

Vernacular: cleanup of definition commands (#6653) by Vincent Laporte and more uniform handling of the Local flag (#1049), by Maxime Dénès. Experimental Show Extraction command (#6926) by Pierre Letouzey. Coercion now accepts Prop or Type as a source (#6480) by Arthur Charguéraud. Export modifier for options allowing to export the option to modules that Import and not only Require a module (#6923), by Pierre-Marie Pédro.

Universes: many user-level and API level enhancements: qualified naming and printing, variance annotations for cumulative inductive types, more general constraints and enhancements of the minimization heuristics, interaction with modules by Gaëtan Gilbert, Pierre-Marie Pédro and Matthieu Sozeau.

Library: Decimal Numbers library (#6599) by Pierre Letouzey and various small improvements.

Documentation: a large community effort resulted in the migration of the reference manual to the Sphinx documentation tool. The new documentation infrastructure (based on Sphinx) is by Clément Pit-Claudel. The migration was coordinated by Maxime Dénès and Paul Steckler, with some help of Théo Zimmermann during the final integration phase. The 14 people who ported the manual are Calvin Beck, Heiko Becker, Yves Bertot, Maxime Dénès, Richard Ford, Pierre Letouzey, Assia Mahboubi, Clément Pit-Claudel, Laurence Rideau, Matthieu Sozeau, Paul Steckler, Enrico Tassi, Laurent Théry, Nikita Zyuizin.

Tools: experimental -mangle-names option to coqtop/coqc for linting proof scripts (#6582), by Jasper Hugunin. Main changes:

Critical soundness bugs were fixed between versions 8.8.0 and 8.8.2, and a PDF version of the reference manual was made available. The Windows installer also includes many more external packages that can be individually selected for installation.

On the implementation side, the `dev/doc/changes.md` file documents the numerous changes to the implementation and improvements of interfaces. The file provides guidelines on porting a plugin to the new version.

More information can be found in the `CHANGES` file. Feedback and bug reports are extremely welcome.

Distribution Installers for Windows 32 bits (i686), Windows 64 bits (x8\_64) and macOS are available. They come bundled with CoqIDE. Windows binaries now include the Bignum library.

Complete sources of the files installed by the Windows installers are made available, to comply with license requirements.

**NEWS OF THE YEAR:** Version 8.8.0 was released in April 2018 and version 8.8.2 in September 2018. This is the third release of Coq developed on a time-based development cycle. Its development spanned 6 months from the release of Coq 8.7 and was based on a public road-map. It attracted many external contributions. Code reviews and continuous integration testing were systematically used before integration of new features, with an important focus given to compatibility and performance issues.

The main advances in this version are cleanups and fixes in the many different components of the system, ranging from low level kernel fixes to advances in the support of notations and tacticals for selecting goals. A large community effort was made to move the documentation to the Sphinx format, providing a more accessible online resource to users.

- Participants: Abhishek Anand, C. J. Bell, Yves Bertot, Frédéric Besson, Tej Chajed, Pierre Courtieu, Maxime Denes, Julien Forest, Emilio Jesús Gallego Arias, Gaëtan Gilbert, Benjamin Grégoire, Jason Gross, Hugo Herbelin, Ralf Jung, Matej Kosik, Sam Pablo Kuper, Xavier Leroy, Pierre Letouzey, Assia Mahboubi, Cyprien Mangin, Érik Martin-Dorel, Olivier Marty, Guillaume Melquiond, Pierre-Marie Pédro, Benjamin C. Pierce, Lars Rasmusson, Yann Régis-Gianas, Lionel Rieg, Valentin Robert, Thomas Sibut-Pinote, Michael Soegtrop, Matthieu Sozeau, Arnaud Spiwack, Paul Steckler, George Stelle, Pierre-Yves Strub, Enrico Tassi, Hendrik Tews, Laurent Théry, Amin Timany, Vadim Zaliva and Théo Zimmermann
- Partners: CNRS - Université Paris-Sud - ENS Lyon - Université Paris-Diderot
- Contact: Matthieu Sozeau
- Publication: [The Coq Proof Assistant, version 8.8.0](#)
- URL: <http://coq.inria.fr/>

## 5.2. Equations

**KEYWORDS:** Coq - Dependent Pattern-Matching - Proof assistant - Functional programming

**SCIENTIFIC DESCRIPTION:** Equations is a tool designed to help with the definition of programs in the setting of dependent type theory, as implemented in the Coq proof assistant. Equations provides a syntax for defining programs by dependent pattern-matching and well-founded recursion and compiles them down to the core type theory of Coq, using the primitive eliminators for inductive types, accessibility and equality. In addition to the definitions of programs, it also automatically derives useful reasoning principles in the form of propositional equations describing the functions, and an elimination principle for calls to this function. It realizes this using a purely definitional translation of high-level definitions to core terms, without changing the core calculus in any way, or using axioms.

**FUNCTIONAL DESCRIPTION:** Equations is a function definition plugin for Coq (supporting Coq 8.6 and 8.7), that allows the definition of functions by dependent pattern-matching and well-founded, mutual or nested structural recursion and compiles them into core terms. It automatically derives the clauses equations, the graph of the function and its associated elimination principle.

Equations is based on a simplification engine for the dependent equalities appearing in dependent eliminations that is also usable as a separate tactic, providing an axiom-free variant of dependent destruction. The main features of Equations include:

Dependent pattern-matching in the style of Agda/Epigram, with inaccessible patterns, with and where clauses. The use of the K axiom or a proof of K is configurable.

Support for well-founded recursion using by rec annotations, and automatic derivation of the subterm relation for inductive families.

Support for mutual and nested structural recursion using with and where auxiliary definitions, allowing to factor multiple uses of the same nested fixpoint definition. It proves the expected elimination principles for mutual and nested definitions.

Automatic generation of the defining equations as rewrite rules for every definition.

Automatic generation of the unfolding lemma for well-founded definitions (requiring only functional extensionality).

Automatic derivation of the graph of the function and its elimination principle. In case the automation fails to prove these principles, the user is asked to provide a proof.

A new dependent elimination tactic based on the same splitting tree compilation scheme that can advantageously replace dependent destruction and sometimes inversion as well. The as clause of dependent elimination allows to specify exactly the patterns and naming of new variables needed for an elimination.

A set of Derive commands for automatic derivation of constructions from an inductive type: its signature, no-confusion property, well-founded subterm relation and decidable equality proof, if applicable.

NEWS OF THE YEAR: Equations 1.0 was released in december this year, after 7 years of (non-continuous) development. It provides the first feature-full version of the software. It has been tried and tested on small to medium scale examples (available on the website). Equations was presented at the Type Theory Tools EUTypes meeting in January 2017 in Paris, and another demo/presentation will be given at PEPM 2018 in Los Angeles in January 2018.

- Participants: Matthieu Sozeau and Cyprien Mangin
- Contact: Matthieu Sozeau
- Publications: [Equations reloaded - Equations for Hereditary Substitution in Leivant's Predicative System F: A Case Study](#) - [Equations: A Dependent Pattern-Matching Compiler](#)
- URL: <http://mattam82.github.io/Coq-Equations/>

### 5.3. Rewr

*Rewriting methods in algebra*

KEYWORDS: Computer algebra system (CAS) - Rewriting systems - Algebra

FUNCTIONAL DESCRIPTION: Rewr is a prototype of computer algebra system, using rewriting methods to compute resolutions and homotopical invariants of monoids. The library implements various classical constructions of rewriting theory (such as completion), improved by experimental features coming from Garside theory, and allows homotopical algebra computations based on Squier theory. Specific functionalities have been developed for usual classes of monoids, such as Artin monoids and plactic monoids.

NEWS OF THE YEAR: Rewr has been extended with the experimental KGB completion algorithm, based on Knuth-Bendix completion procedure improved by techniques coming from Garside theory.

- Participants: Yves Guiraud and Samuel Mimram
- Contact: Yves Guiraud
- Publications: [Higher-dimensional categories with finite derivation type - Higher-dimensional normalisation strategies for acyclicity](#) - [Coherent presentations of Artin monoids](#) - [A Homotopical Completion Procedure with Applications to Coherence of Monoids](#) - [Polygraphs of finite derivation type](#) - [Quadratic normalisation in monoids](#)
- URL: <http://www.lix.polytechnique.fr/Labo/Samuel.Mimram/rewr>



## 5.4. Catex

KEYWORDS: LaTeX - String diagram - Algebra

FUNCTIONAL DESCRIPTION: Catex is a Latex package and an external tool to typeset string diagrams easily from their algebraic expression. Catex works similarly to Bibtex.

NEWS OF THE YEAR: It is now possible to add labels to objects and morphisms

- Participant: Yves Guiraud
- Contact: Yves Guiraud
- URL: <https://www.irif.fr/~guiraud/catex/catex.zip>

## 5.5. Cox

KEYWORDS: Computer algebra system (CAS) - Rewriting systems - Algebra

FUNCTIONAL DESCRIPTION: Cox is a Python library for the computation of coherent presentations of Artin monoids, with experimental features to compute the lower dimensions of the Salvetti complex.

- Participant: Yves Guiraud
- Contact: Yves Guiraud
- Publications: [Coherent presentations of Artin monoids - A Homotopical Completion Procedure with Applications to Coherence of Monoids](#)
- URL: <https://www.irif.fr/~guiraud/cox/cox.zip>

---

## POLSYS Project-Team

# 5. New Software and Platforms

## 5.1. Epsilon

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

- Contact: Dongming Wang
- URL: <http://wang.cc4cm.org/epsilon/index.html>

## 5.2. FGb

KEYWORDS: Gröbner bases - Nonlinear system - Computer algebra

FUNCTIONAL DESCRIPTION: FGb is a powerful software for computing Gröbner bases. It includes the new generation of algorithms 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.

- Participant: Jean Charles Faugère
- Contact: Jean-Charles Faugère
- URL: <http://www-polsys.lip6.fr/~jcf/FGb/index.html>

## 5.3. FGb Light

FUNCTIONAL DESCRIPTION: Gröbner basis computation modulo  $p$  ( $p$  is a prime integer of 16 bits).

- Participant: Jean-Charles Faugère
- Contact: Jean-Charles Faugère
- URL: <http://www-polsys.lip6.fr/~jcf/FGb/index.html>

## 5.4. GBLA

FUNCTIONAL DESCRIPTION: GBLA is an open source C library for linear algebra specialized for eliminating matrices generated during Gröbner basis computations in algorithms like F4 or F5.

- Contact: Jean-Charles Faugère
- URL: <http://www-polsys.lip6.fr/~jcf/GBLA/index.html>

## 5.5. HFEBoost

FUNCTIONAL DESCRIPTION: Public-key cryptography system enabling an authentication of dematerialized data.

- Authors: Jean-Charles Faugère and Ludovic Perret
- Partner: UPMC
- Contact: Jean-Charles Faugère
- URL: <http://www-polsys.lip6.fr/Links/hfeboost.html>

## 5.6. RAGlib

*Real Algebraic Geometry library*

FUNCTIONAL DESCRIPTION: RAGLib is a powerful library, written in Maple, dedicated to solving over the reals polynomial systems. It is based on the FGb library for computing Grobner bases. It provides functionalities for deciding the emptiness and/or computing sample points to real solution sets of polynomial systems of equations and inequalities. This library provides implementations of the state-of-the-art algorithms with the currently best known asymptotic complexity for those problems.

- Contact: Mohab Safey El Din
- URL: <http://www-polsys.lip6.fr/~safey/RAGLib/>

## 5.7. RealCertify

KEYWORDS: Polynomial or analytical systems - Univariate polynomial - Real solving

FUNCTIONAL DESCRIPTION: The package RealCertify aims at providing a full suite of hybrid algorithms for computing certificates of non-negativity based on numerical software for solving linear matrix inequalities. The module univsos handles the univariate case and the module multivsos is designed for the multivariate case.

- Contact: Mohab Safey El Din
- URL: <https://gricad-gitlab.univ-grenoble-alpes.fr/magronv/RealCertify>

## 5.8. SLV

FUNCTIONAL DESCRIPTION: 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 hundreds of Megabytes. Currently the code consists of approx. 5000 lines.

- Contact: Elias Tsigaridas
- URL: <http://www-polsys.lip6.fr/~elias/soft>

## 5.9. SPECTRA

*Semidefinite Programming solved Exactly with Computational Tools of Real Algebra*

KEYWORD: Linear Matrix Inequalities

FUNCTIONAL DESCRIPTION: SPECTRA is a Maple library devoted to solving exactly Semi-Definite Programs. It can handle rank constraints on the solution. It is based on the FGb library for computing Gröbner bases and provides either certified numerical approximations of the solutions or exact representations thereof.

- Contact: Mohab Safey El Din
- URL: <http://homepages.laas.fr/henrion/software/spectra/>

## PROSECCO Project-Team

# 6. New Software and Platforms

## 6.1. Cryptosense Analyzer

**SCIENTIFIC DESCRIPTION:** 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.

**FUNCTIONAL DESCRIPTION:** Cryptosense Analyzer (formerly known as Tookan) is a security analysis tool for cryptographic devices such as smartcards,

- Participants: Graham Steel and Romain Bardou
- Contact: Graham Steel
- URL: <https://cryptosense.com/>

## 6.2. CryptoVerif

*Cryptographic protocol verifier in the computational model*

**KEYWORDS:** Security - Verification - Cryptographic protocol

**FUNCTIONAL DESCRIPTION:** CryptoVerif 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. It also provides an explicit formula that gives the probability of breaking the protocol as a function of the probability of breaking each primitives, this is the exact security framework.

**NEWS OF THE YEAR:** Bruno Blanchet modified ProVerif and CryptoVerif to improve the compatibility between these two tools (see the section on ProVerif). This feature is released in CryptoVerif 2.00.

Bruno Blanchet implemented several extensions of CryptoVerif, in particular: 1) reworked the model of Diffie-Hellman key agreements, in particular to account for the absence of public key validation in popular Diffie-Hellman groups like Curve25519, which is used in many modern protocols, 2) support for the proof of indistinguishability between two games given by the user, 3) facilitate the interactive proofs. Program points, used for instance to insert case distinctions, can now be designated as the line that matches a regular expression, instead of using a number. This is much more stable in case the protocol model is slightly modified. Groups of variables can be designated as all variables that match a regular expression. These features are not released yet.

- Participants: Bruno Blanchet and David Cadé
- Contact: Bruno Blanchet

- Publications: [Composition Theorems for CryptoVerif and Application to TLS 1.3 - Composition Theorems for CryptoVerif and Application to TLS 1.3 - Proved Implementations of Cryptographic Protocols in the Computational Model - Proved Generation of Implementations from Computationally Secure Protocol Specifications - Verified Models and Reference Implementations for the TLS 1.3 Standard Candidate - Verified Models and Reference Implementations for the TLS 1.3 Standard Candidate - Symbolic and Computational Mechanized Verification of the ARINC823 Avionic Protocols - Automated Verification for Secure Messaging Protocols and Their Implementations: A Symbolic and Computational Approach](#)
- URL: <http://cryptoverif.inria.fr/>

### 6.3. F\*

*FStar*

KEYWORDS: Programming language - Software Verification

FUNCTIONAL DESCRIPTION: 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 Fw (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.

- Participants: Antoine Delignat-Lavaud, Catalin Hritcu, Cédric Fournet, Chantal Keller, Karthikeyan Bhargavan and Pierre-Yves Strub
- Contact: Catalin Hritcu
- URL: <https://www.fstar-lang.org/>

### 6.4. miTLS

KEYWORDS: Cryptographic protocol - Software Verification

FUNCTIONAL DESCRIPTION: miTLS is a verified reference implementation of the TLS protocol. Our code fully supports its wire formats, ciphersuites, sessions and connections, re-handshakes and resumptions, alerts and errors, and data fragmentation, as prescribed in the RFCs, it interoperates with mainstream web browsers and servers. At the same time, our code is carefully structured to enable its modular, automated verification, from its main API down to computational assumptions on its cryptographic algorithms.

- Participants: Alfredo Pironti, Antoine Delignat-Lavaud, Cédric Fournet, Jean-Karim Zinzindohoué, Karthikeyan Bhargavan, Pierre-Yves Strub and Santiago Zanella-Béguelin
- Contact: Karthikeyan Bhargavan
- URL: <https://github.com/mitls/mitls-fstar>

### 6.5. ProVerif

KEYWORDS: Security - Verification - Cryptographic protocol

FUNCTIONAL DESCRIPTION: ProVerif 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 verify various security properties (secrecy, authentication, process equivalences).

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.

NEWS OF THE YEAR: Marc Sylvestre extended his interactive simulator of protocols modeled in ProVerif to simulate the semantics of biprocesses, used to prove observational equivalence between two processes. He also made minor improvements to this simulator and to the graphical display of attacks.

Bruno Blanchet modified ProVerif and CryptoVerif to improve the compatibility between these two tools. It is now possible for simple examples to use the same input file with both tools, for instance to try to find attacks in the symbolic model using ProVerif, and if no attack is found, then prove the protocol in the computational model using CryptoVerif. For more complex examples, the differences between the files to provide for each tool are considerably reduced. The cryptographic primitives are specified in distinct libraries, one for ProVerif and one for CryptoVerif, because the assumptions on primitives are very different in the symbolic and computational models. These features are released in ProVerif 2.00.

Vincent Cheval and Bruno Blanchet implemented several extensions of ProVerif: 1) support for integer counters, with incrementation and inequality tests, 2) lemmas and axioms to give intermediate results to ProVerif, which it exploits to help proving subsequent queries, by deriving additional information in the Horn clauses that it uses to perform the proofs, 3) proofs by induction on the length of the trace, by giving as lemma the property to prove, but obviously for strictly shorter traces. These features are not released yet.

- Participants: Bruno Blanchet, Marc Sylvestre and Vincent Cheval
- Contact: Bruno Blanchet
- Publications: [Automated reasoning for equivalences in the applied pi calculus with barriers](#) - [Automated Reasoning for Equivalences in the Applied Pi Calculus with Barriers](#) - [Automated reasoning for equivalences in the applied pi calculus with barriers](#) - [Modeling and Verifying Security Protocols with the Applied Pi Calculus and ProVerif](#) - [Automatic Verification of Security Protocols in the Symbolic Model: The Verifier ProVerif](#) - [Verified Models and Reference Implementations for the TLS 1.3 Standard Candidate](#) - [Verified Models and Reference Implementations for the TLS 1.3 Standard Candidate](#) - [Automated Verification for Secure Messaging Protocols and Their Implementations: A Symbolic and Computational Approach](#) - [Symbolic and Computational Mechanized Verification of the ARINC823 Avionic Protocols](#) - [Symbolic and Computational Mechanized Verification of the ARINC823 Avionic Protocols](#)
- URL: <http://proverif.inria.fr/>

## 6.6. HACL\*

*High Assurance Cryptography Library*

KEYWORDS: Cryptography - Software Verification

FUNCTIONAL DESCRIPTION: HACL\* is a formally verified cryptographic library in F\*, developed by the Prosecco team at Inria Paris in collaboration with Microsoft Research, as part of Project Everest.

HACL stands for High-Assurance Cryptographic Library and its design is inspired by discussions at the HACS series of workshops. The goal of this library is to develop verified C reference implementations for popular cryptographic primitives and to verify them for memory safety, functional correctness, and secret independence.

- Contact: Karthikeyan Bhargavan
- URL: <https://github.com/mitls/hacl-star>

## SECRET Project-Team

# 6. New Software and Platforms

## 6.1. CFS

FUNCTIONAL DESCRIPTION: Reference implementation of parallel CFS (reinforced version of the digital signature scheme CFS). 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.

- Participants: Grégory Landais and Nicolas Sendrier
- Contact: Nicolas Sendrier
- URL: <https://gforge.inria.fr/projects/cfs-signature/>

## 6.2. Collision Decoding

KEYWORDS: Algorithm - Binary linear code

FUNCTIONAL DESCRIPTION: Collision Decoding implements two variants of information set decoding : Stern-Dumer, and MMT. 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.

- Participants: Grégory Landais and Nicolas Sendrier
- Contact: Nicolas Sendrier
- URL: <https://gforge.inria.fr/projects/collision-dec/>

## 6.3. ISDF

FUNCTIONAL DESCRIPTION: Implementation of the Stern-Dumer decoding algorithm, and of a variant of the algorithm due to May, Meurer and Thomae.

- Participants: Grégory Landais and Nicolas Sendrier
- Contact: Anne Canteaut
- URL: <https://gforge.inria.fr/projects/collision-dec/>

**CAGE Project-Team (section vide)**



## **MATERIALS Project-Team**

# **5. New Software and Platforms**

## **5.1. simol**

KEYWORDS: Molecular simulation - Quantum chemistry - Statistical physics - C++ - OpenMP

FUNCTIONAL DESCRIPTION: Molecular simulation software written in C++

- Contact: Gabriel Stoltz

## MATHRISK Project-Team

# 6. New Software and Platforms

## 6.1. PREMIA

KEYWORDS: Financial products - Computational finance - Option pricing

SCIENTIFIC DESCRIPTION: 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 and 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.

FUNCTIONAL DESCRIPTION: Premia is a software designed for option pricing, hedging and financial model calibration.

- Participants: Agnes Sulem, Antonino Zanette, Aurélien Alfonsi, Benjamin Jourdain, Jérôme Lelong and Bernard Lapeyre
- Partners: Inria - Ecole des Ponts ParisTech - Université Paris-Est
- Contact: Agnes Sulem
- URL: <http://www.premia.fr>

## 6.2. Platforms

### 6.2.1. Development of the quantitative platform Premia in 2018

Premia 20 has been delivered to the Consortium Premia on March 12th. It contains the following new algorithms :

#### 6.2.1.1. Optimal Trade Execution, Risk Management, Insurance

- Optimal Execution Under Jump Models For Uncertain Price Impact. S.Moazeni, T.F.Coleman, Y.Li *The Journal of Computational Finance. Vol. 18, Issue 3, 2015.*
- Nested Monte Carlo for Risk Margin computation. L.A. Abbas-Turki, S.Crepey, B.Diallo.
- Efficient Estimation of Sensitivities for Counterparty Credit Risk with the Finite Difference Monte-Carlo Method. C. S.L. de Graaf, D.Kandhai, P.M.A.Sloot. *The Journal of Computational Finance, Volume 21, Issue 1, 2017.*
- Nested Simulation in Portfolio Risk Measurement. M.B.Gordy, S.Juneja *Management Science, Vol 56, Issue 10, 2010*
- Spectral methods for the calculation of risk measures for variable annuity guaranteed benefits. R. Feng, H.W. Volkmer *ASTIN Bull., 44(3), 2014*
- Fast computation of risk measures for variable annuities with additional earnings by conditional moment matching. N. Privault X.Wei *ASTIN Bull., 48(1):171–196, 2018.*

## 6.2.1.2. Equity Derivatives

- Pricing under Rough volatility. C. Bayer, P.Friz, J. Gatheral  
*Quantitative Finance*, Vol. 16, No. 6, 887-904, 2016.
- Hybrid scheme for Brownian semistationary processes. M. Bennedsen, A. Lunde, M.S.Pakkanen  
*Finance and Stochastics* 21(4), 931–965, 2017.
- Antithetic multilevel Monte Carlo estimation for multi-dimensional SDEs without Lévy area simulation. M. B. Giles and L. Szpruch  
*The Annals of Applied Probability*, Vol. 24, No. 4, 2014
- Fourier transform algorithms for pricing and hedging discretely sampled exotic variance products and volatility derivatives under additive processes. W. Zheng and Y. K. Kwok  
*The Journal of Computational Finance*, Volume 18, Issue 2, 2014.
- Efficient Solution of Backward Jump-Diffusion PIDEs with Splitting and Matrix Exponentials. A.Iktin  
*The Journal of Computational Finance*, Volume 19, Issue 3, 2016
- High-Order Splitting Methods for Forward PDEs and PIDEs. A.Iktin  
*International Journal of Theoretical and Applied Finance*, 18(5), 2015
- Pricing Bullet option on local volatility model using GPU L.A. Abbas-Turki
- Pricing Bermudan Options via Multilevel Approximation Methods. D. Belomestny, F. Dickmann, T.Nagapetyan.  
*Siam J. Financial Math.*, Volume 6, 2015.
- Pricing CIR yield options by conditional moment matching. A. Prayoga N. Privault  
*Asia-Pacific Financial Markets*, 24:19–38, 2017

We benefit from the help of the engineer Pierre-Guillaume Raverdy.

## MOKAPLAN Project-Team

# 5. New Software and Platforms

## 5.1. ALG2

FUNCTIONAL DESCRIPTION: ALG2 for Monge Mean-Field Games, Monge problem and Variational problems under divergence constraint. A generalisation of the ALG2 algorithm has been implemented in FreeFem++.

- Contact: Jean-David Benamou
- URL: <https://team.inria.fr/mokaplan/augmented-lagrangian-simulations/>

## 5.2. Mokabajour

FUNCTIONAL DESCRIPTION: We design a software resolving the following inverse problem: define the shape of a mirror which reflects the light from a source to a defined target, distribution and support of densities being prescribed. Classical applications include the conception of solar oven, public lightning, car headlights. . . Mathematical modeling of this problem, related to the optimal transport theory, takes the form of a nonlinear Monge-Ampere type PDE. The numerical resolution of these models remained until recently a largely open problem. MOKABAJOUR project aims to develop, using algorithms invented especially at Inria and LJK, a reflector design software more efficient than geometrical methods used so far. The final step is to realize and physically test prototype reflectors.

- Participants: Boris Thibert, Jean-David Benamou and Quentin Mérigot
- Contact: Jean-David Benamou
- URL: <https://project.inria.fr/mokabajour/>

**QUANTIC Project-Team (section vide)**

## SIERRA Project-Team

# 6. New Software and Platforms

## 6.1. ProxASAGA

KEYWORD: Optimization

FUNCTIONAL DESCRIPTION: A C++/Python code implementing the methods in the paper "Breaking the Nonsmooth Barrier: A Scalable Parallel Method for Composite Optimization", F. Pedregosa, R. Leblond and S. Lacoste-Julien, Advances in Neural Information Processing Systems (NIPS) 2017. Due to their simplicity and excellent performance, parallel asynchronous variants of stochastic gradient descent have become popular methods to solve a wide range of large-scale optimization problems on multi-core architectures. Yet, despite their practical success, support for nonsmooth objectives is still lacking, making them unsuitable for many problems of interest in machine learning, such as the Lasso, group Lasso or empirical risk minimization with convex constraints. In this work, we propose and analyze ProxASAGA, a fully asynchronous sparse method inspired by SAGA, a variance reduced incremental gradient algorithm. The proposed method is easy to implement and significantly outperforms the state of the art on several nonsmooth, large-scale problems. We prove that our method achieves a theoretical linear speedup with respect to the sequential version under assumptions on the sparsity of gradients and block-separability of the proximal term. Empirical benchmarks on a multi-core architecture illustrate practical speedups of up to 12x on a 20-core machine.

- Contact: Fabian Pedregosa
- URL: <https://github.com/fabianp/ProxASAGA>

## 6.2. object-states-action

KEYWORD: Computer vision

FUNCTIONAL DESCRIPTION: Code for the paper Joint Discovery of Object States and Manipulation Actions, ICCV 2017: Many human activities involve object manipulations aiming to modify the object state. Examples of common state changes include full/empty bottle, open/closed door, and attached/detached car wheel. In this work, we seek to automatically discover the states of objects and the associated manipulation actions. Given a set of videos for a particular task, we propose a joint model that learns to identify object states and to localize state-modifying actions. Our model is formulated as a discriminative clustering cost with constraints. We assume a consistent temporal order for the changes in object states and manipulation actions, and introduce new optimization techniques to learn model parameters without additional supervision. We demonstrate successful discovery of seven manipulation actions and corresponding object states on a new dataset of videos depicting real-life object manipulations. We show that our joint formulation results in an improvement of object state discovery by action recognition and vice versa.

- Participants: Jean-Baptiste Alayrac, Josef Sivic, Ivan Laptev and Simon Lacoste-Julien
- Contact: Jean-Baptiste Alayrac
- Publication: [Joint Discovery of Object States and Manipulation Actions](#)
- URL: <https://github.com/jalayrac/object-states-action>

## ANGE Project-Team

# 6. New Software and Platforms

## 6.1. Freshkiss

*FREe Surface Hydrodynamics using KInetic SchemeS*

KEYWORDS: Finite volume methods - Hydrostatic Navier-Stokes equations - Free surface flows

FUNCTIONAL DESCRIPTION: Freshkiss3D is a numerical code solving the 3D hydrostatic and incompressible Navier-Stokes equations with variable density.

- Participants: Fabien Souille, Emmanuel Audusse, Jacques Sainte Marie and Marie-Odile Bristeau
- Partners: UPMC - CEREMA
- Contact: Jacques Sainte Marie

## 6.2. TSUNAMATHS

KEYWORDS: Modeling - Tsunamis

FUNCTIONAL DESCRIPTION: 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.

- Participants: Emmanuel Audusse, Jacques Sainte Marie and Raouf Hamouda
- Contact: Jacques Sainte Marie
- URL: <http://tsunamath.paris.inria.fr/>

## 6.3. Verdandi

KEYWORDS: HPC - Model - Software Components - Partial differential equation

FUNCTIONAL DESCRIPTION: Verdandi is a free and open-source (LGPL) library for data assimilation. It includes various such methods for coupling one or several numerical models and observational data. Mainly targeted at large systems arising from the discretization of partial differential equations, the library is devised as generic, which allows for applications in a wide range of problems (biology and medicine, environment, image processing, etc.). Verdandi also includes tools to ease the application of data assimilation, in particular in the management of observations or for a priori uncertainty quantification. Implemented in C++, the library may be used with models implemented in Fortran, C, C++ or Python.

- Participants: Dominique Chapelle, Gautier Bureau, Nicolas Claude, Philippe Moireau and Vivien Mallet
- Contact: Vivien Mallet
- URL: <http://verdandi.gforge.inria.fr/>

## 6.4. Polyphemus

KEYWORD: Simulation

FUNCTIONAL DESCRIPTION: 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.

- Participants: Sylvain Doré and Vivien Mallet
- Contact: Vivien Mallet
- URL: <http://cera.enpc.fr/polyphemus/>

## **6.5. Urban noise analysis**

KEYWORD: Environment perception

FUNCTIONAL DESCRIPTION: This software processes mobile observations collected by the application Ambiciti (previously known as SoundCity). It can merge simulated noise maps with the mobile observations.

- Authors: Raphaël Ventura, Vivien Mallet and Guillaume Cherel
- Contact: Vivien Mallet

## **6.6. Freshkiss3D**

- Authors: Jacques Sainte Marie, Marie-Odile Bristeau, Anne-Céline Boulanger, Raouf Hamouda, Emmanuel Audusse, Alain Dervieux, Bijan Mohammadi and David Froger
- Partner: UPMC
- Contact: Jacques Sainte Marie



## ARAMIS Project-Team

# 6. New Software and Platforms

## 6.1. Brain Networks Toolbox

KEYWORDS: Neuroimaging - Medical imaging

FUNCTIONAL DESCRIPTION: Brain Networks Toolbox is an open-source package of documented routines implementing new graph algorithms for brain network analysis. It mainly contains Matlab code of new methods developed by the team and associated to publications (e.g., brain network thresholding, extraction of the information redundancy, node accessibility, etc). It requires, as input, adjacency matrices representing brain connectivity networks. Thus, it is independent on the specific approach used to construct brain networks and it can be used to extract network properties from any neuroimaging modality in healthy and diseased subjects.

- Participants: Fabrizio De Vico Fallani, Jeremy Guillon and Mario Chavez
- Contact: Fabrizio De Vico Fallani
- URL: <https://github.com/brain-network/bnt>

## 6.2. Deformetrica

KEYWORDS: 3D modeling - C++ - Automatic Learning - Mesh - Anatomy - Image analysis

SCIENTIFIC DESCRIPTION: Deformetrica is a software for the statistical analysis of 2D and 3D shape data. It essentially computes deformations of the 2D or 3D ambient space, which, in turn, warp any object embedded in this space, whether this object is a curve, a surface, a structured or unstructured set of points, or any combination of them.

Deformetrica comes with two applications:

registration, which computes the best possible deformation between two sets of objects, atlas construction, which computes an average object configuration from a collection of object sets, and the deformations from this average to each sample in the collection.

Deformetrica has very little requirements about the data it can deal with. In particular, it does not require point correspondence between objects!

FUNCTIONAL DESCRIPTION: Deformetrica is a software for the statistical analysis of 2D and 3D shape data. It essentially computes deformations of the 2D or 3D ambient space, which, in turn, warp any object embedded in this space, whether this object is a curve, a surface, a structured or unstructured set of points, or any combination of them.

Deformetrica comes with two applications:

- Registration, which computes the optimal deformation between two sets of objects,
- Atlas construction, which computes an average object configuration from a collection of object sets, and the deformations from this average to each sample in the collection.

Deformetrica has very little requirements about the data it can deal with. In particular, it does not require point correspondence between objects!

- Participants: Alexandre Routier, Ana Fouquier, Barbara Gris, Benjamin Charlier, Cédric Doucet, Joan Alexis Glaunès, Marcel Prastawa, Michael Bacci, Pietro Gori and Stanley Durrleman
- Partners: University of Utah - Université de Montpellier 2 - Université Paris-Descartes
- Contact: Stanley Durrleman
- URL: <http://www.deformetrica.org/>

## 6.3. Clinica

**KEYWORDS:** Neuroimaging - Brain MRI - MRI - Clinical analysis - Image analysis - Machine learning

**SCIENTIFIC DESCRIPTION:** Clinica is a software platform for multimodal brain image analysis in clinical research studies. It makes it easy to apply advanced analysis tools to large scale clinical studies. For that purpose, it integrates a comprehensive set of processing tools for the main neuroimaging modalities: currently MRI (anatomical, functional, diffusion) and PET, in the future, EEG/MEG. For each modality, Clinica allows to easily extract various types of features (regional measures, parametric maps, surfaces, curves, networks). Such features are then subsequently used as input of machine learning, statistical modeling, morphometry or network analysis methods. Processing pipelines are based on combinations of freely available tools developed by the community. It provides an integrated data management specification to store raw and processing data. Clinica is written in Python. It uses the Nipype system for pipelining. It combines widely-used software for neuroimaging data analysis (SPM, Freesurfer, FSL, MRtrix...), morphometry (Deformetrica), machine learning (Scikit-learn) and the BIDS standard for data organization.

**FUNCTIONAL DESCRIPTION:** Clinica is a software platform for multimodal brain image analysis in clinical research studies. It makes it easy to apply advanced analysis tools to large scale clinical studies. For that purpose, it integrates a comprehensive set of processing tools for the main neuroimaging modalities: currently MRI (anatomical, functional, diffusion) and PET, in the future, EEG/MEG. For each modality, Clinica allows to easily extract various types of features (regional measures, parametric maps, surfaces, curves, networks). Such features are then subsequently used as input of machine learning, statistical modeling, morphometry or network analysis methods. Clinica also provides an integrated data management specification to store raw and processing data. Overall, Clinica helps to: i) apply advanced analysis tools to clinical research studies, ii) easily share data and results, iii) make research more reproducible.

**NEWS OF THE YEAR:** - Three clinical studies made with Clinica Clinica : Bertrand et al, JAMA Neurology, 2018 , Jacquemont et al, Neurobiol Aging, 2017, Wen et al, JNNP, 2018 - Clinica presented at OHBM 2018 conference - Clinica was the support for the tutorial "Pattern Recognition for Neuroimaging" at OHBM 2018

- Participants: Jeremy Guillon, Thomas Jacquemont, Pascal Lu, Arnaud Marcoux, Tristan Moreau, Alexandre Routier, Jorge Samper Gonzalez, Junhao Wen, Olivier Colliot, Stanley Durrleman, Michael Bacci, Simona Bottani, Ninon Burgos, Sabrina Fontanella, Pietro Gori, Mauricio Diaz Melo and Elina Thibeau-Sutre
- Partners: Institut du Cerveau et de la Moelle épinière (ICM) - CNRS - INSERM - UPMC
- Contact: Olivier Colliot
- Publications: [Amyloidosis and neurodegeneration result in distinct structural connectivity patterns in mild cognitive impairment](#) - [Yet Another ADNI Machine Learning Paper? Paving The Way Towards Fully-reproducible Research on Classification of Alzheimer's Disease](#) - [Reproducible evaluation of classification methods in Alzheimer's disease: Framework and application to MRI and PET data](#) - [Neurite density is reduced in the presymptomatic phase of C9orf72 disease](#) - [Early cognitive, structural and microstructural changes in c9orf72 presymptomatic carriers before 40 years of age](#)
- URL: <http://www.clinica.run>

## 6.4. Platforms

### 6.4.1. Platform Brain-computer interface

Our team coordinates the developments of the Brain-Computer Interface (BCI) platform at the Centre EEG/MEG of the neuroimaging core facility of the ICM. Several projects, including our NETBCI NSF/NIH/ANR and ATTACK Big-brain theory funded projects, as well as experiments by different researchers of the Institute, are currently being run. To reinforce the impact of the platform we have recently recruited an engineer (J. Gonzalez-Astudillo) for the software and technical development.

## MAMBA Project-Team

# 6. New Software and Platforms

## 6.1. TiQuant

### *Tissue Quantifier*

KEYWORDS: Systems Biology - Bioinformatics - Biology - Physiology

FUNCTIONAL DESCRIPTION: 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. We generated a software for image analysis of histological material and demonstrated its use in analysing liver confocal micrografts, called TiQuant (Tissue Quantifier). 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.

- Author: Dirk Drasdo
- Contact: Dirk Drasdo

## 6.2. TiSim

### *Tissue Simulator*

KEYWORDS: Systems Biology - Bioinformatics - Biology - Physiology

SCIENTIFIC DESCRIPTION: TiSim (Tissue Simulator) is a versatile and efficient simulation environment for tissue models. TiSim is a software for agent-based models of multicellular systems. It permits model development with center-based models and deformable cell models, it contains modules for monolayer and multicellular spheroid simulations as well as for simulations of liver lobules. Besides agent-based simulations, the flow of blood and the transport of molecules can be modelled in the extracellular space, intracellular processes such as signal transduction and metabolism can be simulated, for example over an interface permitting integration of SBML-formulated ODE models. TiSim is written in modern C++ , keeping central model constituents in modules to be able to reuse them as building blocks for new models. For user interaction, the GUI Framework Qt is used in combination with OpenGL for visualisation. The simulation code is in the process of being published. The modeling strategy and approaches slowly reach systems medicine and toxicology. The diffusion of software is a fundamental component as it provides the models that are complex and difficult to implement (implementing a liver lobule model from scratch takes about 2-2.5yrs) in form of a software to the developer and users who like to build upon them. This increases significantly the speed of implementing new models. Moreover, standardization is indispensable as it permits coupling different software tools that may have implemented models at different scales / levels.

FUNCTIONAL DESCRIPTION: TiSim is a software that permits agent-based simulations of multicellular systems. - center-based lattice-free agent-based model - modular - C++, Qt, OpenGL, GUI, batch mode - permits multiscale simulations by integration of molecular pathways (for signaling, metabolisms, drug) into each individual cell - applications so far: monolayer growth, multicellular spheroids - Boolean networks (development time = coding time ( 60 MMs) + model development time ( 264 MMs)) - in follow-up version 1: - liver lobule regeneration - SBML interface - in follow-up version 2: - deformable cell model (by triangulation of cell surface) - deformable rod models - extracellular matrix - vascular flow and transport TiSim can be directly fed by processed image data from TiQuant.

- Participants: Andreas Buttenschoen, Dirk Drasdo, Eugenio Lella, Géraldine Cellière, Johannes Neitsch, Margaretha Palm, Nick Jagiella, Noémie Boissier, Paul Van Liedekerke, Stefan Hoehme and Tim Johann
- Partner: IZBI, Université de Leipzig
- Contact: Dirk Drasdo

## **6.3. Platforms**

### **6.3.1. *TiSim***

New side branches have been developed that integrate the deformable cell model and extracellular matrix model in the *TiSim* software.

### **6.3.2. *TiQuant***

The software has been further extended for machine learning components.

## REO Project-Team

# 6. New Software and Platforms

## 6.1. FELiScE

*Finite Elements for Life Sciences and Engineering problems*

KEYWORDS: Finite element modelling - Cardiac Electrophysiology - Cardiovascular and respiratory systems

FUNCTIONAL DESCRIPTION: FELiScE is a finite element code which the M3DISIM 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 open-source library. FELiScE 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.

- Participants: Axel Fourmont, Benoit Fabreges, Damiano Lombardi, Dominique Chapelle, Irène Vignon-Clementel, Jean-Frédéric Gerbeau, Marina Vidrascu, Matteo Aletti, Miguel Angel Fernandez Varela, Mikel Landajueta Larma, Philippe Moireau and Sébastien Gilles
- Contact: Miguel Angel Fernandez Varela
- URL: <http://felisce.gforge.inria.fr>

## 6.2. SHELDDON

*SHELLs and structural Dynamics with DOmain decomposition in Nonlinear analysis*

FUNCTIONAL DESCRIPTION: SHELDDON 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.

- Participants: Dominique Chapelle, Marina Vidrascu and Patrick Le Tallec
- Contact: Marina Vidrascu
- URL: <https://gforge.inria.fr/projects/shelldon/>

## 6.3. DCIMaL

KEYWORD: Cardiac Electrophysiology

FUNCTIONAL DESCRIPTION: DCIMaL is a Python and C++ software for safety pharmacology studies and particularly field potentials signals measured with micro-electrode array (MEA). The software includes a solver for field potential simulations and a dictionary of entries corresponding to features which can be extracted from real or simulated potential signals. It also includes an algorithm for drug classification (channel blockade or torsadogenic risk) and a tool for estimating ion channel activity (based on the CMAES library). DCIMaL was registered in 2018 at the Agence pour la Protection des Programmes Inter Deposit Digital Number IDDN.FR.001.270003.000.S.P.2018.000.31230

- Participants: Fabien Raphel, Jean-Frédéric Gerbeau and Damiano Lombardi
- Contact: Damiano Lombardi

## 6.4. FELiScE-NS

KEYWORDS: Incompressible flows - Thin-walled solids

FUNCTIONAL DESCRIPTION: FELiScE-NS is a set finite elements solvers for incompressible fluids (fractional-step schemes) and non-linear thin-walled structures (3D shells, and 2D curved beams) developed in the framework of the FELiScE library. FELiScE-NS was registered in 2018 at the Agence pour la Protection des Programmes Inter Deposit Digital Number IDDN.FR.001.270015.000.S.A.2018.000.31200.

- Participants: Benoit Fabreges, Miguel Angel Fernandez Varela, Axel Fourmont, Jean-Frédéric Gerbeau and Marina Vidrascu
- Contact: Miguel Angel Fernandez Varela

## SERENA Project-Team

# 6. New Software and Platforms

## 6.1. CELIA3D

KEYWORDS: Fluid mechanics - Multi-physics simulation

FUNCTIONAL DESCRIPTION: The CELIA3D code simulates the coupling between a compressible fluid flow and a deformable structure. The fluid is handled by a Finite Volume method on a structured Cartesian grid. The solid is handled by a Discrete Element method (Mka3d scheme). The solid overlaps the fluid grid and the coupling is carried out with immersed boundaries (cut cells) in a conservative way.

- Partners: Ecole des Ponts ParisTech - CEA
- Contact: Laurent Monasse
- URL: <http://cermics.enpc.fr/~monassel/CELIA3D/>

## 6.2. DiSk++

KEYWORDS: High order methods - Polyhedral meshes - C++

SCIENTIFIC DESCRIPTION: Discontinuous Skeletal methods approximate the solution of boundary-value problems by attaching discrete unknowns to mesh faces (hence the term skeletal) while allowing these discrete unknowns to be chosen independently on each mesh face (hence the term discontinuous). Cell-based unknowns, which can be eliminated locally by a Schur complement technique (also known as static condensation), are also used in the formulation. Salient examples of high-order Discontinuous Skeletal methods are Hybridizable Discontinuous Galerkin methods and the recently-devised Hybrid High-Order methods. Some major benefits of Discontinuous Skeletal methods are that their construction is dimension-independent and that they offer the possibility to use general meshes with polytopal cells and non-matching interfaces. The mathematical flexibility of Discontinuous Skeletal methods can be efficiently replicated in a numerical software: by using generic programming, the DiSk++ library offers an environment to allow a programmer to code mathematical problems in a way completely decoupled from the mesh dimension and the cell shape.

FUNCTIONAL DESCRIPTION: The software provides a numerical core to discretize partial differential equations arising from the engineering sciences (mechanical, thermal, diffusion). The discretization is based on the "Hybrid high-order" or "Discontinuous Skeletal" methods, which use as principal unknowns polynomials of arbitrary degree on each face of the mesh. An important feature of these methods is that they make it possible to treat general meshes composed of polyhedral cells. The DiSk ++ library, using generic programming techniques, makes it possible to write a code for a mathematical problem independently of the mesh. When a user writes the code for his problem using the basic operations offered by DiSk ++, that code can be executed without modifications on all types of mesh already supported by the library and those that will be added in the future.

- Author: Matteo Cicuttin
- Partner: CERMICS
- Contact: Matteo Cicuttin
- Publication: [Implementation of Discontinuous Skeletal methods on arbitrary-dimensional, polytopal meshes using generic programming](#)
- URL: <https://github.com/wareHHouse/diskpp>

## 6.3. GENFIELD

KEYWORDS: Hydrogeology - Algorithm - Heterogeneity

SCIENTIFIC DESCRIPTION: GENFIELD implements a parallel version of the algorithm initially proposed by [E. Pardo-Iguzquiza and M. Chica-Olmo, *Mathematical Geology*, 25(2):177-217, 1993].

FUNCTIONAL DESCRIPTION: GENFIELD allows the generation of gaussian correlated fields. It is based on the circulant embedding method. Parallelism is implemented using MPI communications. GENFIELD is used in hydrogeology to model natural fields, like hydraulic conductivity or porosity fields.

NEWS OF THE YEAR: In 2018, we have performed scaling tests on ADA cluster. They have revealed that the symmetry of the phases required by this algorithm penalized a lot the parallel efficiency of GENFIELD (cf poster hal-01960444, version 1). We have decided to stop the development of GENFIELD. In 2018, we have developed a complete new software (cf BIL Inria ParaCIRCE) based on another algorithm initially proposed by [C. R. Dietrich and G. N. Newsam. A fast and exact method for multidimensional gaussian stochastic simulations. *Water Resources Research*, 29(8):2861-2869, 1993].

- Participants: Géraldine Pichot, Simon Legrand, Grégoire Lecourt, Jean-Raynald De Dreuzy and Jocelyne Erhel
- Contact: Géraldine Pichot
- Publications: [GENFIELD: A parallel software for the generation of stationary Gaussian random fields - Algorithms for stationary Gaussian random field generation](#)
- URL: [https://gitlab.inria.fr/slegrand/Genfield\\_dev](https://gitlab.inria.fr/slegrand/Genfield_dev)

## 6.4. Mka3d

KEYWORDS: Scientific computing - Elasticity - Elastodynamic equations

FUNCTIONAL DESCRIPTION: The Mka3d method simulates an elastic solid by discretizing the solid into rigid particles. An adequate choice of forces and torques between particles allows to recover the equations of elastodynamics.

- Partners: Ecole des Ponts ParisTech - CEA
- Contact: Laurent Monasse
- URL: <http://cermics.enpc.fr/~monassel/Mka3D/>

## 6.5. NEF-Draw

*Numerical Experiments involving Fractures -Visualisation*

KEYWORD: Fracture network

FUNCTIONAL DESCRIPTION: This software is a visualization tool of discrete fractured networks. It allows the visualization of the network geometry, the mesh of the network together with several quantities of interest (mesh quality, flow solution including wells) computed with the software NEF-Flow.

NEWS OF THE YEAR: This version includes Matlab vectorization of the operations which makes it possible to load flow solution on meshes with more than one million of fractures. It includes a text menu allowing the user to choose between different visualisation options (geometry, mesh together with the aspect ratio or together with the flow solution) A selective visualisation of fractures is also possible, loading only the fractures that carry most of the flow.

- Participant: Géraldine Pichot
- Contact: Géraldine Pichot
- URL: <https://gitlab.inria.fr/gpichot/NEF>

## 6.6. NEF-Flow

KEYWORDS: Hydrogeology - Numerical simulations - 3D

SCIENTIFIC DESCRIPTION: NEF-Flow is a Matlab software for the simulation of steady state single phase flow in Discrete Fracture Networks (DFNs) using the Mixed Hybrid Finite Element (MHFEM) method for conforming and non conforming discretizations.



**FUNCTIONAL DESCRIPTION:** The software NEF-Flow solves the problem of an incompressible fluid flowing through a network of fractures. The software is interfaced with different mesh generators, among which BLSURF from the GAMMA3 team. A mixed hybrid finite element method is implemented.

**NEWS OF THE YEAR:** The last version includes new feature: - wells, sink/source terms boundary conditions - Implementation of P1 non conforming finite elements - New data structures to save the information local to each fracture - New tests per fracture have been added to check the solution - Add wells and sink/sources boundary conditions in the function that check the solution.

- Participants: Géraldine Pichot, Jean-Raynald De Dreuzy and Jocelyne Erhel
- Contact: Géraldine Pichot
- Publication: [A mixed hybrid Mortar method for solving flow in discrete fracture networks](#)
- URL: <https://gitlab.inria.fr/gpichot/NEF>

## 6.7. ParaCirce

*Parallel Circulant Embedding*

**KEYWORDS:** 2D - 3D - Hydrogeology - Gaussian random fields - MPI

**SCIENTIFIC DESCRIPTION:** ParaCirce implements the algorithm proposed by [C. R. Dietrich and G. N. Newsam. A fast and exact method for multidimensional gaussian stochastic simulations. Water Resources Research, 29(8):2861-2869, 1993].

**FUNCTIONAL DESCRIPTION:** ParaCirce implements a parallel Circulant Embedding method for the generation in parallel of 2D or 3D Gaussian Random Fields (second order stationary).

**NEWS OF THE YEAR:** - MPI implementation - Dedicated C++ classes to allow a user-friendly and safe usage of the library - Efficient use of the external library RngStream (L'Ecuyer) for a guarantee of independent realizations and reproducibility. - Splitting of the domain along one direction. The repartition of the field is automatic or defined by the user. - Automatic computation of the padding

- Participants: Géraldine Pichot and Simon Legrand
- Contact: Géraldine Pichot
- URL: <https://gitlab.inria.fr/slegrand/paracirce>

## 6.8. PRune

*ParserRUNnEr*

**KEYWORD:** Test

**FUNCTIONAL DESCRIPTION:** Python tool to parse single or multi configurations parameters files and to automatically run a program and store the results in a predefined tree.

- Participants: Simon Legrand and Géraldine Pichot
- Contact: Simon Legrand

## ALPINES Project-Team

# 6. New Software and Platforms

## 6.1. FreeFem++

*FreeFem++*

SCIENTIFIC DESCRIPTION: FreeFem++ is a partial differential equation solver. It has its own language. freefem scripts can solve multiphysics non linear systems in 2D and 3D.

Problems involving PDE (2d, 3d) from several branches of physics such as fluid-structure interactions require interpolations of data on several meshes and their manipulation within one program. FreeFem++ includes a fast 2d-tree-based interpolation algorithm and a language for the manipulation of data on multiple meshes (as a follow up of bamg (now a part of FreeFem++)).

FreeFem++ is written in C++ and the FreeFem++ language is a C++ idiom. It runs on Macs, Windows, Unix machines. FreeFem++ replaces the older freefem and freefem+.

FUNCTIONAL DESCRIPTION: FreeFem++ is a PDE (partial differential equation) 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.

- Partner: UPMC
- Contact: Frederic Hecht
- URL: <http://www.freefem.org/ff++/>

## 6.2. HPDDM

SCIENTIFIC DESCRIPTION: 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. 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.

FUNCTIONAL DESCRIPTION: 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.

- Participants: Frédéric Nataf and Pierre Jolivet
- Contact: Pierre Jolivet
- URL: <https://github.com/hpddm>

## 6.3. LORASC

*LORASC preconditioner*

KEYWORD: Preconditioner

- Participants: Laura Grigori and Rémi Lacroix
- Contact: Laura Grigori

## 6.4. Platforms

### 6.4.1. HTOOL

KEYWORD: Hierarchical Matrices

**FUNCTIONAL DESCRIPTION:** HTOOL is a C++ header-only library implementing compression techniques (e.g. Adaptive Cross Approximation) using hierarchical matrices. The library uses MPI and OpenMP for parallelism, and is interfaced with HPDDM for the solution of linear systems.

- Partners: CNRS - UPMC - ANR NonlocalDD
- Contact: Pierre Marchand
- URL: <https://github.com/PierreMarchand20/htool>

#### 6.4.2. *BemTool*

**KEYWORD:** Boundary Element Method

**FUNCTIONAL DESCRIPTION:** BemTool is a C++ header-only library implementing the boundary element method for the discretisation of the Laplace, Helmholtz and Maxwell equations, in 2D and 3D. Its main purpose is the assembly of classic boundary element matrices, which can be compressed and inverted through its interface with HTOOL.

- Partners: UPMC - ANR NonlocalDD
- Contact: Xavier Claeys
- URL: <https://github.com/xclaeys/BemTool>

#### 6.4.3. *Geneo4PETSc*

**KEYWORD:** Domain decomposition method

**FUNCTIONAL DESCRIPTION:** Implementation of the GenEO preconditioner with PETSc and SLEPc.

- Partners: CNRS - UPMC - European project NLAFFET
- Contact: Frédéric Nataf
- URL: <https://github.com/geneo4PETSc/geneo4PETSc>

#### 6.4.4. *ffddm*

**KEYWORD:** Domain decomposition method

**FUNCTIONAL DESCRIPTION:** In the acronym ffddm, ff stands for FreeFem++ and ddm for domain decomposition methods. The idea behind ffddm is to simplify the use of parallel solvers in FreeFem++: distributed direct methods and domain decomposition methods.

- Partners: CNRS - UPMC
- Contact: Pierre-Henri Tournier and Frédéric Nataf
- URL: <https://doc.freefem.org/documentation/ffddm/ffddm>

#### 6.4.5. *preAlps*

**KEYWORD:** Preconditioned enlarged Krylov subspace method

**FUNCTIONAL DESCRIPTION:** Contains enlarged Conjugate Gradient Krylov subspace method and Loras preconditioner.

- Partners: Inria
- Contact: Simplic Donfack, Laura Grigori, Olivier Tissot
- URL: <https://github.com/NLAFFET/preAlps>

#### 6.4.6. *FreeFem++ v4*

**KEYWORD:** New version of Freefem++, with new sparse matrix kernel, and with surface finite element.

**FUNCTIONAL DESCRIPTION:**

- Partners: UPMC - Inria
- Contact: Frederic Hecht
- URL: <https://github.com/FreeFem/FreeFem-sources/tree/v4>

**DELYS Team (section vide)**

**DYOGENE Project-Team (section vide)**

## EVA Project-Team

# 6. New Software and Platforms

## 6.1. OpenWSN

KEYWORDS: Internet of things - 6TiSCH - 6LoWPAN - CoAP

FUNCTIONAL DESCRIPTION: OpenWSN is an open-source implementation of a fully standards-based protocol stack for the Internet of Things. It has become the de-facto implementation of the IEEE802.15.4e TSCH standard, has a vibrant community of academic and industrial users, and is the reference implementation of the work we do in the IETF 6TiSCH standardization working group.

- Partner: University of California Berkeley
- Contact: Thomas Watteyne
- URL: <http://www.openwsn.org/>

## 6.2. 6TiSCH Simulator

*High-level simulator of a 6TiSCH network*

KEYWORDS: Network simulator - 6TiSCH

FUNCTIONAL DESCRIPTION: The simulator is written in Python. While it doesn't provide a cycle-accurate emulation, it does implement the functional behavior of a node running the full 6TiSCH protocol stack. This includes RPL, 6LoWPAN, CoAP and 6P. The implementation work tracks the progress of the standardization process at the IETF.

- Contact: Malisa Vucinic

## 6.3. Argus

KEYWORDS: Cloud - Low-Power Wireless - Sniffer

FUNCTIONAL DESCRIPTION: There are three pieces to the Argus:

The Argus Probe is the program which attaches to your low-power wireless sniffer and forwards its traffic to the Argus Broker.

The Argus Broker sits somewhere in the cloud. Based on MQTT, it connects Argus Probes with Argus Clients based on a pub-sub architecture.

Several Argus Clients can be started at the same time. It is a program which subscribes to the Argus Broker and displays the frames in Wireshark.

- Contact: Rémy Leone

## 6.4. SolSystem

*Sensor Object Library System*

KEYWORDS: Low-Power Wireless - Back-End System - SmartMesh IP

FUNCTIONAL DESCRIPTION: The source code is composed of the definition of the SOL structure (<https://github.com/realms-team/sol>), the code that runs on the manager (<https://github.com/realms-team/solmanager>, written in Python) and the code that runs on the server receiving the data (<https://github.com/realms-team/solserver>, written in Python)

- Contact: Keoma Brun-Laguna
- URL: <http://www.solssystem.io/>

## 6.5. 6TiSCH Wireshark Dissector

KEYWORDS: 6TiSCH - Wireshark

FUNCTIONAL DESCRIPTION: Implementation on the dissectors is done through an open-source repository, stable code is regularly contributed back to the main Wireshark code base.

- Contact: Jonathan Munoz

## 6.6. F-Interop

*Remote Conformance and Interoperability Tests for the Internet of Thing*

KEYWORDS: Interoperability - Iot - Conformance testing - Standardization

- Partners: UPMC - IMEC - ETSI - EANTC - Mandat International - Digital Catapult - University of Luxembourg - Device Gateway
- Contact: Rémy Leone

## 6.7. Mercator

KEYWORDS: Deployment - Low-Power Wireless - Testbeds - Connectivity

FUNCTIONAL DESCRIPTION: The firmware is written as part of the OpenWSN project. Scripts and analysis tools are written in Python.

- Contact: Keoma Brun-Laguna

---

## GANG Project-Team

# 6. New Software and Platforms

## 6.1. big-graph-tools

KEYWORD: Graph algorithmics

FUNCTIONAL DESCRIPTION: Gang is developing a software for big graph manipulation. A preliminary library offering diameter and skeleton computation. This library was used to compute the diameters of the worldwide road network (200M edges) and the largest strongly connected component of the Twitter follower-follower graph (23G edges).

- Contact: Laurent Viennot
- URL: <https://who.rocq.inria.fr/Laurent.Viennot/dev/big-graph-tools/>

## 6.2. GRPH

*The high performance graph library for Java*

KEYWORDS: Graph - Graph algorithmics - Java

FUNCTIONAL DESCRIPTION: Grph is an open-source Java library for the manipulation of graphs. Its design objectives are to make it portable, simple to use/extend, computationally/memory efficient, and, according to its initial motivation: useful in the context of graph experimentation and network simulation. Grph also has the particularity to come with tools like an evolutionary computation engine, a bridge to linear programming solvers, a framework for distributed computing, etc.

Grph offers a very general model of graphs. Unlike other graph libraries which impose the user to first decide if he wants to deal with directed, undirected, hyper (or not) graphs, the model offered by Grph is unified in a general class that supports mixed graphs made of undirected and directed simple and hyper edges. Grph achieves great efficiency through the use of multiple code optimization techniques such as multi-core parallelism, caching, adequate data structures, use of primitive objects, exploitation of low-level processor caches, on-the-fly compilation of specific C/C++ code, etc. Grph attempts to access the Internet in order to check if a new version is available and to report who is using it (login name and hostname). This has no impact whatsoever on performance and security.

- Participants: Aurélien Lancin, David Coudert, Issam Tahiri, Luc Hogue and Nathann Cohen
- Contact: Luc Hogue
- URL: <http://www.i3s.unice.fr/~hogie/grph/>



## MIMOVE Project-Team

# 6. New Software and Platforms

## 6.1. SocialBus

*Universal Social Network Bus*

KEYWORDS: Middleware - Interoperability - Social networks - Software Oriented Service (SOA)

FUNCTIONAL DESCRIPTION: Online social network services (OSNSs) have become an integral part of our daily lives. At the same time, the aggressive market competition has led to the emergence of multiple competing siloed OSNSs that cannot interoperate. As a consequence, people face the burden of creating and managing multiple OSNS accounts and learning how to use them, to stay connected. The goal of the Universal Social Network Bus (USNB) is to relieve users from such a burden, letting them use their favorite applications to communicate.

- Authors: Rafael Angarita Arocha, Nikolaos Georgantas and Valérie Issarny
- Contact: Valérie Issarny
- URL: <https://gitlab.inria.fr/usnb/universal-social-network-bus>

## 6.2. WeBrowse

KEYWORDS: Web Usage Mining - Content analysis - Recommendation systems

FUNCTIONAL DESCRIPTION: The amount of information available on the web today, and the fast rate with which new information appears, overwhelm most users. The goal of our research is to assist Web users in discovering content. One of the most powerful means today to help people discover new web content is sharing between members of online communities. In the case of communities of a place (e.g., people who live, study, or work together) people share common interests, but often fail to actively share content. To address this problem, we have developed WeBrowse, a passive crowdsourced content discovery system for communities of a place.

WeBrowse leverages the passive observation of web-clicks (i.e., the URLs users intentionally visit) as an indication of users' interest in a piece of content. Intuitively, the more users click on a URL, the higher the interest in the content on the corresponding page. Our approach is then to leverage the collective clicks in a community to automatically discover relevant content to promote to users of the community.

To implement passive crowdsourcing, one must be in a position to observe the aggregated web-clicks of the community. Luckily, in many communities of a place, users will connect to the Internet from the same network, such as, e.g., the campus/enterprise network or the network of a residential Internet Service Provider (ISP) in a neighborhood. WeBrowse (i) observes web packets flowing through a network link, (ii) passively extracts HTTP logs (i.e., streams recording the headers of HTTP requests), and (iii) detects and decides on-the-fly the set of URLs to show to users.

- Contact: Renata Cruz Teixeira
- URL: <https://team.inria.fr/muse/webbrowse-info-page/>

## 6.3. EEE

*Experiment Execution Engine*

KEYWORD: Iot

**FUNCTIONAL DESCRIPTION:** Experiment Execution Engine (EEE) eases the development of IoT applications that perform analysis of recent or continuously increasing volumes of IoT data from various data stores. To this end, EEE provides APIs for scheduling queries on federated large-scale semantically-enabled IoT data stores. Queries are expressed in the FIESTA-IoT (<http://fiesta-iot.eu>) Experiment Description Specification (FEDSpec), which acts as a Domain Specific Language (DSL). EEE (in combination with Experiment Management Console - EMC) further provides features such as: (a) manage scheduled experiments, (b) (un)subscribe to already existing FEDSpec objects, (c) execute an on-demand query outside a described schedule, (d) monitor execution logs, (e) interact with third-party services such as Analytics and Result Storage, (e) handle dynamic attributes, and (f) API sandbox. EEE is integrated within the FIESTA-IoT Platform. It can be customised depending on needs.

- Authors: Rachit Agarwal and Nikolaos Georgantas
- Contact: Nikolaos Georgantas
- URL: <https://github.com/fiesta-iot/ExperimentExecutionEngine>

## 6.4. EMC

*Experiment Management Console*

KEYWORD: Iot

**FUNCTIONAL DESCRIPTION:** Experiment Management Console (EMC) is a simple easy-to-use user interface that enables experimenters to control their experiments/queries that execute using the Experiment Execution Engine (EEE). EMC provides features such as managing experiment execution, (un)subscribing an existing experiment, and monitoring execution logs.

- Authors: Nikolaos Georgantas and Rachit Agarwal
- Contact: Nikolaos Georgantas
- URL: <https://github.com/fiesta-iot/ExperimentManagementConsole>

## 6.5. VSB

*eVolution Service Bus*

KEYWORDS: Service and Thing choreographies - Middleware protocol interoperability - Enterprise service bus

**FUNCTIONAL DESCRIPTION:** VSB is a development and runtime environment dedicated to complex distributed applications of the Future Internet. Such applications are open, dynamic choreographies of extremely heterogeneous services and Things, including 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). VSB's objective is to seamlessly interconnect, inside choreographies, services and Things that employ heterogeneous interaction protocols at the middleware level, e.g., SOAP Web services, REST Web services, Things using CoAP. This is based on runtime conversions between such protocols, with respect to their primitives and data type systems, while properly mapping between their semantics. This also includes mapping between the public interfaces of services/Things, regarding their operations and data, from the viewpoint of the middleware: the latter means that operations and data are converted based on their middleware-level semantics, while their business semantics remains transparent to the conversion. VSB follows the well-known Enterprise Service Bus (ESB) paradigm. We propose a generic interface description, which we call GIDL, for application components that employ VSB. Based on GIDL, we enable automated synthesis of binding components for connecting heterogeneous services and Things onto VSB.

- Participants: Georgios Bouloukakis, Nikolaos Georgantas and Patient Ntumba
- Contact: Nikolaos Georgantas
- URL: <https://gitlab.ow2.org/chorevolution/evolution-service-bus>

## 6.6. Service traceroute

KEYWORDS: Network monitoring - Network diagnosis

FUNCTIONAL DESCRIPTION: Traceroute is often used to help diagnose when users experience issues with Internet applications or services. Unfortunately, probes issued by classic traceroute tools differ from application traffic and hence can be treated differently by middleboxes within the network. We propose a new traceroute tool, called Service traceroute. Service traceroute leverages the idea from paratrace, which passively listens to application traffic to then issue traceroute probes that pretend to be part of the application flow. We extend this idea to work for modern Internet services with support for identifying the flows to probe automatically, for tracing of multiple concurrent flows, and for UDP flows. We implement command-line and library versions of Service traceroute, which we release as open source.

- Partner: Princeton University
- Contact: Renata Cruz Teixeira
- URL: <https://github.com/wontoniii/service-traceroute>

## 6.7. TA

*TA - Traffic Analysis*

KEYWORDS: Quality of Experience - Network monitoring - Video analysis

FUNCTIONAL DESCRIPTION: System running at the home getaway that analyzes traffic generated by DASH on-demand and live video streams. The system tracks traffic patterns to infer key video QoE metrics such as average bitrate and re-buffering events. Moreover, the system exploits novel algorithms that use probing techniques, i.e. lightweight pings and traceroutes, to detect possible congestion location.

- Participants: Francesco Bronzino and Renata Cruz Teixeira
- Contact: Francesco Bronzino

## 6.8. HostView Mobile

KEYWORDS: Quality of Experience - Network monitoring

FUNCTIONAL DESCRIPTION: HostView for mobile runs on Android devices to monitor user system and network performance together with user feedback on Internet experience.

- Contact: Giulio Grassi

## WHISPER Project-Team

# 6. New Software and Platforms

## 6.1. Coccinelle

KEYWORDS: Code quality - Evolution - Infrastructure software

FUNCTIONAL DESCRIPTION: Coccinelle is a tool for code search and transformation for C programs. It has been extensively used for bug finding and evolutions in Linux kernel code.

- Participants: Gilles Muller, Julia Lawall, Nicolas Palix, Rene Rydhof Hansen and Thierry Martinez
- Partners: LIP6 - IRILL
- Contact: Julia Lawall
- URL: <http://coccinelle.lip6.fr>

## 6.2. Prequel

KEYWORDS: Code search - Git

SCIENTIFIC DESCRIPTION: The commit history of a code base such as the Linux kernel is a gold mine of information on how evolutions should be made, how bugs should be fixed, etc. Nevertheless, the high volume of commits available and the rudimentary filtering tools provided mean that it is often necessary to wade through a lot of irrelevant information before finding example commits that can help with a specific software development problem. To address this issue, we propose Prequel (Patch Query Language), which brings the descriptive power of code matching to the problem of querying a commit history.

FUNCTIONAL DESCRIPTION: Prequel is a tool for searching for complex patterns in the commits of software managed using git.

- Participants: Gilles Muller and Julia Lawall
- Partners: LIP6 - IRILL
- Contact: Julia Lawall
- URL: <http://prequel-pql.gforge.inria.fr/>

## 6.3. Usuba

KEYWORDS: Cryptography - Optimizing compiler - Synchronous language

FUNCTIONAL DESCRIPTION: Usuba is a programming language for specifying block ciphers as well as a bitslicing compiler, for producing high-throughput and secure code.

- Contact: Pierre-Evariste Dagand
- Publication: [Usuba, Optimizing & Trustworthy Bitslicing Compiler](#)
- URL: <https://github.com/DadaIsCrazy/usuba/>

## ALMAAnaCH Team

# 5. New Software and Platforms

## 5.1. Enqi

- Author: Benoît Sagot
- Contact: Benoît Sagot

## 5.2. SYNTAX

KEYWORD: Parsing

FUNCTIONAL DESCRIPTION: Syntax system 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 Sx Pipe 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).

- Participants: Benoît Sagot and Pierre Boullier
- Contact: Pierre Boullier
- URL: <http://syntax.gforge.inria.fr/>

## 5.3. FRMG

KEYWORDS: Parsing - French

FUNCTIONAL DESCRIPTION: FRMG is a large-coverage linguistic meta-grammar of French. It can be compiled (using MGCOMP) into a Tree Adjoining Grammar, which, in turn, can be compiled (using DyALog) into a parser for French.

- Participant: Éric Villemonte De La Clergerie
- Contact: Éric De La Clergerie
- URL: <http://mgkit.gforge.inria.fr/>

## 5.4. MElt

*Maximum-Entropy lexicon-aware tagger*

KEYWORD: Part-of-speech tagger

FUNCTIONAL DESCRIPTION: MElt is a freely available (LGPL) state-of-the-art sequence labeller that is meant to be trained on both an annotated corpus and an external lexicon. It was developed by Pascal Denis and Benoît Sagot within the Alpage team, a joint Inria and Université Paris-Diderot team in Paris, France. MElt allows for using multiclass Maximum-Entropy Markov models (MEMMs) or multiclass perceptrons (multitrons) as underlying statistical devices. Its output is in the Brown format (one sentence per line, each sentence being a space-separated sequence of annotated words in the word/tag format).

MElt has been trained on various annotated corpora, using Alexina lexicons as source of lexical information. As a result, models for French, English, Spanish and Italian are included in the MElt package.

MElt also includes a normalization wrapper aimed at helping processing noisy text, such as user-generated data retrieved on the web. This wrapper is only available for French and English. It was used for parsing web data for both English and French, respectively during the SANCL shared task (Google Web Bank) and for developing the French Social Media Bank (Facebook, twitter and blog data).

- Contact: Benoît Sagot
- URL: <https://team.inria.fr/almanach/melt/>

## 5.5. dyalog-sr

KEYWORDS: Parsing - Deep learning - Natural language processing

FUNCTIONAL DESCRIPTION: DyALog-SR is a transition-based dependency parser, built on top of DyALog system. Parsing relies on dynamic programming techniques to handle beams. Supervised learning exploit a perceptron and aggressive early updates. DyALog-SR can handle word lattice and produce dependency graphs (instead of basic trees). It was tested during several shared tasks (SPMRL'2013 and SEMEVAL'2014). It achieves very good accuracy on French TreeBank, alone or by coupling with FRMG parser. In 2017, DyALog-SR has been extended into DyALog-SRNN by adding deep neuronal layers implemented with the Dynet library. The new version has participated to the evaluation campaigns CONLL UD 2017 (on more than 50 languages) and EPE 2017.

- Contact: Éric De La Clergerie

## 5.6. Crapbank

*French Social Media Bank*

KEYWORDS: Treebank - User-generated content

FUNCTIONAL DESCRIPTION: The French Social Media Bank is a treebank of French sentences coming from various social media sources (Twitter(c), Facebook(c)) and web forums (JeuxVidéos.com(c), Doctissimo.fr(c)). It contains different kind of linguistic annotations: - part-of-speech tags - surface syntactic representations (phrase-based representations) as well as normalized form whenever necessary.

- Contact: Djamé Seddah

## 5.7. DyALog

KEYWORD: Logic programming

FUNCTIONAL DESCRIPTION: 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.

- Participant: Éric Villemonte De La Clergerie
- Contact: Éric Villemonte De La Clergerie
- URL: <http://dyalog.gforge.inria.fr/>

## 5.8. SxPipe

KEYWORD: Surface text processing

SCIENTIFIC DESCRIPTION: Developed for French and for other languages, Sx Pipe 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.

**FUNCTIONAL DESCRIPTION:** SxPipe is a modular and customizable processing chain dedicated to applying to raw corpora a cascade of surface processing steps (tokenisation, wordform detection, non-deterministic spelling correction. . . ). It is used as a preliminary step before ALMAnaCH's parsers (e.g., FRMG) and for surface processing (named entities recognition, text normalization, unknown word extraction and processing...).

- Participants: Benoît Sagot, Djamé Seddah and Éric Villemonte De La Clergerie
- Contact: Benoît Sagot
- URL: <http://lingwb.gforge.inria.fr/>

## 5.9. Mgwiki

**KEYWORDS:** Parsing - French

**FUNCTIONAL DESCRIPTION:** Mgwiki is a linguistic wiki that may be 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.

- Participant: Éric Villemonte De La Clergerie
- Contact: Éric Villemonte De La Clergerie
- URL: <http://alpage.inria.fr/frmgwiki/>

## 5.10. WOLF

*WOrdnet Libre du Français (Free French Wordnet)*

**KEYWORDS:** WordNet - French - Semantic network - Lexical resource

**FUNCTIONAL DESCRIPTION:** The WOLF (Wordnet Libre du Français, Free French Wordnet) is a free semantic lexical resource (wordnet) for French.

The WOLF has been built from the Princeton WordNet (PWN) and various multilingual resources.

- Contact: Benoît Sagot
- URL: <http://alpage.inria.fr/~sagot/wolf-en.html>

## 5.11. vera

**KEYWORD:** Text mining

**FUNCTIONAL DESCRIPTION:** Automatic analysis of answers to open-ended questions based on NLP and statistical analysis and visualisation techniques (vera is currently restricted to employee surveys).

- Participants: Benoît Sagot and Dimitri Tcherniak
- Partner: Verbatim Analysis
- Contact: Benoît Sagot

## 5.12. Alexina

*Atelier pour les LEXiques INformatiques et leur Acquisition*

**KEYWORD:** Lexical resource

**FUNCTIONAL DESCRIPTION:** Alexina is ALMAnaCH's 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 Lefff, a morphological and syntactic lexicon for French.

- Participant: Benoît Sagot
- Contact: Benoît Sagot
- URL: <http://gforge.inria.fr/projects/alexina/>

### **5.13. FQB**

*French QuestionBank*

KEYWORD: Treebank

FUNCTIONAL DESCRIPTION: The French QuestionBanks is a corpus of around 2000 questions coming from various domains (TREC data set, French governmental organisation, NGOs, etc..) it contains different kind of annotations - morpho-syntactic ones (POS, lemmas) - surface syntaxe (phrase based and dependency structures) with long-distance dependency annotations.

The TREC part is aligned with the English QuestionBank (Judge et al, 2006).

- Contact: Djamé Seddah

### **5.14. Sequoia corpus**

KEYWORD: Treebank

FUNCTIONAL DESCRIPTION: The Sequoia corpus contains French sentences, annotated with various linguistic information: - parts-of-speech - surface syntactic representations (both constituency trees and dependency trees) - deep syntactic representations (which are deep syntactic dependency graphs)

- Contact: Djamé Seddah



## COML Team

# 5. New Software and Platforms

## 5.1. abkhazia

KEYWORDS: Speech recognition - Speech-text alignment

FUNCTIONAL DESCRIPTION: The Abkhazia software makes it easy to obtain simple baselines for supervised ASR (using Kaldi) and ABX tasks (using ABXpy) on the large corpora of speech recordings typically used in speech engineering, linguistics or cognitive science research.

- Contact: Emmanuel Dupoux
- URL: <https://github.com/bootphon/abkhazia>

## 5.2. TDE

*Term Discovery Evaluation*

KEYWORDS: NLP - Speech recognition - Speech

SCIENTIFIC DESCRIPTION: This toolbox allows the user to judge of the quality of a word discovery algorithm. It evaluates the algorithms on these criteria : - Boundary : efficiency of the algorithm to found the actual boundaries of the words - Group : efficiency of the algorithm to group similar words - Token/Type: efficiency of the algorithm to find all words from the corpus (types), and to find all occurrences (token) of these words. - NED : Mean of the edit distance across all the word pairs found by the algorithm - Coverage : efficiency of the algorithm to find every discoverable phone in the corpus

FUNCTIONAL DESCRIPTION: Toolbox to evaluate algorithms that segment speech into words. It allows the user to evaluate the efficiency of algorithms to segment speech into words, and create clusters of similar words.

- Contact: Emmanuel Dupoux
- URL: <https://github.com/bootphon/TDE>

## 5.3. ABXpy

KEYWORDS: Evaluation - Speech recognition - Machine learning

FUNCTIONAL DESCRIPTION: The ABX package gives a performance score to speech recognition systems by measuring their capacity to discriminate linguistic contrasts (accents, phonemes, speakers, etc...)

- Contact: Emmanuel Dupoux
- URL: <https://github.com/bootphon/ABXpy>

## 5.4. h5features

KEYWORD: File format

FUNCTIONAL DESCRIPTION: The h5features python package provides easy to use and efficient storage of large features data on the HDF5 binary file format.

- Contact: Emmanuel Dupoux
- URL: <https://github.com/bootphon/h5features>

## RITS Project-Team

# 6. New Software and Platforms

## 6.1. PML-SLAM

KEYWORD: Localization

SCIENTIFIC DESCRIPTION: Simultaneous Localization and Mapping method based on 2D laser data.

- Participants: Fawzi Nashashibi and Zayed Alsayed
- Contact: Fawzi Nashashibi

## 6.2. V2ProVue

*Vehicle-to-Pedestrian*

FUNCTIONAL DESCRIPTION: It is a software developed for the Vehicle-to-Pedestrian (V2P) communications, risk calculation, and alarming pedestrians of collision risk. This software is made of an Android application dedicated to pedestrians and RtMaps modules for the vehicles.

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.

- Contact: Fawzi Nashashibi

## 6.3. SimConVA

*Connected Autonomous Vehicles Simulator*

FUNCTIONAL DESCRIPTION: The software provides an interface between the network simulator ns-3 (<https://www.nsnam.org/>) and the modular prototyping framework RTMaps (<https://intempora.com/>).

This code allows to create an RTMaps component which activates and controls the ns-3 simulator. The component handles the sending and reception of data packets between ns-3 and RTMaps for each vehicle. It also handles the mobility of vehicles in ns-3 using their known position in RTMaps.

- Authors: Pierre Merdrignac, Oyunchimeg Shagdar and Jean-Marc Lasgouttes
- Contact: Jean-Marc Lasgouttes

## VALDA Project-Team

# 5. New Software and Platforms

## 5.1. ProvSQL

KEYWORDS: Databases - Provenance - Probability

FUNCTIONAL DESCRIPTION: The goal of the ProvSQL project is to add support for (m-)semiring provenance and uncertainty management to PostgreSQL databases, in the form of a PostgreSQL extension/module/plugin.

NEWS OF THE YEAR: Support for where-provenance has been completed. Numerous bug fixes. A docker version has been produced, for ease of installation. Demonstration scenarios are included.

- Participants: Pierre Senellart and Yann Ramusat
- Contact: Pierre Senellart
- Publications: [Provenance and Probabilities in Relational Databases: From Theory to Practice - ProvSQL: Provenance and Probability Management in PostgreSQL](#)
- URL: <https://github.com/PierreSenellart/provsql>

## 5.2. WAE

*Web archive explorer*

KEYWORDS: Information extraction - Web archives

FUNCTIONAL DESCRIPTION: The Web archive explorer is a system for extracting, fragmenting and exploring Web archives.

- Contact: Quentin Lobbe
- Publications: [Archives, Web fragments and diasporas. For a disaggregated exploration of web archives related to online representations of diasporas - Where the dead blogs are. A Disaggregated Exploration of Web Archives to Reveal Extinct Online Collectives - Revealing Historical Events out of Web Archives](#)
- URL: <https://github.com/lobbeque/archive-miner>

## 5.3. apxproof

KEYWORD: LaTeX

FUNCTIONAL DESCRIPTION: apxproof is a LaTeX package facilitating the typesetting of research articles with proofs in appendix, a common practice in database theory and theoretical computer science in general. The appendix material is written in the LaTeX code along with the main text which it naturally complements, and it is automatically deferred. The package can automatically send proofs to the appendix, can repeat in the appendix the theorem environments stated in the main text, can section the appendix automatically based on the sectioning of the main text, and supports a separate bibliography for the appendix material.

RELEASE FUNCTIONAL DESCRIPTION: Numerous bug fixes and robustness enhancements, link theorems to their repeated Versions, proper management of equations in repeated theorems

NEWS OF THE YEAR: Major 1.1.0 release adding several features (link theorems to their repeated versions, proper management of equations in repeated theorems), beyond this, bug fixes, robustness enhancements, better support for some document classes.

- Participant: Pierre Senellart
- Contact: Pierre Senellart
- URL: <https://github.com/PierreSenellart/apxproof>

## 5.4. Sgvizler2

KEYWORDS: SPARQL - Data visualization - JavaScript

FUNCTIONAL DESCRIPTION: This project is the reboot in Typescript of project Sgvizler of Martin G. Skjæveland.

- Partners: LRI - Laboratoire de Recherche en Informatique - BorderCloud
- Contact: Karima Rafes
- URL: <https://github.com/BorderCloud/sgvizler2>

## 5.5. SPARQL-PHP

KEYWORDS: SPARQL - PHP

FUNCTIONAL DESCRIPTION: Very simple SPARQL client for PHP.

- Partners: LRI - Laboratoire de Recherche en Informatique - BorderCloud
- Contact: Karima Rafes
- URL: <https://github.com/BorderCloud/SPARQL>

## 5.6. TFT

*Tester for Triplestore*

KEYWORDS: PHP - SPARQL

FUNCTIONAL DESCRIPTION: TFT (Tester for Triplestore) is a script PHP to pass tests through a SPARQL service.

- Partners: LRI - Laboratoire de Recherche en Informatique - BorderCloud
- Contact: Karima Rafes
- URL: <https://github.com/BorderCloud/TFT>

## WILLOW Project-Team

# 6. New Software and Platforms

## 6.1. NCNet

*Neighbourhood Consensus Networks*

KEYWORDS: Computer vision - Machine learning

FUNCTIONAL DESCRIPTION: Open source release of the software package for the NIPS'18 paper by Rocco et al. "Neighbourhood Consensus Networks". This release provides a full implementation of the method, including code for training models, and testing on standard datasets, as well as trained models.

- Participants: Ignacio Rocco Spremolla, Mircea Cimpoi, Akihiko Torii, Relja Arandjelovic, Tomas Pajdla and Josef Sivic
- Contact: Ignacio Rocco Spremolla
- Publication: [Neighbourhood Consensus Networks](#)
- URL: <https://www.di.ens.fr/willow/research/ncnet/>

## 6.2. Mixture-of-Embedding-Experts

KEYWORD: Computer vision

FUNCTIONAL DESCRIPTION: Joint understanding of video and language is an active research area with many applications. Prior work in this domain typically relies on learning text-video embeddings. One difficulty with this approach, however, is the lack of large-scale annotated video-caption datasets for training. To address this issue, we aim at learning text-video embeddings from heterogeneous data sources. To this end, we propose a Mixture-of-Embedding-Experts (MEE) model with ability to handle missing input modalities during training. As a result, our framework can learn improved text-video embeddings simultaneously from image and video datasets. We also show the generalization of MEE to other input modalities such as face descriptors.

- Participants: Ivan Laptev and Josef Sivic
- Contact: Antoine Miech
- Publication: [Learning a Text-Video Embedding from Incomplete and Heterogeneous Data](#)
- URL: <https://www.di.ens.fr/willow/research/mee/>

## 6.3. BodyNet

*BodyNet: Volumetric Inference of 3D Human Body Shapes*

KEYWORDS: Computer vision - Machine learning

FUNCTIONAL DESCRIPTION: BodyNet has the code to train multi-task neural networks for 2D/3D pose estimation, 2D body part segmentation, and 3D volumetric shape estimation of human bodies given single RGB images as input. The release also contains pre-trained models.

- Participants: Gül Varol Simsekli, Ivan Laptev and Cordelia Schmid
- Contact: Gül Varol Simsekli
- Publication: [BodyNet: Volumetric Inference of 3D Human Body Shapes](#)
- URL: <https://www.di.ens.fr/willow/research/bodynet/>

## 6.4. FlexWLoc

*Flexible Weakly supervised action Localization model*

KEYWORDS: Computer vision - Machine learning

FUNCTIONAL DESCRIPTION: Open source release of the software package for the NIPS'18 paper by Chéron et al. "A flexible model for training action localization with varying levels of supervision". This release provides a full implementation of the method, including code for training and testing.

- Participants: Guilhem Chéron, Jean-Baptiste Alayrac, Ivan Laptev and Cordelia Schmid
- Contact: Guilhem Chéron
- Publication: [A flexible model for training action localization with varying levels of supervision](#)
- URL: <https://www.di.ens.fr/willow/research/weakactionloc/>

## 6.5. Pinocchio

KEYWORDS: Robotics - Biomechanics - Mechanical multi-body systems

FUNCTIONAL DESCRIPTION: Pinocchio instantiates state-of-the-art Rigid Body Algorithms for poly-articulated systems based on revisited Roy Featherstone's algorithms. In addition, Pinocchio instantiates analytical derivatives of the main Rigid-Body Algorithms like the Recursive Newton-Euler Algorithms or the Articulated-Body Algorithm. Pinocchio is first tailored for legged robotics applications, but it can be used in extra contexts. It is built upon Eigen for linear algebra and FCL for collision detection. Pinocchio comes with a Python interface for fast code prototyping.

- Partner: CNRS
- Contact: JUSTIN CARPENTIER
- URL: <https://github.com/stack-of-tasks/pinocchio>

## 6.6. weakalign

*End-to-end weakly-supervised semantic alignment*

KEYWORDS: Computer vision - Machine learning

FUNCTIONAL DESCRIPTION: Open source release of the software package for the CVPR'18 paper by Rocco et al. "End-to-end weakly-supervised semantic alignment". This release provides a full implementation of the method, including code for training models, and testing on standard datasets, as well as trained models.

- Participants: Ignacio Rocco Spremolla, Relja Arandjelovic and Josef Sivic
- Contact: Ignacio Rocco Spremolla
- Publication: [End-to-end weakly-supervised semantic alignment](#)
- URL: <https://www.di.ens.fr/willow/research/weakalign/>

## 6.7. InLoc

*Indoor Visual Localization with Dense Matching and View Synthesis*

KEYWORD: Computer vision

FUNCTIONAL DESCRIPTION: Open source release of the software package for the CVPR'18 paper by Taira et al. "Indoor Visual Localization with Dense Matching and View Synthesis". This release provides a full implementation of the method.

- Participants: Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla and Akihiko Torii
- Contact: Josef Sivic
- Publication: [InLoc: Indoor Visual Localization with Dense Matching and View Synthesis](#)
- URL: [https://github.com/HajimeTaira/InLoc\\_demo](https://github.com/HajimeTaira/InLoc_demo)