



RESEARCH CENTER
Paris - Rocquencourt

FIELD

Activity Report 2015

Section Software

Edition: 2016-03-21

ALGORITHMICS, PROGRAMMING, SOFTWARE AND ARCHITECTURE

1. ANTIQUE Project-Team 5
2. AOSTE Project-Team 10
3. CASCADE Project-Team (section vide) 13
4. CRYPT Team (section vide) 14
5. DEDUCTEAM Team 15
6. GALLIUM Project-Team 19
7. MUTANT Project-Team 21
8. PARKAS Project-Team 25
9. PIR2 Project-Team 30
10. POLSYS Project-Team 32
11. PROSECCO Project-Team 33
12. SECRET Project-Team (section vide) 35

APPLIED MATHEMATICS, COMPUTATION AND SIMULATION

13. GAMMA3 Project-Team 36
14. MATHERIALS Project-Team 43
15. MATHRISK Project-Team 44
16. MOKAPLAN Project-Team 48
17. QUANTIC Project-Team (section vide) 49
18. SIERRA Project-Team 50

DIGITAL HEALTH, BIOLOGY AND EARTH

19. ANGE Project-Team 51
20. ARAMIS Project-Team 52
21. CLIME Project-Team 55
22. LIFEWARE Project-Team 57
23. MAMBA Project-Team 58
24. MYCENAE Project-Team (section vide) 59
25. REO Project-Team 60
26. SERENA Team 61

NETWORKS, SYSTEMS AND SERVICES, DISTRIBUTED COMPUTING

27. ALPINES Project-Team 62
28. DYOGENE Project-Team 64
29. EVA Team 65
30. GANG Project-Team 68
31. MIMOVE Team 69
32. MUSE Team 72
33. RAP Project-Team (section vide) 74
34. REGAL Project-Team 75
35. WHISPER Project-Team 77

PERCEPTION, COGNITION AND INTERACTION

36. ALPAGE Project-Team 80

37. RITS Project-Team	85
38. SMIS Project-Team	87
39. WILLOW Project-Team	88

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

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 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: Patrick Cousot, Jérôme Feret, Laurent Mauborgne, Antoine Miné and Xavier Rival
- Partner: CNRS
- Contact: Patrick Cousot
- URL: <http://www.astree.ens.fr/>

5.3. AstréeA

The AstréeA Static Analyzer of Asynchronous Software

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 ARINC 653 OS specification used in embedded industrial aeronautic software. Additionally, AstréeA employs a weakly-consistent memory semantics to model memory accesses not protected by a mutex, in order to take into account soundly hardware and compiler-level program transformations (such as optimizations). AstréeA checks for the same run-time errors as Astrée, with the addition of data-races.

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: Patrick Cousot, Radhia Cousot, Jérôme Feret, Antoine Miné and Xavier Rival est toujours membre de Inria. logiciels Inria): <https://bil.inria.fr/>
- Contact: Patrick Cousot
- URL: <http://www.astreea.ens.fr/>

5.4. ClangML

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: François Berenger, Pippijn Van Steenhoven and Devin Mccoughlin toujours membre de Inria. Inria): <https://bil.inria.fr/>
- Contact: François Berenger
- 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: Caterina Urban and Antoine Miné
- 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

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 350 small size test cases that are used as regression tests.

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

5.8. OPENKAPPA

La platte-forme de modélisation OpenKappa

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.

- Participants: Pierre Boutillier, Vincent Danos, Jérôme Feret, Walter Fontana, Russ Harmer, Jean Krivine and Kim Quyen Ly
- 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. Translation Validation

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 guaranties that the compiled code is correct with respect to the source code. 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. It ensures that no compiler bug did cause incorrect code to be generated.

- Participant: Xavier Rival
- Contact: Xavier Rival

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

5.12. CELIA

The MemCAD static analyzer

FUNCTIONAL DESCRIPTION

CELIA is a tool for the static analysis and verification of C programs manipulating dynamic lists. The static analyzer computes for each control point of a C program the assertions which are true (i.e., invariant) at this control point. The specification language is a combination of Separation Logic with a first order logic over sequences of integers. The inferred properties describe the shape of the lists, their size, the relations between the data (or the sum, or the multiset of data) in list cells. The analysis is inter-procedural, i.e., the assertions computed relate the procedure local heap on entry to the corresponding local heap on exit of the procedure. The results of the analysis can provide insights about equivalence of procedures on lists or null pointer dereferencing. The analysis is currently extended to programs manipulating concurrent data structures.

- Participants: Ahmed Bouajjani, Cezara Drăgoi, Constantin Enea, Mihaela Sighireanu
- Contact: Cezara Drăgoi
- URL: <http://www.liafa.jussieu.fr/celia/>

AOSTE Project-Team

6. New Software and Platforms

6.1. SynDEx

KEYWORDS: Embedded systems - Real time - Optimization - Distributed - 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

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

6.2. TimeSquare

KEYWORDS: Profil MARTE - Embedded systems - UML - IDM

SCIENTIFIC DESCRIPTION TimeSquare offers six main functionalities:

- * graphical and/or textual interactive specification of logical clocks and relative constraints between them,
- * definition and handling of user-defined clock constraint libraries,
- * automated simulation of concurrent behavior traces respecting such constraints, using a Boolean solver for consistent trace extraction,

* call-back mechanisms for the traceability of results (animation of models, display and interaction with waveform representations, generation of sequence diagrams...).

* compilation to pure java code to enable embedding in non eclipse applications or to be integrated as a time and concurrency solver within an existing tool.

* a generation of the whole state space of a specification (if finite of course) in order to enable model checking of temporal properties on it

FUNCTIONAL DESCRIPTION

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

- Participants: Frédéric Mallet, and Julien Deantoni
- Contact: Frédéric Mallet
- URL: <http://timesquare.inria.fr>

6.3. Lopht

KEYWORDS: Real-time scheduling, compilation, ARINC 653, TTEthernet, Many-core, Network-on-chip

SCIENTIFIC DESCRIPTION

Lopht is an acronym for Logical to Physical Time Compiler. Lopht has been designed as an implementation of the AAA methodology. Like SynDEx, Lopht relies on off-line allocation and scheduling techniques to allow real-time implementation of dataflow synchronous specifications (e.g. Scade/Heptagon) onto multiprocessor systems. The main originality is that Lopht takes a compilation-like approach based on:

- Precise modeling of its implementation platforms. For this reason, Lopht targets novel, more complex architectures such as many-core chips and time-triggered embedded systems based on standards such as ARINC 653 and TTEthernet.
- Taking into account complex non-functional specifications covering real-time (release dates and deadlines possibly different from period, major time frame, end-to-end flow constraints), ARINC 653 partitioning, the possibility to preempt or not each task, and finally SynDEx-like allocation
- Tight integration of program analysis, scheduling, and optimization approaches coming from 3 research fields (real-time scheduling, compilation, and synchronous languages) to improve the efficiency of resulting implementations while ensuring functional correctness, the respect of non-functional requirements, and scalability.

FUNCTIONAL DESCRIPTION Lopht is a software tool similar in functioning to a compiler. It takes as input one file defining the functional and non-functional specification of a system (including a model of the execution platform and non-functional requirements). It automatically produces all files needed to build a running implementation (the C code for each processor cores and the configuration files).

- Participants: Dumitru Potop-Butucaru, Keryan Didier
- Contact: Dumitru Potop-Butucaru (dumitru.potop@inria.fr)

6.4. EVT Kopernic

KEYWORD: Embedded systems

EVT Kopernic provides a probabilistic worst case execution time estimation for a program on a processor. The tool takes a set of measurements (execution times of the program on the processor) as input and it provides a probability distribution. The first version released in 2015 is restricted to independent data and a second version has been obtained for dependent data during the last part of the year. A third version provides rules for obtaining the measurements is to be released in the first part of 2016.

- Participants: Liliana Cucu and Adriana Gogonel
- Contact: Liliana Cucu
- URL: Currently restricted distribution

6.5. SAS

Simulation and Analysis of Scheduling

SCIENTIFIC DESCRIPTION

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

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

FUNCTIONAL DESCRIPTION

The SAS software allows the user to perform the schedulability analysis of periodic task systems in the monoprocessor case.

- Participants: Daniel De Rauglaudre and Yves Sorel
- Contact: Yves Sorel
- URL: <http://pauillac.inria.fr/~ddr/sas-dameid/>

CASCADE Project-Team (section vide)

CRYPT Team (section vide)

DEDUCTEAM Team

6. New Software and Platforms

6.1. Introduction

Deducteam develops several kinds of tools or libraries:

- Proof checkers:
 - Dedukti: proof checker for the $\lambda\Pi$ -calculus modulo rewriting
 - Sukerujo: extension of Dedukti with syntactic constructions for records, strings, lists, etc.
 - Rainbow: CPF termination certificate verifier
- Tools for translating into Dedukti's proof format proofs coming from various other provers:
 - Coqine translates Coq proofs
 - Focalide translates Focalize proofs
 - Holide translates OpenTheory proofs (HOL-Light, HOL4, ProofPower)
 - Krajono translates Matita proofs
 - Sigmaid translates ζ -calculus
- Automated theorem provers:
 - iProverModulo: theorem prover based on polarized resolution modulo
 - SuperZenon: extension of Zenon using superdeduction
 - ZenonArith: extension of Zenon using the simplex algorithm for arithmetic
 - ZenonModulo: extension of Zenon using deduction modulo and producing Dedukti proofs
 - Zipperposition: superposition prover featuring arithmetic and induction
 - HOT: automated termination prover for higher-order rewrite systems
- Libraries or generation tools:
 - CoLoR: Coq library on rewriting theory and termination
 - Logtk: library for first-order automated reasoning
 - mSat: modular SAT/SMT solver with proof output
 - Moca: generator of construction functions for types with relations on constructors

In the following, we only details software that received improvements in 2015.

In addition, Shuai Wang developed the ProofCloud prototype, a proof retrieval engine for verified higher order proofs. ProofCloud provides a fast proof searching service for mathematicians and computer scientists for the reuse of proofs and proof packages. Using ProofCloud, he conducted a statistical analysis of the OpenTheory repository.

6.2. Autotheo

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

- Contact: Guillaume Burel
- URL: http://www.ensie.fr/~guillaume.burel/blackandwhite_autotheo.html.en

In 2015, we extended Autotheo so that it prints a derivation of the transformation of the axioms into rewriting rules. This derivation is in TSTP format and includes the CNF conversions obtained from the prover E.

6.3. CoLoR

CoLoR is Coq library on rewriting theory and termination. It provides many definitions and theorems on various mathematical structures (quasi-ordered sets, relations, ordered semi-rings, etc.), data structures (lists, vectors, matrices, polynomials, finite graphs), term structures (strings, first-order terms, lambda-terms, etc.), transformation techniques (dependency pairs, semantic labeling, etc.) and (non-)termination criteria (polynomial and matrix interpretations, recursive path ordering, computability closure, etc.).

- Contact: Frédéric Blanqui
- URL: <http://color.inria.fr/>

In 2015, CoLoR has been enriched and improved in various ways:

- Its compilation time has been improved by about 20%.
- The results on computability have been extended to η -reduction.
- It has been enriched by a library on finite and infinite sets, and a proof of the infinite Ramsey's theorem [54].
- CoLoR is now available on OPAM.

6.4. Coqine

Coqine translates Coq proofs into Dedukti proofs.

- Contact: Guillaume Burel
- URL: http://www.ensiie.fr/~guillaume.burel/blackandwhite_coqInE.html.en

The addition of higher-order pattern matching in Dedukti allowed the encoding of universes.

6.5. Dedukti

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

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

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

- Contact: Olivier Hermant
- URL: <http://dedukti.gforge.inria.fr/>

The new version of Dedukti (v2.5) brings two major improvements.

First the typing of rewrite rules has been completely reworked. It can now check a large class of rewrite rules including rules whose left-hand sides are not algebraic nor well-typed. Moreover the typing context do not need to be given with the rewrite rule anymore, as it is inferred by Dedukti, and therefore it is more convenient for the user.

Second, Dedukti can now be interfaced with automatic confluence checkers in order to check that the rewrite system generated by the rewrite rules together with beta reduction is confluent. This verification is important as the soundness of the program relies on this hypothesis.

6.6. Focalide

Focalide is an extension of the FoCaLiZe compiler which produces Dedukti files.

- Contact: Raphaël Cauderlier
- URL: <http://deducteam.gforge.inria.fr/focalide/>

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

6.7. Holide

Holide translates HOL proofs to Dedukti proofs, using the OpenTheory standard (common to HOL Light and HOL4).

- Contact: Guillaume Burel
- URL: <http://deducteam.gforge.inria.fr/holide/>

Shuai Wang fixed a number of problems, especially in the translation of type variables, allowing us to translate more libraries.

6.8. iProverModulo

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

- Contact: Guillaume Burel
- URL: http://www.ensie.fr/~guillaume.burel/blackandwhite_iProverModulo.html.en

In 2015, we improved its integration with Autotheo.

6.9. Krajono

Krajono translates Matita proofs into Dedukti proofs.

- Contact: Guillaume Burel
- URL: <http://deducteam.gforge.inria.fr/krajono/>

First working version able to translate the Matita library on arithmetics.

6.10. mSAT

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

- Contact: Guillaume Bury
- URL: <https://github.com/Gbury/mSAT>

mSAT now provides a functor for generating a McSat solver, outputs a model or a proof, and provides a push/pop functionality.

6.11. ZenonModulo

Zenon Modulo is an extension of the automated theorem prover Zenon. Compared to Super Zenon, it can deal with rewrite rules both over propositions and terms. Like Super Zenon, Zenon Modulo is able to deal with any first-order theory by means of a similar heuristic.

- Contact: Pierre Halmagrand
- URL: <http://deducteam.gforge.inria.fr/zenonmodulo/>

In 2015, we extended Zenon Modulo to polymorphism. Moreover, it can now take TPTP-TFF1 problems as input, and output Dedukti's proofs.

Guillaume Bury continued to improve an extension of Zenon with arithmetic.

6.12. Zipperposition

Zipperposition is an implementation of the superposition method that relies on the library Logtk for basic logic data structures and algorithms. Zipperposition is designed as a testbed for extensions to superposition, and can currently deal with polymorphic typed logic, integer arithmetic and total orderings.

- Contact: Simon Cruanes
- URL: <http://deducteam.gforge.inria.fr/zipperposition/>

In 2015, we extended Zipperposition to structural induction.

GALLIUM Project-Team

6. New Software and Platforms

6.1. CompCert

Participants: Xavier Leroy [**contact**], Sandrine Blazy [team Celtique], Jacques-Henri Jourdan, Bernhard Schommer [AbsInt GmbH].

The CompCert project investigates the formal verification of realistic compilers usable for critical embedded software. Such verified compilers come with a mathematical, machine-checked proof that the generated executable code behaves exactly as prescribed by the semantics of the source program. By ruling out the possibility of compiler-introduced bugs, verified compilers strengthen the guarantees that can be obtained by applying formal methods to source programs. **AbsInt Angewandte Informatik GmbH** sells a commercial version of CompCert with long-term maintenance.

- URL: <http://compcert.inria.fr/> (academic), <http://www.absint.com/compcert/> (commercial).

6.2. Diy

Participants: Luc Maranget [**contact**], Jade Alglave [Microsoft Research, Cambridge], Keryan Didier.

The **diy** suite (for “Do It Yourself”) 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, ARM or generic (LISA) 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. Recent versions also take a subset of the C language as input, so as to test and simulate the C11 model.

- URL: <http://diy.inria.fr/>

6.3. Menhir

Participants: François Pottier [**contact**], Yann Régis-Gianas [Université Paris Diderot].

Menhir is a LR(1) parser generator for the OCaml programming language. That is, Menhir compiles LR(1) grammar specifications down to OCaml code.

- URL: <http://gallium.inria.fr/~fpottier/menhir/>

6.4. OCaml

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

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 SPARC), a debugger, a documentation generator, a compilation manager, a package manager, and many libraries contributed by the user community.

- URL: <http://ocaml.org/>

6.5. PASL

Participants: Mike Rainey [**contact**], Arthur Charguéraud, Umut Acar.

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.

- URL: <http://deepsea.inria.fr/pasl/>

6.6. Zenon

Participants: Damien Doligez [[contact](#)], Guillaume Bury [CNAM], David Delahaye [CNAM], Pierre Halmagrand [team DEDUCTEAM], Olivier Hermant [MINES ParisTech].

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

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

- URL: <http://opam.ocaml.org/packages/zenon/zenon.0.8.0/>

MUTANT Project-Team

6. New Software and Platforms

6.1. Antescofo

Participants: Arshia Cont, Jean-Louis Giavitto, Philippe Cuvillier, José Echeveste.

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

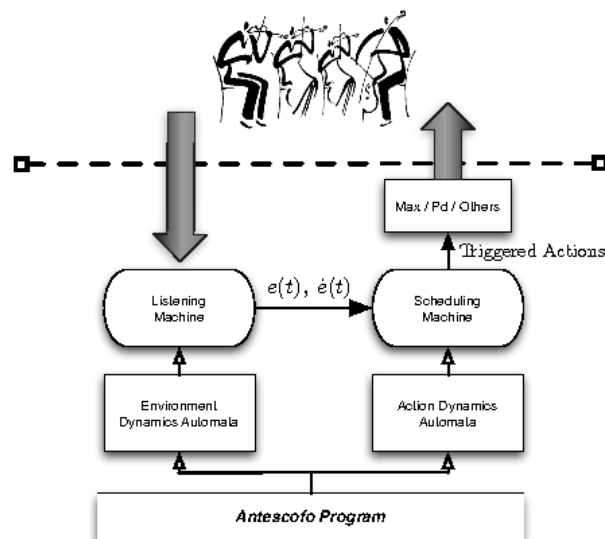


Figure 4. General scheme of Antescofo virtual machine

Antescofo v0.9 was released in November 2015. It contains major additions in the language (see Sections 7.5 and 7.6) as well as machine listening especially for singing voice and highly polyphonic instruments (See Release Notes). [Antescofo Reference Guide](#) is a collaborative document referencing the language and its usage, showcasing the software's latest developments.

- Participants: Arshia Cont, Jean-Louis Giavitto, Philippe Cuvillier and José Echeveste
- Contact: Arshia Cont
- URL: <http://forumnet.ircam.fr/product/antescofo/>

6.2. Ascograph

Participants: Arshia Cont, Grig Burloiu, Robert Piéchaud.

FUNCTIONAL DESCRIPTION

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

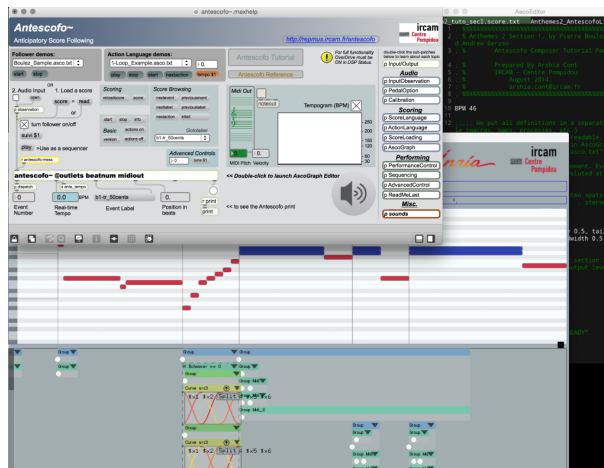


Figure 5. Antescofo and AscoGraph Screen Shorts (Nov. 2015)

In 2015, AscoGraph's User Interaction was redesigned as reported in [12], [13] and furthermore, a new Score Import procedure was developed and released in v0.25 (See Release Notes). See also 7.8 .

- Contact: Arshia Cont
- URL: <http://forumnet.ircam.fr/product/antESCOFO/>

6.3. Antescofo Timed Test Platform

Participants: Clément Poncelet, Florent Jacquemard, Pierre Donat-Bouillud.

The frequent use of Antescofo in live and public performances with human musicians implies strong requirements of temporal reliability and robustness to unforeseen errors in input. To address these requirements and help the development of the system and authoring of pieces by users, we are developing a platform for the automation of testing the behavior of Antescofo on a given score, with of focus on timed behavior. It is based on state of the art techniques and tools for *model-based testing* of embedded systems [37], and makes it possible to automate the following main tasks:

1. offline and on-the-fly generation of relevant input data for testing (i.e. fake performances of musicians, including timing values), with the sake of exhaustiveness,
2. computation of the corresponding expected output, according to a formal specification of the expected behavior of the system on a given mixed score,

3. black-box execution of the input test data on the System Under Test,
4. comparison of expected and real output and production of a test verdict.

The input and output data are timed traces (sequences of discrete events together with inter-event durations). Our method is based on formal models (specifications) in an ad hoc medium-level intermediate representation (IR). We have developed a compiler for producing automatically such IR models from Antescofo high level mixed scores.

Then, in the offline approach, the IR is passed, after conversion to Timed Automata, to the model-checker Uppaal, to which is delegated the above task (1), following coverage criteria, and the task (2), by simulation. In the online approach, tasks (1) and (2) are realized during the execution of the IR by a Virtual Machine developed on purpose. Moreover, we have implemented several tools for Tasks (3) and (4), corresponding to different boundaries for the implementation under test (black box): e.g. the interpreter of Antescofo's synchronous language alone, or with tempo detection, or the whole system.

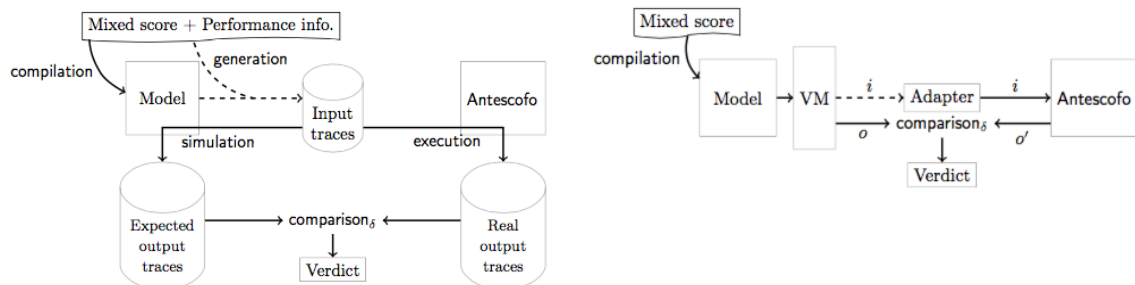


Figure 6. Offline and Online workflows for Antescofo Model Based Testing

Our fully automatic framework has been applied to real mixed scores used in concerts and the results obtained have permitted to identify bugs in Antescofo.

6.4. Rhythm Quantization in Open Music

Participants: Adrien Ycart, Florent Jacquemard, Jean Bresson.

We are developing a new system for rhythm transcription, which is the conversion of sequences of timestamped discrete events into common-western music notation. The input events may e.g. come from a performance on a MIDI keyboard or may also be the result of a computation. Our system privileges the user interactions in order to search for a satisfying balances between different criteria, in particular the precision of the transcription and the readability of the music score in outcome. It is integrated in the graphical environment for computer assisted music composition OpenMusic, and will be released publicly as a library of this system on the the Ircam's Forum.

We have developed a uniform approach for transcription, based on hierarchical representations of notation of duration as rhythm trees, and efficient algorithms for the lazy enumeration of solutions. It has been implemented via a dedicated interface making it possible the interactive exploration of the space of solutions, their visualization and their edition, with a particular focus on the processing of grace-notes and rests.

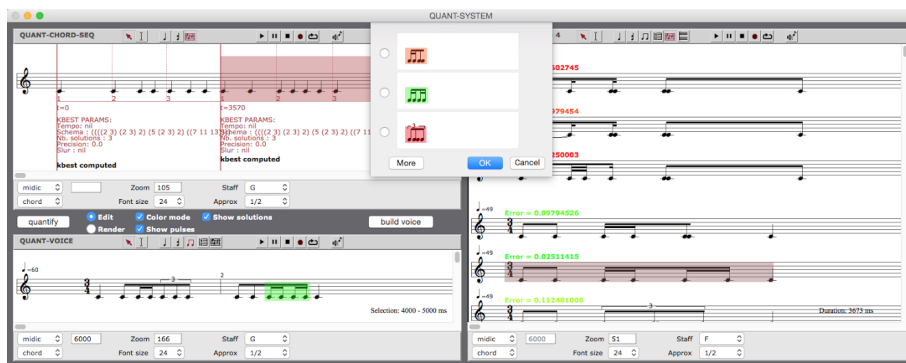


Figure 7. Rhythm Quantization in Open Music

PARKAS Project-Team

5. New Software and Platforms

5.1. Cmmtest: a tool for hunting concurrency compiler bugs

Participant: Francesco Zappa Nardelli [contact].

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

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

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

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

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

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++, libgcj,...). GCC was originally written as the compiler for the GNU operating system. The GNU system was developed to be 100% free software, free in the sense that it respects the user's freedom.

The emphasis as now moved towards LLVM and its Polly framework for polyhedral compilation.

- Participants: Albert Cohen, Riyadh Baghdadi, Mircea Namolaru and Nhat Minh Le
- Contact: Albert Cohen
- URL: <http://gcc.gnu.org/>

5.3. Heptagon

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 Synchronic, 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.

- Participants: Adrien Guatto, Marc Pouzet, Cédric Pasteur, Léonard Gérard, Brice Gelineau, Gwenaél Delaval and Eric Rutten
- Contact: Marc Pouzet
- URL: <http://heptagon.gforge.inria.fr>

5.4. Ott and Lem

lightweight executable mathematics

FUNCTIONAL DESCRIPTION

Ott and Lem are lightweight tools for writing, managing, and publishing large scale semantic definitions, where the scale makes it hard to keep a definition internally consistent, and to keep a tight correspondence between a definition and implementations.

The two tools are complementary. Ott focuses on higher-level programming language semantics. Lem is domain-specific language that resembles a pure subset of Objective Caml, supporting typical functional programming constructs and common logical mechanisms. Both tools can generate OCaml, HOL4, Coq, and Isabelle code. They also generate LaTeX code for inclusion of the language definition in scientific documents. Ott also supports a Lem backend for a tight integration between the two tools.

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

5.5. Lucid Synchronre

FUNCTIONAL DESCRIPTION

Lucid Synchronre 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.

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

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

5.7. 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: Sven Verdoolaege, Tobias Grosser, Michael Kruse, Chandan Reddy, Riyadh Baghdadi and Albert Cohen
- Contact: Sven Verdoolaege
- URL: <http://repo.or.cz/w/ppcg.git>

5.8. 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: Guillaume Baudart, in collaboration with Louis Mandel now at IBM Research
- Contact: Guillaume Baudart
- URL: <http://rml.lri.fr>

5.9. SundialsML

Sundials/ML

KEYWORDS: Simulation - Mathematics - Numerical simulations

FUNCTIONAL DESCRIPTION

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

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

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

NEW PROGRESS

The current version of Sundials/ML comprises about 37,000 lines of OCaml (plus 15,000 lines of api documentation) and 16,000 lines of C (plus 1600 lines of commentary). This year we worked on updating the interface to support Sundials 2.6.x. This involved adding support for a new solver (ARKODE), new modules for sparse matrices (SuperLU_MT and KLU), new nvector (pthreads and OpenMP), and new linear solvers (SPFGMR and PCG), as well as treating several other new or modified features. This work is almost complete and will be released early in 2016. The technical developments required to interface OCaml with this library are explained in a report which has been submitted as a deliverable in the MODRIO project: “D.4.2.16—OCaml interface to the Sundials suite of numerical solvers”. This text will be developed and submitted for journal publication in early 2016.

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

5.10. 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.

NEW PROGRESS

- Development and release of a comprehensive manual: <http://zelus.di.ens.fr/man/>
- Progress on the interaction of multiple numeric solvers (masters internship of V. Andreani).
- New causality analysis (detection of algebraic loops).
- Participants: Marc Pouzet and Timothy Bourke
- Contact: Marc Pouzet
- <http://zelus.di.ens.fr>

5.11. 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: Sven Verdoolaege, Michael Kruse and Albert Cohen
- Contact: Sven Verdoolaege
- URL: <http://repo.or.cz/w/isl.git>

5.12. LaTeX package: Checklistings

FUNCTIONAL DESCRIPTION

User manuals and papers about programming languages usually contain many code samples, often with accompanying compiler messages giving the types of declarations or error messages explaining why certain declarations are invalid.

The checklistings package augments the fancyvrb and listings packages for including source code in LaTeX documents with a way to pass the source code through a compiler and also include the resulting messages in the document. It also integrates with the HeVeA tool developed in the Gallium team: <http://hevea.inria.fr>.

The motivation is to check the code samples in a document for syntax and typing errors and to facilitate the inclusion of inferred types and compiler warnings or errors in a text. This package is intentionally very lightweight and unlike packages like python it is not intended for interacting with an interpreter or including the execution traces of code. While checklistings does not focus on a specific programming language, it is designed to work well with ML-like languages.

We developed this package to improve the quality of our papers and presentations on the Zélus programming language, but it is designed to be general purpose and also works, for instance, with OCaml programs.

- Participants: Timothy Bourke and Marc Pouzet
- Contact: Timothy Bourke
- URL: <http://www.ctan.org/pkg/checklistings>

PL.R2 Project-Team

5. New Software and Platforms

5.1. Coq

KEYWORDS: Proof - Certification - Formalisation

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, ...

- Participants: Benjamin Grégoire, Enrico Tassi, Bruno Barras, Yves Bertot, Pierre Boutillier, Xavier Clerc, Pierre Courtieu, Maxime Dénès, Stéphane Glondou, Vincent Gross, Hugo Herbelin, Pierre Letouzey, Assia Mahboubi, Julien Narboux, Jean-Marc Notin, Christine Paulin-Mohring, Pierre-Marie Pédrot, Loïc Pottier, Matthias Puech, Yann Régis-Gianas, François Ripault, Matthieu Sozeau, Arnaud Spiwack, Pierre-Yves Strub, Benjamin Werner, Guillaume Melquiond and Jean-Christophe Filliâtre
- Partners: CNRS - Université Paris-Sud - ENS Lyon - Université Paris-Diderot
- Contact: Hugo Herbelin
- URL: <http://coq.inria.fr/>

5.1.1. Version 8.5

Cf. Highlights section. Version 8.5 includes as well a number of miscellaneous changes, at the level of tactics, of the specification language, of the Coq tools, of the standard library, altogether amounting to about 150 items in the change log of the version. In particular, Pierre-Marie Pédrot has been working on the overall optimisation of Coq, by tracking hotspots in the code. Coq v8.5 is currently much more efficient than its v8.4 counterpart, and is about as quick as v8.3, while having been expanded with a lot of additional features.

As a counterpart, the complexity of this new version induced a long phase of experimentation which included 3 different beta versions spanned over the whole 2015 year, with the final version being eventually released for the CoqPL workshop in January 2016.

5.1.2. Universes

Matthieu Sozeau followed up his work on universe polymorphism and uncovered important theoretical and practical problems regarding conversion and unification of universe polymorphic definitions in the presence of cumulativity and the $\text{Prop} \leq \text{Type}$ rule, as well as the invariants of the consistency checker. He also collaborated with Maxime Dénès and Benjamin Grégoire (Gallium and Marelle) on adapting the efficient conversion tests to universe polymorphism and with Enrico Tassi (Marelle) on the integration with the asynchronous proof development infrastructure. The universe polymorphic system is part of the 8.5 release.

5.1.3. The Equations plugin

Matthieu Sozeau continued work on the Equations plugin and fixed the remaining bugs preventing full automation of a middle-size example of formalisation – the normalisation proof of a predicate version of System F – together with Cyprien Mangin, during his master's internship. This involved finding a new termination proof for the calculus and making the dependent pattern-matching compilation more robust and axiom-free, using a different encoding of pattern-matching problems. This work was presented at LFMTP'15 in Berlin [29]. Since then, the system has been adapted to work with universe polymorphism and the new features of Coq 8.5.

5.1.4. Proof development in Coq

Pierre Letouzey developed a few new results about some Hofstadter sequences (see <https://oeis.org/A005206> and <https://oeis.org/A123070>). These results have been proved in Coq, and they are presented in the technical report [39].

5.1.5. Proofs of algorithms on graphs

Chen Ran (ISCAS/SKLCS, Beijing) and Jean-Jacques Lévy pursued their work about producing readable formal proofs of graph algorithms. This work is performed in Why3 and partly in Coq. Graph algorithms are a good testbed for experimenting correctness proofs of programs with shared structures. We considered basic algorithms such as depth-first-search, random walk, acyclicity test, articulation points, strongly connected components, minimum spanning trees. In each case, the goal is to provide a simple proof as abstract as possible, although checked by computer. A longer term objective is to give formal proofs which could be inserted in algorithms textbooks. A progress work paper is under submission [41].

5.1.6. Development of programs for parallel and cloud computing

Frédéric Louergue continued his work on the SyDPaCC framework. The goal of this framework is to ease the systematic development of correct parallel programs, in particular large-scale data-intensive applications. The parallel versions of the programs are written with a Coq axiomatisation of Bulk Synchronous Parallel ML (BSML) primitives. New results about SyDPaCC include the design and implementation of a new version of the core of the framework [21]. This new version has been used in a course of École des Jeunes Chercheur/se/s en Informatique Mathématique (EJCIM 2015) [38].

As the SyDPaCC framework currently mixes certified code extracted from Coq and unverified code, Frédéric Louergue and Pierre Letouzey have worked on an extended extraction that generates, when possible, OCaml conditions for preconditions on function arguments. This part is still on-going work.

Frédéric Louergue collaborated with Frédéric Dabrowski and Thomas Pinsard (Univ. Orléans) on the semantics and compilation of languages with nested atomic sections and thread escape. In [18], the focus is on the semantics of programming languages providing these features. The main contribution is the precise definition of atomicity, well-synchronisation and the proof that the latter implies the strong form of the former. A formalisation of the results in the Coq proof assistant is described.

In [27], the compilation of a language with nested atomic sections and thread escape towards a language with threads and locks is addressed. The design decisions of this compilation pass and of the target language were made with respect to the ultimate goal of a mechanised proof of semantic preservation.

Frédéric Louergue collaborated with Allan Blanchard, Nikolai Kosmatov and Matthieu Lemerre (CEA LIST) on the verification of a critical component of a hypervisor. In [23], they present a case study on formal verification of the virtual memory system of the cloud hypervisor Anaxagoras, a microkernel designed for resource isolation and protection. The code under verification is specified and proven in the software verification framework, mostly using automatic theorem proving. The remaining properties are interactively proven with the Coq proof assistant.

Frédéric Louergue collaborated with Asma Guesmi, Pascal Berthomé and Patrice Clemente (INSA Centre Val de Loire) on resources placement in the Cloud taking into account security requirements [28].

5.2. Other software developments

In collaboration with François Pottier (Inria Gallium), Yann Régis-Gianas maintained Menhir, an LR parser generator for OCaml. Yann Régis-Gianas develops the “Hacking Dojo”, a web platform to automatically grade programming exercises. The platform is now used in several courses of the University Paris Diderot. He gets help from the internship of Alexandre Ly, a master student of the Paris Diderot University. In collaboration with Beta Ziliani (LIIS, Cordoba, Argentine), Yann Régis-Gianas, Béatrice Carré and Jacques-Pascal Deplaix develop MetaCoq, an extension of Coq to use Coq as a metalanguage for itself.

POLSYS Project-Team

5. New Software and Platforms

5.1. FGb

FUNCTIONAL DESCRIPTION

FGb is a powerful software for computing Groebner 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://polsys.lip6.fr/~jcf/Software/FGb/index.html>

5.2. FGb Light

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

5.3. 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: Brice Boyer
- URL: <http://www-polsys.lip6.fr/~jcf/Software/index.html>

5.4. RAGLib

FUNCTIONAL DESCRIPTION

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

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

5.5. 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 ~ 5 000 lines.

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

PROSECCO Project-Team

6. New Software and Platforms

6.1. ProVerif

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

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

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

The **PROVERIF** verifier can prove the following properties:

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

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

PROVERIF is freely available on the web, at <http://proverif.inria.fr>, under the GPL license.

6.2. CryptoVerif

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

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

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

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

CRYPTOVERIF is freely available on the web, at <http://cryptoverif.inria.fr>, under the CeCILL license.

6.3. miTLS

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

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

A paper describing the miTLS library was published at IEEE S&P 2013, CRYPTO 2014, and several updates to the software were released in 2015. The software and associated research materials are available from <http://mitls.org>.

6.4. flexTLS

Participants: Karthikeyan Bhargavan [correspondant], Alfredo Pironti, Benjamin Beurdouche.

flexTLS is a TLS testing framework based on miTLS, and is released as part of the miTLS distribution. Unlike miTLS, flexTLS can be configured to run incorrect TLS clients and servers in order to test other TLS implementations. Using flexTLS we analyzed a series of open source TLS implementations and found important vulnerabilities like SKIP and FREAK. We also used flexTLS to build proof-of-concept demos for other attacks such as Logjam.

A paper describing flexTLS was published at Usenix WOOT 2015. The software and associated research materials are available from <http://mitls.org>.

6.5. F*

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

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

A detailed description of F* (circa 2011) appeared in the Journal of Functional Programming [53]. F* has evolved substantially since then. The latest version of F* is written entirely in F*, and bootstraps in OCaml and F#. It is under active development at GitHub: <https://github.com/FStarLang> and the official webpage is at <http://fstar-lang.org>.

6.6. ProScript

Participants: Nadim Kobeissi [correspondant], Karthikeyan Bhargavan, Bruno Blanchet.

Defensive JavaScript (DJS) is a subset of the JavaScript language that guarantees the behaviour of trusted scripts when loaded in an untrusted web page. Code in this subset runs independently of the rest of the JavaScript environment. When properly wrapped, DJS code can run safely on untrusted pages and keep secrets such as decryption keys. ProScript is a typed subset of JavaScript, inspired by DJS, that is focused on writing verifiable cryptographic protocol implementations. In addition to DJS typing, ProScript imposes a functional style that results in more readable and easily verifiable ProVerif models. ProScript has been used to write and verify a full implementation of the TextSecure protocol in JavaScript.

The ProScript compiler and various libraries written in ProScript will be made available from the Prosecco webpage.

SECRET Project-Team (section vide)

GAMMA3 Project-Team

4. New Software and Platforms

4.1. ABL4FLO

FUNCTIONAL DESCRIPTION

KEYWORDS: Boundary layer, Hybrid meshes

SCIENTIFIC DESCRIPTION

Automatic boundary layer mesh generation for complex geometries

FUNCTIONAL DESCRIPTION

ABL4FLO is designed to generate 3D adapted boundary layer meshes by using a cavity-based operator.

- Participant: Adrien Loseille
- Contact: Adrien Loseille
- URL: <https://www.rocq.inria.fr/gamma/Adrien.Loseille/index.php?page=softwares>

4.2. AMA4FLO

FUNCTIONAL DESCRIPTION

KEYWORDS: Anisotropic mesh adaptation, Surface and volume remeshing, Non manifold geometries

SCIENTIFIC DESCRIPTION

Robust and automatic generation of anisotropic meshes in 3D

FUNCTIONAL DESCRIPTION

AMA4FLO is designed to generate adapted meshes with respect to a provided anisotropic sizing field. The surface and the volume mesh is adapted simultaneously to guarantee that a 3D valid mesh is provided on output.

- Participant: Adrien Loseille
- Contact: Adrien Loseille
- URL: <https://www.rocq.inria.fr/gamma/Adrien.Loseille/index.php?page=softwares>

4.3. BL2D

KEYWORDS: Automatic mesher - Delaunay - Anisotropic - Planar domain

SCIENTIFIC DESCRIPTION

The meshing method is of controlled Delaunay type, isotropic or anisotropic. The internal point generation follows an advancing-front logic, and their connection is realised as in a classical Delaunay approach. Quadrilaterals are obtained by a pairing process. The direct construction of degree 2 elements has been made possible via the control of the domain boundary mesh, in order to ensure the desired compatibility.

FUNCTIONAL DESCRIPTION

Planar mesh generator (isotropic or anisotropic, adaptive).

- Participants: Houman Borouchaki and Patrick Laug
- Contact: Patrick Laug
- URL: <https://www.rocq.inria.fr/gamma/Patrick.Laug/logiciels/logiciels.html>

4.4. BL2D-ABAQ

KEYWORDS: Automatic mesher - Delaunay - Anisotropic - Planar domain - error estimation - interpolation

SCIENTIFIC DESCRIPTION

The meshing method is the same as BL2D (see above) in an adaptive process. An error estimation (*a posteriori*) of a solution at the nodes of the current mesh results in a size map. A new mesh satisfying these size specifications (made continuous) is built, and the solution is interpolated on the new mesh.

FUNCTIONAL DESCRIPTION

Planar mesh generator (isotropic or anisotropic, adaptive) for deformable domains, interacting with the ABAQUS solver.

- Participants: Houman Borouchaki, Patrick Laug and Abel Cherouat
- Contact: Patrick Laug
- URL: <https://www.rocq.inria.fr/gamma/Patrick.Laug/logiciels/logiciels.html>

4.5. BLGEOL

KEYWORDS: Automatic mesher - Hex-dominant - Geologic structures

SCIENTIFIC DESCRIPTION

The aim is to generate hex-dominant meshes of geologic structures complying with different geometric constraints: surface topography (valleys, reliefs, rivers), geologic layers and underground workings. First, a reference 2D domain is obtained by projecting all the line constraints into a horizontal plane. Different size specifications are given for rivers, outcrop lines and workings. Using an adaptive methodology, the size variation is bounded by a specified threshold in order to obtain a high quality quad-dominant mesh. Secondly, a hex-dominant mesh of the geological medium is generated by a vertical extrusion, taking into account the surfaces found (interfaces between two layers, top or bottom faces of underground workings). The generation of volume elements follows a global order established on the whole set of surfaces to ensure the conformity of the resulting mesh.

FUNCTIONAL DESCRIPTION

Hex-dominant mesher of geologic structures and storage facilities.

- Participants: Patrick Laug and Houman Borouchaki
- Contact: Patrick Laug
- URL: <https://www.rocq.inria.fr/gamma/Patrick.Laug/logiciels/logiciels.html>

4.6. BLMOL

KEYWORDS: Automatic mesher - Molecular surface

SCIENTIFIC DESCRIPTION

To model a molecular surface, each constituting atom is idealized by a simple sphere. First, a boundary representation (B-rep) of the surface is obtained, i.e. a set of patches and the topological relations between them. Second, an appropriate parameterization and a metric map are computed for each patch. Third, meshes of the parametric domains are generated with respect to an induced metric map, using a combined advancing-front generalized-Delaunay approach. Finally these meshes are mapped onto the entire surface.

FUNCTIONAL DESCRIPTION

Molecular surface mesher.

- Participants: Houman Borouchaki and Patrick Laug
- Contact: Patrick Laug
- URL: <https://www.rocq.inria.fr/gamma/Patrick.Laug/logiciels/logiciels.html>

4.7. BLSURF

KEYWORDS: Automatic mesher - parametric surface - CAD surface

SCIENTIFIC DESCRIPTION

An indirect method for meshing parametric surfaces conforming to a user-specifiable size map is used. First, from this size specification, a Riemannian metric is defined so that the desired mesh is one with unit length edges with respect to the related Riemannian space (the so-called ‘‘unit mesh’’). Then, based on the intrinsic properties of the surface, the Riemannian structure is induced into the parametric space. Finally, a unit mesh is generated completely inside the parametric space such that it conforms to the metric of the induced Riemannian structure. This mesh is constructed using a combined advancing-front Delaunay approach applied within a Riemannian context.

FUNCTIONAL DESCRIPTION

CAD surface mesher.

- Participants: Houman Borouchaki and Patrick Laug
- Contact: Patrick Laug
- URL: <https://www.rocq.inria.fr/gamma/Patrick.Laug/logiciels/logiciels.html>

4.8. FEFLOA-REMESH

KEYWORDS: Anisotropic mesh adaptation, Surface remeshing, Cavity-based operator

SCIENTIFIC DESCRIPTION

Automatic generation of metric-aligned and metric-orthogonal anisotropic meshes in 3D

FUNCTIONAL DESCRIPTION

FEFLOA-REMESH is intended to generate adapted 2D, surface and volume meshes by using a unique cavity-based operator. The metric-aligned or metric-orthogonal approach is used to generate high quality surface and volume meshes independently of the anisotropy involved.

- Participants: Adrien Loseille and Frédéric Alauzet
- Contact: Adrien Loseille
- URL: <https://www.rocq.inria.fr/gamma/Adrien.Loseille/index.php?page=softwares>

4.9. GAMANIC 3D

KEYWORDS: Tetrahedral mesh - Delaunay - Anisotropic size and direction control - Automatic Mesher

SCIENTIFIC DESCRIPTION

Automatic tetrahedral mesher based on an anisotropic Delaunay type point insertion method. A metric field is provided specifying the desired size (edge length) and directional properties.

FUNCTIONAL DESCRIPTION

GAMANIC3D is a volume mesher governed by a (anisotropic) size and directional specification metric field.

- Participants: Houman Borouchaki, Paul Louis George, Frederic Hecht, Jérôme Saltel, Frédéric Alauzet and Adrien Loseille
- Contact: Paul Louis George
- URL: <http://www.meshgems.com/volume-meshing.html>

4.10. GAMHIC 3D

KEYWORDS: Tetrahedral mesh - Delaunay - Isotropic size control - Automatic Mesher

SCIENTIFIC DESCRIPTION

Automatic tetrahedral mesher based on the Delaunay point insertion method. A metric field is provided specifying the desired size (edge length).

FUNCTIONAL DESCRIPTION

GAMHIC3D is a volume mesher governed by a (isotropic) size specification metric field.

- Participants: Houman Borouchaki, Paul Louis George, Frederic Hecht, $\sqrt{\hat{a}}$ ric Saltel, Frédéric Alauzet and Adrien Loseille
- Contact: Paul Louis George
- URL: <http://www.meshgems.com/volume-meshing.html>

4.11. GHS3D

KEYWORDS: Tetrahedral mesh - Delaunay - Automatic Mesher

SCIENTIFIC DESCRIPTION

Automatic tetrahedral mesher based on the Delaunay point insertion method.

FUNCTIONAL DESCRIPTION

GHS3D is an automatic volume mesher

- Participants: Paul Louis George, Houman Borouchaki, $\sqrt{\hat{a}}$ ric Saltel, Frédéric Alauzet, Adrien Loseille and Frederic Hecht
- Contact: Paul Louis George
- URL: <http://www.meshgems.com/volume-meshing.html>

4.12. HEXOTIC

KEYWORDS: Hexahedral mesh - Octree - Automatic mesher

SCIENTIFIC DESCRIPTION

Automatic full hexahedral mesher primarily based on an octree.

FUNCTIONAL DESCRIPTION

HEXOTIC is an automatic hexahedral mesher

- Contact: Loïc Maréchal
- URL: <https://www.rocq.inria.fr/gamma/gamma/Membres/CIPD/Loic.Marechal/Research/Hexotic.html>

4.13. Metrix

KEYWORD: Scientific calculation

SCIENTIFIC DESCRIPTION

Compute a metric field from a given solution field using various error estimates.

FUNCTIONAL DESCRIPTION

Metrix computes metric field from a given solution field using various error estimates. Available error estimates are feature-based and goal-oriented based error estimates for steady or unsteady fields. Metrix also performs operations on metrics: gradation, intersection, natural metric of a mesh.

- Participants: Frédéric Alauzet and Adrien Loseille
- Contact: Frédéric Alauzet
- URL: https://www.rocq.inria.fr/gamma/Frederic.Alauzet/code_eng.html

4.14. Nimbus 3D

KEYWORDS: Surface reconstruction - Point cloud

SCIENTIFIC DESCRIPTION

Given a point cloud, a surface is constructed primarily based on a Delaunay approach.

FUNCTIONAL DESCRIPTION

Nimbus3D is a surface reconstruction method piece of software

- Participants: Paul Louis George and Houman Borouchaki
- Contact: Paul Louis George
- URL: <http://www.meshgems.com/volume-meshing.html>

4.15. VIZIR

KEYWORDS: Mesh and solution visualization

SCIENTIFIC DESCRIPTION

Interactive mesh and solution visualization for linear, and high order curved elements

FUNCTIONAL DESCRIPTION

VIZIR is intended to visualize and modify interactively simplicial, hybrid and high order curved meshes.

- Participants: Julien Castelneau, Adrien Loseille and Alexis Loyer
- Contact: Adrien Loseille
- URL: <http://www-roc.inria.fr/gamma/gamma/vizir/>

4.16. Wolf

KEYWORD: Scientific calculation

SCIENTIFIC DESCRIPTION

General solver platform containing all the Wolf modules: Wolf-Bloom, Wolf-Elast, Wolf-Interpol, Wolf-MovMsh, Wolf-Nsc, Wolf-Shrimp, Wolf-Spyder and Wolf-Xfem.

FUNCTIONAL DESCRIPTION

Wolf is a general solver platform containing all the Wolf modules: Wolf-Bloom, Wolf-Elast, Wolf-Interpol, Wolf-MovMsh, Wolf-Nsc, Wolf-Shrimp, Wolf-Spyder and Wolf-Xfem.

- Participants: Frédéric Alauzet and Adrien Loseille
- Contact: Frédéric Alauzet
- URL: https://www.rocq.inria.fr/gamma/Frederic.Alauzet/code_eng.html

4.17. Wolf-Bloom

KEYWORD: Scientific calculation

SCIENTIFIC DESCRIPTION

Structured boundary layer mesh generator using a pushing approach.

FUNCTIONAL DESCRIPTION

Wolf-Bloom is a structured boundary layer mesh generator using a pushing approach. It start from an existing volume mesh and insert a structured boundary layer by pushing the volume mesh. The volume mesh deformation is solved with an elasticity analogy. Mesh-connectivity optimizations are performed to control volume mesh element quality.

- Participants: Frédéric Alauzet, Adrien Loseille and Dave Marcum
- Contact: Frédéric Alauzet
- URL: https://www.rocq.inria.fr/gamma/Frederic.Alauzet/code_eng.html

4.18. Wolf-Elast

KEYWORD: Scientific calculation

SCIENTIFIC DESCRIPTION

Linear elasticity solver using a P1 Finite-Element method.

FUNCTIONAL DESCRIPTION

Wolf-Elast is a linear elasticity solver using the P1 Finite-Element method. The Young and Poisson coefficient can be parametrized. The linear system is solved using the Conjugate Gradient method with the LUSGS preconditioner.

- Participants: Frédéric Alauzet and Adrien Loseille
- Contact: Frédéric Alauzet
- URL: https://www.rocq.inria.fr/gamma/Frederic.Alauzet/code_eng.html

4.19. Wolf-Interpol

KEYWORD: Scientific calculation

SCIENTIFIC DESCRIPTION

Software transferring scalar, vector and tensor fields from one mesh to another one.

FUNCTIONAL DESCRIPTION

Wolf-Interpol is a tool to transfer scalar, vector and tensor fields from one mesh to another one. Polynomial interpolation (from order 2 to 4) or conservative interpolation operators can be used. Wolf-Interpol also extract solutions along lines or surfaces.

- Participants: Frédéric Alauzet and Adrien Loseille
- Contact: Frédéric Alauzet
- URL: https://www.rocq.inria.fr/gamma/Frederic.Alauzet/code_eng.html

4.20. Wolf-MovMsh

KEYWORD: Scientific calculation

SCIENTIFIC DESCRIPTION

Moving mesh algorithm coupled with mesh-connectivity optimization.

FUNCTIONAL DESCRIPTION

Wolf-MovMsh is a moving mesh algorithm coupled with mesh-connectivity optimization. Mesh deformation is computed by means of a linear elasticity solver or a RBF interpolation. Smoothing and swapping mesh optimization are performed to maintain good mesh quality. It handles rigid bodies or deformable bodies, and also rigid or deformable regions of the domain.

- Participants: Frédéric Alauzet and Adrien Loseille
- Contact: Frédéric Alauzet
- URL: https://www.rocq.inria.fr/gamma/Frederic.Alauzet/code_eng.html

4.21. Wolf-Nsc

KEYWORD: Scientific calculation

SCIENTIFIC DESCRIPTION

Numerical flow solver solving the compressible Navier-Stokes equations.

FUNCTIONAL DESCRIPTION

Wolf-Nsc is numerical flow solver solving steady or unsteady turbulent compressible Euler and Navier-Stokes equations. The available turbulent models are the Spalart-Almaras and the Menter SST k-omega. A mixed finite volume - finite element numerical method is used for the discretization. Second order spatial accuracy is reached thanks to MUSCL type methods. Explicit or implicit time integration are available. It also resolved dual (adjoint) problem and compute error estimate for mesh adaptation.

- Participants: Frédéric Alauzet and Adrien Loseille
- Contact: Frédéric Alauzet
- URL: https://www.rocq.inria.fr/gamma/Frederic.Alauzet/code_eng.html

4.22. Wolf-Shrimp

KEYWORD: Scientific calculation

SCIENTIFIC DESCRIPTION

Mesh partitioner for parallel mesh generation and parallel computation.

FUNCTIONAL DESCRIPTION

Wolf-Shrimp is a generic mesh partitioner for parallel mesh generation and parallel computation. It can partition planar, surface (manifold and non manifold), and volume domain. Several partitioning methods are available: Hilbert-based, BFS, BFS with restart. It can work with or without weight function and can correct the partitions to have only one connected component.

- Participants: Frédéric Alauzet and Adrien Loseille
- Contact: Frédéric Alauzet
- URL: https://www.rocq.inria.fr/gamma/Frederic.Alauzet/code_eng.html

4.23. Wolf-Spyder

KEYWORD: Scientific calculation

SCIENTIFIC DESCRIPTION

Metric-based mesh quality optimizer using vertex smoothing and edge/face swapping.

FUNCTIONAL DESCRIPTION

Wolf-Spyder is a metric-based mesh quality optimizer using vertex smoothing and edge/face swapping.

- Participants: Frédéric Alauzet and Adrien Loseille
- Contact: Frédéric Alauzet
- URL: https://www.rocq.inria.fr/gamma/Frederic.Alauzet/code_eng.html

4.24. Wolf-Xfem

KEYWORD: Scientific calculation

SCIENTIFIC DESCRIPTION

Tool providing the mesh of the intersection between a surface mesh and a volume mesh.

FUNCTIONAL DESCRIPTION

Wolf-Xfem is a tool providing the mesh of the intersection between a surface mesh and a volume mesh.

- Participants: Frédéric Alauzet
- Contact: Frédéric Alauzet
- URL: https://www.rocq.inria.fr/gamma/Frederic.Alauzet/code_eng.html

MATERIALS Project-Team

5. New Software and Platforms

5.1. SIMOL

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

FUNCTIONAL DESCRIPTION

SIMOL (SIMulation of MOlecular systems) is a software written in C++. It is a research code aimed at testing new ideas and algorithms, and provides a unified development platform for the members of the project-team. It is composed of three parts: a common core of input/output functions, linear algebra, random number generators, etc; and two specific applicative branches: one for computational statistical physics and one for quantum chemistry. The methods implemented for computational statistical physics are based on discretizations of ergodic stochastic differential equations such as the Langevin dynamics and its overdamped limit. The systems that can be simulated range from a single isolated particle to Lennard-Jones fluids. For quantum chemistry, the building block is the Hartree-Fock model, solved via fixed-point iterations; and various refinements including greedy methods. A first release should be available in Spring 2016.

- Contact: Cédric Doucet

MATHRISK Project-Team

6. New Software and Platforms

6.1. PREMIA

KEYWORDS: Computational finance - Option pricing

SCIENTIFIC DESCRIPTION

Premia is a software designed for option pricing, hedging and financial model calibration. It is provided with its C/C++ source code and an extensive scientific documentation. The Premia project keeps track of the most recent advances in the field of computational finance in a well-documented way. It focuses on the implementation of numerical analysis techniques for both probabilistic and deterministic numerical methods. An important feature of the platform Premia is the detailed documentation which provides extended references in option pricing.

Premia is thus a powerful tool to assist Research & Development professional teams in their day-to-day duty. It is also a useful support for academics who wish to perform tests on new algorithms or pricing methods without starting from scratch.

Besides being a single entry point for accessible overviews and basic implementations of various numerical methods, the aim of the Premia project is: 1 - to be a powerful testing platform for comparing different numerical methods between each other, 2 - to build a link between professional financial teams and academic researchers, 3 - to provide a useful teaching support for Master and PhD students in mathematical finance.

FUNCTIONAL DESCRIPTION

- Participants: Mathrisk project team and contributors
- Partners: Inria - Ecole des Ponts ParisTech - Université Paris-Est - Consortium Premia
- Contact: Agnès Sulem
- URL: <http://www.premia.fr>
- AMS: 91B28;65Cxx;65Fxx;65Lxx;65Pxx
- License: Licence Propriétaire (genuine license for the Consortium Premia)
- Type of human computer interaction: Console, interface in Nsp, Web interface
- OS/Middleware: Linux, Mac OS X, Windows
- APP: The development of Premia started in 1999 and 16 are released up to now and registered at the APP agency. Premia 16 has been registered on 03/03/2015 under the number IDDN.FR.001.190010.013.S.C.2001.000.31000
- Programming language: C/C++ librairie Gtk
- Documentation: the PNL library is interfaced via doxygen
- Size of the software: 280580 lines for the Src part only, that is 11 Mbyte of code, 130400 lines for PNL, 105 Mbyte of PDF files of documentation.
- interfaces : Nsp for Windows/Linux/Mac, Excel, binding Python, and a Web interface.
- Publications: [12], [61], [69], [77], [80], [49], [59].

6.1.1. Content of Premia

Premia contains various numerical algorithms (Finite-differences, trees and Monte-Carlo) for pricing vanilla and exotic options on equities, interest rate, credit and energy derivatives.

1. Equity derivatives:

The following models are considered:

Black-Scholes model (up to dimension 10), stochastic volatility models (Hull-White, Heston, Fouque-Papanicolaou-Sircar), models with jumps (Merton, Kou, Tempered stable processes, Variance gamma, Normal inverse Gaussian), Bates model.

For high dimensional American options, Premia provides the most recent Monte-Carlo algorithms: Longstaff-Schwartz, Barraquand-Martineau, Tsitsklis-Van Roy, Broadie-Glassermann, quantization methods and Malliavin calculus based methods.

Dynamic Hedging for Black-Scholes and jump models is available.

Calibration algorithms for some models with jumps, local volatility and stochastic volatility are implemented.

2. Interest rate derivatives

The following models are considered:

HJM and Libor Market Models (LMM): affine models, Hull-White, CIR++, Black-Karasinsky, Squared-Gaussian, Li-Ritchken-Sankarasubramanian, Bhar-Chiarella, Jump diffusion LMM, Markov functional LMM, LMM with stochastic volatility.

Premia provides a calibration toolbox for Libor Market model using a database of swaptions and caps implied volatilities.

3. Credit derivatives: Credit default swaps (CDS), Collateralized debt obligations (CDO)

Reduced form models and copula models are considered.

Premia provides a toolbox for pricing CDOs using the most recent algorithms (Hull-White, Laurent-Gregory, El Karoui-Jiao, Yang-Zhang, Schönbucher)

4. Hybrid products

A PDE solver for pricing derivatives on hybrid products like options on inflation and interest or change rates is implemented.

5. Energy derivatives: swing options

Mean reverting and jump models are considered.

Premia provides a toolbox for pricing swing options using finite differences, Monte-Carlo Malliavin-based approach and quantization algorithms.

6.1.2. Premia design

To facilitate contributions, a standardized numerical library (PNL) has been developed by J. Lelong under the LGPL since 2009, which offers a wide variety of high level numerical methods for dealing with linear algebra, numerical integration, optimization, random number generators, Fourier and Laplace transforms, and much more. Everyone who wishes to contribute is encouraged to base its code on PNL and providing such a unified numerical library has considerably eased the development of new algorithms which have become over the releases more and more sophisticated. J. Ph Chancelier, B. Lapeyre and J. Lelong are using Premia and Nsp for Constructing a Risk Management Benchmark for Testing Parallel Architecture [59].

Development of the PNL in 2015 (J. Lelong) . Release 1.70 and 1.71, PNL Library (<http://pnl.gforge.inria.fr>).

1. Release 1.72. of the *PNL* library (<http://pnl.gforge.inria.fr>).
 1. Addition of a CMake module to include the library in other projects.
 2. Improvement of the `pnl_basis` module.
 3. Addition of the non central chi squared distribution to the random number generation toolbox.
 4. Addition of new functions in the linear algebra toolbox to build views.

6.1.3. Algorithms implemented in Premia in 2015

Premia 17 has been delivered to the consortium members in March 2015.

It contains the following new algorithms:

6.1.3.1. Commodities, FX, Insurance, Credit Risk

- Variables Annuities GLWB pricing in the Heston and Black-Scholes/Hull-White models with finite difference techniques.
- Variables Annuities GMAB, GMDB, GMMB pricing with Fourier-cosine techniques.
- A numerical scheme for the impulse control formulation for pricing variable annuities with a Guaranteed Minimum Withdrawal Benefit (GMWB) Z.Chen P.Forsyth
Numerische Mathematik 109, 2008
- Managing Gap Risks in iCPPI for life insurance companies: A risk/return/cost analysis. A.Kalife S.Mouti L.Goudenege
Insurance Markets and Companies: Analyses and Actuarial Computations, Issue 2 2014
- Simulating CVA on American Options. L. Abbas Turki, M.Mikou

6.1.3.2. Equity Derivatives

- Being particular about calibration. J.Guyon and P. Henry-Labordère.
Risk magazine, Jan 2012.
- The Heston Stochastic-Local Volatility Model: Efficient Monte Carlo Simulation. A.W. van der Stoepb, L. A. Grzelakb, C. W. Oosterlee
International Journal of Theoretical and Applied Finance, to appear
- On the Heston model with stochastic interest rates. L. Grzelak C.W.Oosterlee
SIAM J. Fin. Math. 2,2011.
- Alternating direction implicit finite difference schemes for the Heston Hull-White partial differential equation.
The Journal of Computational Finance Volume 16/Number 1, Fall 2012
- Pricing American options in the Heston Hull-White and Hull-White2d Models: an hybrid tree-finite difference approach. M.Briani, L.Caramellino, A.Zanette
- Efficient pricing of Asian options under Lévy processes based on Fourier cosine expansions. Part I: European-style products. B.Zhang C.W.Oosterlee.
SIAM J. Financial Math., 4(1)
- Low-bias simulation scheme for the Heston model by Inverse Gaussian approximation. S. T. Tse J. W. L. Wan.
Quantitative Finance, Volume 13, Issue 6, 2013
- Simple Simulation Scheme for CIR and Wishart Processes P. Baldi, C.Pisani
International Journal of Theoretical and Applied Finance Vol. 16, No. 08, 2013
- Importance sampling for jump processes and applications to finance. L. Badouraly Kassim, J. Lelong and I. Loumrhari.
Journal of Computational Finance, to appear
- A Wiener-Hopf Monte Carlo simulation technique for Lévy process. A. Kuznetsov, A.E.Kyprianou J. C. Pardo and K. van Schaik.
The Annals of Applied Probability, Volume 21, Number 6, 2011.
- A Wiener-Hopf Monte Carlo simulation approach for pricing path-dependent options under Lévy process. O. Kudryavtsev
Preprint.
- An Efficient Binomial Lattice Method for Step Double Barrier Options. E.Appolloni, M.Gaudenzi A.Zanette.
International Journal of Applied and Theoretical Finance Vol.17, Issue No. 6, 2014.

The algorithms

- “Pricing American-Style Options by Monte Carlo Simulation : Alternatives to Ordinary Least Squares” by Stathis Tompaidis and Chunyu Yang
- “Value Function Approximation or Stopping Time Approximation : A comparison of Two Recent Numerical Methods for American Option Pricing using Simulation and Regression” by Lars Stentoft

implemented in 2015 by Céline Labart will be included in the following release.

Moreover, Jérôme Lelong has performed the following tasks:

1. Add an importance sampling based code for jump diffusion models.
2. Improve the internal enumeration mechanism (PremiaEnum).
3. Update gnuplot files generation for reports.
4. Everyday maintenance to fix various bugs.
5. Clean the generation of the documentation process.

MOKAPLAN Project-Team

6. New Software and Platforms

6.1. 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/>

6.2. 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. MOKABAJOURE project aims to develop, using algorithms invented especially at Inria and LJK, a reflector design software more efficient than geometrical methods used so far.

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

6.3. Entropic OT

FUNCTIONAL DESCRIPTION

We design a software to compute fast approximation of optimal transport (and related problems such as barycenters) on geometric domains (either regular Euclidean grid or triangulated meshes). This numerical scheme relies on two key ideas: entropic regularization of the initial linear problem [3] and fast approximate convolution on geometric domains [22] This algorithm is both extremely fast and highly parallelizable, being able to take advantage of GPU computational architectures.

- Gabriel Peyré, Jean-David Benamou, Guillaume Carlier, Marco Cuturi (Kyoto), Justin Solomon.
- Contact: Gabriel Peyré
- URL: <https://github.com/gpeyre/2015-SIGGRAPH-convolutional-ot>

6.4. Jupyter Notebook

FUNCTIONAL DESCRIPTION

Several codes developed by the team are available on an online Jupyter Notebook (Julia and Python) In particular the Semi Discrete Principal Agent Code and also a new Monge-Amère second boundary value problem Finite Difference code.

- Simon Legrand, Xavier Dupuis, Vincent Duval, Jean-David Benamou.
- Contact: Simon Legrand
- URL: <https://mathmarx.paris.inria.fr:8080>

QUANTIC Project-Team (section vide)

SIERRA Project-Team

6. New Software and Platforms

6.1. DICA: Moment Matching for Latent Dirichlet Allocation (LDA) and Discrete Independent Component Analysis (DICA)

The DICA package contains Matlab and C++ (via Matlab mex files) implementations of estimation in the LDA and closely related DICA models [21].

The implementation consists of two parts. One part contains the efficient implementation for construction of the moment/cumulant tensors, while the other part contains implementations of several so called joint diagonalization type algorithms used for matching the tensors. Any tensor type (see below) can be arbitrarily combined with one of the diagonalization algorithms (see below) leading, in total, to 6 algorithms.

Two types of tensors are considered: (a) the LDA moments and (b) the DICA cumulants. The diagonalization algorithms include: (a) the orthogonal joint diagonalization algorithm based on iterative Jacobi rotations, (b) the spectral algorithm based on two eigen decompositions, and (c) the tensor power method.

- Contact: Anastasia Podosinnikova
- URL: <https://github.com/anastasia-podosinnikova/dica>

6.2. LinearFW: Implementation of linearly convergent versions of Frank-Wolfe

This is the code to reproduce all the experiments in the NIPS 2015 paper: "On the Global Linear Convergence of Frank-Wolfe Optimization Variants" by Simon Lacoste-Julien and Martin Jaggi [17], which covers the global linear convergence rate of Frank-Wolfe optimization variants for problems described as in Eq. (1) in the paper. It contains the implementation of Frank-Wolfe, away-steps Frank-Wolfe and pairwise Frank-Wolfe on two applications.

- Contact: Simon Lacoste-Julien
- URL: <https://github.com/Simon-Lacoste-Julien/linearFW>

6.3. CNN-Head-Detection: Context-aware CNNs for person head detection

This is the code for the ICCV 2015 paper "Context-aware CNNs for person head detection" [23]. Person detection is a key problem for many computer vision tasks. While face detection has reached maturity, detecting people under a full variation of camera view-points, human poses, lighting conditions and occlusions is still a difficult challenge. In this work we focus on detecting human heads in natural scenes. Starting from the recent local R-CNN object detector, we extend it with two types of contextual cues. First, we leverage person-scene relations and propose a Global CNN model trained to predict positions and scales of heads directly from the full image. Second, we explicitly model pairwise relations among objects and train a Pairwise CNN model using a structured-output surrogate loss. The Local, Global and Pairwise models are combined into a joint CNN framework. To train and test our full model, we introduce a large dataset composed of 369,846 human heads annotated in 224,740 movie frames. We evaluate our method and demonstrate improvements of person head detection against several recent baselines in three datasets. We also show improvements of the detection speed provided by our model.

- Contact: Anton Osokin
- URL: https://github.com/aosokin/cnn_head_detection

ANGE Project-Team

6. New Software and Platforms

6.1. Freshkiss3D (FREe Surface Hydrodynamics using KInetic SchemeS)

FUNCTIONAL DESCRIPTION

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

- Participants: Jacques Sainte-Marie, Emmanuel Audusse, Marie-Odile Bristeau, Raouf Hamouda, David Froger and Anne-Céline Boulanger
- Partners: UPMC - CEREMA
- Contact: Jacques Sainte-Marie

For a list of recent developments, refer to §7.5.1 .

6.2. TSUNAMATHS

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: Jacques Sainte-Marie, Emmanuel Audusse and Raouf Hamouda
- Contact: Jacques Sainte-Marie
- URL: http://ange.raoufhamouda.com/tsunami/en_motivation.htm

ARAMIS Project-Team

6. New Software and Platforms

6.1. Brain Networks Toolbox

KEYWORDS: Neuroimaging - Medical imaging

FUNCTIONAL DESCRIPTION

Brain Networks Toolbox is a collection of Matlab routines developed to quantify topological metrics of complex brain networks.

- Participants: Mario Chavez and Fabrizio De Vico Fallani
- Contact: Mario Chavez
- URL: <https://sites.google.com/site/fr2eborn/download>

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

6.3. SACHA

Segmentation Automatisée Compétitive de l'Hippocampe et de l'Amygdale

KEYWORDS: Neuroimaging - 3D - Hippocampus - Amygdala - Brain scan - Medical imaging

SCIENTIFIC DESCRIPTION

The current stable version is fully automatic and focused on cross-sectional segmentation. The software can be used both as a command-line program or through a graphical user interface (GUI). The core of the program is coded in C++. It has a dependency to the AIMS library and preprocessing steps rely on processes in Matlab from SPM. The GUI is coded in Python and is based on BrainVISA.

FUNCTIONAL DESCRIPTION

SACHA is a software for the fully automatic segmentation of the hippocampus and the amygdala from MRI 3D T1 brain scans. It has been validated in various populations including healthy controls and patients with Alzheimer's disease, epilepsy and depression. It has been successfully applied to over 3,000 subjects, both controls, from adolescents to elderly subjects, and patients with different types of pathologies.

- Participants: Marie Chupin and Ludovic Fillon
- Contact: Marie Chupin
- URL: <http://www.brainvisa.info>

6.4. WHASA

White matter Hyperintensity Automatic Segmentation Algorithm

KEYWORDS: Health - Neuroimaging - Biomedical imaging

SCIENTIFIC DESCRIPTION

The current stable version is fully automatic and focused on cross-sectional segmentation. The software can be used both as a Matlab command-line or through a graphical user interface (GUI). The core of the program is coded in Matlab. It has a dependency to the SPM environment. The GUI is coded in Python and is based on BrainVISA.

FUNCTIONAL DESCRIPTION

WHASA (“White matter Hyperintensity Automatic Segmentation Algorithm”) is a software for the fully automatic segmentation of age-related white matter hyperintensities from MRI FLAIR and 3D T1 brain scans. It has been validated on a population showing a wide range of lesion load, and is being further evaluated on elderly subjects with few clinical abnormalities and with different acquisition characteristics.

- Participants: Marie Chupin, Ludovic Fillon and Thomas Samaille
- Contact: Marie Chupin
- URL: <http://www.brainvisa.info/>

6.5. qualiCATI

KEYWORDS: Health - Neuroimaging - Medical imaging

SCIENTIFIC DESCRIPTION

QualiCATI requires training for the visual parts, and is closely linked with a team of clinical research assistants. It has been used to analyse about 5000 subjects from about 15 multi centre research projects initiated before or after the CATI started. Other modules will be added in the future to embed new aspects of the MRI protocol proposed by the CATI. The Aramis team is in charge of the second and third modules and jointly in charge of the first module. The software is centered on a graphical user interface (GUI). The whole program is coded in Python within the pyPTK environment. It has dependencies to SPM and brainVISA environments as well as specific tools for DICOM management.

FUNCTIONAL DESCRIPTION

qualiCATI is a software designed for comprehensive quality control of multimodal MRI data acquisition in large multicentre clinical studies. The software is built as a platform receiving several modules, developed by several CATI engineers. The first module is dedicated to acquisition requirement checking and conversion to nifti format. The second module aims at making 3DT1 acquisition quality check more systematic, and relies both on visual inspection and quantitative indices. The third module allows a simultaneous evaluation of the clinical part of the CATI acquisition protocol. The fourth module embeds automatic indices to evaluate resting state fMRI acquisition. The fifth module is dedicated to first preprocessings and quality indices for dMRI. The sixth module is dedicated to qMRI, with visual and automated quality control together with preprocessings. The last module is dedicated to data and project management.

- Participants: Marie Chupin and Hugo Dary
- Contact: Marie Chupin
- URL: <http://www.fil.ion.ucl.ac.uk/spm/>

CLIME Project-Team

5. New Software and Platforms

5.1. Data assimilation library: Verdandi

Participants: Nicolas Claude, Vivien Mallet, Gautier Bureau [M3DISIM], Dominique Chapelle [M3DISIM], Sébastien Gilles [M3DISIM], Philippe Moireau [M3DISIM].

The leading idea is to develop a data assimilation library (see the web site <http://verdandi.sourceforge.net/>) intended to be generic, at least for high-dimensional systems. Data assimilation methods, developed and used by several teams at Inria, are generic enough to be coded independently of the system to which they are applied. Therefore these methods can be put together in a library aiming at:

- making easier the application of methods to a great number of problems,
- making the developments perennial and sharing them,
- improving the broadcast of data assimilation works.

An object-oriented language (C++) has been chosen for the core of the library. A high-level interface to Python is automatically built. The design study raised many questions, related to high dimensional scientific computing, the limits of the object contents and their interfaces. The chosen object-oriented design is mainly based on three class hierarchies: the methods, the observation managers and the models. Several base facilities have also been included, for message exchanges between the objects, output saves, logging capabilities, computing with sparse matrices.

In 2015, version 1.7 was released. We introduced an implementation of nudging. A level-set observation manager was added. Further tests were included. We added the option to build Verdandi as a library.

5.2. Image processing library: Heimdali

Participants: Isabelle Herlin, Dominique Béréziat [UPMC], David Froger [SED].

The initial aim of the image processing library Heimdali was to develop a library based on standard and open source tools, and mostly dedicated to satellite acquisitions.

The leading idea of the library is to allow the following issues:

- making easier the sharing and development of image assimilation softwares. For that purpose, the installation is easily achieved with the package manager Conda.
- developing generic tools for image processing and image assimilation based on ITK (Insight Segmentation and Registration Toolkit <http://www.itk.org>).
- in reverse, providing tools to ITK and contribute to the ITK community.

The main components of Heimdali concern:

- the pre/post processing of image sequences,
- the image assimilation with numerical models,
- the visualization of image sequences.

In 2015, additional functions were introduced in the library in order to allow more pre/post processing tools.

5.3. Polyphemus

Participants: Vivien Mallet, Sylvain Doré [CEREA], Karine Sartelet [CEREA], Yelva Roustan [CEREA].

Polyphemus (see the web site <http://cerea.enpc.fr/polyphemus/>) is a modeling system for air quality. As such, it is designed to yield up-to-date simulations in a reliable framework: data assimilation, ensemble forecasts and daily forecasts. Its completeness makes it suitable for use in many applications: photochemistry, aerosols, radionuclides, etc. It is able to handle simulations from local to continental scales, with several physical models. It is divided into three main parts:

- libraries that gather data processing tools (SeldonData), physical parameterizations (AtmoData) and post-processing abilities (AtmoPy);
- programs for physical pre-processing and chemistry-transport models (Polair3D, Castor, two Gaussian models, a Lagrangian model);
- model drivers and observation modules for model coupling, ensemble forecasting and data assimilation.

Fig. 1 depicts a typical result produced by Polyphemus.

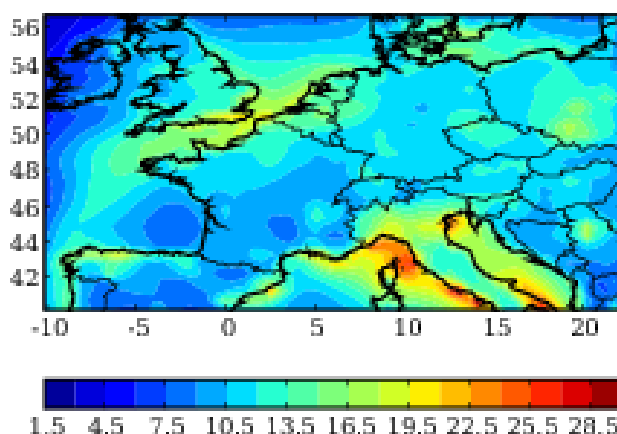


Figure 1. Map of the relative standard deviation (or spread, %) of an ensemble built with Polyphemus (ozone simulations, $\mu\text{g m}^{-3}$). The standard deviations are averaged over the summer of 2001. They provide an estimation of the simulation uncertainties.

Clime is involved in the overall design of the system and in the development of advanced methods in model coupling, data assimilation and uncertainty quantification (through model drivers and post-processing).

In 2015, version 1.9 was released, with all previous developments on the interface between Verdandi and Polyphemus. The other improvements were developed by CEREAs on aerosol modeling.

LIFEWARE Project-Team

6. New Software and Platforms

6.1. BIOCHAM

The Biochemical Abstract Machine

KEYWORDS: Systems Biology - Bioinformatics

FUNCTIONAL DESCRIPTION

The Biochemical Abstract Machine (BIOCHAM) is a software environment for modeling and analyzing biochemical reaction systems, making simulations, performing static analyses, specifying behaviors in temporal logic.

- Participants: François Fages, François-Marie Floch, Thierry Martinez, Sylvain Soliman
- Contact: François Fages
- URL: <http://lifeware.inria.fr/biocham/>

6.2. BIOCHAM-WEB

KEYWORDS: Systems Biology - Bioinformatics

FUNCTIONAL DESCRIPTION

BIOCHAM-web is a web service which makes it possible to try BIOCHAM on line without any installation, through a spreadsheet.

- Participants: François Fages, François-Marie Floch and Thierry Martinez
- Contact: François Fages
- URL: <http://lifeware.inria.fr/biocham/online/>

6.3. CellStar

KEYWORDS: Systems Biology - Bioinformatics

FUNCTIONAL DESCRIPTION

In close collaboration with Kirill Batmanov, Cédric Lhoussaine and Cristian Versari from the LIFL (CNRS/Lille Univ) and with Pascal Hersen (MSC lab; CNRS/Paris 7), we developed CellStar, a tool-chain for image processing and analysis dedicated to segmentation and tracking of yeast cells in brightfield time-lapse microscopy movies. To estimate algorithm quality we developed a benchmark made of manually-verified images illustrating various situations. On this benchmark, CellStar outperformed 5 other state-of-the-art methods. The tool-chain is implemented in MATLAB and is provided together with the Python Yeast Image Toolkit benchmark tool.

- Participants: Pascal Hersen, Grégory Batt, Artémis Llamosi
- Contact: Grégory Batt
- URL: <http://yeast-image-toolkit.biosim.eu/pmwiki.php>

6.4. ClpZinc

FUNCTIONAL DESCRIPTION

CLP2Zinc is a rule-based modeling language for constraint programming. It extends the MiniZinc modeling language with Horn clauses which can be used to express search strategies as constraints in the model. This system is developed in the framework of the ANR Net-WMS-2 project and is a follow-up of the Rules2CP modeling language.

- Participants: Thierry Martinez, François Fages, Philippe Morignot and Sylvain Soliman
- Contact: Thierry Martinez
- URL: <http://lifeware.inria.fr/~tmartine/clp2zinc/>

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 the 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 has been little recognised thus far. 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, allowing 3D reconstructions of liver lobules down to a resolution of less than a micrometer. The software has been made available to the public by publication in ref. [14], together with a new surface reconstruction algorithm based on the morphological Watershed algorithm. We validated that this algorithm allows reconstruction of cell shapes from nucleus and blood microvessel information, and demonstrated that it allows a reliable estimate of liver lobules, the smallest repetitive functional and micro-anatomical liver units, that besides in pig are not anatomically separated.

A separate 2D version of it (TI-Quant-BF-2D) has been used to analyse the invasion pattern of non-small cell lung cancer (NSCLC) cells *in vitro* [24] (see below).

- Contact: Dirk Drasdo
- URL: <http://www.msystbio.com>

6.2. TiSim

Tissue Simulator

KEYWORDS: Systems Biology - Bioinformatics - Biology - Physiology

FUNCTIONAL DESCRIPTION

We advanced the complementary software TiSim (Tissue Simulator) that will soon be provided. TiSim allows agent-based simulations of multicellular systems and can be directly used by processed image data provided by TiQuant.

The software has been tested over the whole year including almost all group members to prepare it for submission and will present a number of application example to introduce a potential user into the software. These will be monolayer and multicellular spheroid growth, a multiscale modeling example and liver regeneration.

- Contact: Dirk Drasdo

MYCENAE Project-Team (section vide)

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

6.2. LIFE-V

KEYWORD: Finite element modelling

FUNCTIONAL DESCRIPTION

LiFE-V is a finite element library providing implementations of state of the art mathematical and numerical methods. It serves both as a research and production library. LiFE-V is the joint collaboration between three institutions: Ecole Polytechnique Fédérale de Lausanne (CMCS) in Switzerland, Politecnico di Milano (MOX) in Italy and Inria (REO) in France. It is a free software under LGPL license.

- Participants: Jean-Frédéric Gerbeau and Miguel Angel Fernandez Varela
- Partners: EPFL - Ecole Polytechnique Fédérale de Lausanne - MOX Politecnico di Milano
- Contact: Miguel Angel Fernández Varela
- URL: <http://www.lifev.org/>

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

SERENA Team

6. New Software and Platforms

6.1. New Software

6.1.1. *FreeFem++ a posteriori package*

Participants: Martin Vohralík, Zuqi Tang.

The scientific calculation code FreeFem++ is an example of a complex software numerical simulation tool relying on traditional matching triangular meshes. It encompasses all specification of the problem, the choice and implementation of the numerical method, the choice and implementation of the linearization method (nonlinear solver), and the choice and implementation of the method of solution of the associated linear systems (linear solver). In the post-doc stay of Z. Tang, we have integrated here some recent advances of the theory of a posteriori error estimation and of adaptive algorithms. In particular, (local) adaptive stopping criteria for the linear and nonlinear solvers have been implemented.

Version 3.42

Programming language: C++

<http://www.freefem.org/ff++/>

<https://who.rocq.inria.fr/Zuqi.Tang/freefem++.html>

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

6.3. DPREPack

KEYWORD: Large scale

FUNCTIONAL DESCRIPTION

This library solves linear systems on parallel computers from PCs based on multicore processors to large scale computers. It implements recent parallel algorithms issued from domain decomposition methods and parallel approximate factorizations.

- Partners: CNRS - UPMC
- Contact: Laura Grigori
- URL: <https://team.inria.fr/alpines/>

Submodules:

- Sparse Toolbox
KEYWORDS: Preconditioner - Interactive method - Linear system
 - Participants: Laura Grigori and Rémi Lacroix
 - Contact: Laura Grigori
 - not yet publicly available

- Block Filtering Decomposition preconditioner
KEYWORDS: Preconditioner - Linear system
FUNCTIONAL DESCRIPTION

Iterative methods are used in many industrial and academic applications to solve large sparse linear systems of equations, and preconditioning these methods is often necessary to accelerate their convergence. Several highly used preconditioners as incomplete LU factorizations are known to have scalability problems, often due to the presence of several low frequency modes that hinder the convergence of the iterative method. To address this problem, we work on filtering preconditioners. A judicious choice of the filtering vector allows to alleviate the effect of low frequency modes, and can accelerate significantly the convergence of the iterative method.

- Participants: Laura Grigori, Rémi Lacroix and Frédéric Nataf
 - Partners: CNRS - UPMC
 - Contact: Laura Grigori
 - not yet publicly available
- LORASC preconditioner
KEYWORD: Preconditioner
 - Participants: Laura Grigori and Rémi Lacroix
 - Contact: Laura Grigori
 - URL: not yet publicly available
 - NFF Nested Filtering Factorization
KEYWORDS: Preconditioner - Interactive method - Linear system
 - Participants: Laura Grigori, Frédéric Nataf and Long Qu
 - Partners: UPMC - Université Paris-Sud
 - Contact: Laura Grigori
 - not yet publicly available

DYOGENE Project-Team

6. New Software and Platforms

6.1. CloNES

CLOsed queueing Networks Exact Sampling

FUNCTIONAL DESCRIPTION

Clones is a Matlab toolbox for exact sampling of closed queueing networks.

- Participant: Christelle Rovetta
- Contact: Christelle Rovetta
- URL: <http://www.di.ens.fr/~rovetta/Clones/index.html>

EVA Team

6. New Software and Platforms

6.1. OpenWSN (Software)

Participants: Thomas Watteyne, Tengfei Chang, Malisa Vucinic, Jonathan Muñoz.

OpenWSN (<http://www.openwsn.org/>) 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.

The OpenWSN ADT started in 2015, with Research Engineer Tengfei Chang who joined the EVA team.

Highlights for 2015:

- Development:
 - Moving the project from UC Berkeley to Inria (**Thomas Watteyne**)
 - Implementation of a layer-2 security based on AES-128 and CCM* (Malisa Vucinic)
 - Implementation of draft-ietf-6tisch-minimal (Tengfei Chang)
 - Implementation of draft-dujovne-6tisch-6top-sf0 (Tengfei Chang)
 - Implementation of draft-wang-6tisch-6top-sublayer (Tengfei Chang)
 - Creation of “Golden Image” used as a reference during interoperability testing (Tengfei Chang)
- Recognition:
 - OpenWSN was selected by ETSI as the reference implementation for IETF 6TiSCH-related standards. It is therefore the base for the ETSI’s Golden Device for 6TiSCH standards, including IEEE802.15.4e TSCH, 6LoWPAN and RPL.
- Events:
 - **Tutorial**
OpenWSN & OpenMote: Hands-on Tutorial on Open Source Industrial IoT. Thomas Watteyne, Xavier Vilajosana, Pere Tuset. IEEE Global Telecommunications Conference (GLOBECOM), San Diego, CA, USA, 6-10 December 2015.
 - **Tutorial**
OpenWSN Tutorial [presented by Xavi Vilsajosana] Workshop Internet Of Things / Equipex FIT IoT-LAB, Lille, France, 15 October 2015.
 - **Hackathon**
OpenWSN/6TiSCH Hackathon, Czech Republic, 19 July 2015.
 - **Interop event**
First ETSI 6TiSCH plugtest (interop event) in Prague, Czech Republic, 17-18 July 2015.

6.2. OPERA and OCARI (Software)

Participants: Erwan Livolant, Pascale Minet.

The OPERA software was developed by the Hipercom2 team in the OCARI project (see <https://ocari.org/>). It includes EOLSR, an energy efficient routing protocol and OSERENA, a coloring algorithm optimized for dense wireless networks. It was registered by the APP. In 2013, OPERA has been made available for download as an open software from the InriaGForge site: https://gforge.inria.fr/scm/?group_id=4665

In 2014, OPERA has been ported on a more powerful platform based on the Atmel transceiver AT86RF233 and on a 32 bits microcontroller Cortex M3. More details and documentation about this software are available in the website made by the Eva team: <http://opera.gforge.inria.fr/index.html>

In 2015, Erwan Livolant maintained the code and corrected some bugs.

6.3. CONNEXION (Software)

Participants: Ines Khoufi, Pascale Minet, Erwan Livolant.

These developments are part of the CONNEXION project. In 2015, Ines khoufi developed two softwares:

- a distributed algorithm, called OA-DVFA, to deploy autonomous and mobile sensor nodes to ensure full coverage of a 2-D area with unknown obstacles. It is based on virtual forces and virtual grid.
- an algorithm, called MRDS, to compute the tours of mobile robots in charge of placing static sensor nodes at the positions given (e.g. points of interest). This is a multi-objective optimization problem: to minimize the deployment duration, to balance the durations of robots tours and to minimize the number of robots used. A genetic heuristic is used to solve this problem.

With regard to the wireless sensor network OCARI, in 2015 we designed and developed the mobility support for OCARI. The solution proposed to support mobility in the OCARI network is simple and limits the overhead induced by mobile nodes. This mobility support is designed to be efficient in its use of resources (e.g. bandwidth, energy, memory). The properties of energy efficiency, determinism, latency and robustness provided by OCARI to static wireless sensor nodes are ensured. In the absence of mobile nodes, the OCARI network behaves exactly as without mobility support and exhibits exactly the same performances. Similarly, the overhead induced by mobile sinks is paid only if mobile sinks are present. Data gathering by the static sink, being the most important objective of the OCARI network from the application point of view, its performances are not altered by mobility support. Data gathering by a mobile sink is a new functionality provided by mobility support. This mobility support has been demonstrated with a mobile robot embedding a sensor node and transferring its data to the static sink via router nodes that depend on the position of the robot.

With Telecom ParisTech, the integration of OCARI in a Service-Oriented Architecture using the OPC-UA/ROSA middleware went on. More precisely, we developed the creation of services corresponding to newly available physical sensor measurements and the suppression of services that are no longer available.

Erwan Livolant developed an OCARI frame dissector plugin for Wireshark (<https://www.wireshark.org>) available from the Git repository at OCARI website (<https://www.ocari.org/gitlab/tools/wireshark.git>). This tool displays the contents of the packets sniffed for the MAC, the NWK and the Application layers, taking into account the specificities of OCARI.

6.4. SAHARA (Software)

Participants: Erwan Livolant, Pascale Minet.

Erwan Livolant developed a SAHARA frame dissector plugin for Wireshark (<https://www.wireshark.org>). This tool displays the contents of the packets sniffed for the MAC and the NWK layers, taking into account the specificities of the SAHARA project.

6.5. FIT IoT-LAB (Platform)

Participant: Thomas Watteyne.

Note well: IoT-lab is NOT strictly speaking a project of Inria-EVA. It is a large project which runs from 2011 to 2021 and which involves the following other partners Inria (Lille, Sophia-Antipolis, Grenoble), INSA, UPMC, Institut Télécom Paris, Institut Télécom Evry, LSIIT Strasbourg. This section highlight Inria-EVA activity and contribution to the IoT-lab testbed in 2015.

- The Paris-Rocquencourt deployment has been stable throughout 2015.
- Thomas Watteyne and the OpenWSN community have been using the platform (all sites, not just Rocquencourt) extensively throughout 2015. Highlights include:
 - Nicola Accettura (then postdoc at UC Berkeley) created scripts to automate running OpenWSN on the IoT-lab, under the co-supervision of Thomas Watteyne and Prof. Kris Pister. Source code is available at <https://github.com/openwsn-berkeley/openwsn-on-iotlab>.
 - This work was presented during the OpenWSN hackathon held in conjunction with the IETF93 standardization meeting in Prague in July 2015.
 - Prof. Diego Dujovne from Universidad Diego Portales (Chile) visited Thomas Watteyne in July 2015 to work on the Mercator project (<https://github.com/openwsn-berkeley/mercator>) to collect Dense Wireless Connectivity Datasets for the IoT on the IoT-lab.
- The Inria-EVA team supported the IoT-lab admin team to remove malfunctioning batteries from the Inria-Rocquencourt deployment in December 2015.
- Thomas Watteyne integrated the IoT-lab admin team in December 2015. Together, they are working on a smaller test deployment with the Inria-EVA premises at Inria-Paris, on which development will be done to:
 - Allow commercial hardware to be plugged into the IoT-lab gateways.
 - Allow multiple motes to be plugged into the same IoT-lab gateway.
 - Use the IoT-lab for deploying and verifying the correct functioning of the OpenWSN implementation on all supported hardware board.
 - Use the IoT-lab for deploying and verifying the correct functioning of the OpenWSN implementation at small/medium/large scale.
- The activities above are lead by Tengfei Chang from the Inria-EVA team, under the supervision of Thomas Watteyne, and in close collaboration with the IoT-lab core team.

GANG Project-Team

6. New Software and Platforms

6.1. Svvpamp: Simulator of Various Voting Algorithms in Manipulating Populations

Svvpamp is a Python package dedicated to the study of voting systems with an emphasis on manipulation analysis. Svvpamp can generate datasets based on a large library of artificial models, or use any kind of real dataset as input. It currently implements more than 20 voting systems. Using state of the art algorithms, it can analyze multiple variants of tactical voting (e.g. absence of weak/strong Nash equilibrium). Svvpamp is free software, under the GNU General Public License version 3. Its documentation includes installation procedure, tutorials, reference guide and instructions for new contributors.

Svvpamp represents about ten thousands lines of code, and according to the Python Software Foundation, 2568 downloads have been reported in the last month (as of December, 10th, 2015). It is available at <https://svvpamp.readthedocs.org>.

Svvpamp [39] will be demonstrated in The Thirtieth Conference on Artificial Intelligence.

6.1.1. Svvpamp self-assesment

A3: ambitious software, usable by people inside and outside the team but without a clear and strong dissemination and support action plan. So3up4: original software reusing known ideas and introducing a few new ideas / original software implementing a fair number of original ideas.

SM3: well-developed software, fairly extensive documentation, reasonable software engineering and testing, attention to usability, dissemination, bug fixes, and user feedback;

EM2: basic maintenance to keep the software alive;

SDL4: public source or binary distribution on the web, organized by the development team;

François Durand is the main contributor (4) in: a) design and architecture (DA) b) coding and debugging (CD) c) maintenance and support (MS) d) team/project management (TPM)

6.2. Big Graph Tools

The team is starting a software development activity around big graph manipulation. A preliminary library offering diameter and skeleton computation is available at <https://who.rocq.inria.fr/Laurent.Viennot/dev/big-graph-tools/>. This library was used to compute the diameters of the worldwide road network (200M edges) and the Twitter follower-followee graph (23G edges).

MIMOVE Team

6. New Software and Platforms

6.1. Introduction

In order to validate our research results and, in certain cases, make them available to specific communities or to the public, our research activities encompass the development of related software as surveyed below.

6.2. VSB: eVolution Service Bus for the Future Internet

Participants: Georgios Bouloukakis, Nikolaos Georgantas [contact], Maël Besson.

URL: <https://tuleap.ow2.org/plugins/git/chorevolution/evolution-service-bus>

The *eVolution Service Bus (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, and Things using CoAP (<https://tools.ietf.org/html/rfc7252>). 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. In this paradigm, a common intermediate bus protocol is used to facilitate interconnection between multiple heterogeneous middleware protocols. Conversion of each protocol to the common bus protocol is done by a component associated to the service/Thing in question and its middleware, called a Binding Component (BC), as it binds the service/Thing to the service bus. We introduce a generic architecture for VSB, which relies on the notion of *Generic Middleware (GM)* connector. GM abstracts interactions among peer components that employ the same middleware protocol through generic *post* and *get* operations, in a unifying fashion for any middleware protocol. We propose an API (application programming interface) for GM and a related generic interface description, which we call *GM-IDL*, for application components that (abstractly) employ GM. Concrete middleware protocols and related interface descriptions of application components that employ these middleware protocols can be mapped to GM API and GM-IDL, respectively. Based on these abstractions, we elaborate a generic architecture for BCs, as well as a related method for BC synthesis and refinement for a concrete choreography that includes services/Things with heterogeneous middleware protocols.

The eVolution Service Bus (VSB) presents a significant rethinking of the architecture and the implementation of a service bus destined to serve dynamic choreographies of services but also Things as first-class entities. More specifically, VSB presents the following advancements:

- VSB is a unified interoperability solution for both services and Things participating in choreographies;
- VSB is flexible and lightweight: it is a completely decentralized network of BCs that are deployed as necessary; hence, no BC is needed when a service/Thing employs the same middleware protocol as the one used as common bus protocol;
- Besides the client-server, publish/subscribe and tuple space paradigms, VSB also provides support for the data streaming paradigm;

- Different protocols can be introduced as VSB's common bus protocol with the same easiness as for integrating support for a new middleware protocol of a service/Thing; additionally, there is no need for relying on and/or providing a full-fledged ESB platform;
- While very modular, VSB's architecture includes only few levels of indirection in the processing of primitives when converting between protocols; this makes it simple, lightweight and fast;
- In VSB, mapping between a concrete middleware protocol and the GM paradigm can be performed in different ways, thus enabling to cover all possible interaction cases; there is no unique, fixed mapping limiting the applicability of the solution;
- BC synthesis follows a systematic method allowing for its automation.

VSB is being developed within the H2020 CHOReVOLUTION project (see § 8.2.1.1) with additional support from the iCONNECT Inria ADT (see § 8.1.1.1). It is also based on previous development carried out in the FP7 CHOReOS project (<http://www.choreos.eu>). VSB is available for download under open source license.

6.3. SoundCity: Urban-scale Noise Monitoring using the Urban Civics Middleware

Participants: Valérie Issarny [contact], Fadwa Rebhi, Animesh Pathak, Sara Hachem.

URL: http://urbancivics.com/soundcity_app.html

The interest in noise pollution and its effects, especially in terms of its influence on citizens' health, has been increasing throughout the years, as illustrated by the plethora of available research and health studies. Traditionally, cities perform periodic noise monitoring to assess the noisiness of neighborhoods and update their city planning policies accordingly. However, in its current state, noise monitoring for city planning takes place once every several years and leverages only a few static sensors (e.g., BruitParif activity, <http://www.bruitparif.fr>). Additionally, the outcome is not representative of citizens' personal exposure to noise as it is coarse grained and requires preplanning to determine where and when the sensing should take place. It is worth mentioning that, even if we assume that the city administration is able to use applications running on the smart phones of the city employees, gathering data by following a pre-determined spatio-temporal schedule – although an excellent strategy in terms of the quality of data gathered – would be expensive and cumbersome to organize.

Noise sensing is also complemented by data modeling approaches to generate maps that allow citizens and governments to visualize the distribution of noise over a region of interest. Those techniques range from basic data plotting to data assimilation techniques that leverage large scale data analytics further enabling the simulation and estimation of missing noise values. However, assimilation techniques are usually restricted to static sensing with predetermined locations and sensing times. This approach limits the capacity of assimilation models to provide estimates with high accuracy.

A solution to the above issues is through mobile participatory sensing, where noise monitoring is no longer restricted to a few expensive sensors. Participatory sensed data can be then leveraged, along with data provided from static sensors, by the assimilation techniques to further reduce the errors in the simulated maps. Toward that goal, we have been building the Urban Civics middleware, which we have further customized for noise monitoring [20]. The refined Urban Civics comprises middleware solutions for noise sensing, crowd-sourcing and data assimilation with the data assimilation component being more specifically developed by the Inria CLIME team. The noise-sensing itself is performed through existing dedicated applications, integrated with our middleware. Such approach enables us to alleviate the burden of domain-specific development and exploit the knowledge of domain experts.

SoundCity and supporting Urban Civics middleware are developed in collaboration with the Inria CLIME team, Ambientic (FR) and the Civic Engine at Berkeley (USA) in the context of CityLab@Inria and Inria@SiliconValley.

6.4. AppCivist-PB: A Platform for Democratic Assembly Customized for Participatory Budgeting

Participants: Valérie Issarny [contact], Cristhian Parra Trepowski, Animesh Pathak.

Participatory budgeting processes are among the most illustrative, real-life experiences of participatory democracy. Participatory Budgeting (PB) has its beginnings in the late 1980s, when some Brazilian cities started to experiment with processes of citizen participation in decisions about how to better allocate part of the city's budget. Although PB takes different forms, they can all be considered as refining the following base process: residents of a city propose spending ideas, volunteers or delegates develop those ideas into proposals, residents then vote on the proposals, and the government finally implements the winning projects. Since the 1980s, PB processes have spread around the world as a set of administrative reforms and, more recently, as a "best practice" in mainstream international development.

Although a large array of ICT tools exist to support citizens' engagement, their use in PB is still limited and scattered. Mostly, ICT have been leveraged for communication for promotion purposes (through multiple channels such as TV, radio and social media) and for facilitating voting for citizens (usually, with custom-made web sites or SMS).

With AppCivist-PB, we want to enable city governments to configure the software assemblies that best match the requirements of the kind of PB campaign they want to support, while leveraging existing software services and components. However, from the overall perspective of participatory democracy, our goal is primarily to facilitate the elaboration of proposals by citizen assemblies that form according to the citizen interests. In other words, we want to support a process that emphasizes collaborative contribution making at all stages of the elaboration of proposals by diverse citizen assemblies, which are primarily created by and for citizens. The collaborative process must in particular facilitate the assembly of groups (or sub-assemblies) on the basis of commonalities among the proposals, which is essential if one wants to sustain city-scale participation and be inclusive of citizen contributions.

AppCivist-PB helps users assemble proposal making and selection workflows, using service-oriented architecture (SOA) principles. The composition principles of SOA allow for various implementations and instances of these workflows, including the possibility of integrating and linking different workflows for the same PB campaign. For example, a city might create and manage its own workflow to receive proposals and facilitate deliberation and voting by registered residents; at the same time, citizen groups (typically activists) can create their own, independent, workflows to co-create, develop, and promote proposals for the city, following their own collaboration practices. Compared to traditional SOA, AppCivist-PB distinguishes itself by enabling the assembly of software services dedicated to the support of online-facilitated participatory democracy by and for relevant citizen assemblies.

The AppCivist-PB platform is developed in collaboration with the Social Apps Labs at CITRIS at University of California Berkeley (USA) in the context of CityLab@Inria and Inria@SiliconValley.

MUSE Team

5. New Software and Platforms

5.1. Fathom

Fathom - browser-based network measurement platform

KEYWORDS: Internet access - Performance measure - Network monitoring

FUNCTIONAL DESCRIPTION

Fathom is a Firefox browser extension that explores the browser as a platform for network measurement and troubleshooting. It provides a wide range of networking primitives directly to in-page JavaScript including raw TCP/UDP sockets, higher-level protocol APIs such as DNS, HTTP, and UPnP, and ready-made functionality such as pings and traceroutes.

- Participants: Anna-Kaisa Pietilainen and Stéphane Archer
- Contact: Anna-Kaisa Pietilainen
- URL: <https://muse.inria.fr/fathom/>

5.2. HostView

FUNCTIONAL DESCRIPTION

End-host performance monitoring and user feedback reporting

- Participants: George Rosca, Anna-Kaisa Pietilainen and Renata Cruz Teixeira
- Contact: Renata Cruz Teixeira
- URL: <https://muse.inria.fr/hostview/>

5.3. Online HoA

Online implementation of home and access throughput bottleneck detection algorithm 'HoA'

FUNCTIONAL DESCRIPTION

Implementation of HoA as collectd plugin for OpenWRT.

- Contact: Renata Cruz Teixeira
- URL: <https://github.com/inria-muse/browserlab>

5.4. SimilarityExplanation

Prototype implementation for explaining a set of similar and recommended movies.

FUNCTIONAL DESCRIPTION

In this web-based prototype for similar movies explanation, we propose two types of browsing for : personalized browsing and non personalized browsing. In the non personalized browsing we suppose that we don't have the user profile. Similar movie sublists are ordered only according to their similarity to the selected movie. For the personalized browsing , we select users that have different profiles from our dataset. We give these users names of actors, according to the types of movies they watch. For each user, we compute the predicted ratings using the matrix factorization model. We select pairs of genres to display to each user based on the preferred genres for the user. In our prototype we identify the preferred genres per user based on the most frequent movie genre pairs that the user has already seen. We then organize the recommended movies with a high rating prediction in sublists, according to the user most preferred genre pairs. When a user selects a movie from the sublists of recommended movies, our application suggests the similar movies presented under four sublists with the added list of words. The sublists are personalized for each user by reordering the movies according to the users predicted ratings.

- Contact: Sara El Aouad
- URL: <http://muse.inria.fr/tagit>

5.5. UCN

User-Centric Networking

FUNCTIONAL DESCRIPTION

The User-Centric Networking (UCN) project is seeking to understand how people consume various kinds of content when using computer networks. Within this project we are undertaking a detailed user study across a range of environments in order to understand the practices involved in consuming media and other content according to context.

- Participants: Renata Cruz Teixeira and Anna-Kaisa Pietilainen
- Contact: Anna-Kaisa Pietilainen
- URL: <https://muse.inria.fr/ucn>

5.6. WeBrowse

FUNCTIONAL DESCRIPTION

WeBrowse is the first passive crowdsourcing-based content curation system. Content curation is the act of assisting users to identify relevant and interesting content in the Internet. WeBrowse requires no active user engagement to promote content. Instead, it extracts the URLs users visit from traffic traversing an ISP network to identify popular content. WeBrowse contains a set of heuristics to identify the set of URLs users visit and to select the subset that are interesting to users.

- Contact: Giuseppe Scavo
- URL: <http://webbrowse.polito.it/>

RAP Project-Team (section vide)

REGAL Project-Team

5. New Software and Platforms

5.1. Antidote

FUNCTIONAL DESCRIPTION

Antidote is the flexible cloud database platform currently under development in the SyncFree European project. Antidote aims to be both a research platform for studying replication and consistency at the large scale, and an instrument for exploiting research results. The platform supports replication of CRDTs, in and between sharded (partitioned) data centres (DCs). The current stable version supports strong transactional consistency inside a DC, and causal transactional consistency between DCs. Ongoing research includes support for explicit consistency [37], [50], for elastic version management, for adaptive replication, for partial replication, and for reconfigurable sharding.

- Participants: Tyler Crain, Marc Shapiro, Serdar Tasiran and Alejandro Tomsic
- Contact: Tyler Crain
- URL: <https://github.com/SyncFree>

5.2. G-DUR

FUNCTIONAL DESCRIPTION

A large family of distributed transactional protocols have a common structure, called Deferred Update Replication (DUR). DUR provides dependability by replicating data, and performance by not re-executing transactions but only applying their updates. Protocols of the DUR family differ only in behaviors of few generic functions. Based on this insight, we offer a generic DUR middleware, called G-DUR, along with a library of finely-optimized plug-in implementations of the required behaviors.

- Participants: Marc Shapiro, Alejandro Tomsic
- Contact: Marc Shapiro
- URL: <https://github.com/msaeida/jessy>

5.3. NumaGiC

FUNCTIONAL DESCRIPTION

NumaGiC is a version of the HotSpot garbage collector (GC) adapted to many-core computers with very large main memories. In order to maximise GC throughput, it manages the trade-off between memory locality (local scans) and parallelism (work stealing) in a self-balancing manner. Furthermore, the collector features several memory placement heuristics that improve locality.

- Participants: Lokesh Gidra, Marc Shapiro, Julien Sopena and Gaël Thomas
- Contact: Marc Shapiro
- URL: <https://scm.gforge.inria.fr/anonscm/git/transgc/>.

5.4. SwiftCloud

FUNCTIONAL DESCRIPTION

Client-side (e.g., mobile or in-browser) apps need local access to shared cloud data, but current technologies either do not provide fault-tolerant consistency guarantees, or do not scale to high numbers of unreliable and resource-poor clients, or both. Addressing this issue, the SwiftCloud distributed object database supports high numbers of client-side partial replicas. SwiftCloud offers fast reads and writes from a causally-consistent client-side cache. It is scalable, thanks to small and bounded metadata, and available, tolerating faults and intermittent connectivity by switching between data centres. The price to pay is a modest amount of staleness. A recent Inria Research Report (submitted for publication) presents the SwiftCloud algorithms, design, and experimental evaluation, which shows that client-side apps enjoy the same guarantees as a cloud data store, at a small cost.

- Participants: Marc Shapiro, Serdar Tasiran, Marek Zawirski and Mahsa Najafzadeh
- Contact: Marc Shapiro
- URL: <git+ssh://scm.gforge.inria.fr//gitroot/swiftcloud>

5.5. PUMA

FUNCTIONAL DESCRIPTION

PUMA is a system that is based on a kernel-level remote caching mechanism that provides the ability to pool VMs memory at the scale of a data center. An important property while lending memory to another VM, is the ability to quickly retrieve memory in case of need. Our approach aims at lending memory only for clean cache pages: in case of need, the VM which lent the memory can retrieve it easily. We use the system page cache to store remote pages such that: (i) if local processes allocate memory the borrowed memory can be retrieved immediately; and (ii) if they need cache the remote pages have a lower priority than the local ones.

- Participants: Maxime Lorrillere, Sébastien Monnet, Pierre Sens, Julien Sopena
- Contact: Maxime Lorrillere
- URL: <https://github.com/mlorrillere/puma>

WHISPER Project-Team

6. New Software and Platforms

6.1. Platforms

6.1.1. Coccinelle

Our recent research is in the area of code manipulation tools for C code, particularly targeting Linux kernel code. This work has led to the Coccinelle tool that we are continuing to develop. Coccinelle serves both as a basis for our future research and the foundation of our interaction with the Linux developer community.

The need to find patterns of code, and potentially to transform them, is pervasive in software development. Examples abound. When a bug is found, it is often fruitful to see whether the same pattern occurs elsewhere in the code. For example, the recent Heartbleed bug in OpenSSL partly involves the same fragment of code in two separate files.⁰ Likewise, when the interface of an API function changes, all of the users of that function have to be updated to reflect the new usage requirements. This generalizes to the case of code modernization, in which a code base needs to be adapted to a new compiler, new libraries, or a new coding standards. Finding patterns of code is also useful in code understanding, *e.g.*, to find out whether a particular function is ever called with a particular lock held, and in software engineering research, *e.g.*, to understand the prevalence of various kinds of code structures, which may then be correlated with other properties of the software. For all of these tasks, there is a need for an easy to use tool that will allow developers to express patterns and transformations that are relevant to their source code, and to apply these patterns and transformations to the code efficiently and without disrupting the overall structure of the code base.

The Coccinelle program matching and transformation tool for C code addresses these needs. Coccinelle has been under development for over 10 years, and is mature software, available in a number of Linux distributions (Ubuntu, Debian, Fedora, etc.). It allows matching and transformation rules to be expressed in terms of fragments of C code, more precisely in the form of a *patch*, in which code to add and remove is highlighted by using + and -, respectively, in the leftmost column, and other, unannotated, code fragments may be provided to describe properties of the context. The C language is extended with a few operators, such as metavariables, for abstracting over subterms, and a notion of positions, which are useful for reporting bugs. The pattern matching rules can interspersed with rules written in Python or OCaml, for further expressiveness. The process of matching patterns against the source code furthermore takes into account some semantic information, such as the types of expressions and reachability in terms of a function's (intraprocedural) control-flow graph, and thus we refer to Coccinelle matching and transformation specifications as *semantic patches*.

Coccinelle was originally motivated by the goal of modernizing Linux 2.4 drivers for use with Linux 2.6, and was originally validated on a collection of 60 transformations that had been used in modernizing Linux 2.4 drivers [8]. Subsequent research involving Coccinelle included a formalization of the logic underlying its implementation [1] and a novel mechanism for identifying API usage protocols [45]. More recently, Coccinelle has served as a practical and flexible tool in a number of research projects that somehow involve code understanding or transformation. These include identifying misuses of named constants in Linux code [47], extracting critical sections into procedures to allow the implementation of a centralized locking service [53], generating a debugging interface for Linux driver developers [23], detecting resource release omission faults in Linux and other infrastructure software [64], and understanding the structure of device driver code in our current DrGene project [66].

⁰<http://git.openssl.org/gitweb/?p=openssl.git;a=commitdiff;h=96db902>

Throughout the development of Coccinelle, we have also emphasized contact with the developer community, particularly the developers of the Linux kernel. We submitted the first patches to the Linux kernel based on Coccinelle in 2007. Since then, over 4500 patches have been accepted into the Linux kernel based on the use of Coccinelle, including around 3000 by over 500 developers from outside our research group. Over 50 semantic patches are available in the Linux kernel source code itself, with appropriate infrastructure for developers to apply these semantic patches to their code within the normal make process. Many of these semantic are also included in a 0-day build-testing system for Linux patches maintained by Intel.⁰ Julia Lawall was invited to the Linux Kernel Summit as a core attendee (invitation only) in 2010 and 2014, and has been invited to the internal 2014 SUSE Labs Conference. She has also presented Coccinelle at developer events such as LinuxCon Europe, Kernel Recipes (Paris), FOSDEM (Brussels), and RTWLS, and has supervised three interns using Coccinelle financed by the Linux Foundation, as part of the Outreachy internship program.

Finally, we are aware of several companies that use Coccinelle for modernizing code bases. These include Metaware in Paris, with whom we had a 5-month contract in 2013-2014 for the customization and maintenance of Coccinelle. We hope to be able to organize other such contracts in the future.

6.1.2. Better Linux

Over the past few years, Julia Lawall and Gilles Muller have designed and developed a number of tools such as Coccinelle, *Diagnosys* [23] [22] and *Hector* [64], to improve the process of developing and maintaining systems code. The *BtrLinux* action aims to increase the visibility of these tools, and to highlight Inria's potential contributions to the open source community. We are developing a web site <https://btrlinux.inria.fr/>, to centralize the dissemination of the tools, collect documentation, and collect results. This action is supported by Inria by the means of a young engineer (ADT), Quentin Lambert. In the case of Coccinelle, we will focus on enhancing its visibility and its dissemination, by using it to find and fix faults in Linux kernel code, and by submitting the resulting patches to the Linux maintainers. Our work on *Diagnosys* and *Hector* is described below.

Diagnosys is a hybrid static and dynamic analysis tool that first collects information about Linux kernel APIs that may be misused, and then uses this information to generate wrapper functions that systematically log at runtime any API invocations or return values that may reflect such misuse. A developer can then use a specific make-like command to build an executable driver that transparently uses these wrapper functions. At runtime, the wrappers write log messages into a crash resilient region of memory that the developer can inspect after any crash. *Diagnosys* is complementary to Coccinelle in the kind of information that it provides to developers. While Coccinelle directly returns a report for every rule match across the code base, often including false positives that have to be manually isolated by the developer, *Diagnosys* only reports on conditions that occur in the actual execution of the code. *Diagnosys* thus produces less information, but the information produced is more relevant to the particular problem currently confronting the developer. As such, it is well suited to the case of initial code development, where the code is changing frequently, and the developer wants to debug a specific problem, rather than ensuring that the complete code base is fault free. *Diagnosys* is a complete functioning system, but it needs to be kept up to date with changes in the kernel API functions. As part of the *BtrLinux* action, we will regularly run the scripts that collect information about how to create the wrappers, and then validate and make public the results.

Hector addresses the problem of leaking resources in error-handling code. Releasing resources when they are no longer needed is critical, so that adequate resources remain available over the long execution periods characteristic of systems software. Indeed, when resource leaks accumulate, they can cause unexpected resource unavailability, and even single leaks can put the system into an inconsistent state that can cause crashes and open the door to possible attacks. Nevertheless, developers often forget to release resources, because doing so often does not make any direct contribution to a program's functionality. A major challenge in detecting resource-release omission faults is to know when resource release is required. Indeed, the C language does not provide any built-in support for resource management, and thus resource acquisition and release are typically implemented using ad hoc operations that are, at best, only known to core developers.

⁰E.g., <http://comments.gmane.org/gmane.linux.kernel.kbuild/269>

Previous work has focused on mining sequences of such functions that are used frequently across a code base, [35], [51] but these approaches have very high rates of false negatives and false positives [48]. We have proposed Hector, a static analysis tool that finds resource-release omission faults based on inconsistencies in the operations performed within a single function, rather than on usage frequency. This strategy allows Hector to have a low false positive rate, of 23% in our experiments, while still being able to find hundreds of faults in Linux and other systems.

Hector was developed as part of the PhD thesis of Suman Saha and was presented at DSN 2013, where it received the William C. Carter award for the best student paper. Hector is complementary to Coccinelle, in that it has a more restricted scope, focusing on only one type of fault, but it uses a more precise static analysis, tailored for this type of fault, to ensure a low false positive rate. Hector, like Coccinelle, is also complementary to Diagnosys, in that it exhaustively reports on faults in a code base, rather than only those relevant to a particular execution, and is thus better suited for use by experienced developers of relatively stable software. Over 70 patches have been accepted into Linux based on the results of Hector. The current implementation, however, is somewhat in a state of disarray. As part of the BtrLinux action, we are currently working on returning the code to working condition and then will actively use it to find faults in Linux. Based on these results, we will either submit appropriate patches to the Linux developers or notify the relevant developer when the corresponding fix is not clear.

6.2. New Software

6.2.1. *coq-bitset library*

As part of Arthur Blot's internship, we have developed the `coq-bitset` library, a certified library implementing bitsets in the Coq proof assistant [17]. It enables abstract and formal reasoning about efficient low-level code within a proof assistant, thus paving the way for further certified results in the field of low-level system code (such as device drivers).

As part of this effort, we have also extended a pre-existing formalization of bit vectors in Coq [41] with a trustworthy extraction mechanism. This enables manipulating and reasoning about native integers in the Coq proof assistant, while supporting an efficient execution in OCaml.

Both libraries have been made available on Github as well as on the Coq-opam repository.

ALPAGE Project-Team

6. New Software and Platforms

6.1. Alexina

Atelier pour les LEXiques INformatiques et leur Acquisition

FUNCTIONAL DESCRIPTION

Alexina is Alpage's Alexina framework for the acquisition and modeling of morphological and syntactic lexical information. The first and most advanced lexical resource developed in this framework is the Lefff, a morphological and syntactic lexicon for French.

- Participants: Benoît Sagot and Laurence Danlos
- Contact: Benoît Sagot
- URL: <http://gforge.inria.fr/projects/alexina/>

6.2. Bonsai

FUNCTIONAL DESCRIPTION

Alpage has developed a statistical parser for French, named Bonsai, trained on the French Treebank. This parser provides both a phrase structure and a projective dependency structure specified in [66] as output. This parser operates sequentially: (1) it first outputs a phrase structure analysis of sentences reusing the Berkeley implementation of a PCFG-LA trained on French by Alpage (2) it applies on the resulting phrase structure trees a process of conversion to dependency parses using a combination of heuristics and classifiers trained on the French treebank. The parser currently outputs several well known formats such as Penn treebank phrase structure trees, Xerox like triples and CONLL-like format for dependencies. The parsers also comes with basic preprocessing facilities allowing to perform elementary sentence segmentation and word tokenisation, allowing in theory to process unrestricted text. However it is believed to perform better on newspaper-like text.

- Participants: Marie-Hélène Candito, Djamé Seddah and Benoit Crabbé
- Contact: Marie-Hélène Candito
- URL: http://alpage.inria.fr/statgram/frdep/fr_stat_dep_parsing.html

6.3. Crapbank

French Social Media Bank

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

6.4. DyALog

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/>

6.5. FDTB1

- Contact: Laurence Danlos

6.6. FQB

French QuestionBank
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

6.7. FRMG

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

6.8. French Question Bank

- Contact: Djamé Seddah

6.9. LexConn

- Contact: Laurence Danlos

6.10. LexViz

FUNCTIONAL DESCRIPTION

In the context of the industrial collaboration of ALPAGE with the company Lingua et Machina, we have extended their WEB plateform Libellex with a new component used to visualize and collaboratively validate lexical resources. In particular, this extension is used to manage terminological lists and lexical networks. The implemented graph-based representation has proved to be intuitive and quite useful for navigating in such large lexical resources (on the order to 10K to 100K entries).

- Participants: Éric Villemonte de La Clergerie and Mickael Morardo
- Contact: Éric Villemonte de La Clergerie

6.11. 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://www.rocq.inria.fr/alpage-wiki/tiki-index.php?page=MElt>

6.12. Mgwiki

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.

- Participants: Éric Villemonte de La Clergerie and Paul Bui-quang
- Contact: Éric Villemonte de La Clergerie
- URL: <http://alpage.inria.fr/frmgwiki/>

6.13. OGRE

Optimized Graph Rewriting Engine

FUNCTIONAL DESCRIPTION

OGRE is a graph rewriting system specifically designed for manipulating linguistic trees and graphs. It relies on a rule specification language for expressing graph rewriting patterns. The transformation is performed in two steps:

First, the system performs simple transformations following the rewriting patterns,

Second, constraints can be applied on edges, which applies transformations depending on their environment that are propagated while all constraints are satisfied.

The system has been designed for the analysis and manipulation of attributed oriented and multi-relational graphs.

- Participants: Corentin Ribeyre, Djamé Seddah, Éric Villemonte de La Clergerie and Marie-Hélène Candito
- Contact: Corentin Ribeyre
- URL: <http://www.corentinribeyre.fr/projects/view/OGRE>

6.14. SYNTAX

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

6.15. Sequoia corpus

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

6.16. SxPipe

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 chain aimed to apply to raw corpora a cascade of surface processing steps. It is used as a preliminary step before Alpage's parsers (e.g., FRMG) and for surface processing (named entities recognition, text normalization, unknown word extraction and processing...).

- Participants: Pierre Boullier, Benoît Sagot, Kata Gábor, Marion Baranes, Pierre Magistry, Éric Villemonte de La Clergerie and Djamé Seddah
- Contact: Benoît Sagot
- URL: <http://lingwb.gforge.inria.fr/>

6.17. VerbeNet

- Contact: Laurence Danlos

6.18. hyparse

Alpage Hybrid Parser

KEYWORDS: Parsing - NLP

FUNCTIONAL DESCRIPTION

Multilingual Phrase Structure Parser

- Contact: Benoit Crabbé
- URL: <http://hyparse.gforge.inria.fr>

6.19. DyALog-sr

DYALOG-sr

KEYWORDS: Parsing - NLP

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.

- Contact: Éric de La Clergerie

RITS Project-Team

6. New Software and Platforms

6.1. DOLAR

FUNCTIONAL DESCRIPTION

This software performs real-time obstacle detection and tracking using laser data scanned with one or several laser sensors with different geometric configurations. Obstacle detection is based on laser data segmentation while obstacle tracking uses PHD-based filtering techniques.

- Contact: Fawzi Nashashibi

6.2. MELOSYM

FUNCTIONAL DESCRIPTION

MELOSYM is the latest laser based Hierarchical ML-SLAM algorithm developed by RITS. It contains all the functions needed to perform the vehicle localization and the mapping of the environment. Windows compatible, it was initially developed under the RTMAPS platform but the version includes a standalone version.

- Participants: Fawzi Nashashibi, Benjamin Lefaudeux, Jianping Xie and Paulo Lopes Resende
- Contact: Benjamin Lefaudeux

6.3. PML-SLAM

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

6.4. STEREOLOC-3D

FUNCTIONAL DESCRIPTION

STEREOLOC is the package performing stereovision based localization and mapping. It performs semi-dense mapping of outdoor large environments and provides real-time estimates of the vehicle position.

- Participants: Benjamin Lefaudeux and Fawzi Nashashibi
- Contact: Fawzi Nashashibi

6.5. SODA

SOftwares for Driving Automation

KEYWORD: Environment perception

FUNCTIONAL DESCRIPTION

This software has been developed in the context of the French ABV (Automatisation Basse Vitesse) project. This package contains the functions that are necessary to automate the vehicle navigation in its secured lane.

- Participants: Paulo Lopes Resende and Fawzi Nashashibi
- Contact: Fawzi Nashashibi

6.6. AutoPathPlan

Automatic Path Planning Generation

FUNCTIONAL DESCRIPTION

Automatic method for a real time path planning generation path for automated vehicles.

- Participants: David Gonzalez Bautista, Joshué Pérez Rastelli and Vicente Milanés Montero
- Contact: Fawzi Nashashibi

6.7. FEMOT

Fuzzy Embedded MOTor

FUNCTIONAL DESCRIPTION

FEMOT is an experimental motor for implementing fuzzy logic controllers, including all the fuzzy stages (fuzzification, inference, and defuzzification). This library has been compiled in Microsoft Visual (MVS) Studio and RtMaps. The proposed library is modular and adaptable to different situations and scenarios, especially for autonomous driving applications. FEMOT allows the development of the fuzzy rules to be written as sentences in an almost natural language. It allows the user to define variables and their fuzzy rules and to join them with other variables in rules to yield crisp signals for the controllers.

This software is used for the arbitration and control for fully automated functions. The behaviour of a human driver can be emulated with this technique. First simulations are showing promising results, and the library allows an easy adaptation in decision marking situations.

- Participants: Joshué Pérez Rastelli and Vicente Milanés Montero
- Contact: Fawzi Nashashibi

6.8. Platools

KEYWORD: Telecommunications

- Participant: Marios Makassikis
- Contact: Thierry Ernst

6.9. V2Provue

Vehicle-to-Pedestrian

KEYWORD: vehicle-to-pedestrian communications

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

6.10. Taxi-col

KEYWORD: Mobile Computing, Transportation

- Participant: Eugenie Lioris
- Contact: Fawzi Nashashibi

SMIS Project-Team

5. New Software and Platforms

5.1. PLUG-DB ENGINE

FUNCTIONAL DESCRIPTION

PlugDB is a complete platform dedicated to a secure and ubiquitous management of personal data. It aims at providing an alternative to a systematic centralization of personal data. The PlugDB engine is a personal database server capable of storing data (tuples and documents) in tables and BLOBs, indexing them, querying them in SQL, sharing them through assertional access control policies and enforcing transactional properties (atomicity, integrity, durability). The PlugDB engine is embedded in a tamper-resistant hardware device combining the security of smartcard with the storage capacity of NAND Flash. The personal database is hosted encrypted in NAND Flash and the PlugDB engine code runs in the microcontroller. Complementary modules allow to pre-compile SQL queries for the applications, communicate with the DBMS from a remote Java program, synchronize local data with remote servers (typically used for recovering the database in the case of a broken or lost devices) and participate in distributed computation (e.g., global queries). PlugDB runs both on secure devices provided by Gemalto and on specific secure devices designed by SMIS and assembled by electronic SMEs. Mastering the hardware platform opens up new research and experiment opportunities (e.g., we have recently integrated a Bluetooth module to communicate wirelessly with PlugDB and a fingerprint module to strongly authenticate users) and allows us to engage ourselves in an open-source/open hardware initiative. Open-SW/open-HW contributes to the trust the community of users can put in any privacy preserving solution and is key to enable a diversity of solutions, hence decreasing the risk of class attacks. PlugDB engine has been registered first at APP (Agence de Protection des Programmes) in 2009 - a new version being registered every two years and the hardware datasheets in 2015. PlugDB has been experimented in the field - notably in the healthcare domain - and we recently set up an educational platform to raise students awareness of privacy protection problems and embedded programming. As a conclusion, PlugDB combines several research contributions from the team, at the crossroads of flash data management, embedded data processing and secure distributed computations. It then strongly federates all members of our team (permanent members, PhD students and engineers). It is also a vector of visibility, technological transfer and dissemination and gives us the opportunity to collaborate with researchers from other disciplines around a concrete privacy enhancing platform.

- Participants: Nicolas Ancaux, Luc Bouganim, Philippe Pucheral, Shaoyi Yin, Yanli Guo, Kevin Jacquemin, Aydogan Ersoz and Quentin Lefebvre
- Contact: Nicolas Ancaux
- URL: <https://project.inria.fr/plugdb/>

WILLOW Project-Team

6. New Software and Platforms

6.1. Visual Place Recognition with Repetitive Structures

A new version of the open-source release of the software package for visual localization in urban environments has been made publicly available in July 2015. The software package implements the method [A. Torii et al., CVPR 2013] (journal version published this year in [5]) for representing visual data containing repetitive structures (such as building facades or fences), which often occur in urban environments and present significant challenge for current image matching methods. This is an extended version that includes geometric verification. The first version was made available in 2013 and has been updated in May 2014. The current version of the software is available at http://www.di.ens.fr/willow/research/reptile/download/reptile_demo_ver04.zip.

6.2. NetVLAD: CNN architecture for weakly supervised place recognition

Open source release of the software package for our paper "NetVLAD: CNN architecture for weakly supervised place recognition" [21]. It provides a full implementation of the method, including code for weakly supervised training of the CNN representation, testing on standard datasets, as well as trained models. Links to all of these are available at our project page <http://www.di.ens.fr/willow/research/netvlad/>.

6.3. 24/7 place recognition by view synthesis

Open source release of the software package for our paper "24/7 place recognition by view synthesis" [16]. It provides code for computing VLAD descriptors, performing feature matching and view synthesis. Link to the code is available at our project page <http://www.ok.ctrl.titech.ac.jp/~torii/project/247/>.

6.4. Weakly Supervised Object Recognition with Convolutional Neural Networks

Open-source release of the software package for weakly supervised object recognition with convolutional neural networks has been made publicly available in May 2015. The software package implements the method [M. Oquab et al., CVPR 2015] [14] for object category recognition and localization using convolutional neural networks with weak supervision (without bounding box annotations). The method (i) outputs accurate image-level labels, (ii) predicts approximate locations (but not extents) of objects, and (iii) performs comparably to its fully-supervised counterparts using object bounding box annotation for training. The current version of the software is available at <http://www.di.ens.fr/willow/research/weakcnn/>.

6.5. Unsupervised Object Discovery and Localization in the Wild

This package contains source code for unsupervised object discovery and localization from image collections. From an arbitrary collection of images in the wild, the method effectively discover dominant object instances and localize them by bounding boxes. The localization accuracy of discovered objects measured at standard benchmarks for object localization is significantly better than the state-of-the-art methods in co-localization, while using no supervision on image collections. The package is available from <http://www.di.ens.fr/willow/research/objectdiscovery/>.

6.6. Joint Static and Dynamic Guidance Filter

Open-source release of the software package for depth upsampling, texture removal, and scale-space filtering has been made publicly available. The software package implements the newly developed method [10] for robust filtering with joint static and dynamic guidance. The software is available at <http://www.di.ens.fr/willow/research/sdfilter/>.