RESEARCH CENTER
Paris

FIELD

Activity Report 2016

Section Software

Edition: 2017-08-25

ALPAGE Project-Team

5. New Software and Platforms

5.1. Alexina

Atelier pour les LEXiques INformatiques et leur Acquisition
FUNCTIONAL DESCRIPTION

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

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

5.2. 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 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, Djame Seddah and Benoît Crabbe
- Contact: Marie-Hélène Candito
- URL: http://alpage.inria.fr/statgram/frdep/fr_stat_dep_parsing.html

5.3. 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: Djame Seddah

5.4. 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: Eric Villemonte De La Clergerie
- Contact: Eric Villemonte De La Clergerie
- URL: http://dyalog.gforge.inria.fr/

5.5. FDTB1

- Contact: Laurence Danlos

5.6. 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: Djame Seddah

5.7. FRMG

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

5.8. Extreme UGC corpus

FUNCTIONAL DESCRIPTION

The Extreme UGC corpus is French three-domain data set focusing on user-generated content, made up of noisy question headlines from a cooking forum, live game chat logs and associated forums from two popular online games (MINECRAFT and LEAGUE OF LEGENDS). Building such an out of domain corpus, allowed us to consider the limits of our current normalization approaches. Currently annotated with part-of-speech, we plan to add other annotations layers.

- Contact: Djame Seddah

5.9. LexConn

- Contact: Laurence Danlos

5.10. LexViz

FUNCTIONAL DESCRIPTION

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

- Participants: Eric Villemonte De La Clergerie and Mickaël Morardo
- Contact: Eric Villemonte De La Clergerie

5.11. 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

5.12. Mgwiki

FUNCTIONAL DESCRIPTION

Mgwiki is a linguistic wiki that may used to discuss linguistic phenomena with the possibility to add annotated illustrative sentences. The work is essentially devoted to the construction of an instance for documenting and discussing FRMG, with the annotations of the sentences automatically provided by parsing them with FRMG. This instance also offers the possibility to parse small corpora with FRMG and an interface of visualization of the results. Large parsed corpora (like French Wikipedia or Wikisource) are also available. The parsed corpora can also be queried through the use of the DPath language.

- Participants: Eric Villemonte De La Clergerie and Paul Bui-Quang
- Contact: Eric Villemonte De La Clergerie
- URL: http://alpage.inria.fr/frmgwiki/

5.13. 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. Itis currently being used to convert existing universal dependencies for French to the upcoming 2.0 scheme to be used for the next “big” CoNLL parsing Shared Task of 2017.

- Participants: Corentin Ribeyre, Djame Seddah, Eric Villemonte De La Clergerie and Marie-Hélène Candito
- Contact: Corentin Ribeyre
- URL: http://www.corentinribeyre.fr/projects/view/OGRE

5.14. 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/

5.15. 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: Djame Seddah

5.16. 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, Eric Villemonte De La Clergerie and Djame Seddah
- Contact: Benoît Sagot
- URL: http://lingwb.gforge.inria.fr/

5.17. Verb∋net

- Contact: Laurence Danlos

5.18. dyalog-sr

KEYWORD: Parsing
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

5.19. hyparse

Alpage Hybrid Parser
KEYWORDS: Parsing - NLP
FUNCTIONAL DESCRIPTION

Multilingual Phrase Structure Parser

- Contact: Benoît Crabbe
- URL: http://hyparse.gforge.inria.fr

5.20. vera

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

ALPINES Project-Team

6. New Software and Platforms

6.1. BFD

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, Remi Lacroix and Frédéric Nataf
- Partners: CNRS - UPMC
- Contact: Laura Grigori
- URL: https://who.rocq.inria.fr/Laura.Grigori/

6.2. CALU : communication optimal algorithms for linear algebra

KEYWORDS: Communication avoiding - Linear algebra
FUNCTIONAL DESCRIPTION

CALU solves linear systems of equations Ax=b using Communication Avoiding LU.

- Contact: Laura Grigori
- URL: https://who.rocq.inria.fr/Laura.Grigori/

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/

6.4. FreeFem++

FeeFrem++
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 2^d-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: Frédéric Hecht
- URL: http://www.freefem.org/ff++/

6.5. 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
- URL: https://github.com/hpddm

6.6. LORASC

LORASC preconditioner

KEYWORD: Preconditioner

- Participants: Laura Grigori and Remi Lacroix
- Contact: Laura Grigori
- URL: not available

6.7. NFF

NFF Nested Filtering Factorization

KEYWORDS: Preconditioner - Interactive method - Linear system

- Participants: Laura Grigori, Frédéric Nataf and Long Qu
- Partners: Université Paris-Sud - UPMC
- Contact: Laura Grigori
- URL: not available

6.8. SparseToolbox

KEYWORDS: Preconditioner - Interactive method - Linear system

- Participants: Laura Grigori and Remi Lacroix
- Contact: Laura Grigori
- URL: not available

ANGE Project-Team

6. New Software and Platforms

6.1. Freshkiss3D (FREe Surface Hydrodynamics using KInetic SchemeS)

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

- Participants: Nora Aïssiouene, Marie-Odile Bristeau, David Froger, Jacques Sainte-Marie, Fabien Souillé
- Formerly: Emmanuel Audusse, Anne-Céline Boulanger, Alain Dervieux, Raouf Hamouda, Bijan Mohammadi
- Partners: CEREMA – UPMC
- Contact: Jacques Sainte-Marie

A review of last developments is given in § 7.5.1 . See [20] for recent numerical results obtained thanks to the Freshkiss3D software.

6.2. TSUNAMATHS

Tsunamaths is an educational platform aiming at simulating historical tsunamis. Real data and mathematical explanations are provided to enable people to better understand the overall process of tsunamis.

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

ANTIQUE Project-Team

6. New Software and Platforms

6.1. 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 Jeannet
- Contact: Antoine Miné
- URL: http://apron.cri.ensmp.fr/library/

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

6.3. 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/

6.4. 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

6.5. 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

6.6. 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

6.7. 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/

6.8. 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

6.9. 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 guara Furthermore, this approach should allow to meet some domain specific software qualification criteria (such as those in DO-178 regulations for avionics software), since it allows proving that successive development levels are correct with respect to each other i.e., that they implement the same specification. Last, this technique also justifies the use of source level static analyses, even when an assembly level certification would be required, since it establishes separately that the source and the compiled code are equivalent.ntees that no compiler bug did cause incorrect code to be generated.

- Participant: Xavier Rival
- Contact: Xavier Rival

6.10. 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

6.11. 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/

6.12. DAFT

DAFT
FUNCTIONAL DESCRIPTION

DAFT is a distributed file management system in user-space, with a command-line interface. DAFT is intended at computational scientists, involved in data-intensive, distributed experiments and when no distributed file-system is available on computing nodes. DAFT is secure; all messages are cryptographically signed and encrypted by default.

- Participants: Francois Berenger and Camille Coti.
- Contact: Francois Berenger and Camille Coti.
- URL: https://github.com/UnixJunkie/daft.

AOSTE Project-Team

5. New Software and Platforms

5.1. EVT Kopernic

Extreme Value Theory for Keeping Worst Reasoning Appropriate for Different Criticalities
FUNCTIONAL DESCRIPTION This software provides a probabilistic bound on the worst case execution time of a program. Its third version, released in March 2016, covers the case of statistically dependent execution times. Currently integrated in Rapitime (Rapita tool chain), a lighter version is under preparation for integration in FUI Waruna framework as well as in the preparation of hybrid versions to be released in 2017 as output of Capacites project.

- Participants: Liliana Cucu and Adriana Gogonel
- Contact: Liliana Cucu
- URL: https://who.rocq.inria.fr/Liliana.Cucu/Software.html

5.2. KPASSA

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

- Participants: Jean Vivien Millo and Robert De Simone
- Contact: Robert de Simone
- URL: http://www-sop.inria.fr/members/Jean-Vivien.Millo/kpassa/index.php

5.3. Lopht

Logical to Physical Time Compiler
SCIENTIFIC DESCRIPTION Lopht is a system-level compiler for embedded systems. Its input is formed of three objects:

- A functional specification in a high-level synchronous language.
- A description of the implementation platform, defining the topology of the parallel execution platform, and the capacity of its elements.
- A set of non-functional requirements, provided under the form of annotations on both functional specification and platform description.

The algorithmic core of Lopht is formed of allocation and scheduling heuristics which rely on two fundamental choices: the use of table-based static scheduling and the use of low-complexity heuristics based on list scheduling. The output of Lopht is formed of all the C code and configuration information needed to allow real deployment on the physical target platform.

FUNCTIONAL DESCRIPTION Accepted input languages for functional specifications include Heptagon and Scade v4. Lopht uses as f ront-end a modified version of the Heptagon compiler developed at Inria. The use of this front-end also allows the use of legacy/business C code satisfying the Heptagon calling convention.

Regarding scheduling, the originality of Lopht resides in a strong focus on classical compiler optimizations e.g. software pipelining), on novel architectural targets (many-core chips and time-triggered embedded systems), and the possibility to handle multiple, complex non-functional requirements covering real-time (release dates a nd deadlines possibly larger than the period, end-to-end flow constraints), ARINC 653 partitioning, the possibility to preempt or not each task, and allocation.
The output of Lopht is formed of all the C code and configuration information needed to allow compilation, linking/loading, and real-time execution on the target platform. Lopht fully automates the creation of tasks, partition, the full synthesis of C code compliant with the target API (e.g. C/APEX for ARINC 653 platforms), including communication code, and OS configuration for each computer), as well as the synthesis of communication schedules for the system

Two Lopht back-ends provide distinct input languages for platform description:

- One for distributed time-triggered architectures using ARINC 653-based processing nodes (SBCs) and Time-Triggered Ethernet networks
- One for many-core processors with support with timing predictability.

An ongoing research effort aims at providing a unified, formal platform description language allowing the unification of these back-ends.

- Participants: Dumitru Potop Butucaru, Raul Gorcitz, and Keryan Didier
- Contact: Dumitru Potop Butucaru

5.4. 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 additionaly allows to select dynamic scheduling policy algorithms such as Earliest Deadline First (EDF). The resulting schedule is displayed as a typical Gantt chart with a transient and a permanent phase, or as a disk shape called "dameid", which clearly highlights the idle slots of the processor in the permanent phase.
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/

5.5. 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, executives kernels are provided for: TMS320C40, PIC18F2680, i80386, MC68332, MPC555, i80C196 and Unix/Linux workstations. Executive kernels for other processors can be achieved at reasonable cost following these examples as patterns.
FUNCTIONAL DESCRIPTION Software for optimising the implementation of embedded distributed real-time applications and generating efficient and correct by construction code

- Participants: Yves Sorel and Meriem Zidouni
- URL: http://www.syndex.org

5.6. 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: Frederic Mallet, and Julien Deantoni
- Contact: Frederic Mallet
- URL: http://timesquare.inria.fr

5.7. Vercors

KEYWORD:

- Participants: Eric Madelaine, Oleksandra Kulankhina, Jimmy Awk, Xudong Qin
- Contact: Eric Madelaine
- URL: http://www-sop.inria.fr/oasis/Vercors

FUNCTIONAL DESCRIPTION The Vercors tools include front-ends for specifying the architecture and behaviour of components in the form of UML diagrams. We translate these high-level specifications, into behavioural models in various formats, and we also transform these models using abstractions. In a final step, abstract models are translated into the input format for various verification toolsets. Currently we mainly use the various analysis modules of the CADP toolset.

We have achieved this year a major version of the platform frontend, named VCE-v4, that is now distributed on our website, and used by some of our partners. This version features a full chain of tools from the design of systems in the graphical component editors (VCE), the checking of static semantics correcteness, the generation of a semantic model suitable for model-checking, and finally the generation of executable code for the Proactive/GCM platform. These new features, and the tool architecture, have been described in [29] and [18].

ARAMIS Project-Team

6. New Software and Platforms

6.1. Clinica

KEYWORDS: Multimodal neuroimaging - anatomical MRI - diffusion MRI - functional MRI - PET - EEG/MEG
FUNCTIONAL DESCRIPTION

Clinica is a software platform for multimodal brain image analysis in clinical research studies. It aims at integrating a comprehensive set of processing tools for the main neuroimaging modalities: MRI (anatomical, functional, diffusion), PET and EEG/MEG. For each modality, it allows to easily extract various types of features (regional measures, parametric maps, surfaces, curves, networks) that can be subsequently used as input of machine learning, statistical modeling, morphometry or network analysis methods. Processing pipelines are based on combinations of freely available tools developed by the community and in-house developments. It provides an integrated data management system to store raw and processing data.

- Participants: Olivier Colliot, Stanley Durrleman, Fabrizio De Vico Fallani, Michael Bacci, Alexandre Routier, Jorge Samper-Gonzalez, Junhao Wen, Jérémy Guillon, Sabrina Fontanella, Thomas Jacquemont
- Contact: Olivier Colliot

6.2. 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

6.3. 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, Cedric Doucet, Michael Bacci and Barbara Gris
- Partners: Université de Montpellier 2 - Université Paris-Descartes - University of Utah
- Contact: Stanley Durrleman
- URL: http://www.deformetrica.org/

6.4. 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

6.5. 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 abnormalisties and with different acquisition characteristics.

- Participants: Marie Chupin, Ludovic Fillon and Thomas Samaille
- Contact: Marie Chupin

6.6. 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, developped by several CATI engineers. The first module is dedicated to acquisition requirement checking and conversion to nifti format. The second module aims at making 3DT1 acquisition quality check more systematic, and relies both on visual inspection and quantitative indices. The third module allows a simultaneous evaluation of the clinical part of the CATI acquisition protocol. The fourth module embeds automatic indices to evaluate resting state fMRI acquisition. The fifth module is dedicated to first preprocessings and quality indices for dMRI. The sixth module is dedicated to qMRI, with visual and automated quality control together with preprocessings. The last module is dedicated to data and project management.

- Participants: Marie Chupin and Hugo Dary
- Contact: Marie Chupin

CASCADE Project-Team (section vide)

CLIME Project-Team

6. New Software and Platforms

6.1. Heimdali

- Participants: Isabelle Herlin, Dominique Bereziat and David Froger
- Contact: Isabelle Herlin

SCIENTIFIC DESCRIPTION

The main components of Heimdali concern:

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

FUNCTIONAL DESCRIPTION

The initial aim of the image processing library Heimdali was to replace an internal Inria library (named Inrimage) by a library based on standard and open source tools, and mostly dedicated to satellite acquisitions.

The leading idea of the library is to allow the following issues:
- making easier the sharing and development of image assimilation softwares. For that purpose, the installation is easily achieved with the package manager Conda.
- developing generic tools for image processing and assimilation based on ITK (Insight Segmentation and Registration Toolkit http://www.itk.org). In reverse provide tools to ITK and contribute to the ITK community. Our software corresponds to issues related to satellite acquisitions but could be of interest for processing medical image sequences.

6.2. Image Forecast

- Authors: Isabelle Herlin and Yann Lepoittevin
- Contact: Isabelle Herlin

SCIENTIFIC DESCRIPTION

From a given number of images, Image Forecast synthetizes the future images at a given and short temporal horizon.

FUNCTIONAL DESCRIPTION

Image forecast includes two components:

- it estimates the dynamics from an image sequence. Various options are available in the software: stationarity, Lagrangian conservation, description of structures. The result is the motion field explaining the temporal evolution of image data.
- the estimated dynamics is applied for forecasting future images at short temporal horizon.

6.3. Polyphemus

- Participants: Sylvain Doré (CEREA, École des Pont ParisTech) and Vivien Mallet
- Contact: Vivien Mallet
- URL: http://cerea.enpc.fr/polyphemus/

FUNCTIONAL DESCRIPTION

Polyphemus is a modeling system for air quality. As such, it is designed to yield up-to-date simulations in a reliable framework: data assimilation, ensemble forecast and daily forecasts. Its completeness makes it suitable for use in many applications: photochemistry, aerosols, radionuclides, etc. It is able to handle simulations from local to continental scales, with several physical models. It is divided into three main parts:

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

Fig. 1 depicts a typical result produced by Polyphemus.

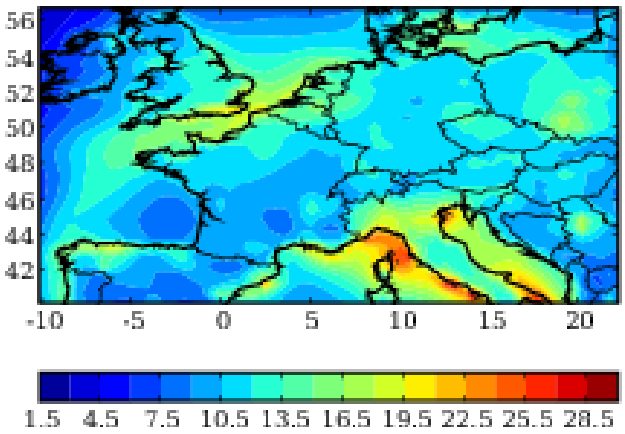

Figure 1. Map of the relative standard deviation (or spread, %) of an ensemble built with Polyphemus (ozone simulations, $\mu g\ m^{-3}$). The standard deviations are averaged over the summer of 2001. They provide an estimation of the simulation uncertainties.

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

6.4. SoundCity - Ambiciti

- Authors: Pierre-Guillaume Raverdy (SED), Fadwa Rebhi (Mimove), Cong Kinh Nguyen (Mimove), Rajiv Bhatia (TheCivicEngine), Vivien Mallet and Valerie Issarny (Mimove)
- Contact: Valerie Issarny (Mimove)

FUNCTIONAL DESCRIPTION

Ambiciti measures the actual noise levels to which you are exposed. It can monitor noise levels throughout the day and inform you about your instantaneous, hourly and daily exposures.
Ambiciti also computes the air quality index in your region or at the exact location where you stand. You can also access to forecasts.

Ambiciti includes a lot of features:

- Measuring noise level, anytime on demand or automatically during the day,
- Air quality indexes, in the past, present and future hours or days,
- Pollution levels for nitrogen dioxide, fine particulate matter and ozone,
- Statistics on exposure to pollutions, hourly, daily, during daytime and nighttime,
- Maps with your noise measurements,
- Hourly air quality maps, at street resolution in Paris, San Francisco, Oakland, Richmond (California), at present time,
- The recommendation of pedestrian routes which minimize the exposure to noise pollution or to air pollution,
- The ability to take pictures with pollution levels on top.

6.5. Urban noise analysis

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

FUNCTIONAL DESCRIPTION

This software merges noise simulations and mobile observations. It can extract a given region of a noise map and filter out the buildings. It extends a previous software for data assimilation of air pollution observations at city scale. This prior software computes the so-called best linear unbiased estimator (BLUE), with a special background error covariance model that depends on the city geometry. The extension for noise introduces special treatments for the errors in mobile observations, and includes more statistical verifications.

The software also comes with a Python module for the management of a large database of mobile noise measurements, especially with many filters relying the observations metadata.

The software finally includes the automatic generation of a report based on intensive measurements in a city district. This report targets participants of crowdsensing experiments.

6.6. Verdandi

- Participants: Vivien Mallet, Gautier Bureau (Medisim), Dominique Chapelle (Medisim), Sébastien Gilles (Medisim) and Philippe Moireau (Medisim)
- Contact: Vivien Mallet
- URL: http://verdandi.gforge.inria.fr/

FUNCTIONAL DESCRIPTION

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

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 Project-Team

6. New Software and Platforms

6.1. OpenWSN (Software)

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

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 2016:

- Development:
  - better (continuous) testing of the existing OpenWSN code.
  - Build a image for a RaspberryPi which contains the OpenVisualizer preinstalled.
  - Port OpenWSN to the First prototype of the "Single-Chip uMote" (SCuM), developed in Prof. Pister's lab at UC Berkeley.
  - Create a Virtual Machine image with all OpenWSN development tools preinstalled.
  - Implementation of 6TiSCH standards as they appear, including revisions.
  - Maintenance of the "Golden Image" used as a reference during interoperability testing
- Recognition:
  - OpenWSN remains ETSI's 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**
    Standards for the Industrial IoT: a Hands-on Tutorial with OpenWSN and OpenMote. Xavier Vilajosana, Pere Tuset-Peiro, Tengfei Chang, **Thomas Watteyne**. IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Valencia, Spain, 4-8 September 2016.
  - **Tutorial**
    Introduction to the IETF 6TiSCH stack with OpenWSN & OpenMote. **Thomas Watteyne**, Xavier Vilajosana, Pere Tuset-Peiro, Tengfei Chang. International Conference on Telecommunications (ICT), Thessaloniki, Greece, 16-18 May 2016.
  - **Interop Event**
    ETSI 6TiSCH 3 plugtests, 15-16 Juny 2016, Berlin, Germany
  - **Interop Event**
    ETSI 6TiSCH 2 plugtests, 2-4 Februay 2016, Paris, France

This software appears in https://bil.inria.fr/.

6.2. OPERA and OCARI (Software)

Participants: Erwan Livolant, Pascale Minet.

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

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

In 2016, Erwan Livolant developed extensions to allow the remote management of the OCARI network and the transmission of commands to sensors and actuators.

6.3. F-Interop (Software)

Participants: Remy Leone, Thomas Watteyne.

F-Interop is revolutionizing the way interoperability events are conducted. We are building a cloud-based system which allows implementors to meet online to test their implementations against one another, verify compliance in a automated way, and verify the performance of their implementations on large scale testbeds. This significantly cuts down time-to-market for standards-based solutions, and eventually leads to more standards-based products on the market. The F-Interop platform starts by targeting 6TiSCH, 6LoWPAN and CoAP standards, but our ambition is for F-Interop to become the standard way of doing interoperability, at least at the IETF IoT level.

This implementation is done as part of the H2020 F-Interop project. More information at http://www.f-interop.eu/.

This software appears in https://bil.inria.fr/.

6.4. 6TiSCH Simulator (Software)

Participants: Malisa Vucinic, Thomas Watteyne.

The 6TiSCH simulator allows one to conduct high-level simulator of an IETF 6TiSCH network and answer questions such as How long does it take the nodes to join the network? What is the average power consumption? What does the latency distribution look like?

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

This implementation is done as part of the H2020 ARMOUR project and the standardization activities of the Inria-EVA team at the IETF. It is published under an open-source BSD license and maintained at https://bitbucket.org/6tisch/simulator/.

This software appears in https://bil.inria.fr/.

6.5. 6TiSCH Wireshark Dissector (Software)

Participants: Jonathan Muñoz, Thomas Watteyne.

The goal of this project is to maintain Wireshark dissectors for 6TiSCH (and 6TiSCH-related) standards up-to-date.

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

This implementation is done as part of the collaboration with Gridbee and the standardization activities of the Inria-EVA team at the IETF. It is published under an open-source BSD license and maintained at https://github.com/openwsn-berkeley/dissectors.

This software appears in https://bil.inria.fr/.

6.6. Mercator (Software)

Participants: Keoma Brun-Laguna, Thomas Watteyne.

Mercator allows one to evaluate the connectivity in a low-power wireless network. It is a collection of tools, including the firmware to load on the devices, the scripts that automate the measurements of the connectivity and the tools to structure and display results.

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

It is published under an open-source BSD license and maintained at https://github.com/openwsn-berkeley/mercator.

This software appears in https://bil.inria.fr/.

6.7. CONNEXION (Software)

Participants: Pascale Minet, Erwan Livolant.

In the CONNEXION project, the integration of the OCARI wireless sensor network in a Service-Oriented Architecture using the OPC-UA/ROSA middleware went on with Telecom ParisTech. More precisely, we developed the remote managment of the OCARI network as well as the possibility to generate commands to the sensors and actuators.

Erwan Livolant developed an OCARI frame dissector plugin for Wireshark (https://www.wireshark.org) available from the Git repository at OCARI website. 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.

6.8. SolSystem (Software)

Participants: Keoma Brun-Laguna, Thomas Watteyne.

SolSystem is a back-end solution for a low-power wireless mesh network based on SmartMesh IP. It defines how low-power wireless devices must format and transfer sensor data, and the tools to gather, store and display sensor data.

This system is used in several deployments, including http://savethepeaches.com/ and http://snowhow.io/.

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

It is published under an open-source BSD license. Information and overview at http://www.solsystem.io/.

This software appears in https://bil.inria.fr/.

6.9. Argus (Software)

Participants: Remy Leone, Thomas Watteyne.

Share your low-power wireless sniffer through the cloud!

Imagine you are a team of low-power wireless enthusiasts developing the next generation of products in standards. One essential tool in your toolkit is a low-power wireless sniffer, which shows you the frames which fly through the air.

Rather than requiring each person in your team to have a sniffer, Argus allows you to put in a share a sniffer through the cloud.

There are three piece to Argus:

- The Argus Probe is the program which attaches to your low-power wireless sniffer and forwards its traffic to the Argus Broker.
- The Argus Broker sits somewhere in the cloud. Based on MQTT, it connect Argus Probes with Argus Clients based on a pub-sub architecture.
- Several Argus Clients can the started at the same time. It is a program which subscribes to the Argus Broker and displays the frames in Wireshark.

It is published under an open-source BSD license, maintained at https://github.com/openwsn-berkeley/argus.

This software appears in https://bil.inria.fr/.

6.10. SAHARA (Software)

Participants: Ines Khoufi, Erwan Livolant, Pascale Minet.

Ines Khoufi developed modules for the simulation of the Time Slotted Channel Hopping (TSCH) on the ns3 simulation tool. These modules include multi-interface management and transmission management according to a given schedule.

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.

6.11. FIT IoT-LAB (Platform)

Participants: Thomas Watteyne, Tengfei Chang.

Note well: IoT-lab is *not* 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 highlights Inria-EVA activity and contribution to the IoT-lab testbed in 2016.

The Inria-EVA team has been using the platform extensively throughout 2016. During the process, we have been interacting closely with the IoT-lab team of engineers.

IoT-lab-related activities include:

- **Running OpenWSN networks at scale**. The IoT-lab has been an incredibly powerful tool to verify the scalability of the protocols and implementations in OpenWSN (Section 6.1). *Lead: Tengfei Chang*.
- **Assessing connectivity with Mercator**. The Mercator project (Section 6.6) is targetted at measuring the connectivity of the IoT-lab platform in time, space and frequency. *Lead: Keoma Brun-Laguna*.
- **16-channel Sniffer with Argus**. We are currently working extending the Argus project (Section 6.9) with support for IoT-lab motes. That is, rather than using a 16-channel sniffer, turn 16 IoT-lab nodes into a distributed multi-frequency sniffer. *Lead: Remy Leone*.
- **IoT-lab for Interop Testing with F-Interop**. Through the H2020 F-Interop project, we are developing the tools (see Section 6.3) to run an F-Interop user's code on the IoT-lab to verify conformance and interoperability. *Lead: Remy Leone*.
- **Scalable Security Solution with H2020 ARMOUR**. Through the H2020 ARMOUR project, Inria-EVA is testing whether security solutions standardized at the IETF are scalable. *Lead: Malisa Vucinic*.

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].

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 specifications, 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. Recent releases ("Seven") provide a new license (Cecill-B), a simplified build process and numerous features, including a simple macro system that connects the C input language and LISA annotations.

- 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. OPAM Builder

Participant: Fabrice Le Fessant.

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

- URL: http://github.com/OCamlPro/opam-builder

6.6. PASL

Participants: Michael 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/

6.7. TLAPS

Participants: Damien Doligez [**contact**], Stefan Merz [team Veridis], Martin Riener [team Veridis].

TLAPS is a platform for developing and mechanically verifying proofs about TLA+ specifications. The TLA+ proof language is hierarchical and explicit, allowing a user to decompose the overall proof into independent proof steps. TLAPS consists of a proof manager that interprets the proof language and generates a collection of proof obligations that are sent to backend verifiers. The current backends include the tableau-based prover Zenon for first-order logic, Isabelle/TLA+, an encoding of TLA+ as an object logic in the logical framework Isabelle, an SMT backend designed for use with any SMT-lib compatible solver, and an interface to a decision procedure for propositional temporal logic.

- URL: https://tla.msr-inria.inria.fr/tlaps/content/Home.html

6.8. 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://zenon-prover.org/

GANG Project-Team

5. New Software and Platforms

5.1. big-graph-tools

FUNCTIONAL DESCRIPTION

Gang is developing software for big graph manipulation. A preliminary library offering diameter and skeleton computation is available online [0]. This library was used to compute the diameters of the worldwide road network (200M edges), skeleton subtrees of the shortest-path trees of continental-sized road networks, as well as the largest strongly connected component of the Twitter follower-followee graph (23G edges).

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

[0] https://who.rocq.inria.fr/Laurent.Viennot/dev/big-graph-tools/

MAMBA Project-Team

6. New Software and Platforms

6.1. TiQuant

Tissue Quantifier
KEYWORDS: Systems Biology - Bioinformatics - Biology - Physiology
Systems biology and medicine on histological scales require quantification of images from histological image modalities such as confocal laser scanning or bright field microscopy. The latter can be used to calibrate the initial state of a mathematical model, and to evaluate its explanatory value, which hitherto has been little recognised. We generated a software for image analysis of histological material and demonstrated its use in analysing liver confocal micrografts, called TiQuant (Tissue Quantifier). The software is part of an analysis chain detailing protocols of imaging, image processing and analysis in liver tissue, permitting 3D reconstructions of liver lobules down to a resolution of less than a micrometer [72]. It is implemented in portable object-oriented ANSI C++. The GUI is based on QT and supports real-time visualisation using OpenGL. TiQuant is embedded in the tissue modelling framework CellSys and thus is tightly linked with TiSim, a versatile and efficient simulation environment for tissue models. TiQuant provides an interface to VolView and further complements its functionality by linking to the open-source libraries ITK and VTK (itk/vtk.org). The image/volume processing chains currently implemented in TiQuant for example include techniques to segment conduit and cell segmentation from 3D confocal micrographs of liver tissue based on the Adaptive Otsu Thresholding method and a number of morphological operators [75]. TiQuant is currently extended by a machine learning component.

FUNCTIONAL DESCRIPTION

We generated a software for image analysis of histological material and demonstrated its use in analysing liver confocal micrografts, called TiQuant (Tissue Quantifier). The software is part of an analysis chain detailing protocols of imaging, image processing and analysis in liver tissue, permitting 3D reconstructions of liver lobules down to a resolution of less than a micrometer.

- Contact: Dirk Drasdo
- URL: http://www.msysbio.com

6.2. TiSim

Tissue Simulator
KEYWORDS: Systems Biology - Bioinformatics - Biology - Physiology
FUNCTIONAL DESCRIPTION

TiSim (Tissue Simulator) is a software for agent-based models of multicellular systems. It permits model development with centre-based models and deformable cell models; it contains modules for monolayer and multicellular spheroid simulations as well as for simulations of liver lobules. Besides agent-based simulations, the flow of blood and the transport of molecules can be modelled in the extracellular space; intracellular processes such as signal transduction and metabolism can be simulated, for example over an interface permitting integration of SBML-formulated ODE models.

TiSim is written in modern C++, keeping central model constituents in modules to be able to reuse them as building blocks for new models. For user interaction, the GUI Framework Qt is used in combination with OpenGL for visualisation. A non-interactive mode to use the software also exists, accepting a combination of XML and HDF5 (hierarchical data format v5) as input, which produces output data in VTP (VTK) and HDF5 format. SBML, SBML_ODESolver and sundials are deployed for the creation and solution of the differential equations of metabolic networks and signalling pathways presented in SBML data format. TiSim permits agent-based simulations of multicellular systems and can be directly fed by processed image data from TiQuant.

- Contact: Dirk Drasdo
- URL: (No url yet)

MATHERIALS Project-Team

5. New Software and Platforms

5.1. SIMOL

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

- Contact: Gabriel Stoltz
- URL: https://gitlab.inria.fr/matherials/simol

MATHRISK Project-Team

6. New Software and Platforms

6.1. PREMIA

KEYWORDS: Financial products - Computational finance - Option pricing

6.2. Scientific Description

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

6.3. Functional Description

- Participants: Mathrisk project team and contributors
- Partners: Ecole des Ponts ParisTech - Inria - 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)
- OS/Middelware: Linux, Mac OS X, Windows
- APP: The development of Premia started in 1999 and 17 are released up to now and registered at the APP agency. Premia 16 has been registered on 0303/2015 under the number IDDN.FR.001.190010.013.S.C.2001.000.31000
- Programming language: C/C++
- Documentation: scientific documentation of all the algorithm implemented. PNL has a 100 pages user documentation
- Size of the software: For the Src part of Premia : 337046 lines , that is 14 Mbyte of code, and 117 Mbyte of PDF files of documentation; For PNL: 747952 lines , that is 25 MO.
- interfaces : Nsp for Windows/Linux/Mac, Excel, binding Python, and a Web interface.
- Publications: [15], [68], [75], [83], [86], [56], [66].

6.4. Content

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.

- **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.

- **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.

- **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)

- **Hybrid products**

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

- **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.

6.5. PNL numerical library

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.

This year, Jérome Lelong has performed the following tasks on the development of PNL:
Releases 1.7.3 and 1.7.4. of the *PNL* library (http://pnl.gforge.inria.fr/).

1. Simplify the use of PNL under Visual Studio. It can either be compiled using CMake or added as an external library to an existing project.
2. Improve the construction of large `PnlBasis` objects and make it possible to deal with non tensor functions.
3. Add complex error functions.

6.6. Consortium Premia

The software Premia is supported by a Consortium of financial institutions created in 1999. The members of the Consortium give an annual financial contribution and receive every year a new version enriched with new algorithms. They participate to the annual meeting where future new developments are discussed.

6.7. Diffusion

All releases of the software Premia (18 in 2016) are registered at the French agency APP. The most recent is provided to the Consortium with an appropriate licence. An opensource version is also available for academic purposes. The software is thus used in many universities, with in France and abroad.

6.8. Algorithms implemented in Premia in 2016

Premia 18 has been delivered to the consortium members in March 2016.

It contains the following new algorithms:

6.8.1. Insurance, Risk Management, Optimal Trade Execution

- A Forward Solution for Computing Derivatives Exposure. M. Ben Taarit, B. Lapeyre.
- Monte Carlo Calculation of Exposure Profiles and Greeks for Bermudan and Barrier Options under the Heston Hull-White Model. Q. Feng, C.W. Oosterlee.
- Dynamic optimal execution in a mixed-market-impact Hawkes price model. A. Alfonsi, P. Blanc. *Finance & Stochastics*
- A Hamilton Jacobi Bellman approach to optimal trade execution. P. Forsyth, *Applied Numerical Mathematics 61, 2011.*

6.8.2. Equity Derivatives

- Value function approximation or stopping time approximation: a comparison of two recent numerical methods for American option pricing using simulation and regression. L. Stentoft, *Journal of Computational Finance, 18(1), 2014.*
- Pricing American-Style Options by Monte Carlo Simulation: Alternatives to Ordinary Least Squares. S. Tompaidis, C. Yang, *Journal of Computational Finance, 18(1), 2014,*
- Solving Optimal Stopping Problems using Martingale Bases. J.Lelong
- The Stochastic Grid Bundling Method: Efficient Pricing of Bermudan Options and their Greeks. S. Jain, C.W. Oosterlee.
- Two-dimensional Fourier cosine series expansion method for pricing financial options. C.W.Oosterlee M.J. Ruijter, *SIAM J. Sci. Comput., 34(5), 2012.*
- Estimation of the parameters of the Wishart process. A.Alfonsi, A.Kebaier, C.Rey, *Preprint*
- The 4/2 Stochastic Volatility Model. M. Grasselli, *Preprint.*
- Ninomiya Victoir Scheme and Multi Level Scheme. A. Al Gerbi, E. Clement, B. Jourdain.
- Importance Sampling for Multilevel Monte Carlo. A.Kebaier J.Lelong
- The evaluation of barrier option prices under stochastic volatility. C.Chiarella, B.Kang, G.H.Meyer, *Computers and Mathematics with Applications 64, 2012.*
- Volatility swaps and volatility options on discretely sampled realized variance. G.Lian, C.Chiarella, P.S.Kalev, *Journal of Economic Dynamics Control 47 2014*
- Efficient variations of the Fourier transform in applications to option pricing. S. Boyarchenko and S.Levendorski, *Journal of Computational Finance, 18(2), 2014.*
- Model-free implied volatility: from surface to index. M. Fukasawa et al., *Int. J. Theor. Appl. Finan. 14, 433, 2011*

- Stratified approximations for the pricing of options on average. N.Privault J.Yu *Journal of Computational Finance.*

Moreover, J. Lelong has ensured everyday maintenance to fix various bugs, especially related to Visual C++; and has get rid of the old bunch of scripts to generate the HTML documentation by implementing the required mechanism directly in TeX. This makes the system much more robust; He has also worked on the continuous integration process with Sébastien Hinderer. Moreover, part of Premia documentation is now generated directly from the source code. The 3000 lines of undocumented C code used so far had become unmaintainable. Now, it is replaced by a much more flexible and efficient Python script.

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], Siddhartha Dutta.

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, Things using CoAP [0]. 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 *GIDL*, 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 GIDL, 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;

[0] https://tools.ietf.org/html/rfc7252

- VSB provides support for the client-server, publish/subscribe, tuple space and data streaming interaction paradigms;
- 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: we have developed related tools that support the automated generation of a BC for a service/Thing from its GIDL interface description.

VSB is being developed within the H2020 CHOReVOLUTION project (see § 8.2.1.1). It is also based on previous development carried out in the FP7 CHOReOS project [0]. VSB is available for download under open source license.

6.3. Ambiciti App & Platform: Monitoring the Exposure to Environmental Pollution

Participants: Valerie Issarny [contact], Cong Kinh Nguyen, Pierre-Guillaume Raverdy, Fadwa Rebhi.

URL: http://ambiciti.io

Is your exposure to noise too high on certain days? How is air pollution in your street? Will air quality improve in the next hours? Do you want to measure the noise pollution on the way between your home and your office? What pollution levels are considered harmful for your health? Ambiciti (previously SoundCity) provides answers to these questions and many others through dedicated Apps and Platforms that leverage Inria research results in the area of mobile distributed systems (from MiMove team) and data assimilation (from Inria CLIME team). The Ambiciti app is available for download on both the App and the Play stores. Starting December 2016, the Ambiciti software solutions are licensed to the Ambiciti start-up.

Monitoring exposure to noise pollution: Noise pollution is a major environmental health problem, with an estimated number of 10,000 premature deaths each year in Europe. The 2002 European environmental noise directive defines a common approach intended to avoid, prevent or reduce the harmful effects of noise. It requires the determination of exposure to environmental noise in major cities, through noise mapping. Until recently, this has been done solely through numerical simulation. Daytime, evening and nighttime averages are generally produced, without distinction between the different days of the year. Also, it is difficult to fill the gap between a noise map and the actual exposure of people where they live and stay. This motivates to monitor noise pollution where and when people are exposed. One promising direction is to make use of the noise sensors that people carry most of the time, i.e., the microphones embedded in their mobile phones.

Ambiciti (previously called SoundCity as it was initially focused on the monitoring of noise pollution) measures the actual noise levels to which individuals are exposed using such an approach, while taking into account the relatively low quality of the collected measurements. Ambiciti can then monitor noise levels throughout the day and inform users about their instantaneous, hourly and daily exposures.

In addition to the monitoring of the individual exposure to the noise pollution using mobile phones, the collective exposure may be derived from crowd-sensing. The adoption rate of mobile phones makes it possible to collect a huge amount of observational data about the noise pollution at the city scale. Recent studies have indeed highlighted the emergence of new environmental monitoring schemes leveraging the combination of mobile phones-embedded sensors and citizen participation. Ambiciti then leverages a mobile phone sensing middleware for collecting noise measurements at the urban scale, which are then assimilated toward the production of real-time pollution maps.

[0] http://www.choreos.eu

Monitoring exposure to air pollution: The Ambiciti app delivers information about the exposure to air pollution, providing hourly air quality maps, which are computed using numerical simulation. Depending on the user's location, the user may have access to hourly air quality maps, at street resolution, in real time and for the next two days. Currently, only Paris (France) and the Bay area enjoy such high resolution maps, but other cities are on the way to be included.

Mobile app features: Ambiciti is easy to use, while featuring various functions to meet different levels of user engagement (from passive monitoring to citizen scientists):

- Measuring noise level, anytime on demand or automatically during the day,
- Providing air quality indexes (according to the EU definition), in the past, present and future hours or days, together with pollution levels for nitrogen dioxide, fine particulate matter and ozone,
- Displaying statistics on exposure to pollutions, hourly, daily, during daytime and nighttime, for both noise and air.
- Displaying maps with own's noise measurements and/or hourly NO2 levels (including street-level resolution for Paris and the San Francisco Bay area)
- Promoting citizen science where communities of users may engage into the intensive measurement of noise in order to analyze a given journey or to map neighborhoods.
- Providing pollution-aware routing.
- Calibration of the smartphone for noise monitoring, automatically with Ambiciti database or manually with a sound level meter,

Privacy: The Ambiciti app has been designed with privacy in mind, which especially holds in the case of noise pollution monitoring. It is in particular important to stress that actual sound samples are never stored on the phone or uploaded to the Ambiciti servers. Only the amplitude of the sound in dB(A) is calculated, and uploaded provided the user's permission. An anonymous identifier is further created for each device for distinguishing between the data sent by different users, while no identifying information is collected. Further detail may be found from the App information page.

The Ambiciti platform is developed in collaboration with the Inria CLIME team together with The Civic Engine and the NUMTECH SMEs in the context of CityLab@Inria and Inria@SiliconValley, and with the support of the EIT Digital Env&You activity.

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

Participants: Valérie Issarny [contact], Cristhian Parra Trepowski, Rafael Angarita.

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.

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, together with the support of the EIT Digital CivicBudget activity.

MOKAPLAN Project-Team

6. New Software and Platforms

6.1. 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 headlightsâŠ Mathematical modeling of this problem, related to the optimal transport theory, takes the form of a nonlinear Monge-Ampere type PDE. The numerical resolution of these models remained until recently a largely open problem. MOKABAJOUR project aims to develop, using algorithms invented especially at Inria and LJK, a reflector design software more efficient than geometrical methods used so far. Different solvers

- Participants: Simon Legrand, Jean-David Benamou, Quentin Merigot and Boris Thibert
- Contact: Jean-David Benamou
- URL: https://project.inria.fr/mokabajour/

6.2. Wasserstein Fisher Rao

Lenaic Chizat

https://github.com/lchizat/optimal-transport

This Julia toolbox provides several tools for solving optimal transport, the unbalanced extensions and related problems.

6.3. OT and Sparse Scaling

Bernhard Schmitzer

http://wwwmath.uni-muenster.de/num/wirth/people/Schmitzer/index.html?content=code

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 Stephane 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://team.inria.fr/muse/

5.3. Online HoA

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

"Home or Access" (HoA) is a system that localizes performance problems in home and access networks. Originally, we implement HoA as custom firmware that collect traces from off-the-shelf home routers. HoA uses timing and buffering information from passively monitored traffic at home routers to detect both access link and wireless network bottlenecks. HoA runs offline on a server to locate last-mile downstream throughput bottlenecks based on the analysis of packet traces collected from home routers. Our attempts to run HoA online on commodity home routers, however, revealed the challenges with performing per-packet analysis on such resource-constrained devices. The online HoA resolves this issue. We design an access bottleneck detector based on lightweight pings of the access link, and a wireless bottleneck detector based on a model of wireless capacity using metrics that are easily available in commodity home routers such as the wireless physical rate and the count of packets/bytes transmitted.

- 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: https://team.inria.fr/muse/

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://team.inria.fr/muse/

5.6. WeBrowse

FUNCTIONAL DESCRIPTION

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

- Contact: Giuseppe Scavo
- URL: http://webrowse.polito.it/

MUTANT Project-Team

5. New Software and Platforms

5.1. Antescofo

Anticipatory Score Following and Real-time Language
FUNCTIONAL DESCRIPTION. Antescofo is a modular polyphonic Score Following system as well as a Synchronous Programming language for musical composition. The first 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 (DSL) within Antescofo allows flexible writing of time and interaction in computer music.

- Participants: Arshia Cont, Jean-Louis Giavitto, Florent Jacquemard and José Echeveste
- Contact: Arshia Cont
- URL: http://forumnet.ircam.fr/product/antescofo/

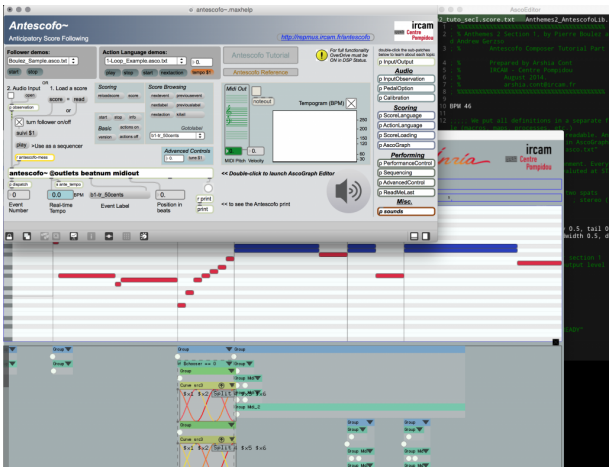

Figure 2. Antescofo and AscoGraph Screenshots

The design of the Antescofo DSL clearly benefits of a strong and continuous involvement in the production of world-class composer pieces and their continuous recreation throughout the world. These interactions motivate new developments, challenge the state of the art and in return, opens new creative dimensions for composers and musicians. The maturity of the system is assessed by the generalization of its use in a large proportion of Ircam new productions, and its use outside Ircam all around the world (Brasil, Chile, Cuba, Italy, China, US, etc.). Antescofo enjoys an active community of 150 active users: http://forumnet.ircam.fr/user-groups/antescofo/

5.2. OMRQ

Library for rhythm transcription integrated in the assisted composition environment OpenMusic.

FUNCTIONAL DESCRIPTION. Rhythm transcription is the conversion of sequence of timed events into the structured representations of conventional Western music notation. Available as a graphical component of OpenMusisc, the library OMRQ privileges user interactions in order to search for an appropriate balance between different criteria, in particular the precision of the transcription and the readability of the musical scores produced.

This system follows a uniform approach, using hierarchical representations of timing notations in the form of rhythm trees, and efficient parsing algorithms for the lazy enumeration of solutions of transcription.

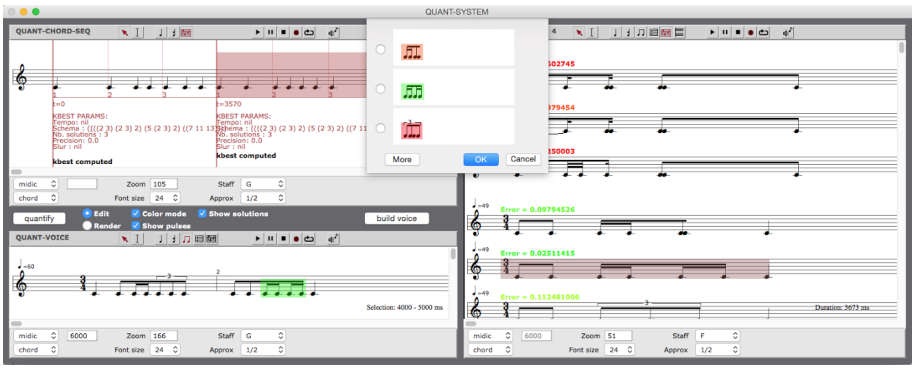

Figure 3. Screenshot of the Open Music Rhythm Quantization library

Its implementation is carried out via a dedicated interface allowing interactive exploration of the solutions space, their visualization and local editing, with particular attention to the processing of grace notes and rests.

- Participants: Florent Jacquemard and Adrien Ycart
- Contact: Florent Jacquemard
- URL: http://repmus.ircam.fr/cao/rq, https://bil.inria.fr/fr/software/view/2904/tab

5.3. Antescofo Timed Test Platform

Timed testing plateform for Antescofo.
FUNCTIONAL DESCRIPTION. 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 [50], 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.

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

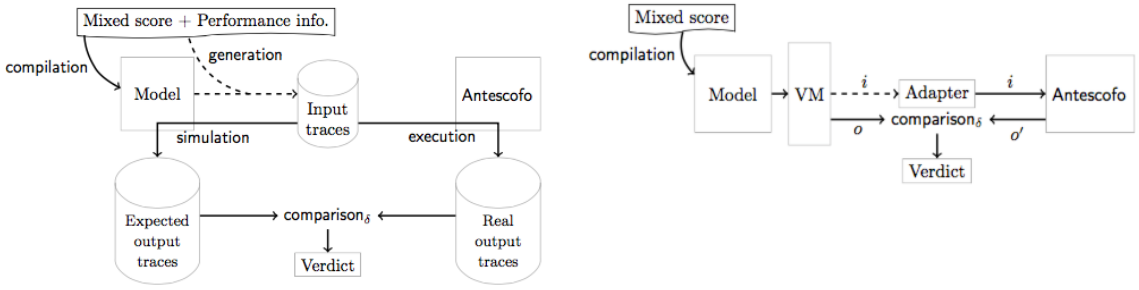

Figure 4. Offline and Online workflows for Antescofo Model Based Testing

These implementations have been conducted as a part of Clément Poncelet's PhD Thesis.

5.4. Ascograph

The Antescofo graphical score editor.

FUNCTIONAL DESCRIPTION. AscoGraph, 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 (Figure 5). 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.

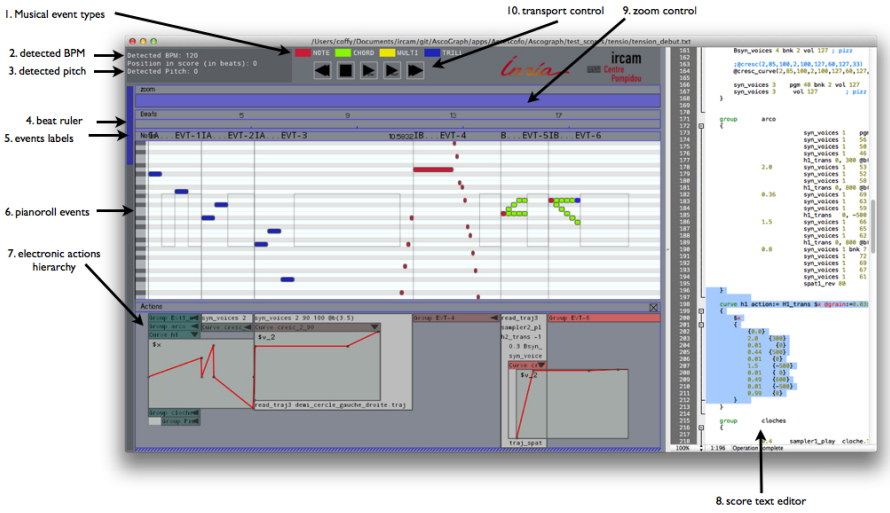

Figure 5. Screenshot of Ascograph, the Antescofo graphical score editor

MYCENAE Project-Team

6. New Software and Platforms

6.1. DynPeak

KEYWORDS: Biology - Health - Physiology
SCIENTIFIC DESCRIPTION

DynPeak is an algorithm for pulse detection and frequency analysis in hormonal time series. A new release of the DynPeak Scilab atom toolbox has been delivered in 2016 https://atoms.scilab.org/toolboxes/Dynpeak/2.1.0

- Participants: Frédérique Clement, Serge Steer, Thierry Martinez
- Partner: INRA
- Contact: Frédérique Clement
- URL: https://team.inria.fr/mycenae/en/software/

PARKAS Project-Team

5. New Software and Platforms

5.1. Cmmtest

FUNCTIONAL DESCRIPTION

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

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

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

compares the recorded trace with a reference trace for the same program, checking if the recorded trace can be obtained from the reference trace by valid eliminations, reorderings and introductions.

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

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

5.2. GCC

KEYWORDS: Compilation - Polyhedral compilation
FUNCTIONAL DESCRIPTION

The GNU Compiler Collection includes front ends for C, C++, Objective-C, Fortran, Java, Ada, and Go, as well as libraries for these languages (libstdc++, libgcj,...). GCC was originally written as the compiler for the GNU operating system. The GNU system was developed to be 100

- Participants: Albert Cohen, Tobias Grosser, Feng Li, Riyadh Baghdadi 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 Synchrone, without type inference, type polymorphism and higher-order. It is thus a Lustre-like language extended with hierchical automata in a form very close to SCADE 6. The intention for making this new language and compiler is to develop new aggressive optimization techniques for sequential C code and compilation methods for generating parallel code for different platforms. This explains much of the simplifications we have made in order to ease the development of compilation techniques.

- Participants: Adrien Guatto, Marc Pouzet, Cédric Pasteur, Léonard Gerard, Brice Gelineau, Gwenael Delaval and Eric Rutten
- Contact: Marc Pouzet

5.4. Lem

lightweight executable mathematics
FUNCTIONAL DESCRIPTION

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

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

5.5. Lucid Synchrone

FUNCTIONAL DESCRIPTION

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

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

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. Ott

FUNCTIONAL DESCRIPTION

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

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

a Coq version of the definition,

an Isabelle version of the definition, and

a HOL version of the definition.

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

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

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

5.8. 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, Riyadh Baghdadi and Albert Cohen
- Contact: Sven Verdoolaege
- URL: http://freshmeat.net/projects/ppcg

5.9. 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, Louis Mandel and Cédric Pasteur
- Contact: Guillaume Baudart
- URL: http://rml.lri.fr

5.10. SundialsML

Sundials/ML
KEYWORDS: Simulation - Mathematics - Numerical simulations
SCIENTIFIC DESCRIPTION

Sundials/ML is an OCaml interface to the Sundials suite of numerical solvers (CVODE, CVODES, IDA, IDAS, KINSOL, ARKODE). It supports all features except for the Hypre and PETSC nvectors (which require additional libraries). Its structure mostly follows that of the Sundials library, both for ease of reading the existing documentation and for adapting existing source code, but several changes have been made for programming convenience and to increase safety, namely:

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

FUNCTIONAL DESCRIPTION

Sundials/ML is an OCaml interface to the Sundials suite of numerical solvers (CVODE, CVODES, IDA, IDAS, KINSOL, ARKODE).

NEW PROGRESS

This year we updated our interface to work with versions 2.6.0 and 2.7.0 of the Sundials library. This included significant work to support the new ARKODE solver, sparse matrices and the KLU and SuperLU/MT linear solvers, OpenMP and Pthreads nvectors, and various new functions and linear solvers in existing solvers. The source files were completely reorganized. The OCaml types for nvectors were adapted to support multiple nvectors. Memory leaks were eliminated and the performance problems investigated. This work was presented at the ACM Workshop on ML [28].

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

5.11. Zélus

SCIENTIFIC DESCRIPTION

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

FUNCTIONAL DESCRIPTION

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

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

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

PI.R2 Project-Team

4. New Software and Platforms

4.1. 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, ...

- Closest participants: Benjamin Grégoire, Enrico Tassi, Bruno Barras, Yves Bertot, Pierre Courtieu, Maxime Dénès, Hugo Herbelin, Matej Košík, Pierre Letouzey, Assia Mahboubi, Cyprien Mangin, Guillaume Melquiond, Jean-Marc Notin, Pierre-Marie Pédrot, Yann Régis-Gianas, Matthieu Sozeau, Arnaud Spiwack, Théo Zimmermann.
- Partners: CNRS - ENS Lyon - Université Paris-Diderot - Université Paris-Sud
- Contact: Matthieu Sozeau
- URL: http://coq.inria.fr/

4.1.1. Coq 8.6

The 8.6 version of Coq was released in December 2016. It initiates a time-based release cycle and concentrates on a smaller set of features than Coq 8.5 for which compatibility and testing were done more intensively. In the πr^2 team, Hugo Herbelin, Cyprien Mangin, Théo Zimmermann and Matthieu Sozeau contributed to the coordination of the development, to the discussion of the roadmap, to the implementation of the features, and to many bugfixes.

Matthieu Sozeau followed up his work on universe polymorphism making the explicit annotation system more accessible and resolving issues in the minimization algorithm used during refinement, resulting in a more predictable system. These improvements were used in the Coq/HoTT library for Homotopy Type Theory, which is described in an upcoming article [26].

Matthieu Sozeau implemented a new variant of the proof-search tactic for typeclasses that is set to replace the existing auto and eauto tactics in the following version. The new variant fully benefits from the features of the underlying proof engine, and allows much more control on proof-search (patterns used consistently, modes for triggering hints, ...). It is at the basis of the work of Théo Zimmermann described below.

4.1.2. The Equations plugin

Cyprien Mangin and Matthieu Sozeau continued work on the Equations plugin, modularizing it so that the use of axioms can be minimized, and making it compatible with developments in Homotopy Type Theory. To achieve this, it has moved to a simplification engine in ML based on telescopes and is able to produce axiom-free proofs of the examples that were previously implicitly using them. This work will be presented at the POPL workshop Type-Theoretic Tools (TTT), next January 2017.

4.1.3. Maintenance

Among other contributions, Hugo Herbelin, Pierre Letouzey, Matej Košík and Matthieu Sozeau worked at the maintenance of the system.

In particular, Pierre Letouzey vastly reworked the build mechanism of Coq, taking advantage of code evolutions driven by Pierre-Marie Pédrot. Pierre Letouzey also administrated (and improved) several machines or systems that are critical for the Coq community (web server, build test server, git repositories ...), in coordination with Inria's SIC support team.

Matej Košík developed a new benchmarking infrastructure based on Jenkins and continuous integration (http://ci.inria.fr). It allows easily testing any developer branch on the benchmark suite prior to integration to the main archive.

4.1.4. Coordination and animation

After 10 years coordinating the Coq development team, Hugo Herbelin handed over the coordination to Matthieu Sozeau.

A Coq working group is organised every two months (5 times a year). Discussions about the development happen, in particular, on `coq-dev@inria.fr`, Coq's GitHub http://github.com/coq and http://coq.inria.fr/bugs. This year, a week-long working group organized in Sophia-Antipolis was devoted to the 8.6 roadmap discussion.

4.1.5. Documentation and stabilization of Coq's programming interface

Matej Košík worked on the programming interfaces of Coq, starting to isolate a subset of key functions to be used by Coq plugin developers.

4.2. Other software developments

In collaboration with François Pottier (Inria Gallium), Yann Régis-Gianas maintained Menhir, an LR parser generator for OCaml. Yann Régis-Gianas 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. Yann Régis-Gianas develops a reference implementation of a syntactic analyzer for the POSIX shell programming language. This analyzer is used by the Colis project to analyze the scripts embedded in the packages of the Debian GNU/Linux distribution. 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.

Yves Guiraud has updated the Catex tool for Latex, whose purpose is to automatise the production of string diagrams from algebraic expressions http://www.irif.fr/~guiraud/catex/catex.zip. Yves Guiraud collaborates with Samuel Mimram (LIX) to develop the prototype Rewr that implements several algorithms developed in the "Effective higher-dimensional algebra" research direction, including the homotopical completion-reduction procedure of [10]. An online version is available at http://www.lix.polytechnique.fr/Labo/Samuel.Mimram/rewr.

POLSYS Project-Team

5. New Software and Platforms

5.1. Epsilon

FUNCTIONAL DESCRIPTION

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

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

5.2. FGb

FUNCTIONAL DESCRIPTION

FGb is a powerful software for computing Groebner bases. It includes the new generation of algorihms for computing Gröbner bases polynomial systems (mainly the F4,F5 and FGLM algorithms).It is implemented in C/C++ (approximately 250000 lines), standalone servers are available on demand. Since 2006, FGb is dynamically linked with Maple software (version 11 and higher) and is part of the official distribution of this software.

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

5.3. FGb Light

FUNCTIONAL DESCRIPTION

Gröbner basis computation modulo p (p is a prime integer of 16 bits).

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

5.4. GBLA

FUNCTIONAL DESCRIPTION

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

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

5.5. HFEBoost

FUNCTIONAL DESCRIPTION

Public-key cryptography system enabling an authentification of dematerialized data.

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

5.6. RAGlib

Real Algebraic Geometry library
FUNCTIONAL DESCRIPTION

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

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

5.7. 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 hundrends of Megabytes. Currently the code consists of $\sim 5\,000$ lines.

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

5.8. SPECTRA

Semidefinite Programming solved Exactly with Computational Tools of Real Algebra
FUNCTIONAL DESCRIPTION

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

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

PROSECCO Project-Team

6. New Software and Platforms

6.1. ProVerif

Participants: Bruno Blanchet [correspondant], Xavier Allamigeon [April–July 2004], Vincent Cheval [Sept. 2011–Dec. 2014], Benjamin Smyth [Sept. 2009–Feb. 2010], Marc Sylvestre [Oct. 2016–].

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–Dec. 2013].

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], Cedric Fournet [Microsoft Research], Markulf Kohlweiss [Microsoft Research], Antoine Delignat-Lavaud [Microsoft Research], Nikhil Swamy [Microsoft Research], Santiago Zanella-Béguelin [Microsoft Research], Jean Karim Zinzindohoué, Benjamin Beurdouche, Alfredo Pironti.

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.

Papers describing the miTLS library was published at IEEE S&P 2013, CRYPTO 2014, and IEEE S&P Journal 2016. miTLS is now being developed on GitHub with dozens of contributors and regular updates. The miTLS team was awarded the Levchin prize for contributions to Real-World Cryptography in 2016. The software and associated research materials are available from http://mitls.org.

6.4. F*

Participants: Alejandro Aguirre, Danel Ahman [University of Edinburgh], Benjamin Beurdouche, Karthikeyan Bhargavan, Antoine Delignat-Lavaud [Microsoft Research], Cédric Fournet [Microsoft Research], Catalin Hritcu, Chantal Keller [Université Paris-Sud], Kenji Maillard, Guido Martínez, Gordon Plotkin, Samin Ishtiaq [Microsoft Research], Markulf Kohlweiss [Microsoft Research], Jonathan Protzenko [Microsoft Research], Tahina Ramananandro [Microsoft Research], Aseem Rastogi [Microsoft Research], Nikhil Swamy [Microsoft Research], Peng Wang [MIT], Santiago Zanella-Béguelin [Microsoft Research], Jean Karim Zinzindohoué.

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

A detailed description of F* (circa 2011) appeared in the Journal of Functional Programming [44]. 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.

6.5. HACL*

Participants: Karthikeyan Bhargavan, Jean Karim Zinzindohoué, Marina Polubelova, Benjamin Beurdouche, Jonathan Protzenko [Microsoft Research].

HACL* is a verified cryptographic library written in F*. It implements modern primitives, including elliptic curves like Curve25519, symmetric ciphers like Chacha20, and MAC algorithms like Poly1305. These primitives are then composed into higher-level constructions like Authenticated Encryption with Additional Data (AEAD) and the NaCl API. All the code in HACL* is verified for memory safety, side channel resistance, and where applicable, also for functional correctness and absence of integer overflow. HACL* code is used as the basis for cryptographic proofs for security in the miTLS project.

HACL* code can be compiled to OCaml using the standard F* compiler, or to C, using the Kremlin backend of F*. The generated C code is as fast as state-of-the-art cryptographic libraries written in C. HACL* is being actively developed on Github; see https://github.com/mitls/hacl-star

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 propertly 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 Signal and TLS 1.3 protocols in JavaScript.

The ProScript compiler and various libraries written in ProScript are being developed on Github and will be publicly released in 2017.

6.7. QuickChick

Participants: Maxime Dénès [Inria Sophia-Antipolis], Catalin Hritcu, John Hughes [Chalmers University], Leonidas Lampropoulos [University of Pennsylvania], Zoe Paraskevopoulou [Princeton University], Benjamin Pierce [University of Pennsylvania].

QuickChick is a verified plugin that integrates property-based testing and proving in the Coq proof assistant. This integration is aimed at reducing the cost of formal verification and at providing stronger, formal foundations to property-based testing. https://github.com/QuickChick/QuickChick

6.8. Luck

Participants: Leonidas Lampropoulos [University of Pennsylvania], Diane Gallois-Wong [ENS and Inria Paris], Catalin Hritcu, John Hughes [Chalmers University], Benjamin Pierce [University of Pennsylvania], Li-Yao Xia [ENS and Inria Paris].

Property-based random testing a la QuickCheck requires building efficient generators for well-distributed random data satisfying complex logical predicates, but writing these generators can be difficult and error prone. We propose a domain-specific language in which generators are conveniently expressed by decorating predicates with lightweight annotations to control both the distribution of generated values and the amount of constraint solving that happens before each variable is instantiated. This language, called Luck, makes generators easier to write, read, and maintain. https://github.com/QuickChick/Luck

6.9. Privacy-preserving federated identity

Participants: Harry Halpin, George Danezis [University College London].

security protocols, secure messaging, decentralization, blockchain

Working with the partners in the NEXTLEAP project, we helped design a protocol for decentralized and privacy-preserving identity and key management, creating both a fix privacy vulnerabilities in OAuth (UnlimitID) and on generic versions of blockchain for usages such as key management (ClaimChain). While this software has been prototyped in 2016 using Python working with NEXTLEAP project partner University College London (George Danezis), we expected either formal analysis using ProVerif or verified implementations using Fstar in 2017, as well and integrate this work with formal verification work done in Prosecco on end-to-end secure messaging.

QUANTIC Project-Team (section vide)

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 [23] , for elastic version management, for adaptive replication, for partial replication, and for reconfigurable sharding.

- Participants: Tyler Crain, Marc Shapiro 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 and Masoud Saeida Ardekani
- 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. Furthemore, the collector features several memory placement heuristics that improve locality.

- Participants: Lokesh Gidra, Marc Shapiro, Julien Sopena and Gaël Thomas
- Contact: Lokesh Gidra
- URL: http://gforge.inria.fr/projects/transgc/

REO Project-Team

6. New Software and Platforms

6.1. cardioXcomp

KEYWORDS: Cardiac Electrophysiology - Safety Pharmacology
FUNCTIONAL DESCRIPTION

cardioXcomp is a software dedicated to the safety pharmacology industry. It is developed in the framework of the joint laboratory (LabCom) "cardioXcomp" with the software company Notocord. Its purpose is to model the electrical potential of cardiomyocytes measured by a microelectrode array (MEA), and to model the effect of drugs on this signal. It was registered in November 2015 at the Agence pour la Protection des Programmes under the Inter Deposit Digital Number IDDNFR.001.480003.000.S.P.2015.000.31230.

- Participants: Jean-Frédéric Gerbeau, Fabien Raphel, Nejib Zemzemi
- Contact: Jean-Frédéric Gerbeau

6.2. FELiScE

Finite Elements for Life SCiences and Engineering problems
KEYWORDS: Finite element modeling - 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 Ángel Fernández Varela, Jean-Frédéric Gerbeau, Philippe Moireau, Marina Vidrascu, Sebastien Gilles, Benoit Fabreges, Axel Fourmont, Mikel Landajuela Larma, Damiano Lombardi, Matteo Aletti, Irene Vignon-Clementel and Faisal Amlani
- Contact: Jean-Frédéric Gerbeau
- URL: http://felisce.gforge.inria.fr

6.3. MODULEF

FUNCTIONAL DESCRIPTION

MODULEF is a legacy finite element library developed at Inria since the 1980's. Here, we limit ourselves to recent developments done within this library.

A numerical method to approximate the constitutive laws for rubber elasticity derived from polymer physics are implemented in Modulef.

It is based on algorithms from stochastic geometry to generate suitable polymer networks, Delaunay tessellation algorithms to deal with steric effects (courtesy of the Inria project-team GAMMA2), the introduction of 1-dimensional finite elements for the polymer-chains in Modulef.

- Participants: Marina Vidrascu and Antoine Gloria
- Contact: Marina Vidrascu
- URL: https://www.rocq.inria.fr/modulef/

6.4. 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/shelddon/

RITS Project-Team

5. New Software and Platforms

5.1. 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.

- Authors: Raoul de Charette, Fawzi Nashashibi and Evangeline Pollard
- Contact: Fawzi Nashashibi

5.2. 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: Joshue Perez Rastelli and Vicente Milanés
- Contact: Fawzi Nashashibi

5.3. 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: Fawzi Nashashibi

5.4. PML-SLAM

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

5.5. Platools

KEYWORD: Telecommunications

- Participant: Marios Makassikis
- Contact: Thierry Ernst

5.6. 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

5.7. 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

5.8. Taxi-col

KEYWORD: Mobile Computing, Transportation

- Participant: Eugenie Lioris
- Contact: Fawzi Nashashibi

5.9. V2Provue

Vehicle-to-Pedestrian
FUNCTIONAL DESCRIPTION

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

On the pedestrian side, the application is relying on GPS data to localize the user and Wi-Fi communications are used to receive messages about close vehicles and send information about the pedestrian positioning. Besides, a service has been developed to evaluate the collision risk with the vehicles near the pedestrian and an HMI based on OpenStreetMap displays all the useful information such as pedestrian and vehicles localization and, collision risk.

On the vehicle side, RtMaps modules allowing V2X communications have been developed. These modules contain features such as TCP/UDP socket transmissions, broadcast, multicast, unicast communications, routing, forwarding algorithms, and application specific modules. In the V2ProVu software, a particular application module has been implemented to create data packets containing information about the vehicle state (position, speed, yaw rate,...) and the V2X communication stack is used to broadcast these packets towards pedestrians. Moreover, the V2proVu application can also receive data from pedestrians and create objects structures that can be shared with the vehicle perception tools.

- Contact: Fawzi Nashashibi

5.10. SimConVA

Participants: Pierre Merdrignac, Oyunchimeg Shagdar, Jean-Marc Lasgouttes.

This software was developed during the SINETIC FUI project. It aims at interfacing the network simulator ns-3 and the prototyping software RTMaps.

The source code of is software is a library to generate an RTMaps component that triggers and controls ns-3. The component handle emission and reception of data packets between RTMaps and ns-3 for every vehicle. It can also deal with the mobility of vehicles in ns-3 based on the localization known in ns-3.

This software was used with the communication stack developed in RITS. It has been shown during the SINETIC project how this it can be used for simulating and emulating cooperative driver assistance systems. Particularly, the software has been tested on cooperative platoons. The tests were conducted on both simulation and real platforms to prove how the such software can be used during the development phase and that it is fully compatible with the architecture already present in the experimental vehicles.

- Contact: Fawzi Nashashibi

SECRET Project-Team

6. New Software and Platforms

6.1. CFS

FUNCTIONAL DESCRIPTION

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

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

6.2. Collision Decoding

KEYWORDS: Algorithm - Binary linear code
FUNCTIONAL DESCRIPTION

Collision Decoding implements two variants of information set decoding : Stern-Dumer, and MMT. To our knowledge it is the best full-fledged open-source implementation of generic decoding of binary linear codes. It is the best generic attack against code-based cryptography.

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

6.3. ISDF

FUNCTIONAL DESCRIPTION

Implementation of the Stern-Dumer decoding algorithm, and of a varaint of the algorithm due to May, Meurer and Thomae.

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

SERENA Team

5. New Software and Platforms

5.1. GEOFRACFLOW

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

- Participants: Géraldine Pichot, Jocelyne Erhel, and Jean-Raynald De Dreuzy
- Contact: Géraldine Pichot
- URL: https://bil.inria.fr/fr/software/view/2653/tab

5.2. SBM

Skew Brownian Motion
SCIENTIFIC DESCRIPTION SBM is a code allowing exact or approximated simulations of the Skew Brownian Motion. This code is used for the simulation, with a Monte-Carlo approach, of a 1D diffusion process with a discontinuous diffusion coefficient. Several benchmark tests are also implemented.

- Participants: Antoine Lejay, Géraldine Pichot
- Contact: Antoine Lejay
- URL: https://gforge.inria.fr/projects/sbm

5.3. Sklml

The OCaml parallel skeleton system
SCIENTIFIC DESCRIPTION Writing parallel programs is not easy, and debugging them is usually a nightmare. To cope with these difficulties, the skeleton programming approach uses a set of predefined patterns for parallel computations. The skeletons are higher-order functional templates that describe the program underlying parallelism. Sklml is a new framework for parallel programming that embeds an innovative compositional skeleton algebra into the OCaml language. Thanks to its skeleton algebra, Sklml provides two evaluation regimes to programs: a regular sequential evaluation (merely used for prototyping and debugging) and a parallel evaluation obtained via a recompilation of the same source program in parallel mode. Sklml was specifically designed to prove that the sequential and parallel evaluation regimes coincide.
FUNCTIONAL DESCRIPTION Sklml is a functional parallel skeleton compiler and programming system for OCaml programs. Slogan is “easy coarse grain parallelization”.

- Participants: Pierre Weis and François Clément
- Contact: François Clément
- URL: http://sklml.inria.fr

SIERRA Project-Team

5. New Software and Platforms

5.1. DICA : Discrete Independent Component Analysis

FUNCTIONAL DESCRIPTION

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.

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

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

FUNCTIONAL DESCRIPTION

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, 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

5.3. cnn_head_detection: Context-aware CNNs for person head detection

FUNCTIONAL DESCRIPTION

Code for ICCV 2015 paper "Context-aware CNNs for person head detection": 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

5.4. Lightning: large-scale linear classification, regression and ranking in Python

FUNCTIONAL DESCRIPTION

Lightning is a Python library for large-scale machine learning. More specifically, the library focuses on linear models for classification, regression and ranking. Lightning is the first project to integrate scikit-learn-contrib, a repository of high-quality projects that follow the same API conventions as scikit-learn. Compared to scikit-learn, the main advantages of lightning are its scalability and its flexibility. Indeed, lightning implements cutting-edge optimization algorithms that allow to train models with millions of samples within seconds on commodity hardware. Furthermore, lightning can leverage prior knowledge thanks to so-called structured penalties, an area of research that has recently found applications in domains as diverse as biology, neuroimaging, finance or text processing. Lightning is available under the 3-clause BSD license at http://contrib.scikit-learn.org/lightning/.

- Contact: Fabian Pedregosa
- URL: http://contrib.scikit-learn.org/lightning/

TAPDANCE Team (section vide)

WHISPER Project-Team

6. New Software and Platforms

6.1. Prequel

KEYWORDS: Code quality - Evolution - Infrastructure software
FUNCTIONAL DESCRIPTION

The commit history of a large, actively developed code base such as the Linux kernel is a gold mine of information on how evolutions should be made, how bugs should be fixed, etc. Nevertheless, the high volume of commits available and the rudimentary filtering tools provided imply that it is often necessary to wade through a lot of irrelevant information before finding example commits that can help with a specific software development problem. To address this issue, we have developed Prequel (Patch Query Language) [20]. Prequel builds on the semantic patch lamguage SmPL developed for Coccinelle, which is now well known to the Linux kernel developer community, to allow developers to scan the changes in a source code development history, taking into account not only the specific changes made, but also the context in which these changes occur. As the history of a code base under active development quickly becomes large, with the Linux kernel incorporating around 13,000 commits on each 2-3 month release cycle, a particular goal in the development of Prequel has been to provide reasonable performance. Currently, most queries in our experiments complete in under minute when running on a single core on a standard laptop. So far, we have applied Prequel to the problem of understanding how to eliminate uses of deprecated functions [20], and are investigating how it may be useful in a systematic driver porting methodology.

Prequel is publicly available under GPLv2. The development of Prequel is supported by OSADL, and Julia Lawall presented Prequel at the 2016 OSADL networking day (https://www.osadl.org/OSADL-Networking-Day-2016.networking-day-2016.0.html).

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

6.2. Coccinelle

KEYWORDS: Code quality - Evolution - Infrastructure software
FUNCTIONAL DESCRIPTION

Coccinelle is a tool for C code program matching and transformation that has been developed by members of the Whisper team over the last 10 years [8]. Coccinelle is widely used by the Linux kernel developer community and for other C software projects. Over the last three years, Coccinelle has benefited from the support of an engineer from the SED. Major improvements in 2016 include support for Python 3, independence from a no-longer-supported interface between Python and OCaml, better support for parallelism, and better support for integrating arbitrary predicates into the matching process. These features significantly improve performance and improve the uniformity of the rule specification language, thus providing a better experience for users. Coccinelle is at the foundation of much of our research work, including the ANR ITrans project, and these improvements will enhance and facilitate our research, accordingly.

Coccinelle is publicly available under GPLv2. In 2016, Julia Lawall presented Coccinelle in an invited keynote at the Linux Security Summit (http://events.linuxfoundation.org/events/archive/2016/linux-security-summit) and at a "birds of a feather" session at Linuxcon Europe (http://events.linuxfoundation.org/events/LinuxCon-europe).

- Participants: Julia Lawall, Gilles Muller, and Thierry Martinez
- Partners: IRILL - LIP6
- Contact: Julia Lawall
- URL: http://coccinelle.lip6.fr

6.3. Hector (BtrLinux)

KEYWORDS: Code quality - Evolution - Infrastructure software
FUNCTIONAL DESCRIPTION

A major source of errors in systems code is resource-release omission, which can lead to memory leaks and to crashes, if the system ends up in an inconsistent state. Currently, many tools exist that detect common patterns in software and detect faults as deviations from those patterns, but most suffer from high rates of false positives. Hector takes the novel approach of detecting inconsistencies local to a single function, and thus has been able to find over 300 faults in Linux kernel code and other C infrastructure software, with a rate of false positives of only 23%. Hector was originally the subject of the PhD thesis of Suman Saha [75]. Over the past two years, improving the robustness of the implementation of Hector has been the focus of ADT (young engineer position) BtrLinux supported by Inria, with the goal of making Hector publicly available and popularizing its use in the Linux kernel developer community. Some Linux kernel patches based on the use of Hector have been integrated into the Linux kernel, and the public release of Hector is in progress. The ADT position also involved the creation and maintenance of the website https://btrlinux.inria.fr/ as a showcase for the work of the Whisper team around Linux kernel development tools.

Building on his experience acquired in the ADT position, Quentin Lambert has recently been offered a position as an engineer at Wolfram MathCore AB.

- Participants: Quentin Lambert, Julia Lawall, and Gilles Muller
- Partners: IRILL - LIP6
- Contact: Julia Lawall
- URL: https://btrlinux.inria.fr/

6.4. ssrbit

FUNCTIONAL DESCRIPTION

ssrbit is a Coq library offering an efficient formalization of bit vectors, a refinement framework for abstractly reasoning about bitsets, and a trustworthy extraction of bit vectors to OCaml integers. Initially developed by Whisper members (Pierre-Évariste Dagand, Julia Lawall), the development has attracted an external contributor (Emilio Jesús Gallego Arias, postdoctoral researcher in CRI Mines-ParisTech), which led to significant improvements. We plan to improve the overall support and documentation so as to provide a full-featured library.

- Participants: Pierre Évariste Dagand, Julia Lawall, and Emilio Jesús Gallego Arias
- Contact: Pierre Évariste Dagand
- URL: https://github.com/ejgallego/ssrbit/

WILLOW Project-Team

6. New Software and Platforms

6.1. 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" [9]. 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/.

6.2. Unsupervised learning from narrated instruction videos

Open source release of the software package for our paper "Unsupervised learning from narrated instruction videos" . It provides a full implementation of the method, including code for weakly supervised training from instruction video, as well as trained models. Links to all of these are available at our project page http://www.di.ens.fr/willow/research/instructionvideos/.

6.3. ContextLocNet: Context-aware deep network models for weakly supervised localization

Open source release of code reproducing the results in our "ContextLocNet: Context-aware deep network models for weakly supervised localization" [11]. It provides code for training models, testing on standard datasets and trained models. It can be found online at https://github.com/vadimkantorov/contextlocnet.

6.4. Long-term Temporal Convolutions for Action Recognition

Open source release of the software package for our paper "Long-term Temporal Convolutions for Action Recognition" [20]. It provides code for training models, testing on standard datasets and trained models. Links are available at our project page http://www.di.ens.fr/willow/research/ltc/.