RESEARCH CENTER
Paris - Rocquencourt

FIELD

Activity Report 2012

Section Software

Edition: 2013-04-24

ABSTRACTION Project-Team

5. Software

5.1. The Apron Numerical Abstract Domain Library

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

Convex polyhedra, Intervals, Linear equalities, Numerical abstract domain, Octagons.

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

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

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

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

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

In 2012, APRON has been used in several researches conducted within or in collaboration with the Abstraction project-team: the design of a sufficient-condition generator [23] and the design of a constraint solver based on abstract domains [25].

5.2. The Astrée Static Analyzer of Synchronous Software

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

Absence of runtime error, Abstract interpretation, Static analysis, Verifier.

ASTRÉE is a static analyzer for sequential programs based on abstract interpretation [41], [32], [42], [34].

The ASTRÉE static analyzer [31], [46][1] www.astree.ens.fr aims at proving the absence of runtime errors in programs written in the C programming language.

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

ASTRÉE discovers all runtime errors including:

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

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

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

ASTRÉE has achieved the following two unprecedented results:

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

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

5.3. The AstréeA Static Analyzer of Asynchronous Software

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

Absence of runtime error, Abstract interpretation, Data races, Interference, Memory model, Parallel software, Static analysis, Verifier.

ASTRÉEA is a static analyzer prototype for parallel software based on abstract interpretation [43], [44], [36]. It started with support from THÉSÉE ANR project (2006–2010) and is continuing within the ASTRÉEA project (2012–2015).

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

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

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

ASTRÉEA is currently being applied to analyze a large industrial avionic software: 1.6 MLines of C and 15 threads, completed with a 2,500-line model of the ARINC 653 OS developed for the analysis. The analysis currently takes a few tens of hours on a 2.9 GHz 64-bit intel server using one core and generates around 1,200 alarms. The low computation time (only a few times larger than the analysis time by ASTRÉE of synchronous programs of a similar size and structure) shows the scalability of the approach (in particular, we avoid the usual combinatorial explosion associated to thread interleavings). Precision-wise, the result, while not as impressive as that of ASTRÉE, is quite encouraging. The development of AstréeA continues within the scope of the ASTRÉEA ANR project (Section 8.1.1.2).

5.4. The OpenKappa modeling plateform

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

Causal traces, Model reduction, Rule-based modelling, Simulation, Static analysis.

OPENKAPPA is a collection of tools to build, debug and run models of biological pathways. It contains a compiler for the Kappa Language [52], a static analyzer [51] (for debugging models), a simulator [50], a compression tool for causal traces [49],[20], and a model reduction tool [4], [48], [53].

OPENKAPPA is developed since 2007 and, the OCaml version currently consists of 46 000 lines of OCaml. Software are available in OCaml and in Java. Moreover, an Eclipse pluggin is available.

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

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

5.5. Translation Validation

Participant: Xavier Rival [correspondent].

Abstract interpretation, Certified compilation, Static analysis, Translation validation, Verifier.

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

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

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

5.6. Zarith

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

Arbitrary precision integers, Arithmetic, OCaml.

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

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

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

ALPAGE Project-Team

5. Software

5.1. Syntax

Participants: Pierre Boullier [correspondant], Benoît Sagot.

See also the web page http://syntax.gforge.inria.fr/.

The (currently beta) version 6.0 of the SYNTAX system (freely available on Inria GForge) includes various deterministic and non-deterministic CFG parser generators. It includes in particular an efficient implementation of the Earley algorithm, with many original optimizations, that is used in several of Alpage's NLP tools, including the pre-processing chain SXPipe and the LFG deep parser SXLFG. This implementation of the Earley algorithm has been recently extended to handle probabilistic CFG (PCFG), by taking into account probabilities both during parsing (beam) and after parsing (n-best computation). SYNTAX 6.0 also includes parsers for various contextual formalisms, including a parser for Range Concatenation Grammars (RCG) that can be used among others for TAG and MC-TAG parsing.

Direct NLP users of SYNTAX for NLP, outside Alpage, include Alexis Nasr (Marseilles) and other members of the (now closed) SEQUOIA ANR project, Owen Rambow and co-workers at Columbia University (New York), as well as (indirectly) all SXPipe and/or SXLFG users. The project-team VASY (Inria Rhône-Alpes) is one of SYNTAX' user for non-NLP applications.

5.2. System DyALog

Participant: Éric Villemonte de La Clergerie [maintainer].

DYALOG *on Inria GForge: http://dyalog.gforge.inria.fr/*

DYALOG provides an environment to compile and execute grammars and logic programs. It is essentially based on the notion of tabulation, i.e. of sharing computations by tabulating traces of them. DYALOG is mainly used to build parsers for Natural Language Processing (NLP). It may nevertheless be used as a replacement for traditional PROLOG systems in the context of highly ambiguous applications where sub-computations can be shared.

The current release **1.13.0** of DYALOG is freely available by FTP under an open source license and runs on Linux platforms for x86 and architectures and on Mac OS intel (both 32 and 64bits architectures).

The current release handles logic programs, DCGs (*Definite Clause Grammars*), FTAGs (*Feature Tree Adjoining Grammars*), FTIGs (*Feature Tree Insertion Grammars*) and XRCGs (*Range Concatenation Grammars* with logic arguments). Several extensions have been added to most of these formalisms such as intersection, Kleene star, and interleave operators. Typed Feature Structures (TFS) as well as finite domains may be used for writing more compact and declarative grammars [120].

C libraries can be used from within DYALOG to import APIs (`mysql`, `libxml`, `sqlite`, ...).

DYALOG is largely used within ALPAGE to build parsers but also derivative softwares, such as a compiler of Meta-Grammars (cf. 5.3). It has also been used for building a parser from a large coverage French TIG/TAG grammar derived from a Meta-Grammar. This parser has been used for the Parsing Evaluation campaign EASy, the two Passage campaigns (Dec. 2007 and Nov. 2009), cf. [117], [118], and very large amount of data (700 millions of words) in the SCRIBO project.

DYALOG and other companion modules are available on Inria GForge.

5.3. Tools and resources for Meta-Grammars

Participant: Éric Villemonte de La Clergerie [maintainer].

mgcomp, MGTOOLS, and FRMG on Inria GForge: http://mgkit.gforge.inria.fr/

DYALOG (cf. 5.2) has been used to implement mgcomp, Meta-Grammar compiler. Starting from an XML representation of a MG, mgcomp produces an XML representation of its TAG expansion.

The current version **1.5.0** is freely available by FTP under an open source license. It is used within ALPAGE and (occasionally) at LORIA (Nancy) and at University of Pennsylvania.

The current version adds the notion of namespace, to get more compact and less error-prone meta-grammars. It also provides other extensions of the standard notion of Meta-Grammar in order to generate very compact TAG grammars. These extensions include the notion of *Guarded nodes*, i.e. nodes whose existence and non-existence depend on the truth value of a guard, and the use of the regular operators provided by DYALOG on nodes, namely disjunction, interleaving and Kleene star. The current release provides a dump/restore mechanism for faster compilations on incremental changes of a meta-grammars.

The current version of mgcomp has been used to compile a wide coverage Meta-Grammar FRMG (version 2.0.1) to get a grammar of around 200 TAG trees [122]. Without the use of guarded nodes and regular operators, this grammar would have more than several thousand trees and would be almost intractable. FRMG has been packaged and is freely available.

To ease the design of meta-grammars, a set of tools have been implemented, mostly by Éric de La Clergerie, and collected in MGTOOLS (version **2.2.2**). This package includes a converter from a compact format to a XML pivot format, an Emacs mode for the compact and XML formats, a graphical viewer interacting with Emacs and XSLT stylesheets to derive HTML views. A new version is under development to provide an even more compact syntax and some checking mechanisms to avoid frequent typo errors.

The various tools on Metagrammars are available on Inria GForge. FRMG is used directly or indirectly (through a Web service or by requiring parsed corpora) by several people and actions (ANR Rhapsodie, ANR Chronoline, ...)

5.4. The Bonsai PCFG-LA parser

Participants: Marie Candito [correspondant], Djamé Seddah.

Web page:
http://alpage.inria.fr/statgram/frdep/fr_stat_dep_parsing.html

Alpage has developed as support of the research papers [74], [65], [66], [11] a statistical parser for French, named Bonsai, trained on the French Treebank. This parser provides both a phrase structure and a projective dependency structure specified in [4] as output. This parser operates sequentially: (1) it first outputs a phrase structure analysis of sentences reusing the Berkeley implementation of a PCFG-LA trained on French by Alpage (2) it applies on the resulting phrase structure trees a process of conversion to dependency parses using a combination of heuristics and classifiers trained on the French treebank. The parser currently outputs several well known formats such as Penn treebank phrase structure trees, Xerox like triples and CONLL-like format for dependencies. The parsers also comes with basic preprocessing facilities allowing to perform elementary sentence segmentation and word tokenisation, allowing in theory to process unrestricted text. However it is believed to perform better on newspaper-like text. The parser is available under a GPL license.

5.5. The MICA parser

Participants: Benoît Sagot [correspondant], Marie Candito, Pierre Boullier, Djamé Seddah.

Web page:
http://mica.lif.univ-mrs.fr/

MICA (Marseille-Inria-Columbia- AT&T) is a freely available dependency parser [57] currently trained on English and Arabic data, developed in collaboration with Owen Rambow and Daniel Bauer (Columbia University) and Srinivas Bangalore (AT&T). MICA has several key characteristics that make it appealing to researchers in NLP who need an off-the-shelf parser, based on Probabilistic Tree Insertion Grammars and on the SYNTAX system. MICA is fast (450 words per second plus 6 seconds initialization on a standard high-end machine) and has close to state-of-the-art performance (87.6% unlabeled dependency accuracy on the Penn Treebank).

MICA consists of two processes: the supertagger, which associates tags representing rich syntactic information with the input word sequence, and the actual parser, based on the Inria SYNTAX system, which derives the syntactic structure from the n-best chosen supertags. Only the supertagger uses lexical information, the parser only sees the supertag hypotheses.

MICA returns n-best parses for arbitrary n; parse trees are associated with probabilities. A packed forest can also be returned.

5.6. Alpage's linguistic workbench, including SxPipe

Participants: Benoît Sagot [correspondant], Rosa Stern, Marion Baranes, Damien Nouvel, Virginie Mouilleron, Pierre Boullier, Éric Villemonte de La Clergerie.

See also the web page http://lingwb.gforge.inria.fr/.

Alpage's linguistic workbench is a set of packages for corpus processing and parsing. Among these packages, the SXPipe package is of a particular importance.

SXPipe [97] is a modular and customizable chain aimed to apply to raw corpora a cascade of surface processing steps. It is used

- as a preliminary step before Alpage's parsers (e.g., FRMG);
- for surface processing (named entities recognition, text normalization...).

Developed for French and for other languages, SXPipe includes, among others, various named entities recognition modules in raw text, a sentence segmenter and tokenizer, a spelling corrector and compound words recognizer, and an original context-free patterns recognizer, used by several specialized grammars (numbers, impersonal constructions, quotations...). In 2012, SXPipe has received a renewed attention in four directions:

- Support of new languages, and most notably German (although this is still at a very preliminary stage of development;
- Analysis of unknown words, in particular in the context of the ANR project EDyLex and of the collaboration with *viavoo*; this involves in particular (ii) 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;
- Development of new local grammars for detecting new types of entities, such as chemical formulae or dimensions, in the context of the PACTE project.

5.7. MElt

Participants: Benoît Sagot [correspondant], Pascal Denis.

MElt is a part-of-speech tagger, initially trained for French (on the French TreeBank and coupled with the Le*fff*), English [78], Spanish, Kurmanji Kurdish [125] and Persian [106], [107]. It is state-of-the-art for French. It is distributed freely as a part of the Alpage linguistic workbench.

In 2012, MElt has underwent two major upgrades:

- It has been successfully trained and used on Italian [35], Spanish [26] and German data. In particular, a statistical parsing architecture for Italian that used MElt in a pre-processing step has obtained the best results in the EVALITA shared task on Italian parsing [35].
- MElt can now be called within a wrapper developed for handling noisy textual data such as user-generated content produced on Web 2.0 platforms (forums, blogs, social media); more precisely, this wrapper is able to "clean" such data, then tag it using MElt, and finally transfer MElt annotations from the "cleaned" data, which could be annotated more easily, to the original noisy data. This architecture has proved useful on French for creating the French Social Media Bank [37], [36]. On English, it has played an important role within both variants of the Alpage parsing architecture that were ranked 2nd and 3rd at the SANCL shared task on parsing user-generated content, organized by Google [38].

5.8. The Alexina framework: the Lefff syntactic lexicon, the Aleda entity database and other Alexina resources

Participants: Benoît Sagot [correspondant], Laurence Danlos.

See also the web page http://gforge.inria.fr/projects/alexina/.

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

Historically, the Le*fff* 1 was a freely available French morphological lexicon for verbs that has been automatically extracted from a very large corpus. Since version 2, the Le*fff* covers all grammatical categories (not just verbs) and includes syntactic information (such as subcategorization frames); Alpage's tools, including Alpage's parsers, rely on the Le*fff*. The version 3 of the Le*fff*, which has been released in 2008, improves the linguistic relevance and the interoperability with other lexical models.

Other Alexina lexicons exist, at various stages of development, in particular for Spanish (the Le*ff*e), Polish, Slovak, English, Galician, Persian, Kurdish, Italian and since this year for German, as well as for Latin and Maltese verbs. These lexicons are used in various tools, including instances of the MElt POS-tagger.

Alexina also hosts *Aleda* [114], [33] a large-scale entity database currently developed for French but under development for English, Spanish and German, extracted automatically from Wikipedia and Geonames. It is used among others in the SxPipe processing chain and its NP named entity recognition, as well as in the Nomos named entity linking system.

5.9. The free French wordnet WOLF

Participants: Benoît Sagot [correspondant], Marion Richard, Sarah Beniamine.

The WOLF (Wordnet Libre du Français) is a wordnet for French, i.e., a lexical semantic database. The development of WOLF started in 2008 [99], [100]. At this time, we focused on benefiting from available resources of three different types: general and domain-specific bilingual dictionaries, multilingual parallel corpora and Wiki resources (Wikipedia and Wiktionaries). This work was achieved in a large part in collaboration with Darja Fišer (University of Ljubljana, Slovenia), in parallel with the development of a free Slovene wordnet, sloWNet. However, it was also impacted by specific collaborations, e.g., on adverbial synsets [101].

2012 results concerning the WOLF are described in the corresponding section.

The WOLF is freely available under the Cecill-C license. It has already been used in various experiments, within and outside Alpage.

5.10. Automatic construction of distributional thesauri

Participant: Enrique Henestroza Anguiano [correspondant].

FREDISTis a freely-available (LGPL license) Python package that implements methods for the automatic construction of distributional thesauri.

We have implemented the context relation approach to distributional similarity, with various context relation types and different options for weight and measure functions to calculate distributional similarity between words. Additionally, FREDISTis highly flexible, with parameters including: context relation type(s), weight function, measure function, term frequency thresholds, part-of-speech restrictions, filtering of numerical terms, etc.

Distributional thesauri for French are also available, one each for adjectives, adverbs, common nouns, and verbs. They have been constructed with FreDist and use the best settings obtained in an evaluation. We use the *L'Est Republicain* corpus (125 million words), *Agence France-Presse* newswire dispatches (125 million words) and a full dump of the French Wikipedia (200 million words), for a total of 450 million words of text.

5.11. Tools and resources for time processing

Participant: Laurence Danlos [correspondant].

Alpage developed the *French TimeBank*, a freely-available corpus annotated with ISO-TimeML-compliant temporal information (dates, events and relations between events).

5.12. System EasyRef

Participants: Éric Villemonte de La Clergerie [maintainer], Corentin Ribeyre.

A collaborative WEB service EASYREF has been developed, in the context of ANR action Passage, to handle syntactically annotated corpora. EASYREF may be used to view annotated corpus, in both EASY or PASSAGE formats. The annotations may be created and modified. Bug reports may be emitted. The annotations may be imported and exported. The system provides standard user right management. The interface has been designed with the objectives to be intuitive and to speed edition.

EASYREF relies on an Model View Controller design, implemented with the Perl Catalyst framework. It exploits WEB 2.0 technologies (i.e. AJAX and JavaScript).

Version 2 has been used by ELDA and LIMSI to annotate a new corpus of several thousands words for PASSAGE.

A preliminary version 3 has been developed by François Guérin and revised by Éric de La Clergerie, relying on Berkeley DB XML to handle very large annotated corpora and to provide a complete query language expanded as XQuery expressions. EASYREF is maintained under Inria GForge.

AOSTE Project-Team

5. Software

5.1. TimeSquare

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

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

TimeSquare offers four main functionalities:

1. graphical and/or textual interactive specification of logical clocks and relative constraints between them;
2. definition and handling of user-defined clock constraint libraries;
3. automated simulation of concurrent behavior traces respecting such constraints, using a Boolean solver for consistent trace extraction;
4. call-back mechanisms for the traceability of results (animation of models, display and interaction with waveform representations, generation of sequence diagrams...).

In practice TimeSquare is a plug-in developed with Eclipse modeling tools. The software is registered by the *Agence pour la Protection des Programmes*, under number IDDN.FR.001.170007.000.S.P.2009.001.10600. It can be downloaded from the site http://timesquare.inria.fr/. It has been integrated in the OpenEmbeDD ANR RNTL platform, and other such actions are under way.

5.2. K-Passa

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

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

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

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

5.3. SynDEx

Participants: Maxence Guesdon, Yves Sorel [correspondant], Cécile Stentzel, Meriem Zidouni.

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

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

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

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

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

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

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

5.4. SAS

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

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

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

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

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

ARLES Project-Team

5. Software

5.1. Introduction

In order to validate our research results, our research activities encompass the development of related prototypes as surveyed below.

5.2. iCONNECT – Emergent Middleware Enablers

Participant: Valérie Issarny [correspondent].

As part of our research work on Emergent Middleware, we have implemented Enablers (or Enabler functionalities) that make part of the overall CONNECT architecture realizing Emergent Middleware in practice [2]. The focus of ARLES work is on the: *Discovery enabler* that builds on our extensive background in the area of interoperable pervasive service discovery; and *Synthesis enabler* that synthesizes mediators that allow networked systems that have compatible functionalities to interact despite mismatching interfaces and/or behaviors.

The CONNECT Discovery Enabler is the component of the overall CONNECT architecture that handles discovery of networked systems (NSs), stores their descriptions (NS models), and performs an initial phase of matchmaking to determine which pairs of systems are likely to be able to interoperate. Such pairs are then passed to the Synthesis Enabler so that mediators can be generated. The Discovery Enabler is written in Java and implements several legacy discovery protocols including DPWS and UPnP.

The proposed solution to mediator synthesis assumes an ontology-based system description *à la* OWL-S, which is made available by the Discovery Enabler, possibly helped by machine learning. The semantically-annotated interfaces of systems that need to communicate are then processed to compute the semantic mapping between their respective operations using a constraint solver. The resulting mapping serves generating a mediator process that further coordinates the behaviors of the systems and guarantees their successful interaction. The mediator is deployed on a dedicated engine able to parse and compose middleware messages and convert them to fit each system expectations regarding the type and order of these messages.

The CONNECT Enablers have been integrated and experimented with by the CONNECT consortium to effectively enable Emergent Middleware. Part of them are available for download under an open source license at the CONNECT Web site at https://www.connect-forever.eu/software.html.

5.3. Service-oriented Middleware for Pervasive Computing

Participants: Nikolaos Georgantas [correspondent], Valérie Issarny [correspondent].

In the past years, we have built a strong foundation of service-oriented middleware to support the pervasive computing vision. This specifically takes the form of a family of middlewares, all of which have been released under the open source LGPL license:

- **WSAMI - A Middleware Based on Web Services for Ambient Intelligence:** WSAMI (Web Services for AMbient Intelligence) is based on the Web services architecture and allows for the deployment of services on wireless handheld devices like smartphones and PDAs.
  URL: http://www-rocq.inria.fr/arles/download/ozone/index.htm
- **Ariadne - A Protocol for Scalable Service Discovery in MANETs:** Ariadne enriches WSAMI with the Ariadne service discovery protocol, which has been designed to support decentralized Web service discovery in multi-hop mobile ad hoc networks (MANETs). Ariadne enables small and resource-constrained mobile devices to seek and find complementary, possibly mobile, Web services needed to complete specified tasks in MANETs, while minimizing the traffic generated and tolerating intermittent connectivity.

URL: http://www-rocq.inria.fr/arles/download/ariadne/index.html

- **MUSDAC - A Middleware for Service Discovery and Access in Pervasive Networks:** The MUlti-protocol Service Discovery and ACcess (MUSDAC) middleware platform enriches WSAMI so as to enable the discovery and access to services in the pervasive environment, which is viewed as a loose and dynamic composition of independent networks. MUSDAC manages the efficient dissemination of discovery requests between the different networks and relies on specific plug-ins to interact with the various middleware used by the networked services.
  URL: http://www-rocq.inria.fr/arles/download/ubisec/index.html
- **INMIDIO - An Interoperable Middleware for Ambient Intelligence:** INMIDIO (INteroperable MIddleware for service Discovery and service InteractiOn) dynamically resolves middleware mismatch. More particularly, INMIDIO identifies the interaction middleware and also the discovery protocols that execute on the network and translates the incoming/outgoing messages of one protocol into messages of another, target protocol.
  URL: http://www-rocq.inria.fr/arles/download/inmidio/index.html
- **COCOA - A Semantic Service Middleware:** COCOA is a comprehensive approach to semantic service description, discovery, composition, adaptation and execution, which enables the integration of heterogeneous services of the pervasive environment into complex user tasks based on their abstract specification. Using COCOA, abstract user tasks are realized by dynamically composing the capabilities of services that are currently available in the environment.
  URL: http://gforge.inria.fr/projects/amigo/
- **ubiSOAP - A Service Oriented Middleware for Seamless Networking:**ubiSOAP brings multi-radio, multi-network connectivity to services through a comprehensive layered architecture: (i) the multi-radio device management and networking layers together abstract multi-radio connectivity, selecting the optimal communication link to/from nodes, according to quality parameters; (ii) the communication layer allows for SOAP-based point-to-point and group-based interactions in the pervasive network; and (iii) the middleware services layer brings advanced distributed resource management functionalities customized for the pervasive networking environment.
  URL: http://www.ist-plastic.org.

5.4. xSOM – Service-oriented middleware for the Future Internet

Participant: Nikolaos Georgantas [correspondent].

Building on our long experience on service-oriented middleware (SOM) for pervasive environments (see § 5.3 above) and given the evolution of such environments towards the Future Internet and the Internet of Things, we have already implemented early results of our related research into an extensible SOM (xSOM) for the Future Internet. xSOM aims at enabling large-scale dynamic compositions of services and things, while being highly extensible for accommodating the extreme heterogeneity of such compositions. xSOM currently supports two major functionalities: (i) a protocol bus-based solution to seamless integration of heterogeneous interaction paradigms for services and things; and (ii) a solution to discovery, access and data fusion over large populations of things.

Regarding (i), we have introduced a protocol interoperability solution comprising representative abstract connector types for the client/server (CS), publish/subscribe (PS) and tuple space (TS) paradigms, as well as their mapping to a higher-level generic application (GA) connector type. We apply these connector abstractions to introduce an enhanced bus paradigm, the eXtensible Service Bus (XSB). XSB features richer interaction semantics than common Enterprise Service Bus (ESB) implementations and incorporates special consideration for semantics-preserving cross-integration of CS, PS and TS. We have carried out a realization of XSB —first on the PEtALS ESB, and then on EasyESB— where we provide templates for systematic and highly facilitated building of binding components for heterogeneous systems (services and things) that are plugged into the XSB. To demonstrate the applicability of our approach, we have implemented a smoke-detection-and-alert system integrating a JMEDS DPWS Web Service (CS), a JMS system based on Apache ActiveMQ (PS), and a Jini JavaSpaces system (TS).

Regarding (ii), we support data queries over large populations of things, notably smartphones, which are getting increasingly ubiquitous and embed a rich collection of sensors. xSOM enables: (i) high-level programming of things on top of heterogeneous smartphone sensors; (ii) thing discovery and access dealing with large numbers of networked things; and (iii) on-the-fly composition of such things and fusion of their data in response to queries. In target settings, e.g., at the scale of a city, not all phones need to register for reporting their data (e.g., ambient sound level); some smartly distributed sampling is sufficient. This enables efficient scalable coverage of the entire city with only a subset of the large phone population being registered. The phone's things registration manager includes a probabilistic decision algorithm for selective registration based on the truncated Lévy walk mobility model. The registration decision is based on the actual density of already registered phones, the coverage quality requirements, and the coverage of the estimated path that the user will take for the next few minutes. We have implemented a demonstrative application enabling a user to know "how lively is this city spot at this moment' based on retrieving and aggregating smartphone ambient sound level data.

Our software will soon be released under open source license as part of the newly launched OW2 initiative on "Future Internet of Software Services" (http://www.ow2.org/view/Future_Internet/).

5.5. Srijan: Data-driven Macroprogramming for Sensor Networks

Participant: Animesh Pathak [correspondent].

Macroprogramming is an application development technique for wireless sensor networks (WSNs) where the developer specifies the behavior of the system, as opposed to that of the constituent nodes. As part of our work in this domain, we are working on *Srijan*, a toolkit that enables application development for WSNs in a graphical manner using data-driven macroprogramming.

It can be used in various stages of application development, *viz.,*

1. Specification of application as a task graph,
2. Customization of the auto-generated source files with domain-specific imperative code,
3. Specification of the target system structure,
4. Compilation of the macroprogram into individual customized runtimes for each constituent node of the target system, and finally
5. Deployment of the auto generated node-level code in an over-the-air manner to the nodes in the target system.

The current implementation of *Srijan* targets both the Sun SPOT sensor nodes and larger nodes with J2SE. Most recently, *Srijan* also includes rudimentary support for incorporating Web services in the application being designed. The software is released under open source license, and available as an Eclipse plug-in at http://code.google.com/p/srijan-toolkit/.

5.6. Yarta: Middleware for supporting Mobile Social Applications

Participant: Animesh Pathak [correspondent].

With the increased prevalence of advanced mobile devices (the so-called "smart" phones), interest has grown in *Mobile Social Ecosystems* (MSEs), where users not only access traditional Web-based social networks using their mobile devices, but are also able to use the context information provided by these devices to further enrich their interactions. We are developing a middleware framework for managing mobile social ecosystems, having a multi-layer middleware architecture consisting of modules, which will provide the needed functionalities, including:

- Extraction of social ties from context (both physical and virtual),
- Enforcement of access control to protect social data from arbitrary access,
- A rich set of MSE management functionalities, using which mobile social applications can be developed.

Our middleware adopts a graph-based model for representing social data, where nodes and arcs describe socially relevant entities and their connections. In particular, we exploit the Resource Description Framework (RDF), a basic Semantic Web standard language that allows representing and reasoning about social vocabulary, and creating an interconnected graph of socially relevant information from different sources.

The current implementation of the Yarta middleware targets both desktop/laptop nodes running Java 2 SE, as well as Android smart phones. The software is released under open source license at https://gforge.inria.fr/projects/yarta/.

5.7. iBICOOP: Mobile Data Management in Multi-* Networks

Participant: Valérie Issarny [correspondent].

Building on the lessons learned with the development of pervasive service oriented middleware and of applications using them, we have been developing the custom iBICOOP middleware. iBICOOP specifically aims at assisting the development of advanced mobile, collaborative application services by supporting interactions between mobile users.

Briefly, the iBICOOP middleware addresses the challenges of easily accessing content stored on mobile devices, and consistent data access across multiple mobile devices by targeting both fixed and mobile devices, leveraging their characteristics (e.g., always on and unlimited storage for home/enterprise servers, ad hoc communication link between mobile devices), and by leveraging the capabilities of all available networks (e.g., ad hoc networks, Internet, Telecoms infrastructure networks). It also relies on Web and Telecoms standards to promote interoperability.

The base architecture of the iBICOOP middleware consists of core modules on top of which we can develop applications that may arise in the up-coming multi-device, multi-user world:

- The *Communication Manager* provides mechanisms to communicate over different available network interfaces of a device — Bluetooth, WiFi, Cellular — and also using different technologies e.g., Web services, HTTP/TCP sockets, ad hoc mode.
- The *Security Manager* uses well-established techniques of cryptography and secure communication to provide necessary security.
- The *Partnership Manager* provides device or user information in the form of *profiles*.
- iBICOOP relies on service location protocols for *naming and discovery* of nearby services on currently active network interfaces that support IP multicast.
- Besides normal file managing tasks, the *Local File Manager* gives the user clear cues to the files that have been replicated across multiple devices or shared among different users by using different icons.

The iBICOOP middleware has been licensed by AMBIENTIC (http://www.ambientic.com/), a start-up that specifically develops innovative mobile distributed services on top of the iBICOOP middleware that allows for seamless interaction and content sharing in today's multi-* networks.
URL: https://www-roc.inria.fr/arles/index.php/ongoing-research-projects/74-data-sharing-and-replication-in-pervasive-networks

AXIS Project-Team

4. Software

4.1. Introduction

From its creation, AxIS has proposed new methods, approaches and software validated experimentally on various applications: Data Mining, Web usage Mining, Information Retrieval, Activity Modeling.
Some of our results are under process to be part of the FocusLab platform (CPER Télius 5.6) which is based on a Service oriented Architecture. The development process of the software part has started in 2011, finding ways to fund human ressources. Such a platform aims the community of Living Labs domain. In [70], we report the usage of the FocusLab platform (hardware and software components) inside various regional and european projects..

4.2. Data Mining

4.2.1. Classification and Clustering Methods

Participants: Marc Csernel, Yves Lechevallier [co-correspondant], Brigitte Trousse [co-correspondant].

We developed and maintained a collection of clustering and classification software, written in C++ and/or Java:

Supervised methods

- a Java library (Somlib) that provides efficient implementations of several SOM(Self-Organizing Map) variants [77], [76], [101], [100], [104], especially those that can handle dissimilarity data (available on Inria's Gforge server (public access) Somlib, developed by AxIS Rocquencourt and Brieuc Conan-Guez from Université de Metz.
- a functional Multi-Layer Perceptron library, called FNET, that implements in C++ supervised classification of functional data [96], [99], [98], [97] (developed by AxIS Rocquencourt).

Unsupervised methods : partitioning methods

- Two partitioning clustering methods on the dissimilarity tables issued from a collaboration between AxIS Rocquencourt team and Recife University, Brazil: CDis and CCClust [84]. Both are written in C++ and use the "Symbolic Object Language" (SOL) developed for SODAS. And one partitioning method on interval data (Div).
- Two standalone versions improved from SODAS modules, SCluster and DIVCLUS-T [74] (AxIS Rocquencourt).

Unsupervised methods : agglomerative methods

- a Java implementation of the 2-3 AHC (developed by AxIS Sophia Antipolis). The software is available as a Java applet which runs the hierarchies visualization toolbox called HCT for Hierarchical Clustering Toolbox (see [75]).

A Web interface developed in C++ and running on our Apache internal Web server .is available for the following methods: SCluster, Div, Cdis, CCClust.
Previous versions of the above software have been integrated in the SODAS 2 Software [95] which was the result of the european project ASSO [6] (2001-2004). SODAS 2 supports the analysis of multidimensional complex data (numerical and non numerical) coming from databases mainly in statistical offices and administration using Symbolic Data Analysis [71]. This software is registrated at APP (Agence de la Protection des Programmes). The latest executive version of the SODAS 2 software, with its user manual can be downloaded at http://www.info.fundp.ac.be/asso/sodaslink.htm [78], [85].

[6] ASSO: Analysis System of Symbolic Official data

As a 2012 result, a release of MND (Dynamic Clustering Method for Multi-Nominal data) algorithm based on previous AxIS research (2003) has been done (cf. section 5.6).

4.2.2. Extracting Sequential Patterns with Low Support

Participant: Brigitte Trousse [correspondant].

Two methods for extracting sequential patterns with low support have been developed by D. Tanasa in his thesis (see Chapter 3 in [103] for more details) in collaboration with F. Masseglia and B. Trousse :

- **Cluster & Divide**,
- and **Divide & Discover** [11].

These methods have been successfully applied from 2005 on various Web logs.

4.2.3. Mining Data Streams

Participants: Brigitte Trousse [correspondant], Mohamed Gaieb.

In Marascu's thesis (2009) [91], a collection of software have been developed for knowledge discovery and security in data streams. Three **clustering methods for mining sequential patterns (Java) in data streams** method have been developped in Java:

- SMDS compares the sequences to each others with a complexity of $O(n^2)$.
- SCDS is an improvement of SMDS, where the complexity is enhanced from $O(n^2)$ to $O(n.m)$ with n the number of navigations and m the number of clusters.
- ICDS is a modification of SCDS. The principle is to keep the clusters' centroids from one batch to another.

Such methods take batches of data in the format "Client-Date-Item" and provide clusters of sequences and their centroids in the form of an approximate sequential pattern calculated with an alignment technique.

In 2010 the Java code of one method called SCDS has been integrated in the MIDAS demonstrator and a C++ version has been implemented by F. Masseglia for the CRE contract with Orange Labs with the deliverability of a licence) with a visualisation module (in Java).

It has been tested on the following data:

- Orange mobile portal logs (100 million records, 3 months) in the context of Midas project (Java version) and the CRE (Orange C++ version)
- Inria Sophia Antipolis Web logs (4 million records, 1 year, Java version)
- Vehicle trajectories (Brinkhoff generator) in the context of MIDAS project (Java version).

In 2011, in the context of the ELLIOT contrcat [cf. Section 6.3.1.1), SCDS has been integrated as a Web service (Java version) in the first version of FocusLab platform (cf. section 5.6) in the ELLIOT context: a demonstration was made on San Rafaelle Hospital media use case at the first ELLIOT review at Brussels.

In 2012 we applied SCDS web service on data issued from co-creation step of two use cases in Logistics (BIBA) and Green Services (ICT Usage Lab). More data are needed to show the relevance of this method, it is planned in 2013 with the experimentation step of Green Services.

The three C++ codes done for the CRE (Orange Labs) have been depositi at APP.

4.3. Web Usage Mining

4.3.1. AWLH for Pre-processing Web Logs

Participants: Yves Lechevallier [co-correspondant], Brigitte Trousse [co-correspondant].

AWLH (AxIS Web Log House) for Web Usage Mining (WUM) is issued from AxISlogminersoftware which implements the mult-site log preprocessing methodology and extrcation of sequential pattern with low support developed by D. Tanasa in his thesis [15] for Web Usage Mining (WUM). In the context of the Eiffel project (2008-2009), we isolated and redesigned the core of AxISlogMiner preprocessing tool (we called it AWLH) composed of a set of tools for pre-processing web log files. The web log files are cleaned before to be used by data mining methods, as they contain many noisy entries (for example, robots requests). The data are stored within a database whose model has been improved.

So AWLH offers:

- Processing of several log files from several servers,
- Support of several input formats (CLF, ECLF, IIS, custom, ...),
- Incremental pre-processing,
- Java API to help integration of AWLH in external application.

An additionnal tool has been developped for capturing user actions in real time based on an open source project called "OpenSymphony ClickStream". An extension version of AWLH called **AWLH-Debate** has been developed for recording and structuring data issued from annotated documents inside discussion forums.

4.3.2. ATWUEDA for Analysing Evolving Web Usage Data

Participants: Yves Lechevallier [correspondant], Brigitte Trousse, Mohamed Gaieb, Yves Lechevallier [correspondant].

ATWUEDA for Web Usage Evolving Data Analysis [80] was developed by A. Da Silva in her thesis [79] under the supervision of Y. Lechevallier. This tool was developed in Java and uses the JRI library in order to allow the application of R which is a programming language and software environment for statistical computing functions in the Java environment.

ATWUEDA is able to read data from a cross table in a MySQL database. It splits the data according to the user specifications (in logical or temporal windows) and then applies the approach proposed in the Da Silva's thesis in order to detect changes in dynamic environment. The proposed approach characterizes the changes undergone by the usage groups (e.g. appearance, disappearance, fusion and split) at each timestamp. Graphics are generated for each analyzed window, exhibiting statistics that characterizes changing points over time.

Version 2.of ATWUEDA (september 2009) is available at Inria's gforce website.

In 2011 we have demonstrated the efficiency of ATWUEDA [82] by applying it on another real case study on condition monitoring data streams of an electric power plant provided by EDF.

ATWUEDA is used by Telecom Paris Tech and EDF [4].
This year we studied how to transform the code of ATWUEDA as a web service for the version 1.2 of FocusLab: in fact we gave up this objective, which would require more resource than we have.

4.4. Information Retrieval

*4.4.1. CBR*Tools for Managing and Reusing Past Experiences based on Historical Data*

Participant: Brigitte Trousse [correspondant].

CBR*Tools [87], [88] is an object-oriented framework [89], [86] for Case-Based Reasoning which is specified with the UMT notation (Rational Rose) and written in Java. It offers a set of abstract classes to model the main concepts necessary to develop applications integrating case-based reasoning techniques: case, case base, index, measurements of similarity, reasoning control. It also offers a set of concrete classes which implements many traditional methods (closest neighbors indexing, Kd-tree indexing, neuronal approach based indexing, standards similarities measurements). CBR*Tools currently contains more than 240 classes divided in two main categories: the core package for basic functionality and the time package for the specific management of the behavioral situations. The programming of a new application is done by specialization of existing classes, objects aggregation or by using the parameters of the existing classes.

CBR*Tools addresses application fields where the re-use of cases indexed by behavioral situations is required. The CBR*Tools framework was evaluated via the design and the implementation of several applications such as Broadway-Web, Educaid, BeCKB, Broadway-Predict, e-behaviour and Be-TRIP.

CBR*Tools is concerned by two past contracts: EPIA and MobiVIP.

CBR*Tools will be available for research, teaching and academic purpose via the FocusLab platform. The user manual can be downloaded at the URL: http://www-sop.inria.fr/axis/cbrtools/manual/.

See also the web page http://www-sop.inria.fr/axis/cbrtools/manual/.

*4.4.2. Broadway*Tools for Building Recommender Systems on the Web*

Participant: Brigitte Trousse [correspondant].

Broadway*Tools is a toolbox supporting the creation of adaptive recommendation systems on the Web or in a Internet/Intranet information system. The toolbox offers different servers, including a server that computes recommendations based on the observation of the user sessions and on the re-use of user groups' former sessions. A recommender system created with Broadway*tools observes navigations of various users and gather evaluations and annotations, to draw up a list of relevant recommendations (Web documents, keywords, etc).

Based on Jaczynski'thesis [87], different recommender systems have been developed for supporting Web browsing, but also browsing inside a Web-based information system or for query formulation in the context of a meta search engine.

4.5. Activity Modeling

4.5.1. K-MADe for Describing Human Operator or User Activities

Participant: Dominique Scapin [correspondant].

K-MADe tool (Kernel of Model for Human Activity Description Environment). The K-MADe is intended for people wishing to describe, analyze and formalize the activities of human operators, of users, in environments (computerized or not), in real or simulated situation; in the field, or in the laboratory. Although all kinds of profiles of people are possible, this environment is particularly intended for ergonomics and HCI (Human Computer Interaction) specialists. It has been developed through collaboration between ENSMA (LISI XSlaboratory) and Inria.
This year a new version v1.2 of K-MAD was released in december. Its history, documentation and tool are available at: http://kmade.sourceforge.net/index.php. This work follows up the findings from the work of Caffiau and al. [73].

BANG Project-Team

5. Software

5.1. Software

5.1.1. Continuation of M3N

A large part of the software currently in use in the project-team was initiated and developed within former projects (Menusin, M3N).

5.1.2. CellSys

Participants: Dirk Drasdo [correspondent], Stefan Höhme [Research Associate, University of Leipzig], Adrian Friebel [PhD student, University of Leipzig], Tim Johann [Software Engineer, University of Leipzig], Nick Jagiella [PhD student].

Computer simulation software for image analysis of tissue samples at histological scales, as well as individual cell (agent)-based models of tumour and tissue growth solved either by systems of coupled equations of motion for each individual cell or by Kinetic Monte Carlo methods [56].
The software CellSys is currently been completely reorganised to permit easier use by external and internal researchers. The idea is to perspectively go open-source and offer consultancy for potential users.

CAD Team

5. Software

5.1. Softwares

1. Tsinghua University, Inria. Software: Modeling Software 2010.
2. Inria, Tsinghua University. Prototype: Tool for Curve Projection on Surfaces 2011.
3. Tsinghua University & Inria. Software: Spline Software Tool 2012.
4. Inria, Tsinghua University. Software: Mesh Segmentation Tool 2012.

We have developed various prototype software but, currently, they are not distributed and used only within the project. Regarding software, we know (see the last LIAMA evaluation report) that we should consider disseminating some of its codes more widely. Using established libraries may improve the impact of some of the results. We did not do it, due to the fast turn over of students in the Chinese team and the lack of Manpower and know-how in the French part.

CASCADE Project-Team

5. Software

5.1. MitMTool

Participants: Patrick Derbez, Jérémy Jean.

The purpose of MITMTOOL is to look for guess-and-determine and meet-in-the-middle attacks on AES and AES-based constructions. This tool allows us to improve known attacks on round-reduced versions of AES, on the LEX stream-cipher on the PELICAN Message Authentication Code and on fault attack on AES. Basically, it solves the problem to find all the solutions of a linear system of equations on the variables x and $S(x)$ where S is an inert function. The tool allows to compute the complexity of some good attack as well as the C code of the attack. We verify that the complexity estimates are accurate using experiments. We also use it to find one solution of the system for chosen-key differential attacks. There are mainly two tools: the first one only looks for guess-and-determine attack and tries to propagate some knowledge and guesses value when it cannot find automatically the value of some variable. The second tool uses the technique of the first tool and more advanced technique to take into account attacks with memory that use the meet-in-the-middle attack.

CLASSIC Project-Team (section vide)

CLIME Project-Team

5. Software

5.1. Polyphemus

Participants: Vivien Mallet, Anne Tilloy.

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

- libraries that gather data processing tools (SeldonData), physical parameterizations (AtmoData) and postprocessing abilities (AtmoPy);
- programs for physical preprocessing and chemistry-transport models (Polair3D, Castor, two Gaussian models, a Lagrangian model);
- drivers on top of the models in order to implement advanced simulation methods such as data assimilation algorithms.

Figure 1 depicts a typical result produced by Polyphemus. Clime is involved in the overall design of the system and in the development of advanced methods in model coupling, data assimilation and ensemble forecast (through drivers and post-processing).

In 2012, Polyphemus received several physical developments on secondary organic aerosols, modeling of pollution due to traffic emissions and coupling of local and regional scale models. Further integration with the data assimilation library Verdandi has been carried out. A Python interface to the Eulerian model Polair3D has been introduced.

5.2. Data assimilation library: Verdandi

Participants: Kévin Charpentier, Marc Fragu [MACS], Vivien Mallet, Dominique Chapelle [MACS], Philippe Moireau [MACS], Sergiy Zhuk, Anne Tilloy.

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

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

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

In 2012, versions 1.2, 1.3 and 1.4 were released. The design of the library has been improved for optimal performance and is now stable. It is possible to write models and observation managers in Python. Efficient support for parallel computations has been introduced. The documentation has been improved.

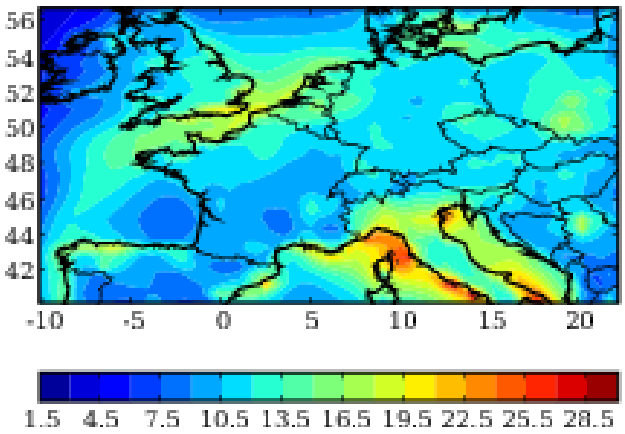

Figure 1. Map of the relative standard deviation (or spread, %) of an ensemble built with Polyphemus (ozone simulations, $\mu g\ m^{-3}$). The standard deviations are averaged over the summer of 2001. They provide an estimation of the simulation uncertainties.

5.3. Urban air quality analysis

Participants: Anne Tilloy, Vivien Mallet.

"Urban Air Quality Analysis" carries out data assimilation at urban scale. It merges the outputs of a numerical model (maps of pollutant concentrations) with observations from an air quality monitoring network, in order to produce the so-called analyses, that is, corrected concentration maps. The data assimilation computes the Best Linear Unbiased Estimator (BLUE), with a call to the data assimilation library Verdandi. The error covariance matrices are parameterized for both model simulations and observations. For the model state error covariances, the parameterization primarily relies on the road network. The software handles ADMS Urban output files, for a posteriori analyses or in an operational context.

CONTRAINTES Project-Team

5. Software

5.1. BIOCHAM, biochemical abstract machine

Participants: François Fages, Steven Gay, Sylvain Soliman.

The Biochemical Abstract Machine BIOCHAM [18] is a modeling environment for systems biology distributed as open-source since 2003. Current version is v3.4, released in October. BIOCHAM uses a compositional rule-based language for modeling biochemical systems, allowing patterns for expressing set of rules in a compact form. This rule-based language is compatible with the Systems Biology Markup Language (SBML) and is interpreted with three semantics correspnding to three abstraction levels:

1. the boolean semantics (presence or absence of molecules),
2. the stochastic semantics (discrete numbers of molecules),
3. the differential semantics (concentrations of molecules).

Based on this formal framework, BIOCHAM features:

- Boolean and numerical simulators (Rosenbrock's method for the differential semantics, Gillespie's algorithm with tau lipping for the stochastic semantics);
- a temporal logic language (CTL for qualitative models and LTL(R_{lin}) with numerical constraints for quantitative models) for formalizing biological properties such as reachability, checkpoints, oscillations or stability, and checking them automatically with model-checking techniques;
- automatic search procedures to infer parameter values, initial conditions and even reaction rules from temporal logic properties;
- automatic detection of invariants, through constraint-based analysis of the underlying Petri net;
- an SBGN-compatible reaction graph editor;
- an event handler allowing the encoding of hybrid models and formalisms.

BIOCHAM is implemented in GNU-Prolog and interfaced to the symbolic model checker NuSMV and to the continuous optimization tool CMAES developed by the EPI TAO.

5.2. Nicotine

Participant: Sylvain Soliman.

Nicotine is a GNU Prolog framework dedicated to the analysis of Petri nets. It was originally built for the computation of invariants using GNU Prolog's CLP(FD) solver [5] but has been further extended to allow import/export of various Petri nets formats. It provides as independent modules different features that can sometimes also be integrated in BIOCHAM, like SEPI computation, or left aside, like unambiguous ODE to Petri net conversion, since a more general heuristic conversion has been developed for BIOCHAM [8], [19].

5.3. STSE, spatio-temporal simulation environment

Participant: Szymon Stoma.

The overall goal of this project is to provide a software platform gathering a set of open-source tools and workflows facilitating spatio-temporal simulations (preferably of biological systems) based on microscopy data. The framework currently contains modules to digitize, represent, analyze, and model spatial distributions of molecules in static and dynamic structures (e.g. growing). A strong accent is put on the experimental verification of biological models by actual, spatio-temporal data acquired using microscopy techniques. Project was initially started at Humboldt University Berlin and moved to Inria with its founder. Project webpage is: http://stse-software.org.

5.4. YeastImageToolkit

Participant: Szymon Stoma.

YeastImageToolkit is an extention of YeastTracker software started originally by Jannis Uhlendorf. It allows following single cells in movies and quantifying fluorescent images based on this tracking as well as creating cell lineages. The software is currently under development and is designed to be a CellProfiler plugin facilitating yeast cell tracking, lineage and fluorescent signal quantification. Project webpage is: http://yeast-segtrack.weebly.com/.

5.5. SBMC, systems biology model-checker

Participant: Szymon Stoma.

Systems Biology Model Checker (SBMC) is a webservice allowing to verify biological models (e.g. signaling pathways stored in SBML files) against their specifications given in Signal Temporal Logics (STL). This project aims at providing to a large audience the methods described in [21] and used to analyse extrensic apoptosis pathway. Project webpage is: SBMC.

5.6. FO-CTL(R_{lin}), first-order computation tree logic over the reals

Participants: François Fages, Thierry Martinez.

FO-CTL(R_{lin}) is a solver for full First-Order Computation Tree Logic with linear arithmetic over the reals in constrained transition systems (CTS). CTS are transition systems where both states and transitions are described with constraints. FO-CTL(R_{lin}) generalizes the implementation done in Biocham of LTL(R_{lin}) for linear traces to branching Kripke structure.

5.7. Rules2CP

Participants: François Fages, Raphaël Martin, Thierry Martinez.

Rules2CP is a rule-based modeling language for constraint programming. It is distributed since 2009 as open-source. Unlike other modeling languages for constraint programming, Rules2CP adopts a single knowledge representation paradigm based on rules without recursion, and a restricted set of data structures based on records and enumerated lists given with iterators. This allows us to model complex constraint satisfaction problems together with search strategies, where search trees are expressed by logical formulae and heuristic choice criteria are defined with preference orderings by pattern-matching on the rules' left-hand sides.

The expressiveness of Rules2CP has been illustrated in the FP6 Strep project Net-WMS by a complete library for packing problems, called PKML (Packing Knowledge Modeling Library), which, in addition to pure bin packing and bin design problems, can deal with common sense rules about weights, stability, as well as specific packing business rules.

5.8. SiLCC, linear concurrent constraint programming

Participant: Thierry Martinez.

SiLCC is an extensible modular concurrent constraint programming language relying upon linear logic. It is a complete implementation of the Linear logic Concurrent Constraint programming paradigm of Saraswat and Lincoln using the formal semantics of Fages, Ruet and Soliman. It is a single-paradigm logical language, enjoying concurrency, imperative traits, and a clean module system allowing to develop hierarchies of constraint systems within the language.

This software prototype is used to study the design of hierarchies of extensible libraries of constraint solvers. SiLCC is also considered as a possible implementation language for restructuring the code of BIOCHAM.

5.9. EMoP, existential modules for Prolog

Participant: Thierry Martinez.

EMoP is an extension of Prolog with first-class modules. These modules have the formal semantics of the LCC modules and provide Prolog with notions of namespaces, closures and objects within a simple programming model. Modules are also the support for user-definition of macros and modular syntax extensions. EMoP is bootstrapped and uses the GNU Prolog compilation chain as back-end.

5.10. CHRat, CHR with ask and tell

Participant: Thierry Martinez.

CHRat is a modular version of the well known Constraint Handling Rules language CHR, called for CHRat for CHR with *ask* and *tell*. Inspired by the LCC framework, this extension of CHR makes it possible to reuse CHRat components both in rules and guards in other CHRat components, and define hierarchies of constraint solvers. CHRat is a bootstrapped preprocessor for CHR which generates code for SWI/Prolog.

5.11. CLPGUI, constraint logic programming graphical user interface

Participant: François Fages.

CLPGUI is a generic graphical user interface written in Java for constraint logic programming. It is available for GNU-Prolog and SICStus Prolog. CLPGUI has been developed both for teaching purposes and for debugging complex programs. The graphical user interface is composed of several windows: one main console and several dynamic 2D and 3D viewers of the search tree and of finite domain variables. With CLPGUI it is possible to execute incrementally any goal, backtrack or recompute any state represented as a node in the search tree. The level of granularity for displaying the search tree is defined by annotations in the CLP program.

CLPGUI has been mainly developped in 2001 and is distributed as third-party software on GNU-Prolog and SICStus Prolog web sites. In 2009, CLPGUI has been interfaced to Rules2CP/PKML and used in the FP6 Strep Net-WMS with a non-released version.

DEDUCTEAM Team

4. Software

4.1. Dedukti

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

Dedukti is designed to be versatile: it must be efficient on proofs that contain many computations—such as proofs by reflection—as well as proofs that do not contain any–such as proofs coming from HOL. These constraints has led us to adopt a Just-In-Time compilation architecture. And instead of designing our own JIT compiler, we have chosen to reuse the cutting-edge LuaJIT compiler. This technological choice, namely devolving the type-checking to Lua, makes Dedukti a proof-checker generator.

This has allowed the introduction of many optimizations: a normalization by evaluation strategy, a higher-order abstract syntax representation of terms and a context-free, bidirectional type-checking algorithm [22].

Dedukti has been developed by Mathieu Boespflug, Olivier Hermant, Quentin Carbonneaux, and Ronan Saillard.

4.2. CoqInE and HOLiDe

Dedukti comes with two companion tools: HOLiDe, an embedding of HOL proofs through the OpenTheory format [51], and CoqInE, an embedding of Coq proofs. Almost all the standard library of HOL and a significant part of that of Coq are checked by Dedukti.

CoqInE now supports the following features of Coq: the raw Calculus of Constructions, inductive types, and fixpoint definitions. It is now able to translate more than 80% of the standard library of Coq [21]. Ongoing work focuses on modules and functors, and on universes.

CoqInE has been developed by Mathieu Boespflug, Guillaume Burel, and Ali Assaf.

HOLiDe supports all the features of HOL, including polymorphism, constant definitions, and type definitions. It is able to translate all of the OpenTheory standard theory library.

HOLiDe has been developed by Ali Assaf.

4.3. iProver Modulo

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

It is developed by Guillaume Burel.

These four systems are available on the website of the team.

FORMES Team

5. Software

5.1. aCiNO

Participants: Fei He [correspondant], Min Zhou.

aCiNO is an SMT (Satisfiability Modulo Theory) solver based on a Nelson-Oppen [62] architecture, and written in C++. Currently, two popular theories are considered: linear real arithmetic (LRA) and uninterpreted functions (UF). A lazy approach is used for solving SMT problem. For efficiency consideration, the solver is implemented in an incremental way. It also invokes an online SAT solver, which is now a modified MiniSAT, so that recovery from conflict is possible.

5.2. CoLoR

Participants: Frédéric Blanqui [correspondant], Kim-Quyen Ly.

CoLoR is a Coq [42] library on rewriting theory and termination of more than 72,000 lines of code [4]. It provides definitions and theorems for:

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

CoLoR is distributed under the CeCILL license on http://color.inria.fr/. Various people participated to its development (see the website for more information).

5.3. CoqMT

Participants: Qian Wang [correspondant], Jean-Pierre Jouannaud.

The proof-assistant Coq is based on a complex type theory, which resulted from various extensions of the Calculus of Constructions studied independently from each other. With the collaboration of Bruno Barras, we decided to address the challenge of proving the real type theory underlying Coq, and even, indeed, of its recent extension CoqMT developed in FORMES by Pierre-Yves Strub. To this end, we have studied formally the theory CoqMTU, which extends the pure Calculus of Constructions by inductive types, a predicative hierarchy of universes, and a decidable theory T for some first-order inductive types [1]. Recently, we were able to announce the complete certification of CoqMTU in Coq augmented with appropriate intuitionistic set-theoretic axioms in order to fight Gödel's incompleteness theorem, a work which has not been published yet. As a consequence, Coq and CoqMTU are the first proof assistants which consistency (relative to intuitionistic set theory IZF augmented with the afore-mentioned axioms) is formally entirely proved (in Coq). While previous formal proofs for Coq and other proof assistants all assumed strong normalization, the present one *proves* strong normalization thanks to the new notion of *strongly-normalizing* model introduced by Bruno Barras. While consistency is done already, decidability of type-checking remains to be done. This is a straightforward consequence for Coq, but a non-trivial task for CoqMTU because of the interaction between inductive types and the first-order theory T. It should however be announced around the turn of the year. We consider this work as a major scientific achievement of the team.

5.4. EDOLA

Participants: Hehua Zhang [correspondant], Ming Gu, Hui Kong.

Joint work with Jiaguang Sun (Tsinghua University, China).

EDOLA [72] is an integrated tool for domain-specific modeling and verification of PLC applications [70]. It is based on a domain-specific modeling language to describe system models. It supports both model checking and automatic theorem proving techniques for verification. The goal of this tool is to possess both the usability in domain modeling, the reusability in its architecture and the capability of automatic verification.

For the moment, we have developed a prototype of the EDOLA language, which can easily describe the features of PLC applications like the scan cycle mechanism, the pattern of environment model, time constraints and five property patterns. TLA+ [56] was chosen as the intermediate language to implement the automatic verification of EDOLA models. A prototype of EDOLA has also been developed, which comes along with an editor to help writing EDOLA models. To automatically verify properties on EDOLA models, it provides the interface for both a model checker TLC [56] and a first-order theorem prover SPASS [71].

5.5. HOT

Participant: Frédéric Blanqui [correspondant].

HOT is an automated termination prover for higher-order rewrite systems based on the notion of computability closure and size annotation [13]. It won the 2012 competition in the category "higher-order rewriting union beta". The sources are not public.

5.6. Moca

Participant: Frédéric Blanqui [correspondant].

Joint work with Pierre Weis (Inria Rocquencourt) and Richard Bonichon (CEA).

Moca is a construction functions generator for OCaml [57] data types with invariants.

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

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

Moca supports two kinds of relations:

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

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

Moca is distributed under QPL on http://moca.inria.fr/.

5.7. Rainbow

Participants: Frédéric Blanqui [correspondant], Kim-Quyen Ly.

Rainbow is a tool for verifying the correctness of termination certificates expressed in the CPF XML format as used in the termination competition. Termination certificates are currently translated and checked in Coq by using the CoLoR library. But a new standalone version is under development using Coq extraction mechanism.

Rainbow is distributed under the CeCILL license on http://color.inria.fr/rainbow.html. See the website for more information.

5.8. SimSoC

Participant: Vania Joloboff [correspondant].

SimSoC is an infrastructure to run simulation models which comes along with a library of simulation models. SimSoC allows its users to experiment various system architectures, study hardware/software partition, and develop embedded software in a co-design environment before the hardware is ready to be used. SimSoC aims at providing high performance, yet accurate simulation, and provide tools to evaluate performance and functional or non functional properties of the simulated system.

SimSoC is based on SystemC standard and uses Transaction Level Modeling for interactions between the simulation models. The current version of SimSoC is based on the open source libraries from the OSCI Consortium: SystemC version 2.2 and TLM 2.0.1 [52], [25]. Hardware components are modeled as TLM models, and since TLM is itself based on SystemC, the simulation is driven by the SystemC kernel. We use standard, unmodified, SystemC (version 2.2), hence the simulator has a single simulation loop.

The second open source version of SimSoC, SimSoC v0.7.1, has been released in November 2010. It contains a full simulator for ARM V5 and PowerPC both running at an average speed of about 80 Millions instructions per second in, and a simulator for the MIPS architecture with an average speed of 20 Mips in mode DT1. It represents about 70,000 lines of source code and includes:

SimSoC is distributed under LGPL on https://gforge.inria.fr/projects/simsoc.

5.9. SimSoC-Cert

Participants: Frédéric Blanqui, Vania Joloboff, Jean-François Monin [correspondant], Xiaomu Shi.

SimSoC-Cert is a set of tools that can automatically generate in various target languages (Coq and C) the decoding functions and the state transition functions of each instruction and addressing mode of the ARMv6 architecture manual [22] (implemented by the ARM11 processor family) but the Thumb and coprocessor instructions. The input of SimSoC-Cert is the ARMv6 architecture manual itself.

Based on this, we first developed *simlight* (8000 generated lines of C, plus 1500 hand-written lines of C), a simulator for ARMv6 programs using no peripheral and no coprocessor. Next, we developed *simlight2*, a fast ARMv6 simulator integrated inside a SystemC/TLM module, now part of SimSoC v0.7.

We can also generate similar programs for SH4 [24] but this is still experimental (work done by Frédéric Tuong in 2011).

Finally, we started to prove that the C code for simulating ARM instructions in Simlight is correct with respect to the Coq model.

GALLIUM Project-Team

5. Software

5.1. OCaml

Participants: Damien Doligez [correspondant], Xavier Clerc [team SED], Alain Frisch [LexiFi], Jacques Garrigue [Nagoya University], Thomas Gazagnaire [OCamlPro], Fabrice Le Fessant [Inria Saclay and OCamlPro], Xavier Leroy, Luc Maranget [EPI Moscova].

OCaml, formerly known as Objective Caml, is our flagship implementation of the Caml language. From a language standpoint, it extends the core Caml language with a fully-fledged object and class layer, as well as a powerful module system, all joined together by a sound, polymorphic type system featuring type inference. The OCaml system is an industrial-strength implementation of this language, featuring a high-performance native-code compiler for several processor architectures (IA32, AMD64, PowerPC, ARM, etc) as well as a bytecode compiler and interactive loop for quick development and portability. The OCaml distribution includes a standard library and a number of programming tools: replay debugger, lexer and parser generators, documentation generator, compilation manager, and the Camlp4 source pre-processor.

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

5.2. CompCert C

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

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

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

5.3. Zenon

Participant: Damien Doligez.

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

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

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

GAMMA3 Project-Team (section vide)

GANG Project-Team (section vide)

HIPERCOM Project-Team

5. Software

5.1. RPL P2P

Participants: Emmanuel Baccelli [correspondant], Oliver Hahm, Matthias Philipp.

P2P-RPL is an implementation of draft-ietf-roll-p2p-rpl, providing reactive discovery of point-to-point routes in low power and lossy networks such as wireless sensor networks. The implementation is based on the Contiki operating system. See also the web page http://contiki-p2p-rpl.gforge.inria.fr/.

- Version: 0.4

5.2. MPR-OSPF

Participants: Emmanuel Baccelli [correspondant], Juan-Antonio Cordero.

MPR-OSPF is an implementation of RFC5449, providing OSPF-compatible routing in hybrid networks composed of both mobile ad hoc routers and fixed wired networks. The implementation is based on Quagga/Zebra. See also the web page http://ospfmanet.gforge.inria.fr.

- Version: 0.4

5.3. OPERA infrastructure

Participants: Cédric Adjih [correspondant], Ichrak Amdouni, Pascale Minet, Ridha Soua.

OPERA-infrastructure is the system support code of OPERA, the Optimized Protocol for Energy efficient Routing with node Activity scheduling.

5.4. OPERA perf simul

Participants: Cédric Adjih [correspondant], Ichrak Amdouni.

OPERA-perf-simul is a set of tools for simulation and performance evaluation as well as large scale tests of OPERA, the Optimized Protocol for Energy efficient Routing with node Activity scheduling.

5.5. OPERA protocol

Participants: Cédric Adjih [correspondant], Ichrak Amdouni, Pascale Minet, Saoucene Mahfoudh Ridene.

OPERA-protocol is the heart of OPERA, the Optimized Protocol for Energy efficient Routing with node Activity scheduling. It includes EOND a neighborhood discovery protocol, EOSTC a protocol byuiding and maintaining a n energy efficient routing tree and SERENA a node coloring algorithm.

5.6. OPERA validation and tools

Participant: Cédric Adjih [correspondant].

OPERA-validation and tools is a set of tools for validation, debugging, analysis and visualization of OPERA protocol, the Optimized Protocol for Energy efficient Routing with node Activity scheduling. It operates either in a real embedded system or in simulation.

IMARA Project-Team

5. Software

5.1. MELOSYM

Participants: Fawzi Nashashibi [correspondant], Benjamin Lefaudeux, Paulo Lopes Resende.

MELOSYM is the acronym for "**M**odélisation de l'**E**nvironnement et **LO**calisation en temps réel pour un **SY**stème **M**obile autonome ou pas, fondé sur des données du capteur laser". This is a SLAM based algorithm for the environment mapping and vehicle localization in real time using laser data. The particularity of the algorithm is its hierarchical approach that improves the accuracy of the system and speeds up the computations.

- Version: V2

5.2. Stereoloc-3D

Participants: Benjamin Lefaudeux, Fawzi Nashashibi [correspondant].

This software is a stereovision based system capable of performing a vehicle or robot ego-localization and 3D environment mapping in real-time. It has also the capability to ensure mobile objects detection and tracking.

- Version: V1

IMEDIA2 Team

5. Software

5.1. IKONA/MAESTRO Software

Participants: Vera Bakic, Laurent Joyeux, Souheil Selmi.

IKONA is a generalist software dedicated to content-based visual information indexing and retrieval. It has been designed and implemented in our team during the last years [22]. Its main functionalities are the extraction, the management and the indexing of many state-of-the-art global and local visual features. It offers a wide range of interactive search and navigation methods including query-by-example, query-by-window, matching, relevance feedback, search results clustering or automatic annotation. It can manage several types of input data including images, videos and 3D models.

Based on a client/server architecture, it is easily deployable in any multimedia search engine or service. The communication between the two components is achieved through a proprietary network protocol. It is a set of commands the server understands and a set of answers it returns to the client. The communication protocol is extensible, i.e. it is easy to add new functionalities without disturbing the overall architecture. can be replaced by any new or existing protocol dealing with multimedia information retrieval.

The main processes are on the server side. They can be separated in two main categories:

- off-line processes: data analysis, features extraction and structuring
- on-line processes: answer the client requests

Several clients can communicate with the server. A good starting point for exploring the possibilities offere by IKONA is our web demo, available at https://www.rocq.inria.fr/cgi-bin/imedia/circario.cgi/bio_diversity?select_db=1. This CGI client is connected to a running server with several generalist and specific image databases, including more than 23,000 images. It features query by example searches, switch database functionality and relevance feedback for image category searches. The second client is a desktop application. It offers more functionalities. More screen-shots describing the visual searching capabilities of IKONA are available at https://www.rocq.inria.fr/cgi-bin/imedia/circario.cgi/demos.

IKONA is a pre-industrial prototype, with exploitation as a final objective. Currently, there does not exist a licensed competitor with the same range of functionalities. It exists several commercial softwares or systems exploiting technologies similar to some functionalities of IKONA but usually not the most advanced ones. We can for example cite the SDK developed by LTU company, the service proposed by AdVestigo company, etc. Many prototypes and demonstrators, industrial or academic, share some functionalities of IKONA but here again not the most advanced (e.g. Google Image Similarity Search Beta, IBM Muffin, etc.).

The main originality of IKONA is its **genericity** (in terms of visual features, metrics, input data, storage format, etc.), its **adaptivity** (to new visual features, new indexing structures or new search algorithms), its innovative **interactive** search functionalities (Local and Global Relevance Feedback, Local Search with Query Expansion, Search results clustering, etc.) and its **scalability** thanks to a generic indexing structure module than can support the integration of any new advances.

Current Users of IKONA include European and National Projects Participants through its integration in prototype multimedia systems, commercial companies through user trials (EXALEAD, INA, BELGA, AFP), General or Specific Public through Web demos (Pl@ntNet leaf identification demo).

IKONA software provides a high degree of visibility to IMEDIA2 scientific works through demos in commercial, scientific and general public events (notably in most Inria national showrooms). It is also the mainstay of several Multimedia Systems developed at the European level, in conjunction with many Leader European Companies and Research Centers.

MATHRISK Team

4. Software

4.1. PREMIA

Participants: Antonino Zanette, Mathrisk Research team, Agnès Sulem [correspondant].

Premia is a software designed for option pricing, hedging and financial model calibration. It is provided with it's C/C++ source code and an extensive scientific documentation. https://www-rocq.inria.fr/mathfi/Premia

The Premia project keeps track of the most recent advances in the field of computational finance in a well-documented way. It focuses on the implementation of numerical analysis techniques for both probabilistic and deterministic numerical methods. An important feature of the platform Premia is the detailed documentation which provides extended references in option pricing.

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

Besides being a single entry point for accessible overviews and basic implementations of various numerical methods, the aim of the Premia project is:

1. to be a powerful testing platform for comparing different numerical methods between each other;
2. to build a link between professional financial teams and academic researchers;
3. to provide a useful teaching support for Master and PhD students in mathematical finance.

- AMS: 91B28;65Cxx;65Fxx;65Lxx;65Pxx
- License: Licence Propriétaire (genuin license for the Consortium Premia)
- Type of human computer interaction: Console, interface in Nsp, Web interface
- OS/Middelware: Linux, Mac OS X, Windows
- APP: The development of Premia started in 1999 and 14 are released up to now and registered at the APP agency.
- Programming language: C/C++ librairie Gtk
- Documentation: the PNL library is interfaced via doxygen
- Size of the software: 11 Mbyte of code split into 275,000 lines. 93 Mbyte of PDF files of documentation
- Publications: [1] [64] [78] [89] [95], [47]

4.1.1. Content of Premia

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

1. **Equity derivatives:**

   The following models are considered:

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

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

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

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

2. **Interest rate derivatives**

   The following models are considered:

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

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

3. **Credit derivatives: CDS, CDO**

   Reduced form models and copula models are considered.

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

4. **Hybrid products**

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

5. **Energy derivatives: swing options**

   Mean reverting and jump models are considered.

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

4.1.2. Premia design

Anton Kolotaev (ADT engineer), supervised by J. Lelong, has developed a web platform allowing online tests (https://quanto.inria.fr/premia/koPremia). This online version allow us to supply benchmarks both for professional R&D teams and academics in mathematical finance. This will considerably increase the impact and the visibility of the software. Up to now, to use the opensource version of the software, one has to download from Premia's website and install it on its own computer and this had become a brake on using Premia. Providing an online version of Premia is an original way of keeping up with the new standards of software usability without focusing too much on a dedicated solution per operation system.
To enable easy an advanced usage of Premia without being an advanced C or C++ programmer, we have started to implement Python bindings. The choice of Python has been quite obvious as Python has become over the past few years a standard cross–platform interpreted language for numerical problems.

Premia has managed to grow up over a period of more than a dozen years; this has been possible only because contributing an algorithm to Premia is subject to strict rules, which have become too stringent. To facilitate contributions, a standardized numerical library (PNL) has been developed under the LGPL since 2009, which offers a wide variety of high level numerical methods for dealing with linear algebra, numerical integration, optimization, random number generators, Fourier and Laplace transforms, and much more. Everyone who wishes to contribute is encouraged to base its code on PNL and providing such a unified numerical library has considerably eased the development of new algorithms which have become over the releases more and more sophisticated. An effort will be made to continue and stabilize the development of PNL.

4.1.3. Algorithms implemented in Premia in 2012

Premia 14 was delivered to the consortium members in March 2012. It contains the following new algorithms:

- **Interest Rate Derivatives**
  - An n-Dimensional Markov-functional Interest Rate Model
    L. Kaisajuntti J. Kennedy. Preprint 2008
  - Efficient log-Levy approximations for Levy-driven Libor model.

A. Papapantoleon, J.Schoenmakers, D. Skovmand.
Preprint 2111, TU Berlin.

- **Energy and Commodities**
  - Efficient pricing of Swing options in Lévy-driven models. O. Kudryavtsev, A. Zanette.
- **Credit Risk Derivatives**
  - Calibration in a local and stochastic intensity model. A. Alfonsi, C. Labart, J. Lelong
- **Equity Derivatives**
  - Forward Variance Dynamics: Bergomi's model revisited. S.M. Ould Aly
  - Volatility of Volatility Expansion for Bergomi's model. S.M. Ould Aly
  - Robust Approximations for Pricing Asian Options and Volatility Swaps Under Stochastic Volatility. M. Forde, A.Jacquier *Applied Mathematical Finance, Volume 17 Issue 3 2010*
  - Small-time asymptotics for implied volatility under the Heston model, M. Forde, A. Mijatovic, and Jaquier, A. *International Journal of Theoretical and Applied Finance, Volume 12, issue 6, 2009*
  - Asymptotic formulae for implied volatility under the Heston model. Forde, F, Jacquier, A, and Mijatovic, A. *Proceedings of the Royal Society A, to appear*.
  - A Mean-Reverting SDE on Correlation Matrices. A. Alfonsi, A.Ahdida
  - Fast and Accurate Long Stepping Simulation of the Heston Stochastic Volatility Model J.H. Chan, M S. Joshi, Preprint
  - High order discretization schemes for stochastic volatility models. B. Jourdain, M. Sbai. *Quantitative Finance, to appear*
  - A Fourier-based Valuation Method for Bermudan and Barrier Options under Heston Model. F.Fang, C. W. Oosterlee. *SIAM J. Finan. Math. 2, 2011, 439-463.*
  - Numerical methods and volatility models for valuing cliquet options.
    H. Windcliff, P.A. Forsyth, K.R. Vetzal, *Applied Mathematical Finance 13 2006.*
  - Pricing Discretely Monitored Asian Options by Maturity Randomization. G.Fusai, D. Marazzina, M. Marena. *SIAM Journal on Financial Mathematics, Vol. 2 2011*
  - Wiener-Hopf techniques for Lookback options in Levy models. O. Kudryavtsev
  - Computing VaR and AVar in Infinitely Divisible Distributions. Y.S. Kim, S. Rachev, M.S. Bianchi, F.J. Fabozzi. *Probability and Mathematical Statistics, Vol. 30, Fasc. 2 2010.*
  - Analytical formulas for local volatility model with stochastic rates. E. Benhamou, E. Gobet, M. Miri 2009
  - Nonparametric Variance Reduction Methods on Malliavin Calculus. B. Lapeyre, A. Turki *SIAM Journal on Financial Mathematics, to appear*
  - Stochastic expansion for the pricing of call options with discrete dividends. P. Etore, E. Gobet. *Applied Mathematical Finance, to appear*
  - American options in high dimension solving EDSR with penalization C. Labart, J. Lelong
  - Static Hedging of Standard Options. P. Carr, L. Wu. Preprint.

The software Premia 14 has been deposited at the APP (Agence pour la Protection des Programmes) with the reference IDDN.FR.001.190010.011.S.C.2001.000.31000.

MICMAC Project-Team (section vide)

MUTANT Project-Team

5. Software

5.1. Antescofo

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

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

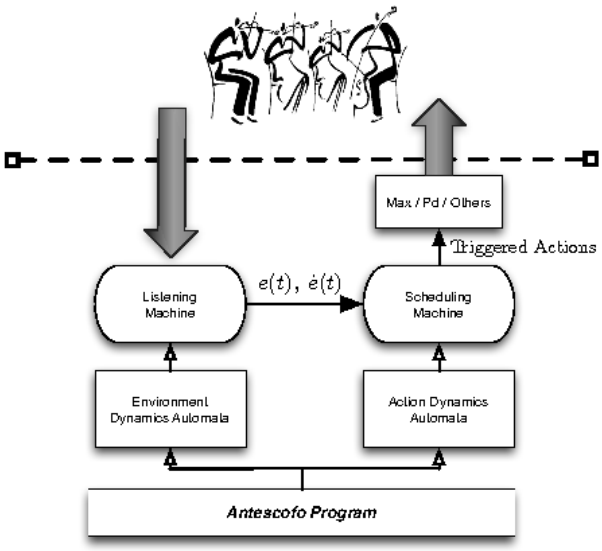

Figure 2. General scheme of Antescofo virtual machine

Antescofo is developed as modules for *Max* and *PureData* real-time programming environments.

5.2. Antescofo Visual Editor

Participants: Thomas Coffy [ADT], Arshia Cont, José Echeveste.

The Antescofo programming language can be extended to visual programing to better integrate existing scores and to allow users to construct complex and embedded temporal structures that are not easily integrated into text. This project has started in October 2012 thanks to Inria ADT Support. The foundations of a visual editor is set and the goal is to release a standalone editor for Antescofo programs in 2013.

PARKAS Project-Team

5. Software

5.1. Lucid Synchrone

Participant: Marc Pouzet [contact].

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

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

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

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

Three major release of the language has been done and the current version is V3 (dev. in 2006). The language is still used for teaching and in our research but we do not develop it anymore. Nonetheless, we have integrated several features from Lucid Synchrone in new research prototypes described below.

5.2. ReactiveML

Participants: Mehdi Dogguy, Louis Mandel [contact], Cédric Pasteur.

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

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

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

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

The language was mainly used for the simulation of mobile ad hoc networks at the Pierre and Marie Curie University and for the simulation of sensor networks at France Telecom and Verimag (CNRS, Grenoble).

In 2012, a new automatic build system for ReactiveML program based on ocamlbuild has been implemented. A new static analysis which checks that programs cooperate has been developed. A full ReactiveML toplevel compiled into JavaScript has been made available at http://rml.lri.fr/tryrml. The ReactiveML distribution has also been cleaned up.

5.3. Heptagon

Participants: Cédric Pasteur [contact], Brice Gelineau, Léonard Gérard, Adrien Guatto, Marc Pouzet.

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

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

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

Heptagon is jointly developed by Gwenael Delaval and Alain Girault from the Inria POP ART team (Grenoble).

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

Participants: Louis Mandel [contact], Adrien Guatto, Marc Pouzet.

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

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

In 2012, a first version of the code generator has been distributed. The typing algorithms has been improved.

5.5. ML-Sundials

Participants: Timothy Bourke, Marc Pouzet [contact].

ML-Sundials library provides an Ocaml interface to the Sundials numerical suite [7] (version 2.4.0). This library is used for solving and initial value problem and includes a zero-crossing detection mechanism. Only the CVODE solver with serial nvectors is currently supported. The structure and naming conventions largely follow the original libraries, both for ease of reading the existing documentation and for converting existing source code, but several changes have been made for programming convenience, namely:

- solver sessions are configured through algebraic data types rather than through multiple function calls,
- error conditions are signalled by exceptions rather than return codes (including in user-supplied callback routines),
- closures (partial applications of higher-order functions) are used to share user data between callback routines, and,
- explicit free commands are not necessary nor provided since Ocaml is a garbage-collected language.

The library is in use in a new synchronous hybrid language we are currently developping.

5.6. GCC

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

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

Licence: GPLv3+ and LGPLv3+

[7] https://computation.llnl.gov/casc/sundials/main.html

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

PARKAS contributes to the polyhedral compilation framework, also known as Graphite. We also distribute an experimental branch for a stream-programming extension of OpenMP, parallel data-flow programming, and automatic parallelization to a data-flow runtime or architecture. This experiment borrows key design elements to synchronous data-flow languages.

Tobias Grosser is the maintainer of the Graphite optimization pass of GCC.

5.7. isl

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

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

Licence: MIT

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

5.8. ppcg

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

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

Licence: LGPLv2.1+

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

5.9. Ott: tool support for the working semanticist

Participant: Francesco Zappa Nardelli [contact].

Languages, semantics, tool support, theorem prouvers.

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

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

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

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

In collaboration with Peter Sewell (Cambridge University).

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

In 2012 we implemented several bug-fixes and we kept the theorem prouver backends up-to date with the prover evolution. We have also been working toward a closer integration with the Lem tool.

The currently relased version is 0.21.2.

5.10. Lem: a tool for lightweight executable semantics

Participant: Francesco Zappa Nardelli [contact].

Languages, semantics, tool support, theorem prouvers.

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 and Isabelle code; the OCaml backend uses a finite set library (and does not yet support inductive relations). A Coq backend is in development.

Lem is already in use at Cambridge and Inria for research on relaxed-memory concurrency. We are currently preparing a feature-complete release with back-ends for HOL4, Isabelle/HOL, Coq, OCaml, and LaTeX. The project web-page is http://www.cl.cam.ac.uk/~so294/lem/.

In collaboration with Scott Owens (U. Kent, UK) and Peter Sewell (U. Cambridge, UK).

5.11. Cmmtest: a tool for hunting concurrency compiler bugs

Participants: Francesco Zappa Nardelli [contact], Robin Morisset, Pankaj Pawan.

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

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

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

Although not yet publicly distributed, cmmtest already 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.

PI.R2 Project-Team

5. Software

5.1. http://coq.inria.frCOQ

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

5.1.1. Version 8.4

Version 8.4 was released in August 2012. It introduced a new proof engine designed and implemented by Arnaud Spiwack and a new extensive modular library of arithmetic contributed by Pierre Letouzey. It also included an extension of the underlying logic with "η-conversion" by Hugo Herbelin and "commutative-cuts compliant guard condition" by Pierre Boutillier, an extension of the pattern-matching compilation algorithm by Hugo Herbelin, an extension of the procedure of simplification of polynomial expressions by Loïc Pottier, a refinement of the type classes mechanism by Matthieu Sozeau, a new communication model by Vincent Gross for the graphical user interface CoqIDE, that Pierre Letouzey, Pierre Boutillier and Pierre-Marie Pédrot further extended.

Several users gracefully contributed improvements of various features (Tom Prince, Enrico Tassi, Daniel Grayson, Hendrik Tews, ...).

5.1.2. Graphical user interface

Pierre Letouzey has finalised and extended the work initiated by Vincent Gross (former ADT engineer) concerning the CoqIDE user interface: CoqIDE and Coq are now separate Unix processes, enhancing the reliability and improving the user experience.

In Fall 2012, Pierre Letouzey also revised the event infrastructure of CoqIDE, from a thread-based model to pure GTK event-loop. This way, CoqIDE is more reactive, less subject to deadlocks (especially under Windows), and the source code is more idiomatic and easier to understand. Interestingly, this work takes advantage of deeper notions such as C.P.S. (continuation passing style).

Pierre Boutillier and Pierre-Marie Pédrot built an abstract communication interface between Coq and CoqIDE based on XML syntax. They also refined the ability to customise CoqIDE. Pierre Boutillier made CoqIDE rely on Gtksourceview.

5.1.3. Proof engine

Arnaud Spiwack has proposed an extension of the expressiveness of tactics based on his previous work for a new proof engine. It allows for more atomic tactics, has a primitive support for backtracking, and allows for tactics which manipulate several goals.

5.1.4. Evaluation algorithms

Pierre Boutillier has proposed a new unfolding algorithm for global constants so that the definition of these ones are unfolded only if it triggers extra reductions. This helps users to keep goals concise during interactive proofs.

5.1.5. Type classes, internal representation

Matthieu Sozeau is adapting the type-classes mechanism to benefit from the new tactic engine and avoid reimplementing a whole proof-search algorithm with backtracking on top of the tactic language. This will bring high benefits in terms of efficiency and ease of use to the users. Forward proof-search for type classes was stabilised and is now used in libraries for better control on the search space, notably in the MathClasses library developed in Nijmegen [69].

An important shortcoming of type classes is the verbosity of the representation of projections from a class, as was illustrated in François Garillot's PhD thesis [48]. Matthieu Sozeau has developed a branch of Coq supporting an efficient representation of these projections based on the idea of bidirectional type checking which is now under stabilisation. This support will also enhance the performance of the assistant on developments using regular parameterised records and dependent sums like the HoTT library on homotopy type theory and the Forcing plugin developed by Sozeau *et al* [32].

5.1.6. Universes

While visiting the Institute for Advanced Study, Matthieu Sozeau implemented a new system of universe polymorphism that makes it possible to develop highly generic theories in the Coq system. Based on ideas from Harper and Pollack's [53] design of polymorphism as an elaboration in the Lego theorem prover, he developed an original algorithm for type inference of universes and implemented it in Coq. Its first application is inside the Homotopy Type Theory (HoTT) research program, as the formalisation of HoTT requires a high level of polymorphism that was not available before. Many other theories will benefit from this, including Sozeau's work on Forcing, B. Barras' (Typical) work on models of type theory or in the Math Classes library mentioned before. It also opens up possibilities to formalise category-theoretic notions without being limited by the universe system, a long-standing barrier in the Coq proof assistant.

5.1.7. The Equations plugin

Matthieu Sozeau continued the maintenance of the Equations plugin and developed a new Forcing plugin for Coq (see below).

5.1.8. Tools

Pierre-Marie Pédrot has written a program using the internal representation of libraries to compact Coq object files. It is based on well-known automata algorithms, representing memory as transition systems. The idea underlying this program is generalisable to any OCaml data structure, provided some conditions on its use are satisfied, and was formalised in a paper that was accepted at JFLA 2013.

5.1.9. Internal architecture of the Coq software

Pierre Letouzey continued a large reorganisation of the internal components of Coq, since these components are currently too much interdependent. This work brought better isolation between some of the Coq components and explicit interfaces between them. This allowed to simplify the compilation of Coq, since it is now easier to build the OCaml syntax extension used when compiling many advanced parts of Coq. Moreover, this clearer architecture should also help new contributors when they discover and interact with this large and complex code-base.

Pierre-Marie Pédrot also made some reorganisations of the code. This includes a clean generic library superseding the one of OCaml, pushing the CAMLP4/5 dependent parts out of the lower strata, as well as benefiting from the OCaml module system to get more uniformity in the naming of usual data structures.

Pierre Letouzey proposed a nicer backtracking infrastructure to Coq, used when the user wants to cancel some recent commands and go back before them. This new infrastructure unifies and improves what was used earlier by ProofGeneral and CoqIDE, the two main user interfaces for Coq.

Pierre Letouzey also dedicated many efforts to improve the support of the Windows platform by Coq.

5.1.10. Efficiency

Pierre-Marie Pédrot has been trying to optimise various parts of the Coq system, including the new tactical system designed by Arnaud Spiwack. Some neat tricks on garbage collection permitted to reach a substantial time improvement in compilation of object files. Various architectural modifications were also made in the process, like trying to get rid of the generic comparison in the code base.

Pierre Letouzey investigated an alternative implementation of the code dealing with Coq universes. These universes are a critical part of Coq: they have direct consequences on Coq safety, and handling them is time-consuming (between 10% to 20% depending on the Coq usage). This alternative implementation looks promising, but still requires some more work and stress-tests before being integrated in mainstream Coq.

5.1.11. Documentation generation

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

5.1.12. General maintenance

Pierre Letouzey has been the main maintainer of Coq with extra contributions from Hugo Herbelin, Pierre Boutillier, Matthieu Sozeau, Pierre-Marie Pédrot, ...

5.1.13. Development Action

An “Action de Développement Technologique” about Coq started September 2011 and continued this year. It gathers the πr^2 team, the Marelle team and the CPR team from CNAM, Hugo Herbelin acting as the coordinator. It supports visits and meetings between developers and aims at strengthening the community of Coq users and contributors.

Yann Régis-Gianas set up an “osqa” server for Frequently Asked Questions.

The ADT Coq supported the internship of François Ripault.

Hugo Herbelin formalised a type-theoretic construction of semi-simplicial sets answering a problem raised early this year by Steve Awodey, Peter LeFanu Lumsdaine and others, in relation with the homotopy models of type theory.

5.2. Pangolin

Participant: Yann Régis-Gianas.

Yann Régis-Gianas maintained a prototype version of Pangolin. He used it to prove concrete complexity bounds for a set of functional programs using the method described in his FOPARA 2011 paper [19].

5.3. Other software developments

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

POLSYS Project-Team

5. Software

5.1. FGb

Participant: Jean-Charles Faugère [contact].

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.
See also the web page http://www-polsys.lip6.fr/~jcf/Software/FGb/index.html.

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

5.2. RAGlib

Participant: Mohab Safey El Din [contact].

RAGLib is a Maple library for computing sampling points in semi-algebraic sets.

5.3. Epsilon

Participant: Dongming Wang [contact].

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

POMDAPI Project-Team

3. Software

3.1. LifeV

Participant: Michel Kern.

LifeV is a finite element (FE) library providing implementations of state of the art mathematical and numerical methods. It serves both as a research and production library. It has been used already in medical and industrial context to simulate fluid structure interaction and mass transport. LifeV is the joint collaboration between four institutions: École Polytechnique Fédérale de Lausanne (CMCS) in Switzerland, Politecnico di Milano (MOX) in Italy, Inria (Pomdapi) in France and Emory University (Sc. Comp) in the U.S.A.

- Version 3.1.1
- Programming language: C++
- http://www.lifev.org/

3.2. M1cg1

– Participant: J. Ch. Gilbert.
– Version: 1.2.
– Programming language: Fortran 77.
– Solves a convex quadratic optimization problem and builds a preconditioning matrix, 8 downloads in 2012.
– https://who.rocq.inria.fr/Jean-Charles.Gilbert/modulopt/optimization-routines/m1cg1/m1cg1.html.

3.3. M1qn3

– Participants: J. Ch. Gilbert, Cl. Lemaréchal.
– Version: 3.3.
– Programming language: Fortran 77.
– Solves a very large scale differentiable optimization problem, 34 downloads in 2012.
– https://who.rocq.inria.fr/Jean-Charles.Gilbert/modulopt/optimization-routines/m1qn3/m1qn3.html.

3.4. Sklml

Participants: François Clément, Pierre Weis.

Easy coarse grain parallelization.

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

- Version: 1.0+pl1
- Programming language: OCaml

3.5. SQPlab

– Participant: J. Ch. Gilbert.
– Version: 0.4.5.
– Programming language: Matlab.
– Solves a constrained differentiable optimization problem, 211 downloads in 2012.
– https://who.rocq.inria.fr/Jean-Charles.Gilbert/modulopt/optimization-routines/sqplab/sqplab.html.

PROSECCO Project-Team

5. Software

5.1. ProVerif

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

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

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

The PROVERIF verifier can prove the following properties:

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

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

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

5.2. CryptoVerif

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

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

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

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

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

5.3. Tookan

Participants: Graham Steel [correspondant], Romain Bardou.

See also the web page http://tookan.gforge.inria.fr/.

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

The first results using Tookan were published in 2010 [47] and a six-month licence was granted to Boeing to use the tool. In 2011, a contract was signed with a major UK bank. Tookan is now the subject of a CSATT transfer action resulting in the hiring of an engineer, Romain Bardou, who started on September 1st, 2011. During 2012 Bardou and Steel implemented a new version of Tookan that is intended to form the technological basis for a spin-off company to be created in 2013.

5.4. miTLS

Participants: Alfredo Pironti [correspondant], Karthikeyan Bhargavan, Cedric Fournet [Microsoft Research], Pierre-Yves Strub [IMDEA], Markulf Kohlweiss [Microsoft Research].

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

A paper describing the miTLS library is under review, and the software is being prepared for imminent release in January 2013.

5.5. WebSpi

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

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

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

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

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

5.6. Defensive JavaScript

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

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

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

RAP Project-Team (section vide)

REGAL Project-Team

5. Software

5.1. Coccinelle

Participants: Christian Clausen, Julia Lawall [correspondent], Arie Middlekoop, Gilles Muller [correspondent], Gaël Thomas, Suman Saha.

Coccinelle is a program matching and transformation engine which provides the language SmPL (Semantic Patch Language) for specifying desired matches and transformations in C code. Coccinelle was initially targeted towards performing collateral evolutions in Linux. Such evolutions comprise the changes that are needed in client code in response to evolutions in library APIs, and may include modifications such as renaming a function, adding a function argument whose value is somehow context-dependent, and reorganizing a data structure.

Beyond collateral evolutions, Coccinelle has been successfully used for finding and fixing bugs in systems code. One of the main recent results is an extensive study of bugs in Linux 2.6 that has permitted us to demonstrate that the quality of code has been improving over the last six years, even though the code size has more than doubled.

http://coccinelle.lip6.fr

5.2. SwiftCloud

Participants: Marc Shapiro [correspondent], Marek Zawirski, Annette Bieniusa, Valter Balegas.

SwiftCloud is a platform for deploying large-scale distributed applications on the edge, close to the users. Internet delays are a problem for interactive distributed applications. Truly optimal responsiveness and availability require mutable shared state replicated near the client, at the network edge. This raises serious challenges of consistency, fault tolerance, and programmability. The SwiftCloud system is designed to address these challenges. Our data model is based on client-side caching of shared mutable objects, made practical with a library of synchronisation-free, yet provably correct object types (CRDTs). The consistency model combines eventual consistency, absence of roll-backs, transactional consistency and session guarantees, but stops short of serialisability. This programming model is practically useful, yet does not require synchronisation, thus ensuring scalability. Maintaining the guarantees at a reasonable cost is especially challenging at large scale. Scalability and programmability are helped by several design decisions detailed in the paper. We validated our approach by building the SwiftCloud platform, by deploying three significant applications, and by measuring their performance in different configurations, in order to explore the benefits of replication at different locations.

SwiftCloud was supported by the ConcoRDanT ANR project (Section 7.1.4) and a Google European Doctoral Fellowship (Section 7.2.2.1).

The code is freely available on http://gforge.inria.fr/ under a BSD license.

5.3. Treedoc

Participants: Marc Shapiro [correspondent], Marek Zawirski, Nuno Preguiça.

A Commutative Replicated Data Type (CRDT) is one where all concurrent operations commute. The replicas of a CRDT converge automatically, without complex concurrency control. We designed and developed a novel CRDT design for cooperative text editing, called Treedoc. It is designed over a dense identifier space based on a binary trees. Treedoc also includes an innovative garbage collection algorithm based on tree rebalancing. In the best case, Treedoc incurs no overhead with respect to a linear text buffer. The implementation has been validated with performance measurements, based on real traces of social text editing in Wikipedia and SVN.

Work in 2010 has focused on studying large-scale garbage collection for Treedoc, and design improvements. Future work includes engineering a large-scale collaborative Wiki, and studying CRDTs more generally.

TreeDoc is supported by the Prose, Streams and ConcoRDanT ANR projects (Sections 7.1.7 , 7.1.6 and 7.1.4 respectively) and by a Google European Doctoral Fellowship (Section 7.2.2.1).

The code is freely available on http://gforge.inria.fr/ under a BSD license.

5.4. Telex

Participants: Marc Shapiro [correspondent], Lamia Benmouffok, Pierre Sutra, Pierpaolo Cincilla.

Developing write-sharing applications is challenging. Developers must deal with difficult problems such as managing distributed state, disconnection, and conflicts. Telex is an application-independent platform to ease development and to provide guarantees. Telex is guided by application-provided parameters: actions (operations) and constraints (concurrency control statements). Telex takes care of replication and persistence, drives application progress, and ensures that replicas eventually agree on a correct, common state. Telex supports partial replication, i.e., sites only receive operations they are interested in. The main data structure of Telex is a large, replicated, highly dynamic graph; we discuss the engineering trade-offs for such a graph and our solutions. Our novel agreement protocol runs Telex ensures, in the background, that replicas converge to a safe state. We conducted an experimental evaluation of the Telex based on a cooperative calendar application and on benchmarks.

The code is freely available on http://gforge.inria.fr/ under a BSD license.

5.5. Java and .Net runtimes for LLVM

Participants: Harris Bakiras, Bertil Folliot, Julia Lawall, Jean-Pierre Lozi, Gaël Thomas [correspondent], Gilles Muller, Thomas Preud homme, Koutheir Attouchi.

Many systems research projects now target managed runtime environments (MRE) because they provide better productivity and safety compared to native environments. Still, developing and optimizing an MRE is a tedious task that requires many years of development. Although MREs share some common functionalities, such as a Just In Time Compiler or a Garbage Collector, this opportunity for sharing has not been yet exploited in implementing MREs. We are working on VMKit, a first attempt to build a common substrate that eases the development and experimentation of high-level MREs and systems mechanisms. VMKit has been successfully used to build two MREs, a Java Virtual Machine and a Common Language Runtime, as well as a new system mechanism that provides better security in the context of service-oriented architectures.

VMKit is an implementation of a JVM and a CLI Virtual Machines (Microsoft .NET is an implementation of the CLI) using the LLVM compiler framework and the MMTk garbage collectors. The JVM, called J3, executes real-world applications such as Tomcat, Felix or Eclipse and the DaCapo benchmark. It uses the GNU Classpath project for the base classes. The CLI implementation, called N3, is its in early stages but can execute simple applications and the “pnetmark” benchmark. It uses the pnetlib project or Mono as its core library. The VMKit VMs compare in performance with industrial and top open-source VMs on CPU-intensive applications. VMKit is publicly available under the LLVM license.

http://vmkit2.gforge.inria.fr/

REO Project-Team

5. Software

5.1. LiFE-V library

Participants: Miguel Ángel Fernández Varela [correspondant], Jean-Frédéric Gerbeau.

LiFE-V [2] is a finite element library providing implementations of state of the art mathematical and numerical methods. It serves both as a research and production library. LiFE-V is the joint collaboration between three institutions: Ecole Polytechnique Fédérale de Lausanne (CMCS) in Switzerland, Politecnico di Milano (MOX) in Italy and Inria (REO) in France. It is a free software under LGPL license.

5.2. Mistral library

Participants: Cristóbal Bertoglio Beltran, Jean-Frédéric Gerbeau [correspondant], Vincent Martin.

Mistral is a finite element library which implements in particular fluid-structure interaction algorithms (ALE and Fictitious domain formulations), fluid surface flow (ALE) and incompressible magnetohydrodynamics equations. Mistral results from a collaboration between Inria and ENPC (CERMICS).

5.3. FELiScE

Participants: Grégory Arbia, Cesare Corrado, Miguel Ángel Fernández Varela, Justine Fouchet-Incaux, David Froger, Jean-Frédéric Gerbeau [correspondant], Damiano Lombardi, Elisa Schenone, Saverio Smaldone, Marina Vidrascu, Irène Vignon-Clementel.

FELiScE – standing for "Finite Elements for Life Sciences and Engineering" – is a new finite element code which the MACS 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 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. https://gforge.inria.fr/projects/felisce/

5.4. SHELDDON

Participant: Marina Vidrascu [correspondant].

SHELDDON (SHELls and structural Dynamics with DOmain decomposition in Nonlinear analysis) is a finite element library based on the Modulef package which contains shell elements, nonlinear procedures and PVM subroutines used in domain decomposition or coupling methods, in particular fluid-structure interaction. (https://gforge.inria.fr/projects/shelddon)

[2] http://www.lifev.org/

SECRET Project-Team (section vide)

SIERRA Project-Team

5. Software

5.1. SPAMS (SPArse Modeling Software)

Participants: Jean-Paul Chieze [correspondant], Guillaume Obozinski [correspondant].

SPAMS (SPArse Modeling Software) is an optimization toolbox for solving various sparse estimation problems: dictionary learning and matrix factorization, solving sparse decomposition prob- lems, solving structured sparse decomposition problems. It is developped by Julien Mairal (former Willow PhD student, co-advised by F. Bach and J. Ponce), with the collaboration of Francis Bach (Inria), Jean Ponce (Ecole Normale Supérieure), Guillermo Sapiro (University of Minnesota), Rodolphe Jenatton (Inria) and Guillaume Obozinski (Inria). It is coded in C++ with a Matlab interface. Recently, interfaces for R and Python have been developed by Jean-Paul Chieze (Inria). Currently 650 downloads and between 1500 and 2000 page visits per month. See http://spams-devel.gforge.inria.fr/.

5.2. SiGMa - Simple Greedy Matching: a tool for aligning large knowledge-bases

Participant: Simon Lacoste-Julien [correspondant].

SiGMa - Simple Greedy Matching: a tool for aligning large knowledge-bases

Version 1. Webpage: http://mlg.eng.cam.ac.uk/slacoste/sigma/.

The tool SiGMa (Simple Greedy Matching) is a knowledge base alignment tool implemented in Python. It takes as input two knowledge bases, each represented as a list of triples of (entity, relationship, entity), in addition to a partial alignment between the relationships from one knowledge base to the other, and gives as output an ordered list of proposed entity matches between the two knowledge base (where the order corresponds heuristically to a notion of certainty about these matches). The matching decisions are made in a greedy fashion, combining information about the relationship graph as well as a pairwise similarity scores defined between the entities. The code handles various sources of information to be used for this score, such as a similarity defined on strings, dates, and other entity properties – and gives a few options to the user.

We also provide two large-scale knowledge base alignment benchmark datasets with tens of thousands of ground truth pairs: YAGO aligned to IMDb as well as Freebase aligned to IMDb.

Participants outside of Sierra: Konstantina Palla, Alex Davies, Zoubin Ghahramani (Machine Learning Group, Department of Engineering, University of Cambridge); Gjergji Kasneci, Thore Graepel (Microsoft Research Cambridge)

See http://mlg.eng.cam.ac.uk/slacoste/sigma/.

5.3. minFunc (2012 version)

Participant: Mark Schmidt [correspondant].

minFunc is a Matlab function for unconstrained optimization of differentiable real-valued multivariate functions using line-search methods. It uses an interface very similar to the Matlab Optimization Toolbox function fminunc, and can be called as a replacement for this function. On many problems, minFunc requires fewer function evaluations to converge than fminunc (or minimize.m). Further it can optimize problems with a much larger number of variables (fminunc is restricted to several thousand variables), and uses a line search that is robust to several common function pathologies.

The default parameters of minFunc call a quasi-Newton strategy, where limited-memory BFGS updates with Shanno-Phua scaling are used in computing the step direction, and a bracketing line-search for a point satisfying the strong Wolfe conditions is used to compute the step direction. In the line search, (safeguarded) cubic interpolation is used to generate trial values, and the method switches to an Armijo back-tracking line search on iterations where the objective function enters a region where the parameters do not produce a real valued output (i.e. complex, NaN, or Inf). See http://www.di.ens.fr/~mschmidt/Software/minFunc.html.

5.4. prettyPlot

Participant: Mark Schmidt [correspondant].

The prettyPlot function is a simple wrapper to Matlab's plot function for quickly making nicer-looking plots. Here are the features: Made the default line styles bigger, and the default fonts nicer. Options are passed as a structure, instead of through plot's large number of different functions. You can pass in cell arrays to have lines of different lengths. You can pass an $n \times 3$ matrix of colors, and cell arrays of line-styles and/or markers. It will cycle through the given choices. All markers are placed on top of (all) lines, you do not have to put a marker on every data point, and you can use different spacing between markers for different lines. You can change only the upper or lower x-limit (y-limit), rather than having to specify both. There is some support for making nicer-looking error lines. See http://www.di.ens.fr/~mschmidt/Software/prettyPlot.html.

5.5. SegAnnot

Participant: Toby Hocking [correspondant].

SegAnnot: an R package for fast segmentation of annotated piecewise constant signals. Tech report and R package. Standard segmentation models for piecewise constant signals do not always agree with an expert's visual interpretation of the signal, as encoded using a set of annotations. This R package implements a dynamic programming algorithm which can be used to quickly find a segmentation model in agreement with expert annotations. Collaboration with Guillem Rigaill (Inria - AgroParisTech). See http://hal.inria.fr/hal-00759129 and http://segannot.r-forge.r-project.org/.

SISYPHE Project-Team

4. Software

4.1. The Matlab System Identification ToolBox (SITB)

Participant: Qinghua Zhang.

This development is made in collaboration with Lennart Ljung (Linköping University, Sweden), Anatoli Juditsky (Joseph Fourier University, France) and Peter Lindskog (NIRA Dynamics, Sweden).

The System Identification ToolBox (SITB) is one of the main Matlab toolboxes commercialized by The Mathworks. Inria participates in the development of its extension to the identification of nonlinear systems which is released since 2007. It includes algorithms for both black box and grey box identification of nonlinear dynamic systems. Inria is mainly responsible for the development of black box identification, with nonlinear autoregressive (NLARX) models and block-oriented (Hammerstein-Wiener) models.

4.2. Inverse Scattering for Transmission Lines (ISTL)

Participants: Michel Sorine, Qinghua Zhang.

ISTL is a software for numerical computation of the inverse scattering transform for electrical transmission lines. In addition to the inverse scattering transform, it includes a numerical simulator generating the reflection coefficients of user-specified transmission lines. With the aid of a graphical interface, the user can interactively define the distributed characteristics of a transmission line. This software is mainly for the purpose of demonstrating a numerical solution to the inverse problem of non uniform transmission lines. Its current version is limited to the case of lossless transmission lines. It is registered at Agence pour la Protection des Programmes (APP) under the number IDDN.FR.001.120003.000.S.P.2010.000.30705.

4.3. CGAO: Contrôle Glycémique Assisté par Ordinateur

Participants: Alexandre Guerrini, Michel Sorine.

This development is made in collaboration with Pierre Kalfon (Chartres Hospital) and the small business enterprise LK2.

This software CGAO developed with LK2 and Hospital Louis Pasteur (Chartres) provides efficient monitoring and control tools that will help physicians and nursing staff to avoid hyperglycaemia and hypoglycaemia episodes in Intensive Care Units. It has been used in a large clinical study, CGAO-REA. More than 3500 patients have been included in CGAO-REA. Commercialization is done by LK2. CGAO has been sold to the company Fresenius Kabi.

4.4. DYNPEAK: a user-friendly interface for the analysis of LH (Luteinizing Hormone) secretion rhythms

Participants: Frédérique Clément, Arnaud Ferré, Vincent Ladevèze, Claire Médigue, Serge Steer, Mouhamadoul-Bachir Syll, Alexandre Vidal, Qinghua Zhang.

DYNPEAK is a Webresource interface dedicated to the analysis of the pulsatile rhythm of secretion of the pituitary hormone LH, that aims at providing the final users (experimentalists and clinicians) with a simple-to-use version of the algorithm developed in [54]. DYNPEAK is developed within the PALOMA project devoted to the development of a prototype for a collaborative platform in Biomathematics.

4.5. LARY_CR: Software package for the Analysis of Cardio Vascular and Respiratory Rhythms

Participants: Claire Médigue, Serge Steer.

LARY_CR is a software package dedicated to the study of cardiovascular and respiratory rhythms [97]. It presents signal processing methods, from events detection on raw signals to the variability analysis of the resulting time series. The events detection concerns the heart beat recognition on the electrocardiogram, defining the RR time series, the maxima and minima on the arterial blood pressure defining the systolic and diastolic time series. These detections are followed by the resampling of the time series then their analyse. This analyse uses temporal and time frequency methods: Fourier Transform, spectral gain between the cardiac and blood pressure series, Smooth Pseudo Wigner_Ville Distribution, Complex DeModulation, temporal method of the cardiovascular Sequences. The objective of this software is to provide some tools for studying the autonomic nervous system, acting in particular in the baroreflex loop; its functionning is reflected by the cardiovascular variabilities and their relationships with the other physiological signals, especially the respiratory activity. Today LARY_CR is used internally, in the framework of our clinical collaborations, and the functions devoted to the rhythms analysis are now freely available through the Scilab Cardiovascular toolbox, as an atom module.

SMIS Project-Team

5. Software

5.1. Introduction

In our research domain, developing software prototypes is mandatory to validate research solutions and is an important vector for publications, demonstrations at conferences and exhibitions as well as for cooperations with industry. This prototyping task is however difficult because it requires specialized hardware platforms (e.g., new generations of smart tokens), themselves sometimes at an early stage of development.

For a decade, we have developed successive prototypes addressing different application domains, introducing different technical challenges and relying on different hardware platforms. PicoDBMS was our first attempt to design a full-fledged DBMS embedded in a smart card [42] [32]. Chip-Secured Data Access (C-SDA) embedded a reduced SQL query engine and access right controller in a secure chip and acted as an incorruptible mediator between a client and an untrusted server hosting encrypted data [37]. Chip-Secured XML Access (C-SXA) was an XML-based access rights controller embedded in a smart card [38]. Prototypes of C-SXA have been the recipient of the e-gate open 2004 Silver Award and SIMagine 2005 Gold award, two renowned international software contests. The next subsections details the two prototypes we are focusing on today.

5.2. PlugDB engine

Participants: Nicolas Anciaux [correspondent], Luc Bouganim, Philippe Pucheral, Shaoyi Yin, Yanli Guo, Lionel Le Folgoc, Alexei Troussov.

More than a stand-alone prototype, PlugDB is part of a complete architecture dedicated to a secure and ubiquitous management of personal data. PlugDB aims at providing an alternative to a systematic centralization of personal data. To meet this objective, the PlugDB architecture lies on a new kind of hardware device called Secure Portable Token (SPT). Roughly speaking, a SPT combines a secure microcontroller (similar to a smart card chip) with a large external Flash memory (Gigabyte sized). The SPT can host data on Flash (e.g., a personal folder) and safely run code embedded in the secure microcontroller. PlugDB engine is the cornerstone of this embedded code. PlugDB engine manages the database on Flash (tackling the peculiarities of NAND Flash storage), enforces the access control policy defined on this database, protects the data at rest against piracy and tampering, executes queries (tackling low RAM constraint) and ensures transaction atomicity. Part of the on-board data can be replicated on a server (then synchronized) and shared among a restricted circle of trusted parties through crypto-protected interactions. PlugDB engine has been registered at APP (Agence de Protection des Programmes) in 2009 [33] and a new version is registered each year. The underlying Flash-based indexing system has also been patented by Inria and Gemalto [43]. It has been demonstrated in a dozen of national and international events including JavaOne and SIGMOD. It is being experimented in the field to implement a secure and portable medical-social folder helping the coordination of medical care and social services provided at home to dependent people. In 2012, we have ported PlugDB-engine on a new hardware platform to 1) become completely independent from Gemalto, 2) have a plug-and-play implementation on Android, 3) serve as a basement to port it on other custom hardware implementation. We have already discussed with hardware companies located in “île de France” to produce new hardware tokens to host future versions of PlugDB-engine. Link: http://www-smis.inria.fr/_DMSP/home.php.

5.3. uFLIP Benchmark

Participants: Luc Bouganim [correspondent], Philippe Bonnet, Bjorn Jónsson, Lionel Le Folgoc.

It is amazingly easy to produce meaningless results when measuring flash devices, partly because of the peculiarity of flash memory, but primarily because their behavior is determined by layers of complex, proprietary, and undocumented software and hardware. uFLIP is a component benchmark for measuring the response time distribution of flash IO patterns, defined as the distribution of IOs in space and time. uFLIP includes a benchmarking methodology which takes into account the particular characteristics of flash devices. The source code of uFLIP, available on the web (700 downloads, 4000 distinct visitors), was registered at APP in 2009 [35]. It has been demonstrated at SIGMOD.

Link: http://www.uflip.org.

TREC Project-Team

5. Software

5.1. Gibbs' Sampler

Participants: François Baccelli, Chung Shue Chen.

The work on the self optimization of cellular networks based on Gibbs' sampler (see Section 6.1.1.5) carried out wit Chung Shue Chen (ALU) in the joint laboratory with Alcatel-Lucent, led to the development of a software prototype that was presented by L. Roullet at the Inria Alcatel-Lucent joint laboratory seminar in November 2012.

5.2. PSI2

Participant: Ana Bušić.

The work on perfect sampling (see Section 6.2.2) has been partially implemented in a software tool PSI2, in collaboration with MESCAL team [Inria Grenoble - Rhône-Alpes]; https://gforge.inria.fr/projects/psi.

WILLOW Project-Team

5. Software

5.1. Patch-based Multi-view Stereo Software (PMVS)

PMVS is a multi-view stereo software that takes a set of images and camera parameters, then reconstructs 3D structure of an object or a scene visible in the images. Only rigid structure is reconstructed. The software outputs a set of oriented points instead of a polygonal (or a mesh) model, where both the 3D coordinate and the surface normal are estimated at each oriented point. The software and its documentation are available at http://www.di.ens.fr/pmvs/. The software is distributed under GPL. A US patent corresponding to this software "Match, Expand, and Filter Technique for Multi-View Stereopsis" was issued on December 11, 2012 and assigned patent number 8,331,615.

5.2. SPArse Modeling Software (SPAMS)

SPAMS v2.3 was released as open-source software in May 2012 (v1.0 was released in September 2009 and v2.0 in November 2010). It is an optimization toolbox implementing algorithms to address various machine learning and signal processing problems involving

- Dictionary learning and matrix factorization (NMF, sparse PCA, ...)
- Solving sparse decomposition problems with LARS, coordinate descent, OMP, SOMP, proximal methods
- Solving structured sparse decomposition problems (ℓ_1/ℓ_2, ℓ_1/ℓ_∞, sparse group lasso, tree-structured regularization, structured sparsity with overlapping groups,...).

The software and its documentation are available at http://www.di.ens.fr/willow/SPAMS/.

5.3. Local dense and sparse space-time features

This is a package with Linux binaries implementing extraction of local space-time features in video. We are preparing a new release of the code implementing highly-efficient video descriptors described in Section 6.4.5 . Previous version of the package was released in January 2011. The code supports feature extraction at Harris3D points, on a dense space-time grid as well as at user-supplied space-time locations. The package is publicly available at http://www.di.ens.fr/~laptev/download/stip-2.0-linux.zip.

5.4. Automatic Mining of Visual Architectural Elements

The code on automatic mining of visual architectural elements (v4.3) described in (Doersch *et al.* SIGGRAPH [6]) has been publicly released online in December 2012 (earlier version v3.0 was released in September 2012) at http://graphics.cs.cmu.edu/projects/whatMakesParis/paris_sigg_release.tar.gz.

5.5. Automatic Alignment of Paintings

The code for automatic alignment of paintings to a 3D model (Russell et al. 2011) was made publicly available in October 2012 at http://www.di.ens.fr/willow/research/paintingalignment/index.html.

5.6. Multi-Class Image Cosegmentation

This is a package of Matlab code implementing multi-class cosegmentation (Joulin *et al.* CVPR 2012 [13] and unsupervised discriminative clustering for image co-segmenting (Joulin *et al.* CVPR 2010) and (Joulin *et al.* NIPS 2010). The aim is to segment a given set of images containing objects from the same category, simultanously and without prior information. The package was last updated in September 2012 and is available at http://www.di.ens.fr/~joulin/code/DALCIM.zip.

5.7. Convex Relaxation of Weakly Supervised Models

This is a package of Matlab code implementing a general multi-class approach to weakly supervised classification described in (Joulin and Bach ICML 2012 [12]). The goal is to avoid local minima typically occuring expectation-maximization like algorithms and to optimize a cost function based on a convex relaxation of the soft-max loss. The package was last updated in September 2012 and is available at http://www.di.ens.fr/~joulin/code/ICML12_Joulin.zip.

5.8. Non-uniform Deblurring for Shaken Images

An online demo of non-uniform deblurring for shaken images implementing the algorithm described in [8] and (Whyte et al. CPCV 2011) was made available in 2012 at http://www.di.ens.fr/willow/research/deblurring/. The demo takes as an input an image uploaded by the user, automatically estimates the blur, and outputs the deblurred image.