Algorithmics, Programming, Software and Architecture - Software - Project-Team ARENAIRE

Algorithmics, Programming, Software and Architecture
1. ARENAIRE Project-Team ................................................................. 4
2. COMPSYS Project-Team .............................................................. 8
3. LICIT Exploratory Action (section vide) .......................................... 13
4. POP ART Project-Team ............................................................... 14
5. VASY Project-Team .................................................................... 19

Applied Mathematics, Computation and Simulation
6. BIPOP Project-Team .................................................................... 22
7. MISTIS Project-Team ................................................................. 25
8. NANO-D Team ........................................................................... 28
9. NECS Project-Team ..................................................................... 29
10. OPALE Project-Team .................................................................. 31

Computational Sciences for Biology, Medicine and the Environment
11. BAMBOO Team .......................................................................... 34
12. Beagle Team ............................................................................... 37
13. DRACULA Project-Team ............................................................. 38
14. IBIS Project-Team ....................................................................... 39
15. MOISE Project-Team .................................................................. 40
16. NUMED Project-Team ................................................................ 42
17. STEEP Exploratory Action ......................................................... 43

Networks, Systems and Services, Distributed Computing
18. AMAZONES Team ....................................................................... 44
19. DNET Team ............................................................................... 47
20. GRAAL Project-Team .................................................................. 48
21. MESCAL Project-Team ............................................................... 51
22. MOAIS Project-Team .................................................................. 53
23. PLANETE Project-Team .............................................................. 55
24. RESO Project-Team ..................................................................... 61
25. SARDES Project-Team ................................................................ 62
26. SWING Team ............................................................................. 65

Perception, Cognition, Interaction
27. ARTIS Project-Team .................................................................... 66
28. E-MOTION Project-Team ............................................................ 70
29. EVASION Project-Team ............................................................. 75
30. EXMO Project-Team ................................................................... 78
31. LEAR Project-Team ..................................................................... 80
32. Morpheo Team ............................................................................ 81
33. PERCEPTION Project-Team ........................................................ 83
34. PRIMA Project-Team ................................................................... 84
35. WAM Project-Team ..................................................................... 89
5. Software

5.1. Introduction

Arénaire proposes various software and hardware realizations that are accessible from the web page http://www.ens-lyon.fr/LIP/Arenaire/Ware/. We describe below only those which progressed in 2011.

![Diagram showing relationships between Arénaire developments.]

Figure 1. Relationships between some Arénaire developments.

5.2. FloPoCo

**Participants:** Florent Dinechin [correspondant], Bogdan Pasca, Laurent-Stéphane Didier.

The purpose of the FloPoCo project is to explore the many ways in which the flexibility of the FPGA target can be exploited in the arithmetic realm. FloPoCo is a generator of operators written in C++ and outputting synthesizable VHDL automatically pipelined to an arbitrary frequency.

In 2011, FloPoCo was turned into a library which can be used as a back-end to high-level synthesis tools. An expression parser that generates a complete pipeline was also added for this context. The integer multiplier and floating-point adder were rewritten, and several new operators were added, including a floating-point power operator, and novel operators for integer and floating-point division by a constant.

Versions 2.2.0, 2.2.1, and 2.3.0 were released in 2011.

5.3. GNU MPFR

Participants: Vincent Lefèvre [correspondant], Paul Zimmermann.

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

MPFR was one of the main pieces of software developed by the old SPACES team at Loria. Since late 2006, with the departure of Vincent Lefèvre to Lyon, it has become a joint project between the Caramel (formerly SPACES then CACAO) and the Arénaire project-teams. MPFR has been a GNU package since 26 January 2009. GNU MPFR 3.0.1 was released on 4 April 2011 and GNU MPFR 3.1.0 was released on 3 October 2011.

The main improvements are the generic tests in a reduced exponent range, the possibility to include the mpfr.h header file several times while still supporting optional functions, and, for the developers, the choice of the native type for the exponent (and various corrections related to these features).

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

5.4. Exhaustive Tests for the Correct Rounding of Mathematical Functions

Participant: Vincent Lefèvre.

The search for the worst cases for the correct rounding (hardest-to-round cases) of mathematical functions (exp, log, sin, cos, etc.) in a fixed precision (mainly double precision) using Lefèvre’s algorithm is implemented by a set of utilities written in Perl, with calls to Maple/intpakX for computations on intervals and with C code generation for fast computations. It also includes a client-server system for the distribution of intervals to be tested and for tracking the status of intervals (fully tested, being tested, aborted).
These programs are run on the LIP network via Grid Engine (SGE). In June 2011, the SGE configuration was changed by the system administrator so that SIGSTOP/SIGCONT signals are sent to the jobs, allowing several users to use SGE at the same time. These signals make Maple crash (segmentation fault), and the Perl scripts needed to be improved to handle these crashes gracefully (by restarting the computations when need be, etc.). This SGE change made other problems appear, such as when the client is first stopped by SGE and is then killed by SGE (without being woke up by SGE), it cannot do its usual clean-up; workarounds were tried, but without success.

The above problems also made an inconsistency in the client-server protocol appear. The validity of the results was not affected, but the protocol had to be redesigned.

### 5.5. CGPE: Code Generation for Polynomial Evaluation

**Participants:** Christophe Mouilleron, Claude-Pierre Jeannerod.

The CGPE project, developed with Guillaume Revy (DALI research team, Université de Perpignan and LIRMM laboratory), aims at generating C codes for fast and certified polynomial evaluation, given various accuracy and architectural constraints. Several improvements for this tool, based on the addition of constraints in the first step of the generation process, were proposed in the PhD thesis of Ch. Mouilleron [12]. These improvements have been implemented, thus allowing us to reduce the whole generation time by about 50% on average.

- ACM: D.2.2 (Software libraries), G.4 (Mathematical software).
- Recommended library or software: MPFI or Gappa.
- License: CeCiLL
- Type of human computer interaction: command-line interface
- OS/Middleware: Unix
- Required library or software: Xerces-C++ XML Parser library and MPFR
- Programming Language: C++
- Status: beta
- Documentation: available in html format on [URL](http://cgpe.gforge.inria.fr/)

### 5.6. FLIP: Floating-point Library for Integer Processors

**Participants:** Claude-Pierre Jeannerod, Jingyan Jourdan-Lu.

FLIP is a C library for the efficient software support of binary32 IEEE 754-2008 floating-point arithmetic on processors without floating-point hardware units, such as VLIW or DSP processors for embedded applications. The current target architecture is the VLIW ST200 family from STMicroelectronics (especially the ST231 cores). This year, we have mostly worked on improving the design and implementation of the following operators with correct rounding “to nearest even”: DP2 (fused dot product in dimension two) and sum of two squares. The impact of the DP2 operator has been evaluated on the UTDSP benchmark, and on some kernels speed-ups of 1.46 have been observed. On the other hand, specializing DP2 to a sum of squares brings a speed-up of 2.

[URL](http://flip.gforge.inria.fr/)

- ACM: D.2.2 (Software libraries), G.4 (Mathematical software)
- AMS: 26-04 Real Numbers, Explicit machine computation and programs.
- APP: IDDN.FR.001.230018.S.A.2010.000.10000
- License: CeCILL v2
- Type of human computer interaction: C library callable, from any C program.
- OS/Middleware: any, as long as a C compiler is available.
- Required library or software: none.
- Programming language: C
5.7. SIPE: Small Integer Plus Exponent

Participant: Vincent Lefèvre.

SIPE (Small Integer Plus Exponent) is a C header file providing a fast floating-point arithmetic with correct rounding to the nearest in very small precision. Implemented operations are the addition, subtraction, multiplication, FMA, and minimum/maximum/comparison functions (of the signed numbers or in magnitude). SIPE has been written for exhaustive tests of simple algorithms in small precision in order to prove results or find conjectures (which could then be proved). In 2011, a research report was written about SIPE [62], including documentation and proof of the implementation; some bugs were fixed at the same time.

- ACM: D.2.2 (Software libraries), G.4 (Mathematical software).
- AMS: 26-04 Real Numbers, Explicit machine computation and programs.
- License: LGPL version 2.1 or later.
- Type of human computer interaction: C header file.
- OS/Middleware: any OS.
- Required library or software: GCC compiler.
- Programming language: C.
- Documentation: Research report RR-7832. [62]
- URL: http://www.vinc17.net/software/sipe.h
COMPSYS Project-Team

5. Software

5.1. Introduction

This section lists and briefly describes the software developments conducted within Compsys. Most are tools that we extend and maintain over the years. They now concern two activities only: a) the development of tools linked to polyhedra and loop/array transformations, b) the development of algorithms within the back-end compiler of STMicroelectronics.

Many tools based on the polyhedral representation of codes with nested loops are now available. They have been developed and maintained over the years by different teams, after the introduction of Paul Feautrier’s Pip, a tool for parametric integer linear programming. This “polytope model” view of codes is now widely accepted: it used by Inria projects-teams Cairn and Alchemy/Parkas, PIPS at École des Mines de Paris, Surf from Stanford University, Compaan at Berkeley and Leiden, PiCo from the HP Labs (continued as PicoExpress by Synfora and now Synopsis), the DTSE methodology at Imec, Sadayappan’s group at Ohio State University, Rajopadhye’s group at Colorado State’s University, etc. More recently, several compiler groups have shown their interest in polyhedral methods, e.g., the GCC group, IBM, and Reservoir Labs, a company that develops a compiler fully-based on the polytope model and on the techniques that we (the French community) introduced for loop and array transformations. Polyhedra are also used in test and certification projects (Verimag, Lande, Vertecs). Now that these techniques are well-established and disseminated in and by other groups, we prefer to focus on the development of new techniques and tools, which are described here.

The other activity concerns the developments within the compiler of STMicroelectronics. These are not stand-alone tools, which could be used externally, but algorithms and data structures implemented inside the LAO back-end compiler, year after year, with the help of STMicroelectronics colleagues. As these are also important developments, it is worth mentioning them in this section. They are also completed by important efforts for integration and evaluation within the complete STMicroelectronics toolchain. They concern exact methods (ILP-based), algorithms for aggressive optimizations, techniques for just-in-time compilation, and for improving the design of the compiler.

5.2. Pip

Participants: Cédric Bastoul [MCF, IUT d’Orsay], Paul Feautrier.

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

5.3. Syntol

Participants: Hadda Cherroun [Former PhD student in Compsys], Paul Feautrier.
Syntol is a modular process network scheduler. The source language is C augmented with specific constructs for representing communicating regular process (CRP) systems. The present version features a syntax analyzer, a semantic analyzer to identify DO loops in C code, a dependence computer, a modular scheduler, and interfaces for CLooG (loop generator developed by C. Bastoul) and Cl@k (see Sections 5.4 and 5.6). The dependence computer now handles casts, records, and the modulo operator in subscripts and conditional expressions. The latest developments are, firstly, a new code generator, and secondly, several experimental tools for the construction of bounded parallelism programs.

- The new code generator, based on the ideas of Boulet and Feautrier [31], generates a counter automaton that can be presented as a C program, as a rudimentary VHDL program at the RTL level, as an automaton in the Aspic input format, or as a drawing specification for the DOT tool.
- Hardware synthesis can only be applied to bounded parallelism programs. Our present aim is to construct threads with the objective of minimizing communications and simplifying synchronization. The distribution of operations among threads is specified using a placement function, which is found using techniques of linear algebra and combinatorial optimization.

5.4. Cl@k
Participants: Christophe Alias, Fabrice Baray [Mentor, Former post-doc in Compsys], Alain Darte.

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

Its application to array contraction has been implemented by Christophe Alias in a tool called Bee (see Section 5.6). Bee uses Rose as a parser, analyzes the lifetimes of the elements of the arrays to be compressed, and builds the necessary input for Cl@k, i.e., the 0-symmetric polytope of conflicting differences. Then, Bee computes the array contraction mapping from the lattice provided by Cl@k and generates the final program with contracted arrays. See previous reports for more details on the underlying theory. Cl@k can be viewed as a complement to the Polylib suite, enabling yet another kind of optimizations on polyhedra. Initially, Bee was the complement of Cl@k in terms of its application to memory reuse. Now, Bee is a stand-alone tool that contains more and more features for program analysis and loop transformations.

5.5. PoCo
Participant: Christophe Alias.

PoCo is a polyhedral compilation framework providing many features to quickly prototype program analysis and optimizations in the polyhedral model. Essentially, PoCo provides:

- C front-end extracting the polyhedral representation of the input program. The parser itself is based on EDG (via ROSE), an industrial C/C++ parser from Edison group used in Intel compilers.
- Extended language of pragmas to feed the source code with compilation directives (a schedule, for example).
- Symbolic layer on polyhedral libraries POLYLIB (set operations on polyhedra) and PIPLIB (parameterized ILP). This feature simplifies drastically the developer task.
- Dependence analysis (polyhedral dependence graph, array dataflow analysis), array region analysis, array liveness analysis.
- C and VHDL code generation based on the ideas of P. Boulet and P. Feautrier [31].
The array dataflow analysis (ADA) of PoCo has been extended to a FADA (Fuzzy ADA) by M. Belaoucha, former PhD student at Université de Versailles. FADALib is available at http://www.prism.uvsq.fr/~bem/fadalib/.

PoCo has been developed by Christophe Alias. It represents more than 19000 lines of C++ code. The tools Bee, Chuba, and RanK presented thereafter make an extensive use of PoCo abstractions.

5.6. Bee

Participants: Christophe Alias, Alain Darte.

Bee is a source-to-source optimizer that contracts the temporary arrays of a program under scheduling constraints. Bee bridges the gap between the mathematical optimization framework described in [32] and implemented in Cl@k (Section 5.4), and effective source-to-source array contraction. Bee applies a precise lifetime analysis for arrays to build the mathematical input of Cl@k. Then, Bee derives the array allocations from the basis found by Cl@k and generates the C code accordingly. Bee is – to our knowledge – the only complete array contraction tool.

Bee is sensitive to the program schedule. This latter feature enlarges the application field of array contraction to parallel programs. For instance, it is possible to mark a loop to be software-pipelined (with an affine schedule) and to let Bee find an optimized array contraction. But the most important application is the ability to optimize communicating regular processes (CRP). Given a schedule for every process, Bee can compute an optimized size for the channels, together with their access functions (the corresponding allocations). We currently use this feature in source-to-source transformations for high-level synthesis (see Section 3.3).

- Bee was made available to STMICROELECTRONICS as a binary.
- Bee will be transferred to the (incubated) start-up Zettice, initiated by Alexandru Plesco.
- Bee is used as an external tool by the compiler GECOS developed in the Cairn team at IRISA.

Bee has been implemented by Christophe Alias, using the compiler infrastructure PoCo. It represents more than 2400 lines of C++ code.

5.7. Chuba

Participants: Christophe Alias, Alain Darte, Alexandru Plesco.

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

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

Chuba has been implemented by Christophe Alias, using the compiler infrastructure PoCo. It represents more than 900 lines of C++. The reduced size of Chuba is mainly due to the high-level abstractions provided by PoCo.

5.8. C2fsm

Participant: Paul Feautrier.
C2fsm is a general tool that converts an arbitrary C program into a counter automaton. This tool reuses the parser and pre-processor of Syntol, which has been greatly extended to handle while and do while loops, goto, break, and continue statements. C2fsm reuses also part of the code generator of Syntol and has several output formats, including FAST (the input format of Aspic), a rudimentary VHDL generator, and a DOT generator which draws the output automaton. C2fsm is also able to do elementary transformations on the automaton, such as eliminating useless states, transitions and variables, simplifying guards, or selecting cut-points, i.e., program points on loops that can be used by RanK to prove program termination.

5.9. RanK

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

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

The input of RanK is an integer automaton, computed by C2fsm (see Section 5.8), representing the control structure of the program to check. RanK uses the Aspic tool, developed by Laure Gonnord during her PhD thesis, to compute automaton invariants. RanK has been used to discover successfully the worst-case time complexity of many benchmarks programs of the community. It uses the libraries Piplib and Polylib.

RanK has been implemented by Christophe Alias, using the compiler infrastructure PoCo. It represents more than 3000 lines of C++.

5.10. Simplifiers

Participant: Paul Feautrier.

The aim of the simple library is to simplify boolean formulas on affine inequalities. It works by detecting redundant inequalities in the representation of the subject formula as an ordered binary decision diagram. It uses PIP for testing the feasibility – or unfeasibility – of a conjunction of affine inequality.

The library is written in Java and is presented as a collection of class files. For experimentation, several front-ends have been written They differ mainly in their input syntax, among which are a C like syntax, the Mathematica and SMTLib syntaxes, and an ad hoc Quast syntax.

5.11. LAO Developments in Aggressive Compilation

Participants: Benoît Boissinot, Florent Bouchez, Florian Brandner, Quentin Colombet, Alain Darte, Benoît Dupont de Dinechin [Kalray], Christophe Guillon [STMicroelectronics], Sebastian Hack [Former post-doc in Compsys], Fabrice Rastello, Cédric Vincent [Former student in Compsys].

Our aggressive optimization techniques are all implemented in stand-alone experimental tools (as for example for register coalescing algorithms) or within LAO, the back-end compiler of STMicroelectronics, or both. They concern SSA construction and destruction, instruction-cache optimizations, register allocation. Here, we report only our more recent activities, which concern register allocation.
Our developments on register allocation within the STMicroelectronics compiler started when Cédric Vincent (bachelor degree, under Alain Darte supervision) developed a complete register allocator in LAO, the assembly-code optimizer of STMicroelectronics. This was the first time a complete implementation was done with success, outside the MCDT (now CEC) team, in their optimizer. Since then, new developments are constantly done, in particular by Florent Bouchez, advised by Alain Darte and Fabrice Rastello, as part of his master internship and PhD thesis. In 2009, Quentin Colombet started to develop and integrate into the main trunk of LAO a full implementation of a two-phases register allocation. This implementation now includes two different decoupled spilling phases, the first one as described in Sebastian Hack’s PhD thesis and a new ILP-based solution (see Section 6.2). It also includes an up-to-date graph-based register coalescing. Finally, since all these optimizations take place under SSA form, it includes also a mechanism for going out of colored-SSA (register-allocated SSA) form that can handle critical edges and does further optimizations (see for example Section 6.3).

5.12. LAO Developments in JIT Compilation

Participants: Benoît Boissinot, Florian Brandner, Alain Darte, Benoît Dupont de Dinechin [Kalray], Christophe Guillou [STMicroelectronics], Fabrice Rastello.

The other side of our work in the STMicroelectronics compiler LAO has been to adapt the compiler to make it more suitable for JIT compilation. This means lowering the time and space complexity of several algorithms. In particular we implemented our fast out-of-SSA translation method, and we programmed and tested various ways to compute the liveness information as described in Section 6.6. Recent efforts (see Section 6.4) also focused on developing a tree-scan register allocator for the JIT part of the compiler, in particular a JIT conservative coalescing. The technique is to bias the tree-scan coalescing, taking into account register constraints, with the result of a JIT aggressive coalescing.

5.13. Low-Level Exchange Format (TireX) and Minimalist Intermediate Representation (MinIR)

Participants: Christophe Guillou [STMicroelectronics], Fabrice Rastello, Benoît Dupont de Dinechin [Kalray].

Most compilers define their own intermediate representation (IR) to be able to work on a program. Sometimes, they even use a different representation for each representation level, from source code parsing to the final object code generation. MinIR (Minimalist Intermediate Representation) is a new intermediate representation, designed to ease the interconnection of compilers, static analyzers, code generators, and other tools. In addition to the specification of MinIR, generic core tools have been developed to offer a basic toolkit and to help the connection of client tools. MinIR generators exist for several compilers, and different analyzers are developed as a testbed to rapidly prototype different static analyses over SSA code. This new common format enables the comparison of the code generator of several production compilers, and simplifies the connection of external tools to existing compilers.

MinIR has been extended into TireX, a Textual Intermediate Representation for EXchanging target-level information between compiler optimizers and whole or parts of code generators (aka compiler back-end). The first motivation for this intermediate representation is to factor target-specific compiler optimizations into a single component, in case several compilers need to be maintained for a particular target (e.g., operating system compiler and application code compiler). Another motivation is to reduce the run-time cost of JIT compilation and of mixed mode execution, since the program to compile is already in a representation lowered to the level of the target processor. Besides the lowering at the target level, the extensions of MinIR include the program data stream and loop scoped information. TireX is currently produced by the Open64/Path64 and the LLVM compilers, with a GCC producer under work. It is consumed by the LAO code generator.

Detailed information, generic core tools, and LLVM IR based generator for MinIR are available at http://www.assembla.com/spaces/minir-dev/wiki. Open64/Path64 emitter for TireX and its LAO back-end are available at https://compilation.ens-lyon.fr/. MinIR was presented at WIR’11 [17].
LICIT Exploratory Action (section vide)
5. Software

5.1. NBac

Participant: Bertrand Jeannet.

NBAC (Numerical and Boolean Automaton Checker) is a verification/slicing tool for reactive systems containing combination of Boolean and numerical variables, and continuously interacting with an external environment. NBAC can also handle the same class of hybrid systems as the HyTech tool. It aims at handling efficiently systems combining a non-trivial numerical behaviour with a complex logical (Boolean) behaviour.

NBAC is connected to two input languages: the synchronous dataflow language LUSTRE, and a symbolic automaton-based language, AUTO/GRAPH, where a system is defined by a set of symbolic hybrid automata communicating via valued channels. It can perform reachability analysis, co-reachability analysis, and combination of the above analyses. The result of an analysis is either a verdict to a verification problem, or a set of states together with a necessary condition to stay in this set during an execution. NBAC is founded on the theory of abstract interpretation.

It has been used for verification and debugging of LUSTRE programs. It is connected to the LUSTRE toolset. It has also been used for controller synthesis of infinite-state systems. The fact that the analyses are approximated results simply in the obtention of a possibly non-optimal controller. In the context of conformance testing of reactive systems, it is used by the test generator STG for selecting test cases.

5.2. Prometheus

Participant: Gregor Goessler.

The BIP component model (Behavior, Interaction model, Priority) has been designed to support the construction of heterogeneous embedded systems involving different models of computation, communication, and execution, at different levels of abstraction. By separating the notions of behavior, interaction model, and execution model, it enables both heterogeneous modeling, and separation of concerns.

The verification and design tool Prometheus implements the BIP component framework. Prometheus is regularly updated to implement new developments in the framework and the analysis algorithms. It has allowed us to carry out several complex case studies from the system-on-chip and bioinformatics domains.

5.3. Implementations of Synchronous Programs

Participant: Alain Girault.

---

13 http://pop-art.inrialpes.fr/people/bjeannet/nbac/
14 http://www-verimag.imag.fr/Lustre-V6.html
5.3.1. Fault Tolerance

We have been cooperating for several years with the INRIA team AOSTE (INRIA Sophia-Antipolis and Rocquencourt) on the topic of fault tolerance and reliability of safety critical embedded systems. In particular, we have implemented several new heuristics for fault tolerance and reliability within their software SYNDEx. Our first scheduling heuristic produces static multiprocessor schedules tolerant to a specified number of processor and communication link failures [64]. The basic principles upon which we rely to make the schedules fault tolerant is, on the one hand, the active replication of the operations [65], and on the other hand, the active replication of communications for point-to-point communication links, or their passive replication coupled with data fragmentation for multi-point communication media (i.e., buses) [66]. Our second scheduling heuristic is multi-criteria: it produces a static schedule multiprocessor schedule such that the reliability is maximized, the power consumption is minimized, and the execution time is minimized [3][17]. Our results on fault tolerance are summarized in a web page [17].

5.4. Apron and BddApron Libraries

Participant: Bertrand Jeannet.

5.4.1. Principles

The APRON library is dedicated to the static analysis of the numerical variables of a program by abstract interpretation [51]. Many abstract domains have been designed and implemented for analysing the possible values of numerical variables during the execution of a program (see Figure 1). However, their API diverge largely (datatypes, signatures, ...), and this does not ease their diffusion and experimental comparison w.r.t. efficiency and precision aspects.

The APRON library aims to provide:

- a uniform API for existing numerical abstract domains;
- a higher-level interface to the client tools, by factorizing functionalities that are largely independent of abstract domains.

From an abstract domain designer point of view, the benefits of the APRON library are:

- the ability to focus on core, low-level functionalities;
- the help of generic services adding higher-level services for free.

For the client static analysis community, the benefits are a unified, higher-level interface, which allows experimenting, comparing, and combining abstract domains.

In 2011, the Taylor1plus domain [62], which is the underlying abstract domain of the tool FLUCTUAT [58] has been improved. Glue code has also been added to enable the connection of an abstract domain implemented in OCaml to the APRON infrastructure written in C (this requires callbacks from C to OCaml that are safe w.r.t. garbage collection). This will enable the integration in APRON of the MaxPlus polyhedra library written by X. Allamigeon [38] in the context of the ANR ASOPT project.

The BDDAPRON library aims at a similar goal, by adding finite-types variables and expressions to the concrete semantics of APRON domains. It is built upon the APRON library and provides abstract domains for the combination of finite-type variables (Booleans, enumerated types, bitvectors) and numerical variables (integers, rationals, floating-point numbers). It first allows to manipulate expressions that freely mix, using BDDs and MTBDDs, finite-type and numerical APRON expressions and conditions. It then provides abstract domains that combines BDDs and APRON abstract values for representing invariants holding on both finite-type variables and numerical variables.

---

16 http://www-rocinria.fr/syndex
17 http://pop-art.inrialpes.fr/~girault/Projets/FT
18 http://apron.cri.ensmp.fr/library/
5.4.2. Implementation and Distribution

The APRON library (Fig. 2) is written in ANSI C, with an object-oriented and thread-safe design. Both multi-precision and floating-point numbers are supported. A wrapper for the OCAML language is available, and a C++ wrapper is on the way. It has been distributed since June 2006 under the LGPL license and available at http://apron.cri.ensmp.fr. Its development has still progressed much since. There are already many external users (ProVal/Démons, LRI Orsay, France — CEA-LIST, Saclay, France — Analysis of Computer Systems Group, New-York University, USA — Sierum software analysis platform, Kansas State University, USA — NEC Labs, Princeton, USA — EADS CCR, Paris, France — IRIT, Toulouse, France) and is currently packaged as a REDHAT and DEBIAN package.

The BDDAPRON library is written in OCAML, using polymorphism features of OCAML to make it generic. It is also thread-safe. It provides two different implementations of the same domain, each one presenting pros and cons depending on the application. It is currently used by the CONCURINTERPROC interprocedural and concurrent program analyzer.

5.5. Prototypes

5.5.1. Logical Causality

Participants: Lacramioara Astefanoaei, Gregor Goessler [contact person].

We have developed LoCA, a new prototype tool written in Scala that implements the analysis of logical causality described in 6.6.2. LoCA currently supports causality analysis in BIP. The core analysis engine is implemented as an abstract class, such that support for other models of computation (MOC) can be added by instantiating the class with the basic operations of the MOC.

5.5.2. Automatic Controller Generation

Participants: Emil Dumitrescu, Alain Girault [contact person].

We have developed a software tool chain to allow the specification of models, the controller synthesis, and the execution or simulation of the results. It is based on existing synchronous tools, and thus consists primarily in the use and integration of SIGALI20 and Mode Automata21. It is the result of a collaboration with Eric Rutten from the SARDIES team.

20 http://www.irisa.fr/vertecs/Logiciels/sigali.html
21 http://www-verimag.imag.fr
Useful component templates and relevant properties can be materialized, on one hand by libraries of task models, and, on the other hand, by properties and synthesis objectives.

5.5.3. Rapture

**Participant:** Bertrand Jeannet.

Rapture is a verification tool that was developed jointly by BRICS (Denmark) and INRIA in years 2000–2002. The tool is designed to verify reachability properties on Markov Decision Processes (MDP), also known as Probabilistic Transition Systems. This model can be viewed both as an extension to classical (finite-state) transition systems extended with probability distributions on successor states, or as an extension of Markov Chains with non-determinism. We have developed a simple automata language that allows the designer to describe a set of processes communicating over a set of channels à la CSP. Processes can also manipulate local and global variables of finite type. Probabilistic reachability properties are specified by defining two sets of initial and final states together with a probability bound. The originality of the tool is to provide two reduction techniques that limit the state space explosion problem: automatic abstraction and refinement algorithms, and the so-called essential states reduction.

5.5.4. The Interproc family of static analyzers

**Participants:** Bertrand Jeannet [contact person], Pascal Sotin.

These analyzers and libraries are of general use for people working in the static analysis and abstract interpretation community, and serve as an experimental platform for the ANR project ASOPT (see § 8.1.2).

A generic fix-point engine written in OCAML. It allows the user to solve systems of fix-point equations on a lattice, using a parameterized strategy for the iteration order and the application of widening. It also implements recent techniques for improving the precision of analysis by alternating

---

22 [http://pop-art.inrialpes.fr/people/bjeannet/rapture/rapture.html](http://pop-art.inrialpes.fr/people/bjeannet/rapture/rapture.html)
post-fixpoint computation with widening and descending iterations in a sound way [ 70 ].

a simple interprocedural static analyzer that infers properties on the numerical variables of programs in a toy language. It is aimed at demonstrating the use of the previous library and the above-described APRON library, and more generally at disseminating the knowledge in abstract interpretation. It is also deployed through a web-interface 25. It is used as the experimental platform of the ASOPT ANR project.

\texttt{FixpointInterprocConcurInterproc} extends Interproc with concurrency, for the analysis of multithreaded programs interacting via shared global variables. It is also deployed through a web-interface 26.

\texttt{PInterproc} extends Interproc with pointers to local variables. It is also deployed through a web-interface 27.

5.5.5. \texttt{Heptagon/BZR}

\textbf{Participant:} Gwenaël Delaval.

\texttt{Heptagon} is a dataflow synchronous language, inspired from \texttt{Lucid Synchrone} 28. Its compiler is meant to be simple and modular, allowing this language to be a good support for the prototyping of compilation methods of synchronous languages. It is developped within the SYNCHRONICSINRIA large-scale action.

\texttt{Heptagon} has been used to built \texttt{BZR} 29, which is an extension of the former with contracts constructs. These contracts allow to express dynamic temporal properties on the inputs and outputs of \texttt{Heptagon} node. These properties are then enforced, within the compilation of a BZR program, by discrete controller synthesis, using the \texttt{Sigali} tool 30. The synthesized controller is itself generated in \texttt{Heptagon}, allowing its analysis and compilation towards different target languages (C, \texttt{Java}, VHDL).

\footnotesize

25 http://pop-art.inrialpes.fr/interproc/interprocweb.cgi
26 http://pop-art.inrialpes.fr/interproc/concurinterprocweb.cgi
27 http://pop-art.inrialpes.fr/interproc/pinterprocweb.cgi
28 http://www.di.ens.fr/~pouzet/lucid-synchrone
29 http://bzr.inria.fr
30 http://www.irisa.fr/vertecs/Logiciels/sigali.html
5. Software

5.1. The CADP Toolbox

Participants: Iker Bellicot, Hubert Garavel [contact person], Yann Genevois, Rémi Hérilier, Frédéric Lang, Radu Mateescu, Christine McKinty, Wendelin Serwe, Damien Thivolle.

We maintain and enhance CADP (Construction and Analysis of Distributed Processes – formerly known as CÆSAR/ALDÉBARAN Development Package) [9], a toolbox for protocols and distributed systems engineering (see http://cadp.inria.fr). In this toolbox, we develop and maintain the following tools:

- **CÆSAR.ADT** [3] is a compiler that translates LOTOS abstract data types into C types and C functions. The translation involves pattern-matching compiling techniques and automatic recognition of usual types (integers, enumerations, tuples, etc.), which are implemented optimally.
- **CÆSAR** [11] is a compiler that translates LOTOS processes into either C code (for rapid prototyping and testing purposes) or finite graphs (for verification purpose). The translation is done using several intermediate steps, among which the construction of a Petri net extended with typed variables, data handling features, and atomic transitions.
- **OPEN/CÆSAR** [4] is a generic software environment for developing tools that explore graphs on the fly (for instance, simulation, verification, and test generation tools). Such tools can be developed independently of any particular high level language. In this respect, OPEN/CÆSAR plays a central role in CADP by connecting language-oriented tools with model-oriented tools. OPEN/CÆSAR consists of a set of 16 code libraries with their programming interfaces, such as:
  - **CAESAR_GRAPH**, which provides the programming interface for graph exploration,
  - **CAESAR_HASH**, which contains several hash functions,
  - **CAESAR_SOLVE**, which resolves boolean equation systems on the fly,
  - **CAESAR_STACK**, which implements stacks for depth-first search exploration, and
  - **CAESAR_TABLE**, which handles tables of states, transitions, labels, etc.

A number of tools have been developed within the OPEN/CÆSAR environment, among which:

- **BISIMULATOR**, which checks bisimulation equivalences and preorders,
- **CUNCTATOR**, which performs on-the-fly steady-state simulation of continuous-time Markov chains,
- **DETERMINATOR**, which eliminates stochastic nondeterminism in normal, probabilistic, or stochastic systems,
- **DISTRIBUTOR**, which generates the graph of reachable states using several machines,
- **EVALUATOR**, which evaluates regular alternation-free μ-calculus formulas,
- **EXECUTOR**, which performs random execution,
- **EXHIBITOR**, which searches for execution sequences matching a given regular expression,
- **GENERATOR**, which constructs the graph of reachable states,
- **PROJECTOR**, which computes abstractions of communicating systems,
- **REDUCTOR**, which constructs and minimizes the graph of reachable states modulo various equivalence relations,
- **SIMULATOR**, **XSIMULATOR**, and **OCIS**, which allow interactive simulation, and
TERMINATOR, which searches for deadlock states.

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

  For instance, one can define recursive functions on sets of states, which allow to specify in XTL evaluation and diagnostic generation fixed point algorithms for usual temporal logics (such as HML [60], CTL [53], ACTL [55], etc.).

- The connection between explicit models (such as BCG graphs) and implicit models (explored on the fly) is ensured by OPEN/CAESAR-compliant compilers, e.g.:
  - **BCG_OPEN**, for models represented as BCG graphs,
  - **CAESAR.OPEN**, for models expressed as LOTOS descriptions,
  - **EXP.OPEN**, for models expressed as communicating automata,
  - **FSP.OPEN**, for models expressed as FSP [66] descriptions,
  - **LNT.OPEN**, for models expressed as LOTOS NT descriptions, and
  - **SEQ.OPEN**, for models represented as sets of execution trace.

The CADP toolbox also includes TGV (*Test Generation based on Verification*), developed by the VERIMAG laboratory (Grenoble) and the VERTECS project team at INRIA Rennes.

The CADP tools are well-integrated and can be accessed easily using either the EUCALYPTUS graphical interface or the SVL [6] scripting language. Both EUCALYPTUS and SVL provide users with an easy and uniform access to the CADP tools by performing file format conversions automatically whenever needed and by supplying appropriate command-line options as the tools are invoked.

### 5.2. The TRAIAN Compiler

**Participants:** Hubert Garavel [contact person], Frédéric Lang.
We develop a compiler named Traian for translating descriptions written in the LOTOS NT language (see § 3.2) into C programs, which will be used for simulation, rapid prototyping, verification, and testing.

The current version of Traian performs lexical analysis, syntactic analysis, abstract syntax tree construction, static semantics analysis, and C code generation for LOTOS NT types and functions.

Although this version of Traian is still incomplete (it does not handle LOTOS NT processes), it already has useful applications in compiler construction [8]. The recent compilers developed by the VASY project team — including AAL, EvaluatoR 4.0 (see § 6.1.6), Exp.open 2.0 (see § 6.1.4), Lnt2Lotos (see § 6.2.2), Ntif (see § 3.2), Pic2Lnt (see § 6.2.3), and SvL (see § 6.1.4) — all contain a large amount of LOTOS NT code, which is then translated into C code by Traian.

Our approach consists in using the Syntax tool (developed at INRIA Rocquencourt) for lexical and syntactic analysis together with LOTOS NT for semantical aspects, in particular the definition, construction, and traversal of abstract trees. Some involved parts of the compiler can also be written directly in C if necessary. The combined use of Syntax, LOTOS NT, and Traian proves to be satisfactory, in terms of both the rapidity of development and the quality of the resulting compilers.

The Traian compiler can be freely downloaded from the VASYWeb site (see http://vasy.inria.fr/traian).
BIPOP Project-Team

5. Software

5.1. Nonsmooth dynamics: Siconos

Participants: Vincent Acary, Maurice Bremond, Olivier Bonnefon.

In the framework of the European project Siconos, Bipop was the leader of the Work Package 2 (WP2), dedicated to the numerical methods and the software design for nonsmooth dynamical systems. The aim of this work is to provide a common platform for the simulation, modeling, analysis and control of abstract nonsmooth dynamical systems. Besides usual quality attributes for scientific computing software, we want to provide a common framework for various scientific fields, to be able to rely on the existing developments (numerical algorithms, description and modeling software), to support exchanges and comparisons of methods, to disseminate the know-how to other fields of research and industry, and to take into account the diversity of users (end-users, algorithm developers, framework builders) in building expert interfaces in Python and end-user front-end through Scilab.

After the requirement elicitation phase, the Siconos Software project has been divided into 5 work packages which are identified to software products:

1. SICONOS/NUMERICS This library contains a set of numerical algorithms, already well identified, to solve non smooth dynamical systems. This library is written in low-level languages (C,F77) in order to ensure numerical efficiency and the use of standard libraries (Blas, Lapack, ...)

2. SICONOS/KERNEL This module is an object-oriented structure (C++) for the modeling and the simulation of abstract dynamical systems. It provides the users with a set of classes to describe their nonsmooth dynamical system (dynamical systems, interactions, nonsmooth laws, ...) and to perform a numerical time integration and solving.

3. SICONOS/FRONT-END. This module is mainly an auto-generated wrapper in Python which provides a user-friendly interface to the Siconos libraries. A scilab interface is also provided in the Front-End module.

4. SICONOS/CONTROL This part is devoted to the implementation of control strategies of non smooth dynamical systems.

5. SICONOS/MULTIBODY. This part is dedicated to the modeling and the simulation of multi-body systems with 3D contacts, impacts and Coulomb’s friction. It uses the Siconos/Kernel as simulation engine but relies on a industrial CAD library (OpenCascade and pythonOCC) to deal with complex body geometries and to compute the contact locations and distances.

Further informations may be found at http://siconos.gforge.inria.fr/

5.2. Humanoid motion analysis and simulation


The HuMANs toolbox offers tools for the modelling, control and analysis of humanoid motion, be it of a robot or a human. It is a C/C++/Scilab/Maple-based set of integrated tools for the generation of dynamical models of articulated bodies with unilateral contact and friction, their simulation with an event-driven integration scheme, their 3D visualization, the computation of stability measures, optimal positions and trajectories, the generation of control laws and observers, the reconstruction of movements from different sensing systems.

5.3. AMELIF

Participants: Pierre-Brice Wieber, François Keith.
The AMELIF framework is an integrative framework that proposes an API for the representation and simulation of virtual scenes including articulated bodies. AMELIF was devised to realize interactive scenario studies with haptic feedback while providing an interface enabling fast and general prototyping of humanoids (avatars or robots). It is entirely developed in C++ and is cross-platform. The framework is articulated around a core library, upon which several modules have been developed for collision detection, dynamic simulation (contact handling in a time stepping scheme), 3D rendering, haptic interaction, posture generation. This framework is developed mostly at the CNRS/AIST UMI JRL, but we started using it in the Bipop team and therefore started contributing actively to its development.

5.4. Optimization

Participant: Claude Lemaréchal.

Essentially two possibilities exist to distribute our optimization software: library programs (say Modulopt codes), communicated either freely or not, depending on what they are used for, and on the other hand specific software, developed for a given application.

The following optimization codes have been developed in the framework of the former Promath project. They are generally available at http://www-rocq.inria.fr/~gilbert/modulopt/ ; M1QN3 is also distributed under GPL.

5.4.1. Code M1QN3

Optimization without constraints for problems with many variables \((n \geq 10^3, \text{ has been used for } n = 10^6)\). Technically, uses a limited-memory BFGS algorithm with Wolfe’s line-search (see Chap. 4 of [3] for the terminology).

5.4.2. Code M2QN1

Optimization with simple bound-constraints for (small) problems: \(D\) is a parallelotope in \(\mathbb{R}^n\). Uses BFGS with Wolfe’s line-search and active-set strategy.

5.4.3. Code N1CV2

Minimization without constraints of a convex nonsmooth function by a proximal bundle method (Chap. XV of [10], Chap. 9 of [3]).

5.4.4. Modulopt

In addition to codes such as above, the Modulopt library contains application problems, synthetic or from the real world. It is a field for experimentation, functioning both ways: to assess a new algorithm on a set of test-problems, or to select among several codes one best suited to a given problem.

5.5. Simulation of fibrous materials

Participants: Florence Bertails-Descoubes, Gilles Daviet.

The goal of the MECHE ADT, which started in September 2009 and was completed in fall 2011, was to develop a software for simulating the dynamics of assemblies of thin rods (such as hair), subject to contact and friction. This software combines a panel of well-accepted models for rods (ranging from reduced coordinates to maximal coordinates models, and including models recently developed by some members of the group) with classical as well as innovative schemes for solving the problem of frictional contact (incorporating the most recent results of the group, as well as the new contact solver we published in [21]). The aim of this software is twofold: first, we were able to compare and analyze the performance of nonsmooth schemes for the frictional contact problem, in terms of realism (capture of dry friction, typically), robustness, and computational efficiency. This study was conducted onto the different rod models that are available in the software. Second, we believe such a software will help us understand the behavior of a fibrous material (such as hair) through virtual experiments, thanks to which we hope to identify and understand some important emergent phenomena. A careful validation study against experiments started to be conducted in 2011 in collaboration with physicists from L’Oréal. Once this discrete elements model will be fully validated, our ultimate goal would be to build a continuous macroscopic model for the hair medium relying on nonsmooth laws.
An associate engineer, Gilles Daviet, has been hired in September 2009 to work full-time on the MECHE project. His contract was extended until October the 31st, funded by the L’Oréal collaboration project.
4. Software

4.1. The ECMPR software

Participant: Florence Forbes.

Joint work with: Radu Horaud and Manuel Iguel.

The ECMPR (Expectation Conditional Maximization for Point Registration) package implements \cite{56} \cite{17}. It registers two (2D or 3D) point clouds using an algorithm based on maximum likelihood with hidden variables. The method can register both rigid and articulated shapes. It estimates both the rigid or the kinematic transformation between the two shapes as well as the parameters (covariances) associated with the underlying Gaussian mixture model. It has been registered in APP in 2010 under the GPL license.

4.2. The LOCUS and P-LOCUS software

Participants: Florence Forbes, Senan James Doyle.

Joint work with: Michel Dojat.

From brain MR images, neuroradiologists are able to delineate tissues such as grey matter and structures such as Thalamus and damaged regions. This delineation is a common task for an expert but unsupervised segmentation is difficult due to a number of artefacts. The LOCUS software and its recent extension P-LOCUS automatically perform this segmentation for healthy and pathological brains An image is divided into cubes on each of which a statistical model is applied. This provides a number of local treatments that are then integrated to ensure consistency at a global level, resulting in low sensitivity to artifacts. The statistical model is based on a Markovian approach that enables to capture the relations between tissues and structures, to integrate a priori anatomical knowledge and to handle local estimations and spatial correlations.

The LOCUS software has been developed in the context of a collaboration between Mistis, a computer science team (Magma, LIG) and a Neuroscience methodological team (the Neuroimaging team from Grenoble Institut of Neurosciences, INSERM). This collaboration resulted over the period 2006-2008 into the PhD thesis of B. Scherrer (advised by C. Garbay and M. Dojat) and in a number of publications. In particular, B. Scherrer received a “Young Investigator Award” at the 2008 MICCAI conference. Its extension for lesion detection is realized by S. Doyle with financial support from Gravit for possible industrial transfer.

The originality of this work comes from the successful combination of the teams respective strengths i.e. expertise in distributed computing, in neuroimaging data processing and in statistical methods.

4.3. The POPEYE software

Participant: Florence Forbes.

Joint work with: Vasil Khalidov, Radu Horaud, Miles Hansard, Ramya Narasimha, Elise Arnaud.

POPEYE contains software modules and libraries jointly developed by three partners within the POP STREP project: INRIA, University of Sheffield, and University of Coimbra. It includes kinematic and dynamic control of the robot head, stereo calibration, camera-microphone calibration, auditory and image processing, stereo matching, binaural localization, audio-visual speaker localization. Currently, this software package is not distributed outside POP.

4.4. The HDDA and HDDC toolboxes

Participant: Stéphane Girard.
Joint work with: Charles Bouveyron (Université Paris 1) and Gilles Celeux (Select, INRIA). The High-Dimensional Discriminant Analysis (HDDA) and the High-Dimensional Data Clustering (HDDC) toolboxes contain respectively efficient supervised and unsupervised classifiers for high-dimensional data. These classifiers are based on Gaussian models adapted for high-dimensional data [53]. The HDDA and HDDC toolboxes are available for Matlab and are included into the software MixMod [52]. Recently, a R package has been developed and integrated in The Comprehensive R Archive Network (CRAN). It can be downloaded at the following URL: http://cran.r-project.org/web/packages/HDclassif/.

4.5. The Extremes freeware
Participants: Laurent Gardes, Stéphane Girard.

Joint work with: Diebolt, J. (CNRS) and Garrido, M. (INRA Clermont-Ferrand-Theix).

The EXTREMES software is a toolbox dedicated to the modelling of extremal events offering extreme quantile estimation procedures and model selection methods. This software results from a collaboration with EDF R&D. It is also a consequence of the PhD thesis work of Myriam Garrido [54]. The software is written in C++ with a Matlab graphical interface. It is now available both on Windows and Linux environments. It can be downloaded at the following URL: http://extremes.gforge.inria.fr/.

4.6. The SpaCEM³ program
Participants: Lamiae Azizi, Senan James Doyle, Florence Forbes.

SpaCEM³ (Spatial Clustering with EM and Markov Models) is a software that provides a wide range of supervised or unsupervised clustering algorithms. The main originality of the proposed algorithms is that clustered objects do not need to be assumed independent and can be associated with very high-dimensional measurements. Typical examples include image segmentation where the objects are the pixels on a regular grid and depend on neighbouring pixels on this grid. More generally, the software provides algorithms to cluster multimodal data with an underlying dependence structure accounting for some spatial localisation or some kind of interaction that can be encoded in a graph.

This software, developed by present and past members of the team, is the result of several research developments on the subject. The current version 2.09 of the software is CeCILL-B licensed.

Main features. The approach is based on the EM algorithm for clustering and on Markov Random Fields (MRF) to account for dependencies. In addition to standard clustering tools based on independent Gaussian mixture models, SpaCEM³ features include:

- The unsupervised clustering of dependent objects. Their dependencies are encoded via a graph not necessarily regular and data sets are modelled via Markov random fields and mixture models (eg. MRF and Hidden MRF). Available Markov models include extensions of the Potts model with the possibility to define more general interaction models.
- The supervised clustering of dependent objects when standard Hidden MRF (HMRF) assumptions do not hold (ie. in the case of non-correlated and non-unimodal noise models). The learning and test steps are based on recently introduced Triplet Markov models.
- Selection model criteria (BIC, ICL and their mean-field approximations) that select the "best" HMRF according to the data.
- The possibility of producing simulated data from:
  - general pairwise MRF with singleton and pair potentials (typically Potts models and extensions)
  - standard HMRF, ie. with independent noise model
  - general Triplet Markov models with interaction up to order 2
- A specific setting to account for high-dimensional observations.
- An integrated framework to deal with missing observations, under Missing At Random (MAR) hypothesis, with prior imputation (KNN, mean, etc), online imputation (as a step in the algorithm), or without imputation.
The software is available at http://spacem3.gforge.inria.fr. A user manual in English is available on the website above together with example data sets. The INRA Toulouse unit is more recently participating to this project for promotion among the bioinformatics community [20].

4.7. The FASTRUCT software

**Participant:** Florence Forbes.

**Joint work with:** Francois, O. (TimB, TIMC) and Chen, C. (former post-doctoral fellow in Mistis).

The FASTRUCT program is dedicated to the modelling and inference of population structure from genetic data. Bayesian model-based clustering programs have gained increased popularity in studies of population structure since the publication of the software STRUCTURE [65]. These programs are generally acknowledged as performing well, but their running-time may be prohibitive. FASTRUCT is a non-Bayesian implementation of the classical model with no-admixture uncorrelated allele frequencies. This new program relies on the Expectation-Maximization principle, and produces assignment rivaling other model-based clustering programs. In addition, it can be several-fold faster than Bayesian implementations. The software consists of a command-line engine, which is suitable for batch-analysis of data, and a MS Windows graphical interface, which is convenient for exploring data.

It is written for Windows OS and contains a detailed user’s guide. It is available at http://mistis.inrialpes.fr/realisations.html.

The functionalities are further described in the related publication:

- Molecular Ecology Notes 2006 [55].

4.8. The TESS software

**Participant:** Florence Forbes.

**Joint work with:** Francois, O. (TimB, TIMC) and Chen, C. (former post-doctoral fellow in Mistis).

TESS is a computer program that implements a Bayesian clustering algorithm for spatial population genetics. It is particularly useful for seeking genetic barriers or genetic discontinuities in continuous populations. The method is based on a hierarchical mixture model where the prior distribution on cluster labels is defined as a Hidden Markov Random Field [59]. Given individual geographical locations, the program seeks population structure from multilocus genotypes without assuming predefined populations. TESS takes input data files in a format compatible to existing non-spatial Bayesian algorithms (e.g. STRUCTURE). It returns graphical displays of cluster membership probabilities and geographical cluster assignments through its Graphical User Interface.

The functionalities and the comparison with three other Bayesian Clustering programs are specified in the following publication:

- Molecular Ecology Notes 2007
NANO-D Team

5. Software

5.1. SAMSON

Figure 6. SAMSON’s architecture.

A major objective of NANO-D is to try and integrate a variety of adaptive algorithms into a unified framework. As a result, NANO-D is developing SAMSON (Software for Adaptive Modeling and Simulation Of Nanosystems), a software platform aimed at including all developments from the group, in particular those described below.

The objective is to make SAMSON a generic application for computer-aided design of nanosystems, similar to existing applications for macrosystem prototyping (CATIA, SolidWorks, etc.).

The current architecture of SAMSON is visible in Figure 6. The code is organized into four main parts: a) the Base (in which “Core” contains, in particular, the heart of the adaptive algorithms: signaling mechanisms specifically designed for SAMSON), b) the Software Development Kit (SDK: a subset of the base that will be provided to module developers), c) Modules, and d) the SAMSON application itself.

Similar to the concept of Mathematica toolboxes, for example, the goal has been to make it possible to personalize the user interface of SAMSON for potentially many distinct applications. For example, we may want to personalize the interface of SAMSON for crystallography, drug design, protein folding, electronics, material science, nano-engineering, etc., by loading different modules at startup, depending on the user application domain.
NECS Project-Team

5. Software

5.1. ORCCAD

Participants: Daniel Simon [correspondant], Soraya Arias [SED], Roger Pissard-Gibollet [SED].

ORCCAD is a software environment that allows for the design and implementation of the continuous and discrete time components of complex control systems, e.g. robotics systems which provided it first ground [64]. It also allows the specification and validation of complex missions to be performed by the system. It is mainly intended for critical real-time applications, in which automatic control aspects (servo loops) have to interact narrowly with the handling of discrete events (exception handling, mode switching). ORCCAD offers a complete and coherent vertical solution, ranging from the high level specification to real-time code generation. The ORCCAD V3 software was designed with proprietary tools that moreover are now becoming obsolete. ORCCAD V4 is currently deeply re-engineered to be compliant with open-source and free software tools (Java/Eclipse). Current targets are Linux (Posix threads) and Xenomai, a real-time development framework cooperating with the Linux kernel (http://www.xenomai.org). ORCCAD is supported by the Support Expérimentations & Développement (SED) service of INRIA-Rhône-Alpes. ORCCAD is used by the experimental robotics platforms of INRIA-Rhône-Alpes and by the Safenecs ANR project in a real-time simulator of a X4 drone. New functionalities and updates are developed jointly by the SED service and researchers of the NECS and SARDÈS teams. Web page: http://orccad.gforge.inria.fr.

5.2. MASim

Participants: J. Dumon [contact person], P. Bellemain [GIPSA-Lab], S. Nicolas [PROLEXIA], N. Maciol [PROLEXIA], F. Martinez [ROBOSOFT], J. Caquas [ROBOSOFT].

MASim is a tool that has been adapted from our former multiagent simulator MUSim (MUSim=MASim + ConnectSim + ConnectIHM). It integrates agent's models, communication media including their limitations, heterogeneous network, and all the variants of the multi-agent control strategies. Besides the models and simulation engine, the simulation can be replayed through a GUI, an interactive graphical interface which is used to visualise and interpret the state of the multi-agent control system and communication topology. The validation scenario is a real-size application enough complex to enforce the pertinence of our results. The simulator MASim is now being used as an open research tool for various applications in the field of multi-agents networked systems, particularly within the FeedNetBack project (see Fig. 4).
Figure 4. A scenario’s view obtained with MASIM.
5. Software

5.1. NUM3SIS

Participants: Régis Duvigneau [correspondant], Thibaud Kloczko, Nora Aïssiouene.

NUM3SIS (http://num3sis.inria.fr) is a modular platform devoted to scientific computing and numerical simulation. It is not restricted to a particular application field, but is designed to host complex multidisciplinary simulations. Main application fields are currently Computational Fluid Dynamics (CFD), Computational Electro-Magnetics (CEM, in collaboration with Nachos Project-Team) and pedestrian traffic simulation.

The most important concept in NUM3SIS is the concept of node. It is a visual wrapper around derivatives of fundamental concepts such as data, algorithm or viewer. Atomic nodes are provided for convenience in order to manipulate computational data (such as grids or fields), apply computational methods (such as the building of a finite-element matrix or the construction of a finite-volume flux) and visualize computational results (such as vector or tensor fields, on a screen or in an immersive space). For a given abstract node, different implementations can be found, each of them being embedded in a plugin system that is managed by a factory.

The second important concept in NUM3SIS is the concept of composition. It consists of the algorithmic pipeline used to link the nodes together. The use of these two concepts, composition and nodes, provides a highly flexible, re-usable and efficient approach to develop new computational scenarios and take benefit from already existing tools. This is a great advantage with respect to classical monolithic softwares commonly used in these fields.

This work is being carried out with the support of two engineers in the framework of an ADT (Action de Développement Technologique) program.

5.2. FAMOSA

Participant: Régis Duvigneau [correspondant].

Opale team is developing the software platform FAMOSA (C++), that is devoted to multidisciplinary design optimization in engineering. It integrates the following components:

- an optimization library composed of various algorithms: several descent methods from steepest-descent method to quasi-Newton BFGS method (deterministic, smooth), the Multi-directional Search Algorithm (deterministic, noisy), the Covariance Matrix Adaption Evolution Strategy (semi-stochastic, multi-modal) and the Efficient Global Optimization method (deterministic, multi-modal). It also contains the Pareto Archived Evolution Strategy to solve multi-objective optimization problems;
- an evaluation library managing the performance estimation process (communication with external simulation tools);
- a metamodel library that contains tools to build a database and kriging models that are used to approximate the objective function for different purposes;
- a scenario library that allows to use the previous components to achieve various tasks:
  - Construct a design of experiments;
  - Construct a metamodel;
  - Find the design that minimizes a cost functional;
  - Find the Pareto front for two cost functionals
  - Play a Nash game to find the equilibrium between two criteria;
  - Apply a multiple gradient descent strategy to improve simultaneously two criteria.
Figure 1. Illustration of the graphic user interface of the NUM3SIS platform: at the top the composition space, at the bottom the visualization space.

The FAMOSA platform is employed by Opale Project-Team to test its methodological developments in multidisciplinary design optimization (MDO). The platform is also used by the Fluid Mechanics Laboratory at Ecole Centrale de Nantes and by the K-Epsilon company (http://www.k-epsilon.com) for hydrodynamic design applications. Moreover, it is presently tested by Peugeot Automotive industry for external aerodynamic design purpose.

5.3. Plugins for AXEL

Participants: Régis Duvigneau [correspondant], Louis Blanchard.

Opale team is developing plugins in the framework of the algebraic modeler Axel, in collaboration with GALAAD team. These developments correspond to two research axes:

- methods for isogeometric analysis and design. In particular, two simulation tools for heat conduction and compressible flows have been implemented, in conjunction with some deterministic and semi-stochastic optimization algorithms for optimum-shape design;
- methods for geometrical modeling of bow shapes for trawler ships.

5.4. Integration platform for multidiscipline optimization applications

Participants: Toan Nguyen, Laurentiu Trifan.

A prototype software integration platform is developed and tested for multidiscipline optimization applications. It is based on a workflow management system called YAWL (http://www.yawlfoundation.org). The goal is to design, develop and assess high-performance distributed scientific workflows featuring resilience, fault-tolerance and exception-handling capabilities. The platform is used to experiment new resilience algorithms, including monitoring and management of application-level errors. The platform is tested against use-cases provided by the industry partners in the OMD2 project supported by the French Agence Nationale de la Recherche. This work is part of Laurentiu Trifan’s PhD thesis. (See Fig. 2.)
Figure 2. Testcase deployment on the Grid5000 infrastructure.
5. Software

5.1. AcypiCyc

Participants: Hubert Charles [EPI], Patrice Baa Puyoule [Contact, Patrice.Baa-Puyoulet@lyon.inra.fr], Stefano Colella [Contact, stefano.colella@lyon.inra.fr], Ludovic Cottret, Marie-France Sagot [EPI], Augusto Vellozo [Contact, augusto@cycadsys.org], Amélie Véron.

Database of the metabolic network of *Acrystosiphon pisum*.
http://acypicyc.cycadsys.org/

5.2. BaobabLuna

Participants: Marília Braga [Contact, mdvbraga@gmail.com], Marie-France Sagot [EPI], Eric Tannier.

Manipulation of signed permutations in the context of genomic evolution.
http://pbil.univ-lyon1.fr/software/luna/

5.3. Cassis

Participants: Christian Baudet [EPI, Contact, christian.baudet@univ-lyon1.fr], Christian Gautier [EPI], Claire Lemaitre [Contact, claire.lemaitre@inria.fr], Marie-France Sagot [EPI], Eric Tannier.

Algorithm for precisely detecting genomic rearrangement breakpoints.
http://pbil.univ-lyon1.fr/software/Cassis/

5.4. Cravela

Participants: Ana Teresa Freitas, Nuno Mendes [EPI, Contact, ndm@kdbio.inesc-id.pt], Marie-France Sagot [EPI, Contact, marie-france.sagot@inria.fr].

Framework for the identification and evaluation of miRNA precursors (finished), targets (in development) and regulatory modules (in development).
http://www.cravela.org/

5.5. C3P

Participants: Frédéric Boyer, Anne Morgat [EPI, ext. member], Alain Viari [EPI, Contact, alain.viari@inria.fr].

Merging two or more graphs representing biological data (e.g. pathways, ...).
http://www.inrialpes.fr/helix/people/viari/cccpart

5.6. CycADS

Participants: Hubert Charles [EPI], Patrice Baa Puyoule [Contact, Patrice.Baa-Puyoulet@lyon.inra.fr], Stefano Colella [Contact, stefano.colella@lyon.inra.fr], Ludovic Cottret, Marie-France Sagot [EPI], Augusto Vellozo [Contact, augusto@cycadsys.org].

Cyc annotation database system.
http://www.cycadsys.org/

5.7. Ed’Nimbus

Participants: Pierre Peterlongo [Contact, pierre.peterlongo@inria.fr], Marie-France Sagot [EPI].

Algorithm for detecting and filtering repeats in sequences prior to multiple alignments.
5.8. GeM

**Participants:** Gisèle Bronner, Christian Gautier [EPI, Contact, christian.gautier@univ-lyon1.fr], Bruno Spataro.

Database for comparative genomic analysis of complete vertebrate genomes.

http://pbil.univ-lyon1.fr/gem/gem_home.php

5.9. Gobbolino

**Participants:** Vicente Acuña [EPI], Etienne Birmelé [EPI, délégation], Ludovic Cottret, Pierluigi Crescenzi, Fabien Jourdan, Vincent Lacroix, Alberto Marchetti-Spaccamela [EPI, ext. member], Andrea Marino, Paulo Vieira Milreu [EPI, Contact, pvmilreu@gmail.com], Marie-France Sagot [EPI], Leen Stougie [EPI, ext. member].

Algorithm to enumerate all metabolic stories in a metabolic network given a set of metabolites of interest. Code available on request.

5.10. kisSnp

**Participants:** Vincent Lacroix [EPI], Pierre Peterlongo [Contact, pierre.peterlongo@inria.fr], Nadia Pisanti, Marie-France Sagot [EPI], Nicolas Schnel.

Algorithm for identifying SNPs without a reference genome by comparing raw reads.

http://alcovna.genouest.org/kissnp/

5.11. kisSplice

**Participants:** Rayan Chikhi, Janice Kielbassa [EPI], Vincent Lacroix [Contact, EPI], Pierre Peterlongo [Contact, pierre.peterlongo@inria.fr], Gustavo Sacomoto [EPI], Marie-France Sagot [EPI], Raluca Uricaru.

Algorithm for de-novo calling alternative splicing events from RNA-seq data.

http://alcovna.genouest.org/kissplice/

5.12. MetExplore

**Participants:** Michael Barrett, Hubert Charles [EPI], Ludovic Cottret [Contact, Ludovic.Cottret@toulouse.inra.fr], Fabien Jourdan, Marie-France Sagot [EPI], Florence Vinson, David Wildridge.

Web server to link metabolomic experiments and genome-scale metabolic networks.

http://metexplore.toulouse.inra.fr/metexplore/

5.13. Migal

**Participants:** Julien Allali [Contact, julien.allali@labri.fr], Marie-France Sagot [EPI].

Algorithm for comparing RNA structures.


5.14. MotusWEB

**Participants:** Ludovic Cottret, Fabien Jourdan, Vincent Lacroix [EPI, Contact, vincent.lacroix@univ-lyon1.fr], Odile Rogier, Marie-France Sagot [EPI].

Algorithm for searching and inferring coloured motifs in metabolic networks (web-based version - offers different functionalities from the downloadable version).

http://pbil.univ-lyon1.fr/software/motus_web/

5.15. Motus

**Participants:** Ludovic Cottret, Fabien Jourdan, Vincent Lacroix [EPI, Contact, vincent.lacroix@univ-lyon1.fr], Odile Rogier, Marie-France Sagot [EPI].
Algorithm for searching and inferring coloured motifs in undirected graphs (downloadable version - offers different functionalities from the web-based version).
http://pbil.univ-lyon1.fr/software/motus/

5.16. PhEVER

Participants: Christian Gautier [EPI], Vincent Lotteau, Leonor Palmeira [Contact, mlpalmeira@ulg.ac.be], Chantal Rabourdin-Combe, Simon Penel.
Database of homologous gene families built from the complete genomes of all available viruses, prokaryotes and eukaryotes and aimed at the detection of virus/virus and virus/host lateral gene transfers.
http://pbil.univ-lyon1.fr/databases/phever/

5.17. PepLine

Participants: Jérôme Garin, Alain Viari [EPI, Contact, alain.viari@inria.fr].
Pipeline for the high-throughput analysis of proteomic data.
http://www.grenoble.prabi.fr/protehome/software/pepline

5.18. Pitufo

Participants: Vicente Acuña [EPI], Ludovic Cottret [Contact, Ludovic.Cottret@toulouse.inra.fr], Alberto Marchetti-Spaccamela [EPI, ext. member], Paulo Vieira Milreu [EPI, Contact, pvmilreu@gmail.com], Marie-France Sagot [EPI], Leon Stougie [EPI, ext. member], Fabio Vidal-Martinez.
Algorithm to enumerate all minimal sets of precursors of target compounds in a metabolic network.
http://sites.google.com/site/pitufosoftware/

5.19. PSbR

Participants: Yoan Diekmann, Marie-France Sagot [EPI, Contact, marie-france.sagot@inria.fr], Eric Tannier.
Algorithm for testing the evolution and conservation of common clusters of genes.
http://pbil.univ-lyon1.fr/members/sagot/htdocs/team/software/PSbR/

5.20. Repseek

Participants: Guillaume Achaz [Contact, achaz@abi.snv.jussieu.fr], Eric Coissac, Alain Viari [EPI].
Finding approximate repeats in large DNA sequences.
http://wwwabi.snv.jussieu.fr/~public/RepSeek/

5.21. Smile

Participants: Laurent Marsan, Marie-France Sagot [EPI, Contact, marie-france.sagot@inria.fr].
Motif inference algorithm taking as input a set of biological sequences.

5.22. Tuiuiu

Participants: Alair Pereira do Lago, Pierre Peterlongo [Contact, pierre.peterlongo@inria.fr], Nadia Pisanti, Gustavo Sacomoto [EPI], Marie-France Sagot [EPI].
Multiple repeat search filter with edit distance.
http://mobyle.genouest.org/cgi-bin/Mobyle/portal.py?form=tuiuiu

5.23. UniPathway

Participants: Eric Coissac, Anne Morgat [EPI, Contact, anne.morgat@inria.fr], Alain Viari [EPI].
Database of manually curated pathways developed with the Swiss-Prot group.
http://www.unipathway.org
4. Software

4.1. aevol (artificial evolution)

Participants: Guillaume Beslon, Stephan Fischer, Carole Knibbe, David P. Parsons, Bérénice Batut.

- Contact: Carole Knibbe (carole.knibbe@inrialpes.fr).
- Aevol is a simulation software dedicated to the study of genome evolution. It allows to carry out in silico experimental evolution. Populations of digital organisms reproduce and mutate randomly, with both small mutations and large chromosomal rearrangements, in a steady or varying environment. A curve-fitting task is used to determine the fitness of the organisms and thus their rate of reproduction. The number of genes, their order, their sequences, their intergenic distances are all free to evolve.
- URL: http://gforge.liris.cnrs.fr/projects/aevol/

4.2. DMT4SP (Data Mining Tool For Sequential Patterns)

Participant: Christophe Rigotti.

- Contact: Christophe.Rigotti@insa-lyon.fr.
- Summary: The dmt4sp prototype is a command line tool to extract episodes and episode rules, supporting various constraints, over a single sequence or several sequences of events. Three kinds of patterns can be extracted: (1) serial episodes, (2) serial episode rules having a single event type in the consequent, and (3) quantitative episodes (aka grouping of “homogeneous” occurrences of the serial episodes with respect to the time gap between events).
- URL: http://liris.cnrs.fr/~crigotti/dmt4sp.html
5. Software

5.1. CelDyn

Participants: Nikolai Bessonov, Vitaly Volpert [correspondant].

Software "Celdyn" is developed in order to model cell population dynamics for biological applications. Cells are represented either as soft spheres or they can have more complex structure. Cells can divide, move, interact with each other or with the surrounding medium. Different cell types can be introduced. When cells divide, the types of daughter cells are specified. A user interface is developed.
4. Software

4.1. Genetic Network Analyzer (GNA)

Participants: Hidde de Jong [Correspondent], Michel Page, François Rechenmann, Delphine Ropers.

Genetic Network Analyzer (GNA) is the implementation of a method for the qualitative modeling and simulation of gene regulatory networks developed in the IBIS project. The input of GNA consists of a model of the regulatory network in the form of a system of piecewise-linear differential equations, supplemented by inequality constraints on the parameters and initial conditions. From this information, GNA generates a state transition graph summarizing the qualitative dynamics of the system. In order to analyze large graphs, GNA allows the user to specify properties of the qualitative dynamics of a network in temporal logic, using high-level query templates, and to verify these properties on the state transition graph by means of standard model-checking tools, either locally installed or accessible through a remote web server. GNA is currently distributed by the company Genostar, but remains freely available for academic research purposes. The current version is GNA 8.3. In comparison with the previously distributed versions, GNA 8.3 has the following additional functionalities. First, it supports the editing and visualization of regulatory networks, in an SBGN-compatible format, and second it semi-automatically generates a prototype model from the network structure, thus accelerating the modeling process. For more information, see http://www-helix.inrialpes.fr/gna.

4.2. WellReader

Participants: Guillaume Baptist, Johannes Geiselmann, Jérôme Izard, Hidde de Jong [Correspondent], Delphine Ropers.

WellReader is a program for the analysis of gene expression data obtained by means of fluorescent and luminescent reporter genes. WellReader reads data files in an XML format or in a format produced by microplate readers, and allows the user to detect outliers, perform background corrections and spline fits, compute promoter activities and protein concentrations, and compare expression profiles across different conditions. WellReader has been written in MATLAB and is available under an LGPL licence, both as source code (M files) and compiled code (platform-specific binary files). For more information, see: http://ibis.inrialpes.fr/article957.html.
5. Software

5.1. Adaptive Grid Refinement

Participants: Laurent Debreu, Marc Honnorat.

AGRIF (Adaptive Grid Refinement In Fortran, [71]) is a Fortran 90 package for the integration of full adaptive mesh refinement (AMR) features within a multidimensional finite difference model written in Fortran. Its main objective is to simplify the integration of AMR potentialities within an existing model with minimal changes. Capabilities of this package include the management of an arbitrary number of grids, horizontal and/or vertical refinements, dynamic regridding, parallelization of the grids interactions on distributed memory computers. AGRIF requires the model to be discretized on a structured grid, like it is typically done in ocean or atmosphere modelling. As an example, AGRIF is currently used in the following ocean models: MARS (a coastal model developed at IFREMER-France), ROMS (a regional model developed jointly at Rutgers and UCLA universities), OPA-NEMO ocean modelling system (a general circulation model used by the French and European scientific community) and HYCOM (a regional model developed jointly by University of Miami and the French Navy).

In 2011, a new contract has been signed with IFREMER to optimize parallel capabilities of the software. AGRIF is licensed under a GNU (GPL) license and can be downloaded at its web site (http://ljk.imag.fr/MOISE/AGRIF/index.html).

5.2. DatIce

Participants: Bénédicte Lemieux-Dudon, Habib Toye Mahamadou Kele.

The Datice code ([76], [77]) is designed to estimate consistent chronologies of several deep ice cores (i.e., depth-age relationships of the ice matrix and trapped gas). A cost function derived from Bayes theorem puts in competition the chronological constraints brought by heterogeneous observations (stratigraphic links between cores, gas and ice age markers, delta-depth markers, etc.), and the background dating scenarios simulated with glaciological models (firn densification and ice flow models). The minimization of the cost function provides optimal estimations of three key quantities from which dating scenarios can be derived: the past accumulation rate, the close-off depth which is the depth where the gas is trapped into ice, and the total thinning function. Uncertainties of the analysed dating scenarios (key quantities and chronologies) are assessed on the basis of the Bayesian formulation. This approach is innovative because:

- it relies on data assimilation techniques to calculate ice core chronologies and uncertainties;
- it applies to a large number of heterogeneous observations;
- it ensures consistency between the chronologies of several cores and the consistency between the gas and ice age scales.

The code has been used in several recent publications (see [68], [87] for example).

5.3. SDM toolbox

Participant: Antoine Rousseau.
The computation of the wind at small scale and the estimation of its uncertainties is of particular importance for applications such as wind energy resource estimation. To this aim, we develop a new method based on the combination of an existing numerical weather prediction model providing a coarse prediction, and a Lagrangian Stochastic Model adapted from a pdf method introduced by S.B. Pope for turbulent flows. This Stochastic Downscaling Method (SDM http://sdm.gforge.inria.fr/) is thus aimed to be used as a refinement toolbox of large-scale numerical models. SDM requires a specific modelling of the turbulence closure, and involves various simulation techniques whose combination is totally new (such as Poisson solvers, optimal transportation mass algorithm, original Euler scheme for confined Langevin stochastic processes, and stochastic particle methods). In 2011, we worked on the comparison of the SDM model (endowed with a physical geostrophic forcing and a wall log law) with simulations obtained with a LES method (Méso-NH code) for the atmospheric boundary layer (from 0 to 750 meters in the vertical direction), in the neutral case, see [58].

5.4. CompModSA package

Alexandre Janon is a contributor of the package CompModSA - Sensitivity Analysis for Complex Computer Models (see http://cran.r-project.org/web/packages/CompModSA/index.html). This package is useful for conducting sensitivity analysis of complex computer codes when model evaluations are somewhat expensive (e.g. take longer than a couple of seconds to run) but a reasonable number (50 or more) of model evaluations can be obtained at sampled input values.

5.5. NEMO-TAM

Tangent and adjoint models for the NEMO platform of the oceanic modelling that have been developed by the MOISE team have been published now under Cecill license and distributed by the NEMO consortium.
5. Software

5.1. Zebre

Participant: Thierry Dumont [correspondant].

Thierry Dumont is currently developing a toolbox to solve stiff reaction diffusion equations using splitting methods, together with refined numerical schemes for ODEs (RADO 5).

5.2. OptimChemo

Participants: Violaine Louvet [correspondant], Emmanuel Grenier.

OptimChemo is a user-friendly software designed to study numerically the effect of multiple chemotherapies on simple models of tumour growth and to optimize chemotherapy schedules.
4. Software

4.1. TEOS: Tranus Exploration and Optimization Software

Participants: Anthony Tschirhard, Mathieu Vadon, Elise Arnaud, Emmanuel Prados.

The TEOS software offers a set of tools to help the calibration of the land use and transport integrated model TRANUS. It uses some exploration and optimization procedures of the relevant parameters.
AMAZONES Team

5. Software

5.1. Logos

Participants: Julien Ponge, Stéphane Frénot.

Logos is a development project linked to the LISE ANR grant. Its goal is to generate execution logs from OSGi services interactions. The main idea is to intercept every service call and generate an entry in a log file. The log file system should be used in the LISE context which is related to legal issues. Generated Logos logs should be: Complete, encoded with a cryptographic algorithm, compact and immutable.

The software is currently used as a Amazones internal test suite. It is fully tested on standard OSGi architectures.

5.2. Logminer

Participants: Julien Ponge, Stéphane Frénot.

LogMiner is a toolbox, written in Scala in current development. The LogMiner framework takes Logos inputs and generates service usage automata. The goal of logminer is to represent application activity in a synthetic way in order to identify behavioral changes while updating the system. When one updates its applications on its environment, the logminer framework enables observation and identifies variations in service usages.

The software is currently under development it integrates a automata generator and a data visualisation modules.

5.3. Eimc

Participants: Zheng Hu, Stéphane Frénot, Bernard Tourancheau [Projet Swing].

Eimc is an architecture for managing sensor dedicated to legacy equipment management. The project aims at designing a dynamic framework that integrates sensors from the surrounding environment and detects new equipments from their physical behavior. For instance, a fridge vibrates when the compressor is working. The frequency of vibrations distinguishes a fridge from a washing machine. The framework designs a Complex Based Event processing architecture where we need to focus on the number of manageable equipments, the number of deployed sensors and the number of physical measurements that can be handled.

The project is a joint project with Orange Labs, and a PhD student Zheng Hu. He is co-directed by Stéphane Frénot and Bernard Tourancheau from Amazones and Swing teams.

5.4. Aoraï

Participant: Nicolas Stouls.

Developed at CEA-LIST, Frama-C is an extensible and collaborative platform dedicated to source-code analysis of C software. The Aoraï [49] plug-in for Frama-C [31] provides a method to automatically annotate a C program according to a behavioral property P such that, if the annotations are verified, then we ensure that the program respects P.

The computation process is divided into two steps: the specification generation from the property and the constraints propagation for static simplification. According to the classical invariant verification granularity, observable states of a program correspond to each call or return statements of an operation. Each state of the program is associated to a set of transitions in an internal representation of the property, managed as a Büchi automata. Starting from a super-set of authorized behaviors, some static simplifications can be done in order to generate sufficient pre/post-conditions on each operation.
The classical method to validate generated annotations is to use the Jessie plug-in and the Why tool, using theorem provers.

A new research report [50] has been published and some new developments have been done in order to increase consequently the efficiency of the tool.

5.5. STOP

**Participants:** François Goichon, Stéphane Frénot, Pierre Parrend.

STOP is a security-oriented program analysis toolkit developed by François Goichon as part of his masters thesis. He was supervised by Stéphane Frénot and Pierre Parrend from FZI, Karlsruhe.

The tool implements a novel static analysis technique called *Service-oriented Tainted Object Propagation*, described in more detail in the Results section.

5.6. IzPack

**Participant:** Julien Ponge.

IzPack [47] is a software installer creation framework for the Java platform. Its main differentiator with respect to the other installation solutions is that it generates cross-platform installers that can adapt themselves to the underlying operating system so as to still provide tight integration. It was also designed to be highly customizable and extensible.

IzPack is nearing its 10 years landmark. It is hosted at the Coddehaus [32] Foundation and released under the terms of the Apache Software License version 2.0. Its users community non-exhaustively comprise Spring-Source, JBoss / RedHat, Oracle / Sun Microsystems, the Scala language, XWiki, Terracotta or Silverpeas.

The project was originally created by now INRIA Amazones team member Julien Ponge, who still leads the project. In 2010, it was presented at the Devoxx conference.

5.7. WSNet

**Participants:** Guillaume Chelius [INRIA D-NET Team, project leader], Antoine Fraboulet, Loïc Lemaître [INRIA SensTools IJD].

WSNet is a modular wireless network simulator. It incorporates the following aspects: (i) accurate simulation of the radio channel: Supports MIMO, multi-interface, multi-channel, etc. (ii) Simulation environment: simulation of the interaction between sensors and their environment: measurement and control, simulation of device power consumption. Furthermore, WSNet can be interfaced with the WSim sensor node emulator to form a distributed emulation of a sensor network.

WSNet source code is registered at the Agency For The Protection Of Programs (APP IDDN 06-370013-000). Licence: CeCILL (2). See also the web page [http://wsnet.gforge.inria.fr/](http://wsnet.gforge.inria.fr/).

5.8. WSim

**Participants:** Guillaume Chelius [INRIA D-NET Team], Antoine Fraboulet [Project leader], Loïc Lemaître [INRIA SensTools IJD], Julien Carpentier [INRIA ORSI IJD].

WSIM is a platform emulator for embedded systems allowing performance evaluation and programming assistance during the application design stages of distributed wireless sensor networks. WSIM is a simulation tool enabling a rapid and relevant feedback on features and quality of embedded software in constrained systems. Its simulation model allows to interface with other tools like WSNet to build complex simulation environments.

WSim source code is registered at the Agency For The Protection Of Programs (APP IDDN 06-370012-000). Licence: CeCILL (2). See also the web page [http://wsim.gforge.inria.fr/](http://wsim.gforge.inria.fr/).
5.9. Esimu

**Participant:** Antoine Fraboulet.

eSimu is a complete system energy model based on non-intrusive measurements. This model aims at being integrated in fast cycle accurate simulation tools to give energy consumption feedback for embedded systems software programming. Estimations take into account the whole system consumption including peripherals. Experiments on a complex ARM9 platform show that our model estimates are in error by less than 10% from real system consumption, which is precise enough for source code application design, while simulation speed remains fast. eSimu can be used as a standalone tool or in conjunction with WSim.

Licence: CeCILL (2). See also the web page [http://esimu.gforge.inria.fr/](http://esimu.gforge.inria.fr/).

5.10. ABR

**Participants:** Frédéric Le Mouël, Stéphane Frénot.

The Ambient Bundle Repository (ABR) is an OSGi extension, compliant with the Bundle Repository API. Instead of proposing a centralized discovery as the default bundle repository implementation, ABR abstracts different discovery protocols (UPnP, ...) and publishes/subscribes a local repository containing bundles in a device geographically-close environment. ABR implements mobility models to track mobile devices, to warn the user deploying bundles of the remaining presence time of bundles and to anticipate a possible bundle deployment non-ending.

5.11. AxSeL

**Participants:** Amira Ben Hamida, Frédéric Le Mouël, Stéphane Frénot.

While installing and executing applications on mobile devices, the issue of the limit of resources is quickly encountered.

AxSeL (A conteXtual Service Loader) is an OSGi prototype extension that modifies the bundle loading at deployment time for a context-aware service loading at run time. The approach is based on a service graph colouring process. We represent an application as a bi-dimensional dynamic graph with services and bundles dependencies. The colouring decision provides an optimal deployment configuration of the application in a given context. Context listening mechanisms capture changes and propagate recolouring and redeployment processes.

Context elements currently implemented and monitored are the hardware memory and disk sizes. Application currently implemented and tested is a service-oriented PDF viewer that is adapting its display to available device resources [6].

This prototype is a part of the PhD thesis of Amira Ben Hamida [29].

5.12. QuestMonitor

**Participants:** Stéphane Grumbach, Ahmad Ahmad-Kassem, Fuda Ma.

QuestMonitor [28] is a visualization tool that allows to visualize dynamic networks, and monitor the execution of protocols written in the data centric language Netlog. The language allows to specify protocols which in sometimes their behavior, in dynamic networks, are tricky to understand. QuestMonitor allows to monitor all the communication between the nodes, the evolution of the data stores on each node, as well as the execution of the declarative code. It also allows to color the virtual data structures, such as routes, backbones, etc. Together with the code editing facility, it constitutes a good tool for rapid prototyping.

Amazones team aimed at bridging the gap between "high-level" developed architectures that we called the northBound and "low-level" run-time, the southBound. Northbound architecture rely on virtual machines and advanced development languages, whereas southbound architectures rely on micro-kernels and drivers development. Our results are mainly initial studies since we fixed our research team on November 2010, and the end notification arrived on June 2011. Although our time frame was short we managed to gain knowledge in three areas linked to Amazones goals.
5. Software

5.1. WSNet

Participant: Guillaume Chelius [correspondant].

WSNet is a wireless sensor network simulator that was designed to offer the following features:

- a modular, flexible and accurate simulation of the radio physical medium;
- support for the simulation of environmental phenomena;
- support for interaction between nodes and their environment (sensor-actuator architecture);
- interconnection with the sensor platform emulator WSim to support the distributed emulation of wireless sensor networks.

WSNet is currently in its second release. The number of WSNet users is still growing and several research works reference the software. Many pointers can be found on the project website. Maintenance and support of the software is handled by the D-NET project but also by several contributors from the CITI laboratory (INSA de Lyon), Orange R&D. The WSNet community is quietly spreading in France as well as abroad.

5.2. WSNet-3

Participant: Guillaume Chelius [correspondant].

Driven by the feedback gathered among WSNet users, we have started the development of the third WSNet release. While still private, the project web page is available. The objectives behind this new development is:

- to ease the simulation of new radio architectures / standards: e.g. MIMO schemes, UWB, multi-interfaces system;
- to ease the writing of new modules through the use of High Level Languages such as Python or Ruby for the development of protocols, etc;
- to ease the debugging and compilation of results during a simulation.

These developments are handled by a core of developers from different affiliations (INSA de Lyon, Orange R&D, INRIA) lead by the D-NET team.

5.3. Sensor Network Tools: drivers, OS and more

Participants: Guillaume Chelius, Eric Fleury [correspondant], Clément Burin des Rosiers, Sandrine Avakian, Guillaume Roche.

As a outcomes of the ANR SensLAB project and the INRIA ADT SensTOOLS and SensAS, several softwares (from low level drivers to OSes) were delivered and made available to the research community. The main goal is to lower the cost of developing/deploying a large scale wireless sensor network application. All software are gathered under the SensLAB web site: http://www.senslab.info/ web page where one can find:

- low C-level drivers to all hardware components;
- ports of the main OS, mainly TinyOS, FreeRTOS and Contiki;
- ports and development of higher level library like routing, localization.
5. Software

5.1. DIET

**Participants:** Yves Caniou, Eddy Caron [correspondent], Frédéric Desprez, Maurice Djibril Faye, Adrian Muresan, Jonathan Rouzaud-Cornabas.

Huge problems can now be processed over the Internet thanks to Grid and Cloud middleware systems. The use of on-the-shelf applications is needed by scientists of other disciplines. Moreover, the computational power and memory needs of such applications may of course not be met by every workstation. Thus, the RPC paradigm seems to be a good candidate to build Problem Solving Environments on the Grid or Cloud. The aim of the DIET project (http://graal.ens-lyon.fr/DIET) is to develop a set of tools to build computational servers accessible through a GridRPC API.

Moreover, the aim of a middleware system such as DIET is to provide a transparent access to a pool of computational servers. DIET focuses on offering such a service at a very large scale. A client which has a problem to solve should be able to obtain a reference to the server that is best suited for it. DIET is designed to take into account the data location when scheduling jobs. Data are kept as long as possible on (or near to) the computational servers in order to minimize transfer times. This kind of optimization is mandatory when performing job scheduling on a wide-area network. DIET is built upon Server Daemons. The scheduler is scattered across a hierarchy of Local Agents and Master Agents. Applications targeted for the DIET platform are now able to exert a degree of control over the scheduling subsystem via plug-in schedulers [85]. As the applications that are to be deployed on the Grid vary greatly in terms of performance demands, the DIET plug-in scheduler facility permits the application designer to express application needs and features in order that they be taken into account when application tasks are scheduled. These features are invoked at runtime after a user has submitted a service request to the MA, which broadcasts the request to its agent hierarchy.

DIET has been validated on several applications. Example of them have been described in Sections 4.3 through 4.5.

5.1.1. DIET Security

We have worked on extending DIET to include security mechanisms. The first work was to provide authentication of users and components within DIET without breaking DIET distributed architecture. Our security mechanism must also be simple to use by the end users but we need a strong authentication. Recently, we have opted for Kerberos as it provided a Single Sign One that eases the security from the user point of view. Moreover, Kerberos provides strong authentication and works with heterogeneous systems. Work in progress is to integrate Kerberos within DIET. First, it will be used to provide traceability of user’s actions and authentication of all DIET inner components. Then, it will be integrated in an authorization mechanism and other higher level security mechanisms.

5.1.2. GridRPC Data Management API

The GridRPC paradigm is an OGF standard, but the API appeared to lack of precision in order to make a GridRPC code portable to any GridRPC compliant middleware. Additionally required data have to be present on the client side (this can involve a potential transfer from where the data is stored onto the client), and transfers must be performed during the GridRPC call, both degrading performance, and can even make a calculus unfeasible.

Thus the GridRPC community has interests in Data Management within the GridRPC paradigm – Because of previous works performed in the DIET middleware concerning Data Management, Eddy Caron is co-chair of the GridRPC working group.
In consequence, we worked on a Data Management API which has been presented to almost all OGF sessions since OGF’21. Since September 2011, the proposal is an OGF standard, published at http://www.ogf.org/documents/GFD.186.pdf under the title “Data Management API within the GridRPC. Y. Caniou and others, via GRIDRPC-WG”. Some work are still in progress, like 1) the implementation of a library and its integration into GridRPC middleware, in order to publish a proof of concept of both realization and collaboration between two different GridRPC middleware supervising different domain platforms, and 2) a specific OGF document describing some parts of implementation to achieve code portability.

5.1.3. Latest Releases

- November 14th 2011, DIET 2.8 release.
- June 16th 2011, DIET 2.7 release.
- March 7th 2011, DIET 2.6.1 release
- January 14th 2011, DIET 2.6 release

5.2. MUMPS

Participants: Maurice Brémond, Guillaume Joslin, Jean-Yves L’Excellent [correspondent], Mohamed Sid-Lakhdar, Bora Uçar.

MUMPS (for MUltifrontal Massively Parallel Solver, see http://graal.ens-lyon.fr/MUMPS ) is a software package for the solution of large sparse systems of linear equations. The development of MUMPS was initiated by the European project PARASOL (Esprit 4, LTR project 20160, 1996-1999), whose results and developments were public domain. Since then, research and developments have been supported by CERFACS, CNRS, ENS Lyon, INPT-ENSEEIHT-IRIT (main contributor), INRIA, and University of Bordeaux.

MUMPS implements a direct method, the multifrontal method, and is a parallel code capable of exploiting distributed-memory computers; its main originalities are its performance, its numerical robustness and the wide range of functionalities available.

The latest release is MUMPS 4.10.0 (May 2011). Its main new functionalities concern the determinant, the possibility to compute entries of the inverse of a sparse matrix and an option to discard factors. Some memory and performance improvements have also been obtained thanks to specific users’ testcases. This year, we have also worked on generic tools and scripts for experimentation, validation and performance study.


5.3. HLCMi

Participants: Julien Bigot, Cristian Klein, Christian Pérez [correspondent], Vincent Pichon.

HLCMi is an implementation of the HLCM component model defined during the PhD of Julien Bigot. HLCM is a generic extensible component model with respect to component implementations and interaction concerns. Moreover, HLCM is abstract; it is its specialization—such as HLCM/CCM—that define the primitive elements of the model, such as the primitive components and the primitive interactions.

HLCMi is making use of Model-driven Engineering (MDE) methodology to generate a concrete assembly from an high level description. It is based on the Eclipse Modeling Framework (EMF). HLCMi contains 700 Emfatic lines to describe its models and 7000 JAVA lines for utility and model transformation purposes. HLCMi is a general framework that supports several HLCM specialization: HLCM/CCM, HLCM/JAVA, HLCM/C++ (known as L2C) and HLCM/Charm++ (known as Gluon++).

5.4. BitDew

Participants: Gilles Fedak [correspondent], Haiwu He, Bing Tang, José Francisco Saray Villamizar, Mircea Moca, Lu Lu.
BitDEW is an open source middleware implementing a set of distributed services for large scale data management on Desktop Grids and Clouds. BitDEW relies on five abstractions to manage the data: i) replication indicates how many occurrences of a data should be available at the same time on the network, ii) fault-tolerance controls the policy in presence of hardware failures, iii) lifetime is an attribute absolute or relative to the existence of other data, which decides the life cycle of a data in the system, iv) affinity drives movement of data according to dependency rules, v) protocol gives the runtime environment hints about the protocol to distribute the data (http, ftp or bittorrent). Programmers define for every data these simple criteria, and let the BitDEW runtime environment manage operations of data creation, deletion, movement, replication, and fault-tolerance operation.

The current status of the software is the following: BitDEW is open source under the GPLv3 or Cecill licence at the user’s choice, 10 releases were produced in the last two years, and it has been downloaded approximatively 6000 times on the INRIA forge. Known users are Université Paris-XI, Université Paris-XIII, University of Florida, Cardiff University and University of Sfax. In term of support, the development of BitDew is partly funded by the INRIA ADT BitDew and by the ANR MapReduce projects. Thanks to this support, we have developed and released the first prototype of the MapReduce programming model for Desktop Grids on top of BitDew. In 2011, 5 versions of the software have been released, including the version 1.0.0 considered as the first stable release of BitDew. Our most current work focuses on providing reliable storage on top of hybrid distributed computing infrastructures.

5.5. XtremWeb

Participants: Gilles Fedak [correspondent], Haiwu He, Bing Tang, Simon Delamare.

XtremWeb is an open source software for Desktop Grid computing, jointly developed by INRIA and IN2P3. XtremWeb allows to build lightweight Desktop Grid by gathering the unused resources of Desktop Computers (CPU, storage, network). Its primary features permit multi-users, multi-applications and cross-domains deployments. XtremWeb turns a set of volatile resources spread over LAN or Internet into a runtime environment executing high througput applications.

XtremWeb is a highly programmable and customizable middleware which supports a wide range of applications (bag-of tasks, master/worker), computing requirements (data/CPU/network-intensive) and computing infrastructures (clusters, Desktop PCs, multi-Lan) in a manageable, scalable and secure fasion. Known users include LIFL, LIP, LIG, LRI (CS), LAL (physics Orsay), IBBMC (biology), Université Paris-XIII, Université de Guadeloupe, IFP (petroleum), EADS, CEA, University of Wisconsin Madison, University of Tsukuba (Japan), AIST (Australia), UCSD (USA), Université de Tunis, AlmerGrid (NL), Fundecyt (Spain), Hobai (China), HUST (China).

There are two branches of XtremWeb: XtremWeb-HEP is a production version developed by IN2P3. It features many security improvements such as X509 support which allows its usage within the EGEE context. XtremWeb-CH is a research version developed by HES-SO, Geneva, which aims at building an effective Peer-To-Peer system for CPU time consuming applications.

XtremWeb has been supported by national grants (ACI CGP2P) and by major European grants around Grid and Desktop Grid such as FP6 CoreGrid: European Network of Excellence, FP6 Grid4all, and more recently FP7 EDGeS : Enabling Desktop Grid for E-Science and FP7 EDGI: European Desktop Grid Initiative.

On going developments include: providing Quality-of-Service for Desktop Grids (SpeQuIoS), inclusion of the BitDew middleware to distribute data as well as inclusion of virtualization tenchnologies.
5. Software

5.1. Tools for cluster management and software development

Participant: Olivier Richard [correspondant].

The KA-Tools is a software suite developed by MESCAL for exploitation of clusters and grids. It uses a parallelization technique based on spanning trees with a recursive starting of programs on nodes. Industrial collaborations were carried out with Mandrake, BULL, HP and Microsoft.

5.1.1. KA-Deploy

KA-Deploy is an environment deployment toolkit that provides automated software installation and reconfiguration mechanisms for large clusters and light grids. The main contribution of KA-Deploy toolkit is the introduction of a simple idea, aiming to be a new trend in cluster and grid exploitation: letting users concurrently deploy computing environments tailored exactly to their experimental needs on different sets of nodes. To reach this goal KA-Deploy must cooperate with batch schedulers, like OAR, and use a parallel launcher like Taktuk (see below).

5.1.2. Taktuk

Taktuk is a tool to launch or deploy efficiently parallel applications on large clusters, and simple grids. Efficiency is obtained thanks to the overlap of all independent steps of the deployment. We have shown that this problem is equivalent to the well known problem of the single message broadcast. The performance gap between the cost of a network communication and of a remote execution call enables us to use a work stealing algorithm to realize a near-optimal schedule of remote execution calls. Currently, a complete rewriting based on a high level language (precisely Perl script language) is under progress. The aim is to provide a light and robust implementation. This development is lead by the MOAIS project-team.

5.2. OAR: Batch scheduler for clusters and grids

Participant: Olivier Richard [correspondant].

The OAR project focuses on robust and highly scalable batch scheduling for clusters and grids. Its main objectives are the validation of grid administration tools such as Taktuk, the development of new paradigms for grid scheduling and the experimentation of various scheduling algorithms and policies.

The grid development of OAR has already started with the integration of best effort jobs whose purpose is to take advantage of idle times of the resources. Managing such jobs requires a support of the whole system from the highest level (the scheduler has to know which tasks can be canceled) down to the lowest level (the execution layer has to be able to cancel awkward jobs). The OAR architecture is perfectly suited to such developments thanks to its highly modular architecture. Moreover, this development is used for the CiGri grid middleware project.

The OAR system can also be viewed as a platform for the experimentation of new scheduling algorithms. Current developments focus on the integration of theoretical batch scheduling results into the system so that they can be validated experimentally.

See also the web page http://oar.imag.fr.

5.3. FTA: Failure Trace Archive

Participant: Derrick Kondo [correspondant].
With the increasing functionality, scale, and complexity of distributed systems, resource failures are inevitable. While numerous models and algorithms for dealing with failures exist, the lack of public trace data sets and tools has prevented meaningful comparisons. To facilitate the design, validation, and comparison of fault-tolerant models and algorithms, we led the creation of the Failure Trace Archive (FTA), an on-line public repository of availability traces taken from diverse parallel and distributed systems.

While several archives exist, the FTA differs in several respects. First, it defines a standard format that facilitates the use and comparison of traces. Second, the archive contains traces in that format for over 20 diverse systems over a time span of 10 years. Third, it provides a public toolbox for failure trace interpretation, analysis, and modeling. The FTA was released in November 2009. It has received over 11,000 hits since then. The FTA has had national and international impact. Several published works have already cited and benefited from the traces and tools of the FTA. Simulation toolkits for distributed systems, such as SimGrid (CNRS, France) and GridSim (University of Melbourne, Australia), have incorporated the traces to allow for simulations with failures.

5.4. SimGrid: simulation of distributed applications
Participants: Arnaud Legrand [correspondant], Lucas Schnorr, Pierre Navarro, Sascha Hunold, Laurent Bobelin.

SimGrid is a toolkit that provides core functionalities for the simulation of distributed applications in heterogeneous distributed environments. The specific goal of the project is to facilitate research in the area of distributed and parallel application scheduling on distributed computing platforms ranging from simple network of workstations to Computational Grids.

We have released one new major version (3.6) of SimGrid (June 2011) and two minor versions (June and October 2011). These versions include our current work on visualization, analysis of large scale distributed systems, and extremely scalable simulation. See also the web page http://simgrid.gforge.inria.fr/.

5.5. TRIVA: interactive trace visualization
Participants: Lucas Schnorr [correspondant], Arnaud Legrand.

TRIVA is an open-source tool used to analyze traces (in the Pajé format) registered during the execution of parallel applications. The tool serves also as a sandbox for the development of new visualization techniques. Some features include: Temporal integration using dynamic time-intervals; Spatial aggregation through hierarchical traces; Scalable visual analysis with squarified treemaps; A Custom Graph Visualization.

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

5.6. \(\psi\) and \(\psi^2\): perfect simulation of Markov Chain stationary distributions
Participant: Jean-Marc Vincent [correspondant].

\(\psi\) and \(\psi^2\) are two software tools implementing perfect simulation of Markov Chain stationary distributions using coupling from the past. \(\psi\) starts from the transition kernel to derive the simulation program while \(\psi^2\) uses a monotone constructive definition of a Markov chain. They are available at http://www-id.imag.fr/Logiciels/psi/.
MOAIS Project-Team

5. Software

5.1. KAAPI

Participants: Thierry Gautier [correspondant], Vincent Danjean, Pierre Neyron.

KAAPI means Kernel for Adaptative, Asynchronous Parallel and Interactive programming. It is a C++ library that allows to execute multithreaded computation with data flow synchronization between threads. The library is able to schedule fine/medium size grain program on distributed machine. The data flow graph is dynamic (unfold at runtime). Target architectures are clusters of SMP machines. Main features are: * It is based on work-stealing algorithms; * It can run on various processors; * It can run on various architectures (clusters or grids); * It contains non-blocking and scalable algorithms.

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

- ACM: D.1.3
- License: CeCILL
- OS/Middleware: Unix (Linux, MacOSX, ...)
- Programming language: C/C++, Fortran

5.2. OAR

Participants: Pierre Neyron [correspondant MOAIS], Grégory Mounié.

OAR is a batch scheduler developed by Mescal team (correspondant: Olivier Richard). The MOAIS team develops the central automata and the scheduling module that includes successive evolutions and improvements of the policy. OAR is used to schedule jobs both on the CiGri (Grenoble region) and Grid5000 (France) grids. CiGri is a production grid that federates about 500 heterogeneous resources of various Grenoble laboratories to perform computations in physics. MOAIS has also developed the distributed authentication for access to Grid5000.

See also the web page http://oar.imag.fr.

5.3. SOFA

Participant: Bruno Raffin [correspondant].

SOFA is an Open Source framework primarily targeted at real-time simulation, with an emphasis on medical simulation. It is mostly intended for the research community to help develop newer algorithms, but can also be used as an efficient prototyping tool. Based on an advanced software architecture, it allows to: * create complex and evolving simulations by combining new algorithms with algorithms already included in SOFA; * modify most parameters of the simulation (deformable behavior, surface representation, solver, constraints, collision algorithm, etc.) by simply editing an xml file; * build complex models from simpler ones using a scene-graph description; * efficiently simulate the dynamics of interacting objects using abstract equation solvers; * reuse and easily compare a variety of available methods.

See also the web page http://www.sofa-framework.org/.

- ACM: J.3
- Programming language: C/C++

5.4. TakTuk - Adaptive large scale remote execution deployment

Participants: Guillaume Huard [correspondant], Pierre Neyron.
TakTuk is a tool for deploying remote execution commands to a potentially large set of remote nodes. It spreads itself using an adaptive algorithm and sets up an interconnection network to transport commands and perform I/Os multiplexing/demultiplexing. The TakTuk algorithms dynamically adapt to environment (machine performance and current load, network contention) by using a reactive algorithm that mix local parallelization and work distribution. Characteristics:

- adaptivity: efficient work distribution is achieved even on heterogeneous platforms thanks to an adaptive work-stealing algorithm
- scalability: TakTuk has been tested to perform large size deployments (hundreds of nodes), either on SMPs, regular clusters or clusters of SMPs
- portability: TakTuk is architecture independent (tested on x86, PPC, IA-64) and distinct instances can communicate whatever the machine they’re running on
- configurability: mechanics are configurable (deployment window size, timeouts, ...) and TakTuk outputs can be suppressed/formatted using I/O templates

Outstanding features:

- auto-propagation: the engine can spread its own code to remote nodes in order to deploy itself
- communication layer: nodes successfully deployed are numbered and perl scripts executed by TakTuk can send multicast communications to other nodes using this logical number
- information redirection: I/O and commands status are multiplexed from/to the root node. http://taktuk.gforge.inria.fr under GNU GPL licence.

5.5. KRASH - Kernel for Reproduction and Analysis of System Heterogeneity

Participants: Guillaume Huard [correspondant], Swann Perarnau.

KRASH is a tool for reproducible generation of system-level CPU load. This tool is intended for use in shared memory machines equipped with multiple CPU cores that are usually exploited concurrently by several users. The objective of KRASH is to enable parallel application developers to validate their resources use strategies on a partially loaded machine by replaying an observed load in concurrence with their application. To reach this objective, KRASH relies on a method for CPU load generation which behaves as realistically as possible: the resulting load is similar to the load that would be produced by concurrent processes run by other users. Nevertheless, contrary to a simple run of a CPU-intensive application, KRASH is not sensitive to system scheduling decisions. The main benefit brought by KRASH is this reproducibility: no matter how many processes are present in the system the load generated by our tool strictly respects a given load profile. This last characteristic proves to be hard to achieve using simple methods because the system scheduler is supposed to share the resources fairly among running processes. http://krash.ligforge.imag.fr under GNU GPL licence.

5.6. Cache Control

Participants: Guillaume Huard [correspondant], Swann Perarnau.

Cache Control is a Linux kernel module enabling user applications to restrict their memory allocations to a subset of the hardware memory cache. This module reserves and exports available physical memory as virtual devices that can be mmap’d to. It gives to calling processes physical memory using only a subset of the cache (similarly to page coloring). It actually creates cache partitions that can be used simultaneously by a process to control how much cache a data structure can use.
5. Software

5.1. ns-3

Participant: Daniel Camara [correspondant].

ns-3 is a discrete-event network simulator for Internet systems, targeted primarily for research and educational use. ns-3 is free software, licensed under the GNU GPLv2 license, and is publicly available for research, development, and use. ns-3 includes a solid event-driven simulation core as well as an object framework focused on simulation configuration and event tracing, a set of solid 802.11 MAC and PHY models, an IPv4, UDP, and TCP stack and support for nsc (integration of Linux and BSD TCP/IP network stacks).

See also the web page http://www.nsnam.org.

- Version: ns-3.7
- Keywords: networking event-driven simulation
- License: GPL (GPLv2)
- Type of human computer interaction: programmation C++/python, No GUI
- OS/Middleware: Linux, cygwin, osX
- Required library or software: standard C++ library: GPLv2
- Programming language: C++, python
- Documentation: doxygen

5.2. EphPub

Participants: Mohamed Ali Kaafar [correspondant], Claude Castelluccia.

EphPub (Ephemeral Publishing) (previously called EphCom) implements a novel key storage mechanism for time-bounded content, that relies on the caching mechanism of the Domain Name System (DNS). Features of EphPub include: EphPub exploits the fact that DNS servers temporarily cache the response to a recursive DNS query for potential further requests. EphPub provides higher security than Vanish, as it is immune to Sybil attacks. EphPub is easily deployable and does not require any additional infrastructure, such as Distributed Hash Tables. EphPub comes with high usability as it does not require users to install and execute any extra additional software. EphPub lets users define data lifetime with high granularity. We provide EphPub as an Android Application to provide ephemeral exchanged SMS, emails, etc. and as a Firefox or Thunderbird extensions so as to support ephemeral publication of any online document.

For more details about the different software products, see http://planete.inrialpes.fr/projects/ephemeral-publication/.

- Version: v0.1.2-beta
- ACM: K.4.1
- AMS: 94Axx
- Keywords: Ephemeral communications, Right to Forget, Future Internet Architecture, Privacy
- Software benefit: We provide a Firefox Extension that easily allows users to manage disappearing emails. We also provide a command-line tool to manage disappearing files.
- APP: Under APP deposit internal process
- License: GPL
- Type of human computer interaction: Firefox extension + Unix Console
- OS/Middleware: Firefox under any OS
- Required library or software: Python Ext
- Programming language: Python
- Documentation: No detailed documentation has been released so far. A detailed howto can be consulted however at: http://code.google.com/p/disappearingdata/source/browse/wiki/EphCOM_Firefox_Extension.wiki?r=77
5.3. Username Tester

Participants: Claude Castelluccia [correspondant], Mohamed Ali Kaafar, Daniele Perito.

Usernames are ubiquitous on the Internet. Almost every website uses them to identify its users and, by design, they are unique within each service. In web services that have millions or hundreds of millions of users, it might become difficult to find a username that has not already been taken. For instance, you might have experienced that a specific username you wanted was already taken. This phenomenon drives users to choose increasingly complex and unique usernames.

We built a tool to estimate how unique and linkable usernames are and made it available on this page for you to check. For example, according to our tool, “ladygaga” or “12345678” only carry 24 and 17 bits of entropy, respectively. They are therefore not likely to be unique on the Internet. On the other hand, usernames such as “pdjkwerl” or “yourejerky” carry about 40 bits of entropy and are therefore very good identifiers.

Type your username (for example “zorro1982” or “dan.perito”) to discover how unique it is. This tool can help you to select an username that has low entropy and can’t be used to track you on the Internet.

Alternatively, try typing two usernames separated by a space. The tool will give an estimation on whether the two usernames are linkable. The tool is accessible here: http://planete.inrialpes.fr/projects/how-unique-are-your-usernames/

5.4. DroidMonitor

Participants: Claude Castelluccia [correspondant], Mohamed Ali Kaafar, Anasthesia Fedane.

In nowadays world the technological progress evolves very quickly. There are more and more new devices, fully equipped with the latest innovations. The question is: do we adopt our main privacy concerns according to these new technologies as quickly as they grow and become widely available for us?…

We developed a novel tool, private data leakage monitoring tool, DroidMonitor. It aims to serve as an educational tool for regular Android Smartphones users to make them aware of existing privacy threats while they are using Location-Based Services. It can be downloaded here: http://planete.inrialpes.fr/android-privacy/

5.5. NEPI

Participants: Thierry Turletti [correspondant], Alina Quereilhac, Claudio Freire.

NEPI stands for Network Experimentation Programming Interface. NEPI implements a new experiment plane used to perform ns-3 simulations, planetlab and emulation experiments, and, more generally, any experimentation tool used for networking research. Its goal is to make it easier for experimenters to describe the network topology and the configuration parameters, to specify trace collection information, to deploy and monitor experiments, and, finally, collect experiment trace data into a central datastore. NEPI is a python API (with an implementation of that API) to perform all the above-mentioned tasks and allows users to access these features through a simple yet powerful graphical user interface called NEF. During the year 2011 we improved the robustness in the experiment control scheme, and we added support for new experimentation environments. We released and registered a second version of the NEPI software (IDDN.FR.001.06003.001.S.A.2010.000.10600). Details on the improvements made can be found in [48]. See also the web page http://nepihome.org.

- Version: 1.0
- ACM: C.2.2, C.2.4
- Keywords: networking experimentation
- License: GPL (2)
- Type of human computer interaction: python library, QT GUI
- OS/Middelware: Linux
- Programming language: python
5.6. Reference implementation for SFA Federation of experimental testbeds

Participants: Thierry Parmentelat [correspondant], Baris Metin, Julien Tribino.

We are codevelopping with Princeton University a reference implementation for the Testbed-Federation architecture known as SFA for Slice-based Federation Architecture. During 2011 we have focused on the maturation of the SFA codebase, with several objectives in mind, better interoperability between the PlanetLab world and the EmuLab, a more generic shelter that other testbeds can easily leverage in order to come up with their own SFA-compliant wrapper and support for ‘ reservable’ mode, which breaks the usual best-effort PlanetLab model. For more details about this contribution see section

See also the web page http://planet-lab.eu

- Version: myplc-5.0-rc26
- Keywords: networking testbed virtual machines
- License: Various Open Source Licences
- Type of human computer interaction: Web-UI, XMLRPC-based API, Qt-based graphical client
- OS/Middelware: Linux-Fedora
- Required library or software: Fedora-14 for the infrastructure side; the software comes with a complete software suite for the testbed nodes
- Programming languages: primarily python, C, ocaml
- Documentation: most crucial module plcapi is self-documented using a local format & related tool. See e.g. https://www.planet-lab.eu/db/doc/PLCAPI.php
- Codebase: http://git.onelab.eu

5.7. MultiCast Library Version 3

Participant: Vincent Roca [correspondant].

MultiCast Library Version 3 is an implementation of the ALC (Asynchronous Layered Coding) and NORM (NACK-Oriented Reliable Multicast Protocol) content delivery Protocols, and of the FLUTE/ALC file transfer application. This software is an implementation of the large scale content distribution protocols standardized by the RMT (Reliable Multicast Transport) IETF working group and adopted by several standardization organizations, in particular 3GPP for the MBMS (Multimedia Broadcast/Multicast Service), and DVB for the CBMS (Convergence of Broadcast and Mobile Services). Our software is used in operational, commercial environments, essentially in the satellite broadcasting area and for file delivery over the DVB-H system where FLUTE/ALC has become a key component. See http://planete-bcast.inrialpes.fr/ for more information.

5.8. OpenFEC.org: because open, free AL-FEC codes and codecs matter

Participants: Vincent Roca [correspondant], Jonathan Detchart [engineer], Ferdaouss Mattoussi [PhD student].

The goals of the OpenFEC.org http://openfec.org are:
- to share IPR-free, open, AL-FEC codes, to share high performance, ready-to-use, open, free, C-language, software codecs and to share versatile and automated performance evaluation environments.

This project can be useful to users who do not want to know the details of AL-FEC schemes but do need to use one of them in the software they are designing, or by users who want to test new codes or new encoding or decoding techniques, and who do know what they are doing and are looking for, or by users who need to do extensive tests for certain AL-FEC schemes in a given use-case, with a well defined channel model.
5.9. BitHoc

Participants: Chadi Barakat [correspondant], Thierry Turletti, Amir Krifa.

BitHoc (BitTorrent for wireless ad hoc networks) enables content sharing among spontaneous communities of mobile users using wireless multi-hop connections. It is an open source software developed under the GPLv3 licence. A first version of BitHoc has been made public. We want BitHoc to be the real testbed over which we evaluate our solutions for the support and optimization of file sharing in a mobile wireless environment where the existence of an infrastructure is not needed. The proposed BitHoc architecture includes two principal components: a membership management service and a content sharing service. In its current form it is composed of PDAs and smartphones equipped with WIFI adapters and Windows Mobile 6 operating system.

See also the web page http://planete.inria.fr/bithoc

- Version: 1.2
- Keywords: Tracker-less BitTorrent for mobile Ad Hoc networks
- License: GPL (GPLv3)
- Type of human computer interaction: Windows Mobile 6 GUI
- OS/Middleware: Windows Mobile 6
- Required library or software: OpenSSL (http://www.openssl.org/, GPL), C++ Sockets (http://www.alhem.net/Sockets/, GPL)
- Programming languages: C++, C#
- Documentation: doxygen

5.10. TICP

Participant: Chadi Barakat [correspondant].

TICP is a TCP-friendly reliable transport protocol to collect information from a large number of network entities. The protocol does not impose any constraint on the nature of the collected information: availability of network entities, statistics on hosts and routers, quality of reception in a multicast session, weather monitoring, etc. TICP ensures two main things: (i) the information to collect arrives entirely and correctly to the collector where it is stored and forwarded to upper layers, and (ii) the implosion at the collector and the congestion of the network are avoided by controlling the rate of sending probes. The congestion control part of TICP is designed with the main objective to be friendly with applications using TCP. Experimental results show that TICP can achieve better performance than using parallel TCP connections for the data collection. The code of TICP is available upon request, it is an open source software under the GPLv3 licence.

See also the web page http://planete.inria.fr/ticp/

- Version: 1.0
- Keywords: Information Collection, Congestion and Error Control
- License: GPL (GPLv3)
- Type of human computer interaction: XML file
- OS/Middleware: Linux/Unix
- Required library or software: C/C++ Sockets
- Programming languages: C/C++
- Documentation: Text
5.11. Experimentation Software

WisMon
WisMon is a Wireless Statistical Monitoring tool that generates real-time statistics from a unified list of packets, which come from possible different probes. This tool fulfills a gap on the wireless experimental field: it provides physical parameters on realtime for evaluation during the experiment, records the data for further processing and builds a single view of the whole wireless communication channel environment. WisMon is available as open source under the Cecill license, at http://planete.inria.fr/software/WisMon/.

WEX Toolbox
The Wireless Experimentation (WEX) Toolbox aims to set up, run and make easier the analysis of wireless experiments. It is a flexible and scalable open-source set of tools that covers all the experimentation steps, from the definition of the experiment scenario to the storage and analysis of results. Sources and binaries of the WEX Toolbox are available under the GPLv2 licence at https://twiki-sop.inria.fr/twiki/bin/view/Projets/Planete/WEXToolkit. WEX Toolbox includes the CrunchXML utility, which aims to make easier the running and the analysis of wireless experimentations. In a nutshell, it implements an efficient synchronization and merging algorithm, which takes XML (or PDML) input trace files generated by multiple probes, and stores only the packets fields that have been marked as relevant by the user in a MySQL database –original pcap traces should be first formatted in XML using wireshark. These operations are done in a smart way to balance the CPU resources between the central server (where the database is created) and the different probes (i.e., PC stations where the capture traces are located). CrunchXML is available under the GNU General Public License v2 at http://twiki-sop.inria.fr/twiki/bin/view/Projets/Planete/CrunchXML.

WiMAX ns-3
This simulation module for the ns-3 network simulator is based on the IEEE 802.16-2004 standard. It implements the PMP topology with TDD mode and aims to provide detailed and standard compliant implementation of the standard, supporting important features including QoS scheduling services, bandwidth management, uplink request/grant scheduling and the OFDM PHY layer. The module is available under the GNU General Public License at http://code.nsnam.org/iamine/ns-3-wimax. It will be included in the official 3.8v release of ns-3.

MonLab
Monitoring Lab is a platform for the emulation and monitoring of traffic in virtual ISP networks. It is supported by the FP7 ECODE project and is available for download at the web page of the tool http://planete.inria.fr/MonLab under the terms of the GPL licence. MonLab presents a new approach for the emulation of Internet traffic and for its monitoring across the different routers of the emulated ISP network. In its current version, the traffic is sampled at the packet level in each router of the platform, then monitored at the flow level. We put at the disposal of users real traffic emulation facilities coupled to a set of libraries and tools capable of Cisco NetFlow data export, collection and analysis. Our aim is to enable running and evaluating advanced applications for network wide traffic monitoring and optimization. The development of such applications is out of the scope of this research. We believe that the framework we are proposing can play a significant role in the systematic evaluation and experimentation of these applications’ algorithms. Among the direct candidates figure algorithms for traffic engineering and distributed anomaly detection. Furthermore, methods for placing monitors, sampling traffic, coordinating monitors, and inverting sampling traffic will find in our platform a valuable tool for experimentation.

MobiTrade
MobiTrade is the ns-3 and Android implementation of our solution in [41] for trading content between wireless devices. The application provides a utility driven trading system for efficient content dissemination on top of a disruption tolerant network. While simple tit-for-tat (TFT) mechanisms can force nodes to give one to get one, dealing with the inherent tendency of peers to
take much but give back little, they can quickly lead to deadlocks when some (or most) of interesting content must be somehow fetched across the network. To resolve this, MobiTrade proposes a trading mechanism that allows a node (merchant) to buy, store, and carry content for other nodes (its clients) so that it can later trade it for content it is personally interested in. To exploit this extra degree of freedom, MobiTrade nodes continuously profile the type of content requested and the collaboration level of encountered devices. An appropriate utility function is then used to collect an optimal inventory that maximizes the expected value of stored content for future encounters, matched to the observed mobility patterns, interest patterns, and collaboration levels of encountered nodes. See also http://planete.inria.fr/MobiTrade.
**RESO Project-Team**

5. **Software**

5.1. **CloudWeaver suite**

*Participants:* Paulo Gonçalves, Guilherme Koslovski, Fabienne Anhalt.

The following list of softwares, whose development was initiated at RESO, constitutes the main outcome of the research work delivered by Guilherme Koslovski (PhD, july 2011) [8] and Fabienne Anhalt (PhD, july 2011) [7]. These products are also part of the technological transfer to Lyatiss (headed by Pascale Vicat-Blanc); embedded in the *CloudWeaver Suite*, they implement the solutions for virtual resources orchestration and infrastructure services.

- **VXAlloc** Dynamic allocation of virtual resources (Patent INPI:10/01626, 2010, Lyatiss, INRIA, ENS Lyon)
- **VXCap** Partitioning of complex physicals infrastructures (Patent INPI:10/01624, 2010, Lyatiss, INRIA, ENS Lyon)
- **HiperNet** Automatic configuration of virtual networks, by programming virtual routers and configuring virtual links according to service requirements (APPcode: IDDN.FR.001.260010.000.S.P.2009.000.10700, 2009, LYaTiss, INRIA ENS Lyon)
- **VXDL parser** Interpretation and XML traduction of virtual infrastructures specifications (APPcode: IDDN.FR.001.260009.000.S.P.2009.000.10800)

Due to non disclosure agreement between INRIA and Lyatiss, access to these software is now submitted to patent restriction.

5.2. **Queueing-systems**

*Participant:* Thomas Begin.

This tool aims at providing a simple web based interface to promote the use of our proposed solutions to numerically solve classical queueing systems. It currently implements the solution to get the distribution for the number of customers along with customary performance parameters for a queue with multiple servers, general arrivals, exponential services and a possibly finite buffer, (i.e., $Ph//M//c/N$-like queue). The steady-state solution to this queue is based on a simple and stable recurrence [50] and was performed in collaboration with Pr. Brandwajn (UCSC). We will include new features and new models to this tool in the near future. Associated URL is: [http://queueing-systems.ens-lyon.fr](http://queueing-systems.ens-lyon.fr)

5.3. **ECOFEN simulation framework**

*Participants:* Anne-Cecile Orgerie, Laurent Lefevre.

The problem when evaluating new network architectures and protocols is that large testbed platforms are really expensive and difficult to manage. That is why we have designed ECOFEN whose user’s entries are the network topology and traffic. Based on configurable measurements of different network component (routers, switches, NICs, etc.), it provides the power consumption of the overall network including the end-hosts as well as the power consumption of each equipment over time. The ECOFEN simulator supports green network leverages such as Adaptive Link Rate and on/off. The aim of ECOFEN is to compute and expose the energy consumed by a network under a given traffic. Firstly based on NS2 and now developed on NS3, this simulator has been made in collaboration with Dino Lopez-Pacheco [29].
4. Software

4.1. AAC_tactics

Participants: Thomas Braibant, Damien Pous [correspondant].

AAC_tactics is a plugin for the Coq proof-assistant that implements new proof tactics for rewriting modulo associativity and commutativity. It is available at http://sardes.inrialpes.fr/~braibant/aac_tactics and as part of the Coq distribution.

- ACM: D.2.4 Software/Program Verification
- Keywords: Rewriting, rewriting modulo AC, proof tactics, proof assistant
- Software benefit: AAC_tactics provides novel efficient proof tactics for rewriting modulo associativity and commutativity.
- License: LGPL
- Type of human computer interaction: N/A
- OS/Middleware: Windows, Linux, MacOS X
- Programming language: Coq

4.2. ATBR

Participants: Thomas Braibant, Damien Pous [correspondant].

ATBR (Algebraic Tools for Binary Relations) is library for the Coq proof assistant that implements new proof tactics for reasoning with binary relations. Its main tactics implements a decision procedure for inequalities in Kleene algebras. It is available at http://sardes.inrialpes.fr/~braibant/atbr and as part of the Coq distribution contributed modules.

- ACM: D.2.4 Software/Program Verification
- Keywords: Binary relations, Kleene algebras, proof tactics, proof assistant
- Software benefit: ATBR provides new proof tactics for reasoning with binary relations.
- License: LGPL
- Type of human computer interaction: N/A
- OS/Middleware: Windows, Linux, MacOS X
- Programming language: Coq

4.3. MoKa

Participant: Sara Bouchenak [correspondant].
MoKÀ is a software framework for the modeling and capacity planning of distributed systems. It first provides a set of tools to build analytical models that describe the behavior of distributed computing systems, in terms of performance, availability, cost. The framework allows to include several model algorithms and to compare them regarding their accuracy and their efficiency. Furthermore, MoKÀ provides a set of tools to build capacity planning methods. A capacity planning method allows to find a distributed system configuration that guarantees given quality-of-service objectives. MoKÀ is able to include different capacity planning algorithms and to compare them regarding their efficiency and the optimality of their results. MoKÀ is available at: http://sardes.inrialpes.fr/research/moka.

- ACM: C.2.4 Distributed Systems, C.4 Performance of Systems, D.2.9 Management
- Keywords: Caching, multi-tier systems, consistency, performance
- Software benefit: a novel end-to-end caching protocol for multi-tier services.
- License: TBD
- Type of human computer interaction: command-line interface
- OS/Middleware: Windows, Linux, MacOS X
- Programming language: Java

4.4. ConSer

Participant: Sara Bouchenak [correspondant].

ConSER is a software framework for the modeling and the concurrency and admission control of servers systems. It implements a fluid-based model that exhibits the dynamics and behavior of a server system in terms of service performance and availability. ConSER implements various novel admission control laws for servers such as AM-C, PM-C, AA-PM-C and PA-AM-C. A control law produces the server concurrency level that allows to trade-off and meet given service level objectives. ConSER’s modeling and control laws algorithms are implemented following a proxy-based approach for more transparency.

- ACM: C.4 Performance of Systems; D.2.9 Management
- Keywords: System management, capacity planning, performance management
- Software benefit: MoKà provides modeling, capacity planning and performance management facilities for application server clusters. Thanks to its model-based capacity planning, MoKà is able to enforce service level objectives while minimizing the service cost.
- License: LGPL
- Type of human computer interaction: web interface
- OS/Middleware: Windows, Linux, MacOS X
- Programming language: Java, AspectJ

4.5. e-Caching

Participants: Dàmian Serrano, Sara Bouchenak [correspondant].

E-CACHING is a software framework for higher scalability of multi-tier Internet services through end-to-end caching of dynamic data. It provides a novel caching solution that allows to cache different types of data (e.g. Web content, database query results, etc.), at different locations of multi-tier Internet services. The framework allows to combine different caches and, thus, to provide higher scalability of Internet services. E-CACHING maintains the integrity of the cached data through novel distributed caching algorithms that guarantee the consistency of the underlying data.

- ACM: C.2.4 Distributed Systems, C.4 Performance of Systems
- Keywords: Caching, multi-tier systems, consistency, performance
- Software benefit: a novel end-to-end caching protocol for multi-tier services, consistency management, performance improvement.
- License: TBD
- Type of human computer interaction: command-line interface
- OS/Middleware: Windows, Linux, MacOS X
- Programming language: Java
4.6. MRB

Participants: Amit Sangroya, Dàmian Serrano, Sara Bouchenak [correspondant].

MRB is a software framework for benchmarking the performance and dependability of MapReduce distributed systems. It includes five benchmarks covering several application domains and a wide range of execution scenarios such as data-intensive vs. compute-intensive applications, or batch applications vs. interactive applications. MRB allows to characterize application workload, faultload and dataload, and it produces extensive performance and dependability statistics.

- ACM: C.2.4 Distributed Systems, C.4 Performance of Systems
- Keywords: Benchmark, performance, dependability, MapReduce, Hadoop, Cloud Computing
- Software benefit: the first performance and dependability benchmark suite for MapReduce systems.
- License: TBD
- Type of human computer interaction: GUI and command-line interface
- OS/Middleware: Windows, Linux, MacOS X
- Programming language: Java, Unix Shell scripts

4.7. BZR

Participants: Eric Rutten [correspondant], Gwenaël Delaval [POP ART team].

BZR is a reactive language, belonging to the synchronous languages family, whose main feature is to include discrete controller synthesis within its compilation. It is equipped with a behavioral contract mechanisms, where assumptions can be described, as well as an "enforce" property part: the semantics of the latter is that the property should be enforced by controlling the behaviour of the node equipped with the contract. This property will be enforced by an automatically built controller, which will act on free controllable variables given by the programmer.

BZR is now further developed with the Pop-Art team, where G. Delaval got a position. It has been designed and developed in the Sardes team in relation with the research topic on Model-based Control of Adaptive and Reconfigurable Systems. It is currently applied in different directions: component-based design and the Fractal framework; real-time control systems and the Orccad design environment; operating systems and administration loops in virtual machines; hardware and reconfigurable architecture (FPGAs).

See also the web page [http://bzr.inria.fr](http://bzr.inria.fr).

- ACM: D.3.3 [Programming Languages]: Language Constructs and Features—Control structures; C.3 [Special-purpose and Application-based Systems]: Real-time and embedded systems; D.2.2 [Software Engineering]: Design Tools and Techniques—Computer-aided software engineering, State diagrams; D.2.4 [Software Engineering]: Software / Program Verification—Formal methods, Programming by contract
- Keywords: Discrete controller synthesis, modularity, components, contracts, reactive systems, synchronous programming, adaptive and reconfigurable systems
- Software benefit: the first integration of discrete control synthesis in a compiler, making it usable at the level of the programming language.
- License: TBD
- Type of human computer interaction: programming language and command-line interface
- OS/Middleware: Linux
- Programming language: Caml, generates C or Java or Caml executable code
SWING Team

4. Software

4.1. Introduction

SWING develops several tools supporting its research like SOCLIB and Wiplan. Moreover, SWING is an active contributor to WSnet (http://wsnet.gforge.inria.fr/) a multi-hop wireless network discrete event simulator. WSnet was created in the ARES team and it is now supported by the D-NET team of INRIA Rhône-Alpes. SWING is one of the most important contributor for the design of protocol libraries in WSnet.

4.2. SOCLIB

Participant: Tanguy Risset [correspondant].

SocLib is a library of simulation models for virtual components (IP cores) for Systems on Chip. Many simulation models are under development, SocLib currently contains simulation models for processors (Mips, ARM), memories and network on chips (Spin and DSpin developed at LIP6 laboratory. SocLib permits to simulate at cycle accurate application running on embedded computing systems such as mobile phones. Swing use this platform to prototype design techniques either for embedded software or for hardware parts of signal processing applications.

See also the web page https://www.soclib.fr/trac/dev/wiki.

4.3. Wiplan

Participants: Jean-Marie Gorce [correspondant], Guillaume Villemaud, Meiling Luo, Dmitry Umansky, Tao Wang.

Wiplan is a software including an Indoor propagation engine and a wireless LAN optimization suite, which has been registered by INSA-Lyon. The heart of this software is the propagation simulation core relying on an original method, MR-FDPF (multi-resolution frequency domain ParFlow). The discrete ParFlow equations are translated in the Fourier domain providing a wide linear system, solved in two steps taking advantage of a multi-resolution approach. The first step computes a cell-based tree structure referred to as the pyramid. In the second phase, a radiating source is simulated, taking advantage of the pre-processed pyramidal structure. Using of a full-space discrete simulator instead of classical ray-tracing techniques is a challenge due to the inherent high computation re-quests. However, we have shown that the use of a multi-resolution approach allows the main computation load to be restricted to a pre-processing phase. Extensive works have been done to make predictions more realistic. The network planning and optimization suite is based on a multi-criteria model relying on a Tabu solver. The development of the wiplan software is a part of the european project iPLAN (IAPP-FP7 project).

See also the web page http://wiplan.citi.insa-lyon.fr.
ARTIS Project-Team

5. Software

5.1. Introduction

ARTIS insists on sharing the software that is developed for internal use. These are all listed in a dedicated section on the web site http://artis.imag.fr/Software.

5.2. libQGLViewer: a 3D visualization library

libQGLViewer is a library that provides tools to efficiently create new 3D viewers. Simple and common actions such as moving the camera with the mouse, saving snapshots or selecting objects are not available in standard APIs, and libQGLViewer fills this gap. It merges in a unified and complete framework the tools that every one used to develop individually. Creating a new 3D viewer now requires 20 lines of cut-pasted code and 5 minutes. libQGLViewer is distributed under the GPL licence since January 2003, and several hundreds of downloads are recorded each month.

5.3. PlantRad

Participant: Cyril Soler [contact].

PlantRad is a software program for computing solutions to the equation of light equilibrium in a complex scene including vegetation. The technology used is hierarchical radiosity with clustering and instantiation. Thanks to the latter, PlantRad is capable of treating scenes with a very high geometric complexity (up to millions of polygons) such as plants or any kind of vegetation scene where a high degree of approximate self-similarity permits a significant gain in memory requirements. Its main domains of applications are urban simulation, remote sensing simulation (See the collaboration with Noveltis, Toulouse) and plant growth simulation, as previously demonstrated during our collaboration with the LIAMA, Beijing.

5.4. High Quality Renderer

Participant: Cyril Soler [contact].

In the context of the European project RealReflect, the ARTIS team has developed the HQR software based on the photon mapping method which is capable of solving the light balance equation and of giving a high quality solution. Through a graphical user interface, it reads X3D scenes using the X3DToolKit package developed at ARTIS, it allows the user to tune several parameters, computes photon maps, and reconstructs information to obtain a high quality solution. HQR also accepts plugins which considerably eases the development of new algorithms for global illumination, those benefiting from the existing algorithms for handling materials, geometry and light sources. HQR is freely available for download.

5.5. MobiNet

Participants: Fabrice Neyret [contact], Joëlle Thollot.

The MobiNet software allows for the creation of simple applications such as video games, virtual physics experiments or pedagogical math illustrations. It relies on an intuitive graphical interface and language which allows the user to program a set of mobile objects (possibly through a network). It is available in public domain for Linux, Windows and MacOS, and originated in a collaboration with the EVASION project-team.

The main aim of MobiNet is to allow young students at high school level with no programming skills to experiment, with the notions they learn in math and physics, by modeling and simulating simple practical problems, and even simple video games. This platform has been massively used during the Grenoble INP "engineer weeks" since 2002: 150 senior high school pupils per year, doing a 3 hour practice. This work is partly funded by Grenoble INP. Various contacts are currently developed in the educational world. Besides "engineer weeks", several groups of "monitors" PhD students conducts experimentations based on MobiNet with a high scool class in the frame of the courses. Moreover, presentation in workshops and institutes are done, and a web site repository is maintained.

5.6. Freestyle

Freestyle is a software for Non-Photorealistic Line Drawing rendering from 3D scenes. It is designed as a programmable interface to allow maximum control over the style of the final drawing: the user "programs" how the silhouettes and other feature lines from the 3D model should be turned into stylized strokes using a set of programmable operators dedicated to style description. This programmable approach, inspired by the shading languages available in photorealistic renderers such as Pixar’s RenderMan, overcomes the limitations of integrated software with access to a limited number of parameters and permits the design of an infinite variety of rich and complex styles. The system currently focuses on pure line drawing as a first step. The style description language is Python augmented with our set of operators. Freestyle was developed in the framework of a research project dedicated to the study of stylized line drawing rendering from 3D scenes. This research has lead to two publications [31], [30].

In 2008, Freestyle get a new life, completely outside ARTIS or INRIA: it was the basis of one of the 6 Google Summer of Code projects awarded to the Blender Foundation 4! The goal of the project was to integrate Freestyle to the well known free 3D modeler Blender, as its standard NPR line-drawing renderer. Maxime Curioni (under the mentoring of Jean-Luc Peurière from the Blender Foundation), is currently making the integration. First beta versions are publicly available, and tested by enthusiasts around the web.

5.7. Diffusion Curves

Participant: Joëlle Thollot [contact].

---

4 http://www.blender.org/
We provide an implementation of the vector drawing tool described in the 2008 Diffusion Curves Siggraph paper. This prototype is composed of the Windows binary, along with the required shader programs (ie. in source code). The software is available for download ⁵ for free, for non-commercial research purposes.

5.8. TiffIO: Qt 3 binding for TIFF images

Participant: Jean-Dominique Gascuel [contact].

TiffIO is a plug-in that add TIFF images read/write capabilities to all Qt3 and Qt4 applications using the refernce QImage class. TiffIO come with a self-test suite, and have been compiled and used successfully on a wide variety of systems, compilers and Qt version combination. A demo application enables to quickly test image loading and viewing on any platform. All TIFF operations are based on libtiff 3.8.0, this plugin is just a wrapper that enable to use it transparently from the QImage class, and the architecture defined by Qt.

TiffIO has been downloaded by a large number of developer, and integrated in a variety of commercial or internal tools, such as by Pixar. TiffIO is freely available for download ⁶.

5.9. VRender: vector figures

Participant: Cyril Soler [contact].

The VRender library is a simple tool to render the content of an OpenGL window to a vectorial device such as Postscript, XFig, and soon SVG. The main usage of such a library is to make clean vectorial drawings for publications, books, etc.

In practice, VRender replaces the z-buffer based hidden surface removal of OpenGL by sorting the geometric primitives so that they can be rendered in a back-to-front order, possibly cutting them into pieces to solve cycles.

VRender is also responsible for the vectorial snapshot feature of the QGLViewer library. VRender is released under the LGPL licence and is freely available for download ⁷.

⁵ http://artis.imag.fr/Publications/2008/OBW/BTS08
⁶ http://artis.imag.fr/Software/TiffIO
⁷ http://artis.imag.fr/Software/VRender
5.10. ProLand

**Participants:** Fabrice Neyret [contact], Eric Bruneton.

Proland (for procedural landscape) is a software platform originally developed at the Evasion team-project by Eric Bruneton, and currently funded by the ANR-JCJC SimOne. The goal of this platform is the real-time quality rendering and editing of large landscapes. All features can work with planet-sized terrains, for all viewpoints from ground to space. Most of the work published by Eric Bruneton and Fabrice Neyret has been done within Proland, and a large part has been integrated in the main branch. Several licences have been transfered to companies. A free software version is about to be distributed. Eric Bruneton was hired by Google-Zürich in september 2011, but will be able to keep some participation in the project.

5.11. GigaVoxel

**Participants:** Fabrice Neyret [contact], Morgan Armand, Eric Bruneton, Cyril Crassin, Pascal Guehl, Eric Heitz.

Gigavoxel is a software platform initiated from the PhD work of Cyril Crassin, and currently funded by the ANR CONTINT RTIGE. The goal of this platform is the real-time rendering of very large very detailed scenes. Performances permit showing details over deep zooms and walk through very crowdy scenes (which are rigid, for the moment). The principle is GPU ray-tracing of volumetric-encoded multiscale data with minimal just-in time generation of data (accounting visibility and needed resolution) kept in a cache on GPU. The representation eases the cheap management of soft shadows, depth of field, anti-aliasing and geometric LOD. Beside the representation, data management and base rendering algorithm themself, we also worked on realtime light transport, and on quality prefiltering of complex data. This work led to numerous publications ([16], [22], [23]). Several licences have been sold to companies. A free software version is about to be distributed.
4. Software

4.1. PROTEUS

Participants: Amaury Nègre, Juan Lahera-Perez.

This toolkit offers a automatic mobile robot driver, some sensors drivers (sensors as Sick laser, GPS, motion tracker, mono or stereo camera), and a 3D Simulator.

The latest developments have been focuses on the robotics simulator. This simulator is based on the simulation and 3D rendering engine "mgEngine" (http://mgengine.sourceforge.net/) embedded with the physics engine "bullets physics" (http://bulletphysics.org) for realistic robot dynamic simulation. We also worked on the interface with the robotics middleware "ROS" (http://www.ros.org) in order to offer interoperability with many robotics applications. This software is developed in C++ and the simulator operates with the Lua scripting language.

The simulation software is used in the ANR Proteus (http://www.anr-proteus.fr), as a simulation engine for the PROTEUS Toolkit.

Figure 1. Screenshot of the Mobile Robot Simulator. Simulation of a Cycab robot in the "Pavin" environment provided by the LASMEA.

- Version: 2.0
- APP:IDDN.FR.001.510040.000.S.P.2005.000.10000
- Programming language: C/C++, Lua

4.2. AROSDYN

ArosDyn (http://arosdyn.gforge.inria.fr/) is a system which integrates our recently developed techniques to provide a real-time collision risk estimation in a dynamic environment. The main features of this software are:

1. The deliberated design provides high maintainability, scalability, and reusability of the models and algorithms.
2. The software has a user interface (UI) which is user-friendly.
3. The software facilitates the parameter tuning of the models.
4. It uses the GPU to accelerate the computation.
5. Working together with the Hugr middleware (http://gforge.inria.fr/projects/cycabtk), it can run on our experimental vehicle in real-time.

The software is developed in C/C++ in Linux and its architecture is shown in Fig. 2.

![Figure 2. Architecture of ArosdynTestSuite software](image)

In this example, we demonstrate a typical sensor fusion application. We retrieve the raw data from the Hugr middleware and store them in individual sensor objects. Then, by using this framework, we integrate the IBEO Bayesian Occupancy Filter (BOF) sensor model, the stereo sensor processor model, the stereo BOF sensor model, and the BOF model together. Finally, different aspects of the computational results are visualized in several viewers. At the same time, all the parameters used by the algorithms can be tuned online.

Several windows of this application are shown in Fig. 3. Here we demonstrate the main window, the 2D viewer of the stereo camera and the lidar, the disparity map of the stereo vision and the compounded BOF grid which is the result of the sensor fusion.

Another important property of this software is a large part of the computation task executed on GPU. As the processing of stereo image and the computation in the BOF can be highly parallelized, we run these tasks on the GPU to improve the time performance, as shown in Fig. 4. In this way, the software can work in real-time.
Figure 3. Windows of the ArosdynTestSuite software
The GPU calculation is based on CUDA library and is carried out in an independent thread. The schematic graph of the GPU computational thread is shown in Fig. 5.

Furthermore, thanks to the deliberated design of the software, we can easily add new models to it and let them work together. The fast detection and tracking algorithm (FCTA) and the Gaussian process based collision assessment algorithm are added into this framework.

4.3. Bayesian Occupancy Filter


The BOF toolbox is a C++ library that implements the Bayesian Occupancy Filter. It is often used for modelling dynamic environments. It contains the relevant functions for performing bayesian filtering in grid spaces. The output from the BOF toolbox are the estimated probability distributions of each cell’s occupancy and velocity. Some basic sensor models such as the laser scanner sensor model or Gaussian sensor model for gridded spaces are also included in the BOF toolbox. The sensor models and BOF mechanism in the BOF toolbox provides the necessary tools for modelling dynamic environments in most robotic applications. This toolbox is patented under two patents: “Procédé d’assistance à la conduite d’un véhicule et dispositif associé”
n. 0552735 (9 September 2005) and “Procédé d’assistance à la conduite d’un véhicule et dispositif associé amélioré” n. 0552736 (9 September 2005) and commercialized by ProBayes.

- Version: 1
- Programming language: C/C++

4.4. PROBT

People involved: Juan-Manuel Ahuactzin, Kamel Mekhnacha, Pierre Bessière, Emmanuel Mazer, Manuel Yguel, Christian Laugier.

ProBT is both available as a commercial product (ProBAYES.com) and as a free library for public research and academic purposes (http://emotion.inrialpes.fr/BP/spip.php?rubrique6). Formerly known as OPL, ProBT is a C++ library for developing efficient Bayesian software. It is available for Linux, Unix, PC Windows (Visual C++), MacOS9, MacOSX and Irix systems. The ProBT library (http://www.probayes.com/) has two main components: (i) a friendly Application Program Interface (API) for building Bayesian models, and (ii) a high-performance Bayesian Inference Engine (BIE) allowing to execute all the probability calculus in exact or approximate way. ProBT is now commercialized by our start-up Probayes; it represents the main Bayesian programming tool of the e-Motion project-team, and it is currently used in a variety of external projects both in the academic and industrial field (e.g. for the European project BACS and for some industrial applications such as Toyota or Denso future driving assistance systems).
5. Software

5.1. MyCorporisFabrica

Participants: Ali-Hamadi Dicko, François Faure, Olivier Palombi.

My Corporis Fabrica (MyCF) is an anatomical knowledge database (see fig. 1). During 2011, we have added new anatomical entities and improved some parts of FMA (Foundational Model of Anatomy). The FMA’s license is now under Creative Commons licenses (CC-by: Licensees may copy, distribute, display and perform the work and make derivative works based on it only if they give the author or licensor the credits in the manner specified by these). The license of MyCF is not yet defined. Our new contribution this year, is the creation of a brand new ontology about human functions. Based on the International Classification of Functioning, Disability and Health, also known as ICF, we have organized human functions through a tree of 4330 items. A original journal paper must be submitted soon. MyCF browser is now available on line: http://www.mycorporisfabrica.org/. The MyCF’s generic programming framework can be used for other domains. The link with semantic and 3D models matches research activities of IMAGINE towards interactive digital creation media. Anatomy can be seen as a study case.

![Figure 1](image.png)

Figure 1. My Corporis Fabrica is an anatomical knowledge database developed in our team.

5.2. SOFA

Participants: Guillaume Bousquet, Ali Hamadi Dicko, François Faure, François Jourdes.
SOFA is a C++ library primarily targeted at medical simulation research. Based on an advanced software architecture, it allows to (1) create complex and evolving simulations by combining new algorithms with algorithms already included in SOFA; (2) modify most parameters of the simulation – deformable behavior, surface representation, solver, constraints, collision algorithm, etc. – by simply editing an XML file; (3) build complex models from simpler ones using a scene-graph description; (4) efficiently simulate the dynamics of interacting objects using abstract equation solvers; and (5) reuse and easily compare a variety of available methods. The GPU capabilities of SOFA have been demonstrated at a SIGGRAPH talk [16] (see fig. 2) and presented in a book chapter [27].

SOFA is currently used by company Digital Trainers to develop basic skill endoscopic simulators. A start-up company based on SOFA, InSimo, is being created in the Strasbourg IHU, and is expected to start in first semester 2012.

**Figure 2. GPU methods in SOFA for detailed deformable objects at interactive rates.**

### 5.3. AESTEM Studio
**Participants:** Adrien Bernhardt, Marie-Paule Cani, Maxime Quiblier.

AESTEM Studio is dedicated to free form shape modeling through interactive sketching and sculpting gestures. The goal is to provide a very intuitive way to create 3D shapes, as easy to use for the general public as roughly sketching a shape or modeling it with a piece of clay. This software is developed in the framework of a research contract with the company Axiatec. It enables to create a 3D shape by successively painting in 2D and smoothly blending different components: the painting step takes place at different scales and from different viewing angles. 3D is inferred from a 2D painted region by using an isotropic implicit surface along the skeleton of the region. Then, implicit blending, restricted to the intersection areas, is computed to connect the new component with the existing ones. This relies on our researches on free-form sketch-based modeling using geometric skeletons and on convolution surfaces. Our prototype is written in C++. It uses the Ogre open-source library and our new library Convol dedicated to convolution surfaces. Future extensions will include the combination of sketching with modeling gestures related to clay sculpting, such as deforming a shape through pulling, pushing, bending or twisting gestures.

### 5.4. Convol
**Participants:** Marie-Paule Cani, Maxime Quiblier, Cédric Zanni.
Convol is a new C++ library we develop for easing our work on implicit surfaces – and more particularly on the sub-class of convolution surfaces. It enables us to make our latest research results soon available to the rest of the group and easily usable in our industrial partnerships. Convol incorporates all the necessary material for constructive implicit modeling: skeleton-based distance and convolution primitives, with closed form solution for the field values and gradient whenever possible; a variety of blending operators; and several methods for tessellating an implicit surface into a mesh, and for refining the later in highly curved regions. This development is funded by INRIA as support to our research group.
EXMO Project-Team

5. Software

Exmo’s work can be implemented in software: in particular, we have developed an API for expressing ontology alignment (§ 5.1) and a library of ontology distances and similarities OntoSim (§ 5.2).

5.1. Alignment API: manipulating ontology alignments

Participants: Jérôme Euzenat [Contact], Jérôme David, Cássia Trojahn dos Santos.

We have designed a format for expressing alignments in a uniform way [ ]. The goal of this format is to be able to share available alignments on the web. It should help systems using alignments, e.g., mergers, translators, to take advantage of any alignment algorithm and it will help alignment algorithms to be used in many different tasks. This format is expressed in RDF, so it is freely extensible, and has been defined by a DTD (for RDF/XML), an OWL ontology and an RDF Schema.

The API itself [ 3 ] is a Java description of tools for accessing the common format. It defines five main interfaces (OntologyNetwork, Alignment, Cell, Relation and Evaluator) and proposes the following services:

- Storing, finding, and sharing alignments;
- Piping matching algorithms (improving an existing alignment);
- Manipulating alignments (thresholding and hardening);
- Generating processing output (transformations, axioms, rules);
- Comparing alignments.

We provide an implementation for this API which can be used for producing transformations, rules or bridge axioms independently from the algorithm which produced the alignment. The proposed implementation features:

- a base implementation of the interfaces with all useful facilities;
- a library of sample matchers;
- a library of renderers (XSLT, SWRL, OWL, C-OWL, SEKT mapping language);
- a library of evaluators (various generalisation of precision/recall, precision/recall graphs);
- a library of wrapper for several ontology API;
- a parser for the format.

To instantiate the API, it is sufficient to refine the base implementation by implementing the align() method. Doing so, the new implementation will benefit from all the services already implemented in the base implementation.

We have developed on top of the Alignment API an Alignment server that can be used by remote clients for matching ontologies and for storing and sharing alignments. It is developed as an extensible platform which allows to plug-in new interfaces. The Alignment server can be accessed through HTML, web service (SOAP and REST) and agent communication interfaces.

This year, within the SEALS project (see § 8.2.1), we have developed a flexible test generation framework within the Alignment API which allows for generating new evaluation datasets [ 12 ].

The Alignment API is used in the Ontology Alignment Evaluation Initiative data and result processing. It is also used by more than 30 other teams worldwide.

The Alignment API is freely available since december 2003 under the LGPL licence at http://alignapi.gforge.inria.fr
5.2. The OntoSim library

Participants: Jérôme David [Contact], Jérôme Euzenat.

OntoSim is a library offering similarity and distance measures between ontology entities as well as between ontology themselves. It materialises our work towards better ontology proximity measures.

There are many reasons for measuring a distance between ontologies. For example, in semantic social networks, when a peer looks for a particular information, it could be more appropriate to send queries to peers having closer ontologies because it will be easier to translate them and it is more likely that such a peer have the information of interest. OntoSim provides a framework for designing various kinds of similarities. In particular, we differentiate similarities in the ontology space from those in the alignment space. The latter ones make use of available alignments in an ontology network while the former only rely on ontology data. OntoSim is provided with 4 entity measures which can be combined using various aggregation schemes (average linkage, Hausdorff, maximum weight coupling, etc.), 2 kinds of vector space measures (boolean and TF.IDF), and 4 alignment space measures. In addition, the framework embeds external similarity libraries which can be combined to our owns.

This year, we have implemented the measures of agreement and disagreement between ontologies recently proposed by Mathieu d’Aquin (Open university).

OntoSim is based on an ontology interface allowing for using ontology parsed with different APIs.

OntoSim is written in Java and is available under the LGPL license at http://ontosim.gforge.inria.fr .
LEAR Project-Team

5. Software

5.1. Face recognition

Participants: Jakob Verbeek [correspondant], Guillaume Fortier.

In a collaboration with Technosens (a start-up based in Grenoble) we are developing an efficient face recognition library. During 18 months Guillaume Fortier, financed by INRIA’s technology transfer program, streamlines code developed by different team members on various platforms. This encompasses detection of characteristic points on the face (eyes, nose, mouth), computing appearance features on these points, and learning metrics on the face descriptors that are useful for face verification (faces of the same person are close, faces of different people are far away). The code will be ported to run in real-time on the mini-pc system of Technosens that implements advanced user interfaces to TV-top videophone systems.

5.2. Large-scale image search

Participants: Matthijs Douze [correspondant], Mohamed Ayari, Cordelia Schmid.

LEAR’s image search demonstration was extended to 100M images. The image dataset was provided by Exalead. Search at this scale is possible due to the Fisher vector representation and the pqcodes software. The search time on a single core is about 250 ms.

In collaboration with Hervé Jégou, from the INRIA Texmex team, we stabilized and improved the pqcodes software package. The software was extended to implement matrix multiplications in the PQ-compressed domain. A non-exclusive license on pqcodes was sold to Technicolor. Another agreement is under negotiation with Morpho (a company owned by Safran).

LEAR’s implementation of the Fisher descriptor was improved in several ways. A new method to train the GMM was developed and the computation time of second-order derivatives (w.r.t. $\sigma$) was significantly reduced. Furthermore, the extraction of dense SIFT descriptors was improved in quality and speed.

5.3. Video descriptors

Participants: Heng Wang, Cordelia Schmid.

We have developed and made on-line available software for video description based on dense trajectories and motion boundary histograms [18]. The trajectories capture the local motion information of the video. A state-of-the-art optical flow algorithm enables a robust and efficient extraction of the dense trajectories. Descriptors are aligned with the trajectories and based on motion boundary histograms (MBH) which are robust to camera motion.
5. Software

5.1. Platforms

5.1.1. The Grimage platform

The Grimage platform is an experimental multi-camera platform dedicated to spatio-temporal modeling including immersive and interactive applications. It hosts a multiple-camera system connected to a PC cluster, as well as visualization facilities including head mounted displays. This platform is shared by several research groups, most prominently MOAIS, MORPHEO and PERCEPTION. In particular, Grimage allows challenging real-time immersive applications based on computer vision and interactions between real and virtual objects, Figure 1.

5.1.2. Virtualization Gate

Vgate is an immersive environment that allows full-body immersion and interaction with virtual worlds. It is a joint initiative of computer scientists from computer vision, parallel computing and computer graphics from several research groups at INRIA Grenoble Rhône-Alpes, and in collaboration with the company 4D View Solutions. The MORPHEO team is leading this project.

Figure 1. Platforms: on the left the Grimage acquisition; on the right the vgate immersive environment.

5.1.3. Multicamera platform for video analysis of mice behavior

This project is a follow-up of the experimental set-up developed for a CNES project with Mathieu Beraneck from the CESeM laboratory (centre for the study of sensorimotor control, CNRS UMR 8194) at the Paris-Descartes University. The goal of this project was to analyze the 3D body postures of mice with various vestibular deficiencies in low gravity condition (3D posturography) during a parabolic flight campaign. The set-up has been now adapted for new experiments on motor-control disorders for other mice models. This experimental platform is currently under development for a broader deployment for high throughput phenotyping with the technology transfer project ETHOMICE. This project involves a closed relationship with the CESeM laboratory and the European Mouse Clinical Institute in Strasbourg (Institut Clinique de la Souris, ICS).
5.2. Software packages

5.2.1. LucyViewer

Lucy Viewer [http://4drepository.inrialpes.fr/lucy_viewer/](http://4drepository.inrialpes.fr/lucy_viewer/) is an interactive viewing software for 4D models, i.e., dynamic three-dimensional scenes that evolve over time. Each 4D model is a sequence of meshes with associated texture information, in terms of images captured from multiple cameras at each frame. Such data is available from various websites over the world, including the 4D repository website hosted by INRIA Grenoble [http://4drepository.inrialpes.fr/](http://4drepository.inrialpes.fr/). The software was developed in the context of the European project iGlance, and is available as open-source software under the GNU LGP Licence.

5.3. Databases

5.3.1. 4D repository ([http://4drepository.inrialpes.fr/](http://4drepository.inrialpes.fr/))

This website hosts dynamic mesh sequences reconstructed from images captured using a multi-camera setup. Such mesh-sequences offer a new promising vision of virtual reality, by capturing real actors and their interactions. The texture information is trivially mapped to the reconstructed geometry, by back-projecting from the images. These sequences can be seen from arbitrary viewing angles as the user navigates in 4D (3D geometry + time). Different sequences of human / non-human interaction can be browsed and downloaded from the data section. A software to visualize and navigate these sequences is also available for download.
5. Software

5.1. Mixed camera platform

We started to develop a multiple camera platform composed of both high-definition color cameras and low-resolution depth cameras. This platform combines the advantages of the two camera types. On one side, depth (time-of-flight) cameras provide relatively accurate 3D scene information. On the other side, color cameras provide information allowing for high-quality rendering. The software package developed during the year 2011 contains the calibration of TOF cameras, alignment between TOF and color cameras, and image-based rendering. These software developments are performed in collaboration with the Samsung Advanced Institute of Technology. The multi-camera platform and the basic software modules are products of 4D Views Solutions SAS, a start-up company issued from the PERCEPTION group.

5.2. Audiovisual robot head

We have developed two audiovisual (AV) robot heads: the POPEYE head and the NAO stereo head. Both are equipped with a binocular vision system and four microphones. The software modules comprise stereo matching and reconstruction, sound-source localization and audio-visual fusion. POPEYE has been developed within the European project POP (http://perception.inrialpes.fr/POP) in collaboration with the project-team MISTIS and with two other POP partners: the Speech and Hearing group of the University of Sheffield and the Institute for Systems and Robotics of the University of Coimbra. The NAO stereo head is being developed under the European project HUMA VIPS (http://humavips.inrialpes.fr) in collaboration with Aldebaran Robotics (which manufactures the humanoid robot NAO) and with the University of Bielefeld, the Czech Technical Institute, and IDIAP. The software modules that we develop are compatible with both these robot heads.
5. Software

5.1. OMiSCID Middleware for Distributed Multi-Modal Perception

Participants: Patrick Reignier, Dominique Vaufreydaz [correspondant], Amaury Negre, Remi Barraquand.

OMiSCID is new lightweight middleware for dynamic integration of perceptual services in interactive environments. This middleware abstracts network communications and provides service introspection and discovery using DNS-SD ([DNS-based Service Discovery](#)). Services can declare simplex or duplex communication channels and variables. The middleware supports the low-latency, high-bandwidth communications required in interactive perceptual applications. It is designed to allow independently developed perceptual components to be integrated to construct user services. Thus our system has been designed to be cross-language, cross-platform, and easy to learn. It provides low latency communications suitable for audio and visual perception for interactive services.

OMiSCID has been designed to be easy to learn in order to stimulate software reuse in research teams and is revealing to have a high adoption rate. To maximize this adoption and have it usable in projects involving external partners, the OMiSCID middleware has been released under an open source licence. To maximize its target audience, OMiSCID is available from a wide variety of programming languages: C++, Java, Python and Matlab. A website containing informations and documentations about OMiSCID has been set up to improve the visibility and promote the use of this middleware.

The OMiSCID graphical user interface (GUI) is an extensible graphical application that facilitates analysis and debugging of service oriented applications. The core functionality of this GUI is to list running services, their communication channels and their variables. This GUI is highly extensible and many modules (i.e. plugins) have been created by different members of the team: figure 4 shows an example of some of these modules. OMiSCID GUI is based on the Netbeans platform and thus inherits from its dynamic installation and update of modules.

5.2. 3D Bayesian Tracker

Participants: James Crowley [correspondant], Amaury Negre, Lukas Rummelhard.

The 2DBT and 3DBT tracking systems are autonomic perceptual components originally created for the IST CAVIAR project and the IST CHIL projects. Both systems are autonomous perceptual components managed by a autonomic supervisor. The Autonomic supervisor provides self monitoring, self repair, self configuration, auto-regulation of parameters and self-description.

The INRIA 3D Bayesian body tracker is used to detect, locate and track multiple 3D entities in real time. It is configured and optimized for detecting and tracking people within rooms using multiple calibrated cameras. The system currently uses corner mounted cartesian cameras, ceiling mounted cameras with wide angle lenses and panoramic cameras placed on tables. Cameras may be connected and disconnected while the component is running, but they must be pre-calibrated to a common room reference frame. We are currently experimenting with techniques for Bayesian estimation of camera parameters for auto-calibration.

This perceptual component can be configured to monitor and track the activity within a smart space. The tracker receives its observations from 2D detection process that can use any available pixel level detection algorithm. The tracker currently integrates information from adaptive background subtraction, motion detection, skin color detection, and local appearance using scale normalised Gaussian derivatives. A common scenario is to use the motion to detect and initialise tracking, adaptive background subtraction to track 3D bodies, and skin color to track hands and faces. Cameras may be connected dynamically.
Figure 4. OMiSCID GUI showing a list of running services and some modules for service interconnections, variable plotting, live video stream display and variable control
Figure 5. The 3D Bayesian tracker integrates observations from multiple sensors
This work is currently supported by ICT Labs thematic actions on Smart Spaces and Smart Energy systems.

The original system 3DBT has been declared with the APP "Agence pour la Protection des Programmes" under the Interdeposit Digital number IDDN.FR.001.490023.000.S.P.2006.000.10000. A revised declaration for the latest version of the system is currently being prepared.

5.3. Stereo Viewfinder

Participants: Frédéric Devernay [correspondant], Elise Mansilla, Loic Lefort, Sergi Pujades.

This software has been filed with the APP "Agence pour la Protection des Programmes" under the Interdeposit Digital number IDDN.FR.001.370083.000.S.P.2007.000.10000

5.4. Tracking Focus of Attention for Large Screen Interaction

Participants: Claudine Combe, John Alexandre Ruiz Hernandez, Varun Jain, James Crowley [correspondant].

Large multi-touch screens may potentially provide a revolution in the way people can interact with information in public spaces. Technologies now exist to allow inexpensive interactive displays to be installed in shopping areas, subways and urban areas. Thesis displays can provide location aware access to information including maps and navigation guidance, information about local businesses and and commercial activities. While location information is an important component of a users context, information about the age and gender of a user, as well as information about the number of users present can greatly enhance the value of such interaction for both the user and for local commerce and other activities.

The objective of this task is to leverage recent technological advances in real time face detection developed for cell phones and mobile computing to provide a low-cost real time visual sensor for observing users of large multi-touch interactive displays installed in public spaces. The initial requirements for this system were expressed by the recent INRIA start-up HiLabs, created in 2008. By the end of 2010, HiLabs had installed over 100 interactive displays in public spaces, mostly in the form of interactive shop windows for travel agents, real-estate agents and banks. HiLabs customers indicated a potential important gain in market if such displays could be made aware of the number, gender and age of users.

The software developed for this activity builds on face detections software that has recently been developed by INRIA for the French OSEO project MinImage. MinImage was a five year, multi-million euro project to develop next generation technologies for integrated digital imaging devices to be used in cellphones, mobile and lap-top computing devices, and digital cameras, that has begun in February of 2007. The project scope included research on new forms of retinas, integrated optics, image formation and embedded image processing. INRIA was responsible for embedded algorithms for real time applications of computer vision.

Within MinImage, INRIA developped embedded image analysis algorithms using image descriptors that are invariant to position, orientation and scale and robust to changes in viewing angle and illumination intensity. INRIA proposed use of a simple hardware circuit to compute a scale invariant Gaussian pyramid as images acquired by the retina. Sums and differences of image samples from the pyramid provide invariant image descriptors that can be used for a wide variety of computer vision applications including detection, tracking and recognition of visual landmarks, physical objects, commercial logos, human bodies and human faces. Detection and tracking of human faces was selected as benchmark test case. This work has been continued with support from EIT ICTlabs, to provide context information for interaction with large multi-touch interactive displays installed in public spaces.

Multitouch interactive displays are increasingly used in outdoor and public spaces. This objective of this task is to provide a visual observation system that can detect and count users of a multitouch display and to estimate information such as the gender, and age category of each user. us rendering the system sensitive to environmental context.
SuiviDeCiblesCouleur locates individuals in a scene for video communications. FaceStabilisationSystem renormalises the position and scale of images to provide a stabilised video stream. SuiviDeCiblesCouleur has been declared with the APP "Agence pour la Protection des Programmes" under the Interdeposit Digital number IDDN.FR.001.370003.000.S.P.2007.000.21000.

A revised APP is under preparation for new versions of this software for face detection, face tracking, gender and age estimation, and orientation estimation.
5. Software

5.1. Amaya

Participant: Vincent Quint.

Amaya is an open source web editor, i.e. a tool for creating and updating documents directly on the web. Browsing features are seamlessly integrated with editing features in a uniform environment that allows users to save files locally and on remote servers as well. This follows the original vision of the web as a space for collaboration and not just a one-way publishing medium.

Amaya started as a joint effort with W3C to showcase web technologies in a fully-featured web client. The main motivation for developing Amaya was originally to provide a framework that can integrate many web technologies during their development, with the goal of demonstrating these technologies in action while taking advantage of their combination in a single, consistent environment.

Amaya now implements a number of web technologies, such as HTML and the XHTML family, CSS style sheets, generic XML, MathML (for mathematical expressions), and SVG (for vector graphics). It allows all those document formats to be edited simultaneously in compound documents. It also includes a collaborative annotation application based on RDF, XLink, and XPointer.

It is a unique tool for manipulating simultaneously different kinds of content through a formatted representation of documents, while closely following standard formats. Developed jointly with W3C, the software is distributed worldwide through the W3C servers and many mirrors. It is also part of several Linux distributions.

Amaya is also used as a platform for experimenting and distributing new editing techniques and document formats developed in WAM. It provides a full implementation of the XTiger language and its constraint-driven editing feature. It also helps users to create their own document types defined as XTiger templates.

5.2. XML Reasoning Solver

Participants: Pierre Genevès, Nabil Layaïda.

The XML Reasoning Solver is a tool for the static analysis of XPath queries and XML schemas based on the latest theoretical advances [13]. It allows automated verification of properties that are expressed as logical formulas over trees. A logical formula may for instance express structural constraints or navigation properties (like e.g. path existence and node selection) in finite trees.

The tool can solve many fundamental XML problems such as satisfiability of XPath expressions in the presence of XML schemas, containment and equivalence of XPath expressions, and many other problems that can be formulated with XPath expressions and schemas (DTDs, XML Schemas, Relax-NG).

The system is implemented in Java and uses symbolic techniques (binary decision diagrams) in order to enhance its performance. It is capable of comparing path expressions in the presence of real-world DTDs (such as the W3C SMIL and XHTML language recommendations, for instance). The cost ranges from several milliseconds for comparison of XPath queries without tree types, to several seconds for queries under very large, heavily recursive, type constraints, such as the XHTML DTD. These measurements shed light for the first time on the cost of solving static analysis problems in practice. Furthermore, the analyzer generates XML counter-examples that allow program defects to be reproduced independently from the analyzer.

5.3. Timesheets Library

Participants: Fabien Cazenave, Cécile Roisin.
The goal of the Timesheets library is to synchronize HTML5 content using declarative synchronization languages defined by W3C standards (namely, SMIL Timing and Synchronization and SMIL Timesheets).

With the raise of HTML5 which natively supports continuous content (audio, video) there is a dramatic need for handling synchronization, animation and user interaction in an efficient and homogeneous way. As web browsers do not support SMIL, except for SVG Animation (which rely on the SMIL BasicAnimation module), multimedia web authoring remains difficult and relies on code-based, non-standard solutions.

Therefore we are developing a generic, cross-browser JavaScript implementation for scheduling the dynamic behavior of HTML5 content that can be described with declarative SMIL markup. Using a declarative language makes sense for the most common tasks, that currently require JavaScript programming:

- it is much easier for web authors and for web authoring tool developers;
- it is a much better way to achieve good accessibility and indexability;
- it is easier to maintain, since no specific JavaScript code is used.

### 5.4. Mobile Audio Language

#### Participants: Yohan Lasorsa, Jacques Lemordant.

#### 5.4.1. MAUDL library

The MAUDL library (Mobile AUDio Language) [15] is an evolution of the ARIA library whose primary target was games on mobile.

Augmented Reality Audio applications use sound objects to create a soundscape. A sound object is a time structure of audio chunks whose duration is on the time scale of 100 ms to several seconds. These sound objects have heterogeneous and time-varying properties. In order to describe Interactive Audio (IA) contents, we created MAUDL, an XML language inspired by iXMF that is well adapted to the design of dynamic soundtracks for navigation systems.

MAUDL prevents audio information overwhelming through categorization at the declarative level and the use of priority queues at the execution level. This allows to take account of speed when walking, and rapid hand gestures when interrogating the environment for example. MAUDL can be used as an authoring time interchange file format for interactive mobile applications or as a runtime file format that is actually loaded through the web and played directly in the mobile. MAUDL is a cue-oriented interactive audio system, audio services being requested using named events and the systems response to each event being determined by the audio artist. The current version of the API supports iOS and further support for other mobile platforms (Android) is planned.

#### 5.4.2. 3D Audio Pointer

A virtual 3D audio pointer provides an intuitive guide to the user of a mobile application, reducing the need for cognitive work when compared to vocal instructions. We have built such a pointer using the MAUDL language. It gives the user the azimuth using HRTF spatialized audio cues, with additional hints taking the form of variations in the sound used. It allows to superpose other kinds of audio contents, such as voice while the pointer is active, to indicate distance for example. This audio object is suitable for different sorts of navigation systems, such as POIs browsers, self-guided audio tours, or predefined route following applications.

### 5.5. Mixed Reality Browser

#### Participants: Audrey Colbrant, Yohan Lasorsa, Jacques Lemordant, David Liodenot, Mathieu Razafimahazo.

The Mixed Reality Browser (MRB) is a geolocalized web browser running on mobile devices.
The concept of Mixed Reality comes from the fact that the real/virtual dichotomy is not sharp, but interpola-
tively smooth over a virtuality continuum. Idealized notions of reality and virtuality can be thought of as
endpoints on a continuum, an instance of the former approach corresponding for example to a see-through
display with natural sounds, an instance of the latter to texture-mapped image-based rendering (panoramas)
with synthetic sound objects.

Augmented Reality (AR) mode refers to all cases in which the auditory or visual display of an otherwise real
environment is augmented by means of virtual sound or graphic objects. The converse case on the virtuality
continuum is Augmented Virtuality (AV), where a virtual world, one that is generated primarily by computer,
like with synthetic 3D graphic or synthetic panoramic, is being augmented with the audio-visual content of
points of interest (POIs).

The introduction of mobile augmented reality browsers has forced a rethink on what kind of reality should
be offered. Mobility induces a need for telepresence and simulation to free the user or the developer of
the necessity to go every time in the real world. Mobility is the main reason behind the concept of Mixed
Reality Browsers. By its intrinsic characteristics, MRB supports advance MR applications like mobile remote
maintenance and assisted navigation.