Activity Report 2011

Section Software

Edition: 2012-03-23
# Algorithmics, Programming, Software and Architecture

1. CAMUS Team ................................................................. 4
2. CARAMEL Project-Team .................................................. 7
3. CARTE Project-Team ....................................................... 10
4. CASSIS Project-Team ...................................................... 11
5. PAREO Project-Team ....................................................... 14
6. TRIO Project-Team ......................................................... 16
7. VEGAS Project-Team ....................................................... 19
8. VERIDIS Team ............................................................... 21

# Applied Mathematics, Computation and Simulation

9. CALVI Project-Team ......................................................... 23
10. CORIDA Project-Team ..................................................... 24
11. TOSCA Project-Team ...................................................... 25

# Computational Sciences for Biology, Medicine and the Environment

12. BIGS Project-Team ........................................................ 26
13. CORTEX Project-Team ..................................................... 27
14. MASAIE Project-Team (section vide) ................................ 30

# Networks, Systems and Services, Distributed Computing

15. ALGORILLE Project-Team ................................................ 31
16. MADYNES Project-Team ................................................ 34
17. SCORE Team ............................................................... 36

# Perception, Cognition, Interaction

18. ALICE Project-Team ....................................................... 37
19. MAGRIT Project-Team ..................................................... 39
20. MAIA Project-Team ........................................................ 40
21. ORPAILLEUR Project-Team ............................................... 41
22. PAROLE Project-Team ..................................................... 45
23. Sémagramme Team ........................................................ 49
24. TALARIS Project-Team .................................................... 52
5. Software

5.1. PolyLib

PolyLib is a C library of polyhedral functions, that can manipulate unions of rational polyhedra of any dimension, through the following operations: intersection, difference, union, convex hull, simplify, image and preimage. It was the first to provide an implementation of the computation of parametric vertices of a parametric polyhedron, and the computation of an Ehrhart polynomial (expressing the number of integer points contained in a parametric polytope) based on an interpolation method.

It is used by an important community of researchers (in France and the rest of the world) in the area of compilation and optimization using the polyhedral model. Vincent Loechner is the maintainer of this software. It is distributed under GNU General Public License version 3 or later, and it has a Debian package maintained by Serge Guelton (Symbiose Projet, IRISA).

5.2. ZPolyTrans

ZPolyTrans is a C library and a set of executables, that permits to compute the integer transformation of a union of parametric $\mathbb{Z}$-polyhedra (the intersection between lattices and parametric polyhedra), as a union of parametric $\mathbb{Z}$-polyhedra. The number of integer points of the result can also be computed. It is build upon PolyLib and Barvinok library. This work is based on some theoretical results obtained by Rachid Seghir and Vincent Loechner, that will be published in ACM TACO in 2011.

It allows for example to compute the number of solutions of a Presburger formula by eliminating existencial integer variables, or to compute the number of different data accessed by some array accesses contained in an affine parametric loop nest.

The authors of this software are Rachid Seghir (Univ. Batna, Algeria) and Vincent Loechner. It is distributed under GNU General Public License version 3 or later.

5.3. NLR

We have developed a program implementing our loop-nest recognition algorithm, detailed in [7]. This standalone, filter-like application takes as input a raw trace and builds a sequence of loop nests that, when executed, reproduce the trace. It is also able to predict forthcoming values at an arbitrary distance in the future. Its simple, text-based input format makes it applicable to all kinds of data. These data can take the form of simple numeric values, or have more elaborate structure, and can include symbols. The program is written in standard ANSI C. The code can also be used as a library.

We have used this code to evaluate the compression potential of loop nest recognition on memory address traces, with very good results. We have also shown that the predictive power of our model is competitive with other models on average. The software is available upon request to anybody interested in trying to apply loop nest recognition. It has been distributed to a dozen of colleagues around the world.

We plan on using this software as the base for a new tool we currently design, for the analysis of parallel traces.

5.4. Dynamic version selector

We are developing a toolchain to automatically select between different versions of parallel loop nests, as described in subsection 6.2. It generates the profiling code and selection code from a loop nest source code and different schedules, expressed in the CLooG format.

---

8 http://icps.u-strasbg.fr/PolyLib
9 http://ZPolyTrans.gforge.inria.fr
Benoît Pradelle (PhD) wrote this toolchain, based on python scripts. It is not yet distributed.

5.5. Binary files decompiler

Our research on efficient memory profiling has lead us to develop a sophisticated decompiler. This tool analyzes x86-64 binary programs and libraries, and extracts various structured representations of the code. It works on a routine per routine basis, and first builds a loop hierarchy to characterize the overall structure of the algorithm. It then puts the code into Static Single Assignment (SSA) form to highlight the fine-grain data-flow between registers and memory. Building on these, it performs the following analyzes:

- All memory addresses are expressed as symbolic expressions involving specific versions of register contents, as well as loop counters. Loop counter definitions are recovered by resolving linearly incremented registers and memory cells, i.e., registers that act as induction variables.
- Most conditional branches are also expressed symbolically (with registers, memory contents, and loop counters). This captures the control-flow of the program, but also helps in defining what amounts to loop “trip-counts”, even though our model is slightly more general, because it can represent any kind of iterative structure.

This tool embodies several passes that, as far as we know, do not exist in any existing similar tool. For instance, it is able to track data-flow through stack slots in most cases. It has been specially designed to extract a representation that can be useful in looking for parallel (or parallelizable) loops [48]. It is the basis of several of our studies.

Because binary program decompilation is especially useful to reduce the cost of memory profiling, our current implementation is based on the Pin binary instrumenter. It uses Pin’s API to analyze binary code, and directly interfaces with the upper layers we have developed (e.g., program skeletonization, or minimal profiling). However, we have been careful to clearly decouple the various layers, and to not use any specific mechanism in designing the binary analysis component. Therefore, we believe that it could be ported with minimal effort, by using a binary file format extractor and a suitable binary code parser. It is also designed to abstract away the detailed instruction set, and should be easy to port (even though we have no practical experience in doing so).

We feel that such a tool could be useful to other researchers, because it makes binary code available under abstractions that have been traditionally available for source code only. If sufficient interest emerges, e.g., from the embedded systems community, or from researchers working on WCET, or from teams working on software security, we are willing to distribute and/or to help make it available under other environments.

5.6. Dynamic dependency analyser

We have recently started developing a dynamic dependence analyzer. Such a tool consumes the trace of memory (or object) accesses, and uses the program structure to list all the data dependences appearing during execution. Data dependences in turn are central to the search for parallel sections of code, with the search for parallel loops being only a particular case of the general problem. Most current works of these questions are either specific to a particular analysis (e.g., computing dependence densities to select code portions for thread-level speculation), or restricted to particular forms of parallelism (e.g., typically to fully parallel loops). Our tool tries to generalize existing approaches, and focuses on the program structures to provide helpful feedback either to a user (as some kind of “smart profiler”), or to a compiler (for feedback-directed compilation). For example, the tool is able to produce a dependence schema for a complete loop nest (instead of just a loop). It also targets irregular parallelism, for example analyzing a loop execution to estimate the expected gain of parallelization strategies like inspector-executor.

We have developed this tool in relation to our minimal profiling research project. However, the tool itself has been kept independent of our profiling infrastructure, getting data from it via a well-defined trace format. This intentional design decision has been motivated by our work on distinct execution environments: first
on our usual x86-64 benchmark programs, and second on less regular, more often written in Java, real-world applications. The latter type of applications is likely the one that will most benefit from such tools, because their intrinsic execution environment does not offer enough structure to allow effective static analysis techniques. Parallelization efforts in this context will most likely rely on code annotations, or specific programming language constructs. Programmers will therefore need tools to help them choose between various constructs. Our tool has this ambition. We already have a working tool-chain for C/C++/Fortran programs (or any binary program). We are in the process of developing the necessary infrastructure to connect the dynamic dependence profiler to instrumented Java programs. Other managed execution environments could be targeted as well, e.g., Microsoft’s .Net architecture, but we have no time and/or workforce to devote to such time-consuming engineering efforts.

5.7. VMAD software and LLVM

For dynamic analysis and optimization of programs, we are developing a virtual machine called VMAD, and specific passes to the LLVM compiler suite, plus a modified Clang frontend. It is fully described in subsection 6.1.

We implemented for now a memory access predictor in loop nests, based on the computation of linear interpolation functions. The profiling is very fast compared to other existing tools, as it samples only the first few iterations of each loop in the nest, then it is deactivated to return to the original, faster version. Other tools like PIN or PEBIL do not support such activation/deactivation mechanism.

New annotations for the final user, taken as input by LLVM, and new VMAD modules will be developed, as these tools have been designed to be very evolving.

Alexandra Jimborean (PhD), Matthieu Herrmann (Master student) and Luis Mastrangelo (Master student) are the main contributors of this software. It is not yet distributed.

5.8. Polyhedral prover

Participants: Nicolas Magaud, Julien Narboux, Éric Violard [correspondant].

We are currently developing a formal proof of program transformations based on the polyhedral model. We use the CompCert verified compiler [54] as a framework. This tool is written in the specification language of Coq. It is connected to the activity described in section 6.6.
5. Software

5.1. Introduction

A major part of the research done in the CARAMEL team is published within software. On the one hand, this enables everyone to check that the algorithms we develop are really efficient in practice; on the other hand, this gives other researchers — and us of course — basic software components on which they — and we — can build other applications.

5.2. GNU MPFR

Participant: Paul Zimmermann [contact].

GNU MPFR is one of the main pieces of software developed by the CARAMEL team. Since end 2006, with the departure of Vincent Lefèvre to ENS Lyon, it has become a joint project between CARAMEL and the ARÉNAIRE project-team (INRIA Grenoble - Rhône-Alpes). GNU MPFR is a library for computing with arbitrary precision floating-point numbers, together with well-defined semantics, and is distributed under the LGPL license. All arithmetic operations are performed according to a rounding mode provided by the user, and all results are guaranteed correct to the last bit, according to the given rounding mode.

Several software systems use GNU MPFR, for example: the GCC and GFortran compilers; the SAGE computer algebra system; the KDE calculator Abakus by Michael Pyne; CGAL (Computational Geometry Algorithms Library) developed by the Geometrica project-team (INRIA Sophia Antipolis - Méditerranée); Gappa, by Guillaume Melquiond; Sollya, by Sylvain Chevillard, Mioara Joldes and Christoph Lauter; Genius Math Tool and the GEL language, by Jiri Lebl; Giac/Xcas, a free computer algebra system, by Bernard Parisse; the iRRAM exact arithmetic implementation from Norbert Müller (University of Trier, Germany); the Magma computational algebra system; and the Wcalc calculator by Kyle Wheeler.

The main developments in 2011 are the release of version 3.0.1 in April, and the release of version 3.1.0 (the “canard à l’orange” release) in October. The main changes in GNU MPFR 3.1.0 are the following: thread local storage (TLS) support is now detected automatically, the squaring and division routines got a major speed up thanks to Mulders’ algorithm [20], and a new divide-by-zero exception was introduced. Note that the automatic TLS support did exhibit several compiler bugs (http://www.loria.fr/~zimmerma/software/compilerbugs.html). We had a developers meeting in January 13-14, and in August GNU MPFR was presented at the GNU Hackers Meeting in Paris.

5.3. MPC

Participant: Paul Zimmermann [contact].

MPC is a floating-point library for complex numbers, which is developed on top of the GNU MPFR library, and distributed under the LGPL license. It is co-written with Andreas Enge (LFANT project-team, INRIA Bordeaux - Sud-Ouest). A complex floating-point number is represented by $x + iy$, where $x$ and $y$ are real floating-point numbers, represented using the GNU MPFR library. The MPC library provides correct rounding on both the real part $x$ and the imaginary part $y$ of any result. MPC is used in particular in the TRIP celestial mechanics system developed at IMcce (Institut de Mécanique Céleste et de Calcul des Éphémérides), and by the Magma computational number theory system.

A new version, MPC 0.9 (Epilobium montanum), was released in February 2011, with new functions, some speed-ups, a few bug fixes, and a logging feature for debugging. Since version 4.5 of GCC, released in May 2010, GCC requires MPC to compute constant complex expressions at compile-time (constant folding), like it requires GNU MPFR since GCC 4.3.
5.4. GMP-ECM

Participants: Cyril Bouvier, Paul Zimmermann [contact].

GMP-ECM is a program to factor integers using the Elliptic Curve Method. Its efficiency comes both from the use of the GMP library, and from the implementation of state-of-the-art algorithms. GMP-ECM contains a library (LIBECM) in addition to the binary program (ECM). The binary program is distributed under GPL, while the library is distributed under LGPL, to allow its integration into other non-GPL software. The Magma computational number theory software and the SAGE computer algebra system both use LIBECM.

During his internship of 4 months in 2011, Cyril Bouvier developed a version of ECM for GPUs. The code was written for NVIDIA GPUs using CUDA. First, the code was written for all NVIDIA cards, and later, it was optimized for the newer Fermi cards. As there is no modular arithmetic library (like GMP) available for GPU, it was necessary to implement a modular arithmetic using array of unsigned integers from scratch, while taking into account constraints of GPU programming. The code was optimized for factoring 1024 bits integers. For now, the code has a throughput roughly four times bigger than GMP-ECM on one core. This code is not yet fully integrated in GMP-ECM but is available in the GMP-ECM svn repository.

The implementation of ECM on GPU uses a different algorithm for scalar multiplication (the binary ladder instead of PRAC) and a different parametrization. This new approach was implemented for CPU in GMP-ECM. It results in a speedup in the execution time of GMP-ECM for finding big factors (more than 20 digits). It will be integrated in the next release of GMP-ECM.

5.5. Finite fields

Participants: Pierrick Gaudry, Emmanuel Thomé [contact].

mpFq is (yet another) library for computing in finite fields. The purpose of mpFq is not to provide a software layer for accessing finite fields determined at runtime within a computer algebra system like Magma, but rather to give a very efficient, optimized code for computing in finite fields precisely known at compile time. mpFq is not restricted to a finite field in particular, and can adapt to finite fields of any characteristic and any extension degree. However, one of the targets being the use in cryptology, mpFq somehow focuses on prime fields and on fields of characteristic two.

mpFq’s ability to generate specialized code for desired finite fields differentiates this library from its competitors. The performance achieved is far superior. For example, mpFq can be readily used to assess the throughput of an efficient software implementation of a given cryptosystem. Such an evaluation is the purpose of the “EBats” benchmarking tool. mpFq entered this trend in 2007, establishing reference marks for fast elliptic curve cryptography: the authors improved over the fastest examples of key-sharing software in genus 1 and 2, both over binary fields and prime fields. These timings are now comparison references for other implementations.

The library’s purpose being the generation of code rather than its execution, the working core of mpFq consists of roughly 18,000 lines of Perl code, which generate most of the C code. mpFq is distributed at http://mpfq.gforge.inria.fr/.

The mpFq library has undergone no change in 2011.

5.6. gf2x

Participants: Pierrick Gaudry, Emmanuel Thomé [contact], Paul Zimmermann.

GF2X is a software library for polynomial multiplication over the binary field, developed together with Richard Brent (Australian National University, Canberra, Australia). It holds state-of-the-art implementation of fast algorithms for this task, employing different algorithms in order to achieve efficiency from small to large operand sizes (Karatsuba and Toom-Cook variants, and eventually Schönhage’s or Cantor’s FFT-like algorithms). GF2X takes advantage of specific processors instruction (SSE, PCLMULQDQ).

1 http://www.ecrypt.eu.org/ebats/
The current version of GF2X is 1.0, released in 2010 under the GNU GPL. Since 2009, GF2X can be use as an auxiliary package for the widespread software library NTL, as of version 5.5.

There has been no update of GF2X in 2011, but the software is still maintained. An LGPL-licensed portion of GF2X is also part of the CADO-NFS software package.

5.7. CADO-NFS

Participants: Jérémie Detrey, Pierrick Gaudry, Lionel Muller, Emmanuel Thomé [contact], Paul Zimmermann.

CADO-NFS is a program to factor integers using the Number Field Sieve algorithm (NFS), developed in the context of the ANR-CADO project (November 2006 to January 2010).

NFS is a complex algorithm which contains a large number of sub-algorithms. The implementation of all of them is now complete, but still leaves some places to be improved. Compared to existing implementations, the CADO-NFS implementation is already a reasonable player. Several factorizations have been completed using our implementations.

Since 2009, the source repository of CADO-NFS is publicly available for download. On October 28, 2011, the 1.1 version of CADO-NFS has been released. Several improvements to the program have been obtained, in practically all areas of the program. In particular, the polynomial selection code described by Thorsten Kleinjung at the CADO workshop in 2008 is now used within CADO-NFS, together with some efficient root-sieve code written by Shi Bai (Australian National University). Overall, CADO-NFS keeps improving its competitiveness over alternative code bases. The lattice siever now supports a sieving region of $2^{11} (I = 16)$; its code has been deeply reorganized to allow future improvements that we have in mind but were difficult to implement (proper sieving of powers, sieve according to the parities of the coordinates of the location). The executables in the linear algebra step have been reorganized (now using shared libraries), and now use a code generation mechanism built on top of the MPFQ library for the arithmetic parts. This is in particular meant to ease future accommodation of other base fields that GF(2), which is a requirement for adapting CADO-NFS to discrete logarithm computation. The MPI performance of the linear algebra code has been optimized. Some experimental scripts have been added to execute the sieve on a cluster; these scripts rely on the OAR job scheduler being used, and exploit its “besteffort” mode.

The largest factorizations performed by CADO-NFS in 2011 are a 170-digit integer from aliquot sequence 660 and a 171-digit integer from aliquot sequence 966.

5.8. AVIsogenies

Participants: Gaëtan Bisson [contact], Romain Cosset.

AVISOGENIES (Abelian Varieties and Isogenies) is a Magma package for working with abelian varieties, with a particular emphasis on explicit isogeny computation; it has been publicly released under the LGPLv2+ license in 2010.

Its prominent feature is the computation of $(\ell, \ell)$-isogenies between Jacobian varieties of genus-2 hyperelliptic curves over finite fields; practical runs have involved values of $\ell$ in the hundreds. It also provides procedures for exploring and drawing isogeny graphs, and for computing various complex-multiplication-related structures, such as Shimura’s Gothic C group.

In 2011, two incremental versions have been released. They provide the following new features: the characteristic $2$ is now supported, and the complete addition laws of [23] have been implemented.

The package can be obtained at http://avisogenies.gforge.inria.fr/.
5. Software

5.1. Morphus/MMDEX
An anti-virus software based on morphological analysis, Dépôt APP du logiciel MMDEX, 2009, IDDN.FR.001.300033.000.R.P.2009.000.10000

5.2. PYMS
Online disassembler. http://pys86.appspot.com/

5.3. TraceSurfer

5.4. Crème Brûlée
Crème Brûlée is an experimental Javascript dynamic instrumentation engine. http://code.google.com/p/cremebrulee/
CASSIS Project-Team

5. Software

5.1. Protocol Verification Tools

Participants: Pierre-Cyrille Héam, Olga Kouchkarenko, Michaël Rusinowitch, Mathieu Turuani, Laurent Vigneron.

5.1.1. AVISPA

AVISPA has been one of the 4 partners involved in the European project AVISPA, which has resulted in the distribution of a tool for automated verification of security protocols, named AVISPA Tool. It is freely available on the web\(^2\) and it is well supported. The AVISPA Tool compares favourably to related systems in scope, effectiveness, and performance, by (i) providing a modular and expressive formal language for specifying security protocols and properties, and (ii) integrating 4 back-ends that implement automatic analysis techniques ranging from protocol falsification (by finding an attack on the input protocol) to abstraction-based verification methods for both finite and infinite numbers of sessions.

5.1.2. CL-AtSe

We develop, as a first back-end of AVISPA, CL-AtSe, a Constraint Logic based Attack Searcher for cryptographic protocols. The CL-AtSe approach to verification consists in a symbolic state exploration of the protocol execution, for a bounded number of sessions. This necessary restriction (for decidability, see \([85]\)) allows CL-AtSe to be correct and complete, i.e., any attack found by CL-AtSe is a valid attack, and if no attack is found, then the protocol is secure for the given number of sessions. Each protocol step is represented by a constraint on the protocol state. These constraints are checked lazily for satisfiability, where satisfiability means reachability of the protocol state. CL-AtSe includes a proper handling of sets (operations and tests), choice points, specification of any attack states through a language for expressing secrecy, authentication, fairness, non-abuse freeness, advanced protocol simplifications and optimizations to reduce the problem complexity, and protocol analysis modulo the algebraic properties of cryptographic operators such as XOR (exclusive or) and Exp (modular exponentiation). The handling of XOR and Exp has required to implement an optimized version of the combination algorithm of Baader & Schulz \([76]\) for solving unification problems in disjoint unions of arbitrary theories.

CL-AtSe has been successfully used \([75]\) to analyse France Telecom R&D, Siemens AG, IETF, or Gemalto protocols in funded projects. It is also employed by external users, e.g., from the AVISPA’s community. Moreover, CL-AtSe achieves very good analysis times, comparable and sometimes better than state-of-the art tools in the domain (see \([90]\) for tool details and precise benchmarks).

5.1.3. TA4SP

We have developed, as a second back-end of AVISPA, TA4SP (Tree Automata based on Automatic Approximations for the Analysis of Security Protocols), an automata based tool dedicated to the validation of security protocols for an unbounded number of sessions. This tool provides automatic computations of over- and under-approximations of the knowledge accessible by an intruder. This knowledge is encoded as a regular tree language and protocol steps and intruder abilities are encoded as a term rewriting system. When given a reachability problem such as secrecy, TA4SP reports that (1) the protocol is safe if it manages to compute an over-approximation of intruder’s knowledge that does not contain a secret term or (2) the protocol is unsafe in the rewrite model if it manages to compute an underapproximation of intruder’s knowledge containing a secret term or (3) I don’t know otherwise. TA4SP has verified 28 industrial protocols and case (3) occurred only once, for Kaiochow protocol version 2.

\(^2\) http://www.avispa-project.org
TA4SP handles protocols using operators with algebraic properties. Thanks to a recent quadratic completion algorithm new experimental results have been obtained, for example for the Encrypted Key Exchange protocol (EKE2) using the exponential operator.

Recently, TA4SP was used in [89] to analyse a hierarchy of authentication properties.

5.2. Testing Tools

**Participants:** Fabrice Bouquet, Frédéric Dadeau, Philippe Paquelier.

In December 2008, we have started the redevelopment of our original testing tools environment, with two objectives: first, refactoring the existing developments, and, second, providing an open platform aiming at gathering together the various developments, increasing the reusability of components. The resulting platform, named Hydra, is a Eclipse-like platform, based on Plug-ins architecture. Plug-ins can be of five kinds: *parser* is used to analyze source files and build an intermediate format representation of the source; *translator* is used to translate from a format to another or to a specific file; *service* denotes the application itself, i.e. the interface with the user; *library* denotes an internal service that can be used by a service, or by other libraries; *tool* encapsulates an external tool. The following services have been developed so far:

- **BZPAnimator:** performs the animation of a BZP model (a B-like intermediate format);
- **Angluin:** makes it possible to perform a machine learning algorithm (à la Angluin) in order to extract an abstraction of a system behavior;
- **UML2SMT:** aims at extracting first order logic formulas from the UML Diagrams and OCL code of a UML/OCL model to check them with a SMT solver.

These services involve various libraries (sometimes reusing each other), and rely on several *tool* plug-ins that are: SMTProver (encapsulating Z3 solver), PrologTools (encapsulating CLPS-B solver), Grappa (encapsulating a graph library). We are currently working on transferring the existing work on test generation from B abstract machines, JML, and statecharts using constraint solving techniques.

5.3. Collaborative Tools

**Participants:** Abdessamad Imine, Asma Cherif.

The collaborative tools allow us to manage collaborative works on shared documents using flexible access control models. These tools have been developed in order to validate and evaluate our approach on combining collaborative edition with optimistic access control.

- **P2PEdit.** This prototype is implemented in Java and supports the collaborative editing of HTML pages and it is deployed on P2P JXTA platform. In our prototype, a user can create a HTML page from scratch by opening a new collaboration group. Other users (peers) may join the group to participate in HTML page editing, as they may leave this group at any time. Each user can dynamically add and remove different authorizations for accessing to the shared document according the contribution and the competence of users participating in the group. Using JXTA platform, users exchange their operations in real-time in order to support WYSIWIS (What You See Is What I See) principle. Furthermore, the shared HTML document and its authorization policy are replicated at the local memory of each user. To deal with latency and dynamic access changes, an optimistic access control technique is used where enforcement of authorizations is retroactive.

- **P2PCalendar.** To extend our collaboration and access control models to mobile devices, we implemented a shared calendar on iPhone OS which is decentralized and scalable (i.e. it can be used over both P2P and ad-hoc networks) [58]. This application aims to make a collaborative calendar where users can simultaneously modify events (or appointments) and control access on events. The access rights are determined by the owner of an event. The owner decides who is allowed to access the event and what privileges they have. Likewise to our previous tool, the calendar and its authorization policy are replicated at every mobile device.

---

3 http://www.sun.com/software/jxta/
5.4. Other Tools

Several software tools described in previous sections are using tools that we have developed in the past. For instance BZ-TT uses the set constraints solver CLPS. Note that the development of the SMT prover haRVey has been stopped. The successor of haRVey is called veriT and is developed by David Déharbe (UFRN Natal, Brasil) and Pascal Fontaine (Veridis team).
5. Software

5.1. ATerm

Participant: Pierre-Etienne Moreau [correspondant].

ATerm (short for Annotated Term) is an abstract data type designed for the exchange of tree-like data structures between distributed applications.

The ATerm library forms a comprehensive procedural interface which enables creation and manipulation of ATerms in C and Java. The ATerm implementation is based on maximal subterm sharing and automatic garbage collection.

A binary exchange format for the concise representation of ATerms (sharing preserved) allows the fast exchange of ATerms between applications. In a typical application—parse trees which contain considerable redundant information—less than 2 bytes are needed to represent a node in memory, and less than 2 bits are needed to represent it in binary format. The implementation of ATerms scales up to the manipulation of ATerms in the giga-byte range.

The ATerm library provides a comprehensive interface in C and Java to handle the annotated term data-type in an efficient manner.

We are involved (with the CWI) in the implementation of the Java version, as well as in the garbage collector of the C version. The Java version of the ATerm library is used in particular by Tom.

The ATerm library is documented, maintained, and available at the following address: http://www.meta-environment.org/Meta-Environment/ATerms.

5.2. Tom

Participants: Jean-Christophe Bach, Horatiu Cirstea, Pierre-Etienne Moreau [correspondant], Claudia Tavares.

Since 2002, we have developed a new system called Tom [49], presented in [27], [28]. This system consists of a pattern matching compiler which is particularly well-suited for programming various transformations on trees/terms and XML documents. Its design follows our experiences on the efficient compilation of rule-based systems [45]. The main originality of this system is to be language and data-structure independent.

This means that the Tom technology can be used in a C, C++ or Java environment. The tool can be seen as a Yacc-like compiler translating patterns into executable pattern matching automata. Similarly to Yacc, when a match is found, the corresponding semantic action (a sequence of instructions written in the chosen underlying language) is triggered and executed. Tom supports sophisticated matching theories such as associative matching with neutral element (also known as list-matching). This kind of matching theory is particularly well-suited to perform list or XML based transformations for example.

In addition to the notion of rule, Tom offers a sophisticated way of controlling their application: a strategy language. Based on a clear semantics, this language allows to define classical traversal strategies such as innermost, outermost, etc.. Moreover, Tom provides an extension of pattern matching, called anti-pattern matching. This corresponds to a natural way to specify complements (i.e. what should not be there to fire a rule). Tom also supports the definition of cyclic graph data-structures, as well as matching algorithm and rewriting rules for term-graphs.

5.3. Cat

**Participant:** Yves Guiraud [correspondant].

Cat is a library for polygraphic calculus, written in Caml. It has been used, in a joint work with F. Blanqui, to produce an automatic termination prover for first-order functional programs. It translates such a rewriting system into a polygraph and tries to find a derivation proving its termination, using the results of [6], [39]. If possible, it seeks a derivation that proves that the program is polynomial [29], [3]. Cat is also at the basis of Catex.

5.4. Catex

**Participant:** Yves Guiraud [correspondant].

Catex is a tool for (pdf)Latex, used in the same way as Bibtex, that automatically produces string diagrams from their algebraic expression. It follows the same design as Tom, a Catex file being a Latex file enriched with formal islands corresponding to those algebraic expressions, such as:

\begin{verbatim}
\deftwocell[red]{delta : 1 -> 2}
\deftwocell[orange]{mu : 2 -> 1}
\deftwocell[orange]{tau : 2 -> 2}
\twocell{(delta *0 delta) *1 (1 *0 tau *0 1) *1 (mu *0 mu)}
\end{verbatim}

Catex dissolves such an island into Latex code, using the PGF/Tikz package. Executed on the result, (pdf)Latex produces the following diagram:

![Figure 4.](image)

Catex is distributed through the page: [http://www.loria.fr/~guiraudy/catex](http://www.loria.fr/~guiraudy/catex). We want to extend Catex in two directions. First, to produce diagrams not only for Latex but also for web publications. Then, Catex will be adapted to a tool for the automatic production, in scientific papers, of certified algebraic computations, which are a three-dimensional equivalent of string diagrams.
5. Software

5.1. MPIGate: Multi-Protocols Interface and Gateway for telehomecare and environment monitoring and control

Participants: Shahram Nourizadeh, Hugo Cruz Sanchez, Ye-Qiong Song.

For developing AAL (Ambient Assisted Living) or more generally the environment monitoring and control systems, heterogeneous wireless and wired networks will be used. To solve firstly the interoperability problems, and then to ensure the application required QoS, we developed a software prototype called MPIGate. MPIGate includes two important components: a user interface for telehomecare and home automation, and a gateway for ensuring the interworking of the different networks. In 2010, MPIGate has been laureate of the 12th national contest for the creation of innovative technology companies by the ministry of higher education and research (“Emergence” category). During 2011, MPIGate has been implemented on an embedded linux board and integrated into LORIA smart room platform within CPER IS project (http://infositu.loria.fr) [25]. [45]. In its current version, the gateway ensures the communication between IP (Ethernet and Wifi), home automation network (KNX), Bluetooth and Zigbee. Heterogeneous sensors can be now easily used through MPIGate interface for further building the activity monitoring of the elderly person living along at home or other application scenarios.

5.2. SAMOVAR

Participants: Adrien Guénard, Lionel Havet, Françoise Simonot-Lion.

Wireless Sensor and Actuator Networks (WSANs) combine sensors and actuators interconnected by wireless networks in order to perform distributed sensing and acting tasks. Closed-loop controllers can therefore be deployed on WSANs. Such systems have to meet specific requirements in terms of performance, dependability, energy and cost which raises great challenges due to the unreliability of wireless communications. A way to ensure that a system meets the required properties is to model it and go through its analysis. Building a model requires both deep knowledge on the system as well as on the used framework. Therefore there is a need for frameworks well-suited to the targeted systems and to the properties to verify. We proposed an approach meeting these conditions and a simulation framework, Samovar, based on Matlab / Simulink, allowing the modeling of the network protocols (Mac and routing services) and the resources sharing policy thanks to the TrueTime toolbox. Several classes of components (application, nodes, networks and middleware) and a clear semantics for their composition are identified. Furthermore, the design of Samovar was also driven by the need to easily transfer software component model between the concrete systems and its simulated model. The modeling and simulation method as well as the Samovar framework were assessed on several case studies: cooperating robots, intelligent living environment, embedded controllers on UAV robots... The simulation framework is available from http://samovar.loria.fr/. This work is supported by INRIA through the ADT SAMOVAR.

5.3. ANR Open-PEOPLE platform

Participants: Sophie Alexandre, Jonathan Ponroy, Kévin Roussel, Olivier Zendra.

The aim of Open-PEOPLE is to provide a platform for estimating and optimizing the power and energy consumption of systems. The Open-PEOPLE project formally started in April 2009. Two systems administrator and software developers had been hired initially: Sophie Alexandre and Kévin Roussel. Another system administrator and software developer, Jonathan Ponroy, joined them in 2010 when he finished his work on the ANR MORE project where he worked previously. Sophie Alexandre contract ended in February 2011.
Since the beginning of the Open-PEOPLE project, we had made significant progress in setting up the infrastructure for the software part of the platform, for which INRIA Nancy Grand Est is responsible. We had included new features to be able to fully integrate and test software developed as Eclipse plugins, relying on the Buckminster tool. We had also created a specific extension set for SVN and Hudson, called OPCIM (Open-PEOPLE Continuous Integration Mechanism). OPCIM had been registered at APP on 13/04/2010 with number IDDN.FR.001.150008.000.S.P.2010.000.10000.

Concerning the Open-PEOPLE platform itself, we had first tackled the high-level work, working with our partners on the definition of the requirements of the platform according to the needs of industry. We had then realized the specification work to define the global perimeter of our platform, according to the previous requirements. As part of this work had also been designed exchanges formats between the various tools. We had also designed at INRIA Nancy Grand Est a Tools integration Protocol, which specified requirements for external tools to be integrated in our platform. All this design work had been materialized in several reports which were deliveries provided to ANR.

We had also designed and developed an authentication component (Eclipse plugin) for the platform, so as to be able to provide a unique, secured access gate to the platform to all the tools that are or shall be integrated into it.

We had also started and almost finished developing an Internet portal giving access and control to the Open-PEOPLE Hardware Platform, located at our partner’s UBS in Lorient. Our portal features included user account management facilities, on the admin side, and on the user side, the ability to create, save, edit, reuse and of course submit jobs, make reservations for the hardware platform resources and get back tests results.

Finally, we had started working on two important parts of the software platform.

First, a way to unify the user experience despite the fact the platform federates several tools which were not developed to interact together. This implied an important and in-depth study of the wanted ergonomy for the platform, which involved taking into account both user needs and habits and the features of the available software tools.

The second work which had begun in 2011 was the design (then implementation) of the communications of between the various tools of the platform. This skeleton will be a key part of our platform, and the quality of its design will have a tremendous impact on its maintainability and its extensibility.

Note that the Open-PEOPLE project had been successfully evaluated on 14/09/2010 by ANR. Developments done during the first two years in the project are detailed in the 2009 and 2010 activity reports. In 2011, these developments went on.

We continued the work to solidify our development platform supporting our work and that of our partners. We produced a finer grained definition of the software platform functionalities, and a more precise definition of the tools integration protocol. We worked towards the corresponding implementation documents, adding two new deliverables about the architecture of the software platform and the ergonomics of the software platform.

For the latter, we extensively interviewed user about ergonomics and designed several GUI mockups. We progressed on the implementation of the software platform, especially with respect to the internet portal to remote-control the hardware platform. We participated to the definition of the hardware platform and its functionalities, and participated actively to the work on the Specification document for HW / SW interfacing.

We provided the first concrete design and implementation of the HW/SW platform interfacing, with our implementation of the remote control portal for the HW platform. This remote control module was completed in Fall 2011.

We also participated to the work pertaining to basic components model homogenization, by reviewing this in the context of the software platform architecture and implementation, which resulted in several incremental improvements of the underlying models. Finally, progressing towards the first release of the software part of the Open-PEOPLE platform, we realized an ergonomic study for the consumption laws editors, with mockups and user interviews and validation. We worked on the implementation of the editors for the consumption laws, which required learning new environments and development tools (related to the EMF framework and the AADL, QUDV and MathML models). As a consequence, we completed the implementation of the GUI and
engine to create units and quantities. We finalized the architecture needed to integrate external modules in the platform.

With this progress, the first release of the whole Open-PEOPE software platform platform is expected early 2012.

5.4. VITRAIL

Participants: Damien Bodenes, Pierre Caserta, Olivier Zendra.

The aim of the VITRAIL operation is to provide tools for the advanced and immersive visualization of programs. It partners with the University of Montréal, University of Montpellier and Pareo team of INRIA Nancy Grand Est.

Last years, in VITRAIL, we had developed software to instrument and trace Java programs at the bytecode level. We then had developed an analysis tool able to exploit these traces to compute relevant software metrics. We had hired Damien Bodenes as software developer, and had begun the work on a prototype able to render a 3D world, symbolizing software, onto various visualization hardware, with the possibility to change the display metaphor. The main part of our development work had been in 2009 the choice and validation of the technology, and a first architecture. In 2010, the development had go on at a good pace, building on chosen technologies and architecture. This had brought new experience, and with the first actual runs of our platform, we had realized that with the Irrlicht platform we had chosen, we could reach unforeseeable problem when scaling up. We had thus decided to reverse our choice to the Ogre3D 3D engine at the beginning of 2010. Our development had then progressed steadily.

We had released in 2010 a first prototype of our platform, with all the underlying architecture, able to provide navigation features and interaction capacities limited to the driving of the navigation, as per our plans. This had included dual screen management.

Our first prototype, using 2 large 2D screens, with a city metaphor, had been demonstrated during the "Fête de la Science" in November 2010 and had received a lot of attention and enthusiasm from the general public. About 55 persons per day had visited our booth and got demonstrations.

We had also progressed significantly in our Java bytecode tracer, by improving its granularity, the completeness of the traced information, and its performance as well. We have a unique tool which is able to trace both program classes and JDK classes, at basic block level. In addition, it does so with a dynamic instrumentation of classes, which means there is no need to have an instrumented version of the class files on disk. This is very convenient, especially when changing machine of JVM, or when upgrading either the JDK or the program itself. In addition, the performance is good enough that the instrumented programs are still fully usable in an interactive way, without bothering the user. To the best of our knowledge, this is the only Java bytecode tracer that offers these features nowadays.

Our software development had lead to several registrations with APP:

- VITRAIL - Visualizer had been first registered on 29/12/2009 under number IDDN.FR.001.530021.000.S.P.2009.000.10000.
- VITRAIL - Tracer, was registered at APP on 20/09/2010 with number IDDN.FR.001.380001.000.S.P.2010.000.10000.

In 2011, we acquired a workstation and three 30 inches computer screens, to be able to set up a "boxed 3D workstation", that would provide display in front and on both sides of the operator. This would constitute the next step in our experiments, by improving immersion with a larger field of vision (on the sides). The software developments to do this are ongoing. We also integrated a WiiMote interaction device to our system, but our experiments found that its spacial resolution was too poor for our needs.

We finally improved significantly our VITRAIL prototype in 2011, especially by designing and implementing a new representation for the relations between software (hence visual) elements, with limited clutter and the possibility to regroup links and see their direction.
5. Software

5.1. QI: Quadrics Intersection

QI stands for “Quadrics Intersection”. QI is the first exact, robust, efficient and usable implementation of an algorithm for parameterizing the intersection of two arbitrary quadrics, given in implicit form, with integer coefficients. This implementation is based on the parameterization method described in [10], [32], [33], [34] and represents the first complete and robust solution to what is perhaps the most basic problem of solid modeling by implicit curved surfaces.

QI is written in C++ and builds upon the LiDIA computational number theory library [27] bundled with the GMP multi-precision integer arithmetic [26]. QI can routinely compute parameterizations of quadrics having coefficients with up to 50 digits in less than 100 milliseconds on an average PC; see [10] for detailed benchmarks.

Our implementation consists of roughly 18,000 lines of source code. QI has being registered at the Agence pour la Protection des Programmes (APP). It is distributed under the free for non-commercial use INRIA license and will be distributed under the QPL license in the next release. The implementation can also be queried via a web interface [28].

Since its official first release in June 2004, QI has been downloaded six times a month on average and it has been included in the geometric library EXACUS developed at the Max-Planck-Institut für Informatik (Saarbrücken, Germany). QI is also used in a broad range of applications; for instance, it is used in photochemistry for studying the interactions between potential energy surfaces, in computer vision for computing the image of conics seen by a catadioptric camera with a paraboloidal mirror, and in mathematics for computing flows of hypersurfaces of revolution based on constant-volume average curvature.

5.2. Isotop: Topology and Geometry of Planar Algebraic Curves

ISOTOP is a Maple software for computing the topology of an algebraic plane curve, that is, for computing an arrangement of polylines isotopic to the input curve. This problem is a necessary key step for computing arrangements of algebraic curves and has also applications for curve plotting. This software has been developed since 2007 in collaboration with F. Rouillier from INRIA Paris - Rocquencourt. It is based on the method described in [31] which incorporates several improvements over previous methods. In particular, our approach does not require generic position.

Isotop is registered at the APP (June 15th 2011) with reference IDDN.FR.001.240007.000.S.P.2011.000.10000. This version is competitive with other implementations (such as ALC1X and INSULATE developed at MPII Saarbrücken, Germany and TOP developed at Santander Univ., Spain). It performs similarly for small-degree curves and performs significantly better for higher degrees, in particular when the curves are not in generic position.

We are currently working on an improved version integrating our new bivariate polynomial solver [22].

5.3. CGAL: Computational Geometry Algorithms Library

Born as a European project, CGAL (http://www.cgal.org) has become the standard library for computational geometry. It offers easy access to efficient and reliable geometric algorithms in the form of a C++ library. CGAL is used in various areas needing geometric computation, such as: computer graphics, scientific visualization, computer aided design and modeling, geographic information systems, molecular biology, medical imaging, robotics and motion planning, mesh generation, numerical methods...
In computational geometry, many problems lead to standard, though difficult, algebraic questions such as computing the real roots of a system of equations, computing the sign of a polynomial at the roots of a system, or determining the dimension of a set of solutions. We want to make state-of-the-art algebraic software more accessible to the computational geometry community, in particular, through the computational geometric library CGAL. On this line, we contributed a model of the Univariate Algebraic Kernel concept for algebraic computations [30] (see Sections 8.2.2 and 8.4). This CGAL package improves, for instance, the efficiency of the computation of arrangements of polynomial functions in CGAL [36]. We are currently developing a model of the Bivariate Algebraic Kernel based on our new bivariate polynomial solver [22]. This work is done in collaboration with F. Rouillier at INRIA Paris - Rocquencourt and L. Peñaranda at the university of Athens.
5. Software

5.1. The veriT solver

Participants: Diego Caminha Barbosa de Oliveira, David Déharbe, Pascal Fontaine [correspondant], Bruno Woltzenlogel Paleo.

The veriT solver is an SMT (Satisfiability Modulo Theories) solver developed in cooperation with David Déharbe from the Federal University of Rio Grande do Norte in Natal, Brazil. The solver can handle large quantifier-free formulas containing uninterpreted predicates and functions, and arithmetic on integers and reals. It features a very efficient decision procedure for difference logic, as well as a simplex-based reasoner for full linear arithmetic. It also has some support for user-defined theories, quantifiers, and lambda-expressions. This allows users to easily express properties about concepts involving sets, relations, etc. The prover can produce an explicit proof trace when it is used as a decision procedure for quantifier-free formulas with uninterpreted symbols and arithmetic. To support the development of the tool, a regression platform using INRIA’s grid infrastructure is used; it allows us to extensively test the solver on thousands of benchmarks in a few minutes.

The veriT solver is available as open source under the BSD license, and distributed through the web site http://www.veriT-solver.org. It entered for the third time the international competition of SMT solvers SMT-COMP 2011, a joint event with the SMT workshop 2011 and the CAV conference. As in the previous competitions, it performed decently against the other participating SMT solvers. It embeds an original symmetry reduction technique that greatly improved its efficiency on some categories of formulas. This technique was immediately incorporated also in other competing solvers, in particular Z3 (Microsoft) and CVC3 (University of New-York and University of Iowa).

Efforts in 2011 have been focused on the extension of the expressiveness of the tool (with improvements in the handling of quantifiers), and on its efficiency (which was significantly improved at different levels, including a purpose-built SAT solver underlying veriT). A lot of work was also devoted to improve the proof production of the tool, with the definition of a precise proof language. This proof language has been presented to the community as a standard for describing SMT proofs [17]. We are collaborating on this with Laurent Théry and Benjamin Grégoire (Marelle, INRIA Sophia-Antipolis), Laurent Voisin (Systerel), and Frédéric Besson (Celtique, INRIA Rennes).

Future research and implementation efforts will be directed to furthermore extend the accepted language, and increase the efficiency. We target applications where validation of formulas is crucial, such as the validation of TLA+ and B specifications, and work together with the developers of the respective verification platforms to make veriT even more useful in practice.

The software will be supported by an INRIA ADT, which will start at the beginning of 2012.

5.2. The TLA+ proof system

Participants: Stephan Merz, Hernán-Pablo Vanzetto.

TLAPS, the TLA+ proof system, is a platform for developing and mechanically verifying TLA+ proofs. It is developed at the Joint MSR-INRIA Centre. The TLA+ proof language is declarative and based on standard mathematical logic; it supports hierarchical and non-linear proof construction and verification. TLAPS consists of a proof manager that interprets the proof language and generates a collection of proof obligations that are sent to backend verifiers that include theorem provers, proof assistants, SMT solvers, and decision procedures.
TLAPS is publically available at http://msr-inria.inria.fr/~doligez/tlaps/, it is distributed under a BSD-like license. It handles the non-temporal part of TLA+ with the exception of computing enabledness predicates and can currently be used to prove safety, but not liveness properties. Its backends include a tableau prover for first-order logic, an encoding of TLA+ in the proof assistant Isabelle, as well as an SMT translation and a custom decision procedure for Presburger arithmetic. Our main contribution in 2011 has been the implementation of a new SMT backend that handles formulas including linear arithmetic, elementary set theory, functions, tuples, and records (see section 6.4). Other efforts in 2011 concerned improvements and stabilization of the fingerprinting technique that avoids re-proving proof obligations that have remained unchanged since a previous prover run.
5. Software

5.1. SeLaLib

SeLaLib (the Semi-Lagrangian Library) is a library providing numerical methods for the kinetic models of plasma physics, in particular different types of Vlasov equations including the gyrokinetic model coupled to field solvers based on the Poisson, Maxwell or gyrokinetic quasi-neutraly equations. The PDEs are solved on structured mapped meshes or a collection of patches of such meshes, where the meshes are defined by a mapping for a cartesian logical grid. It is developed with an ADT and is strongly related to the INRIA large scale initiative Fusion.

One of its aims is to provide numerical building blocks for the GYSELA code developed at CEA Cadarache in collaboration with the Calvi project-team. GYSELA is used by physicists for simulating the development of turbulence in magnetic fusion plasmas in particular in view of the ITER project.
5. Software

5.1. Simulation of viscous fluid-structure interactions

Participants: Takéo Takahashi [correspondant], Jean-François Scheid, Jérôme Lohéac.

A number of numerical codes for the simulation for fluids and fluid-structure problems has been developed by the team. These codes are mainly written in MATLAB Software with the use of C++ functions in order to improve the sparse array process of MATLAB. We have focused our attention on 3D simulations which require large CPU time resources as well as large memory storage. In order to solve the 3D Navier-Stokes equations which model the viscous fluid, we have implemented an efficient 3D Stokes sparse solver for MATLAB and a 3D characteristics method to deal with the nonlinearity of Navier-Stokes equations. This year, we have also started to unify our 2D fluid-structure codes (fluid alone, fluid with rigid bodies and fluid with fishes).

Another code has been developed in the case of self-propelled deformable object moving into viscous fluid. Our aim is to build a deformable ball which could swim in a viscous fluid. In order to do this we have started a collaboration with a team from the CRAN (Research Centre for Automatic Control). This software solves numerically 3D Stokes equations using finite elements methods. The source code is written for use with MATLAB thanks to a C++ library developed by ALICE.

- Version: v0.5
- Programming language: MATLABc++
5. Software

5.1. CarbonQuant

Participants: Mireille Bossy [correspondant], Jacques Morice, El Hadj Aly Dia.

CarbonQuant is a simulator project of CO2 allowances prices on a EU-ETS type market, by an indifference price approach.

It aims to demonstrate the high potentiality of stochastic control solvers, to quantify sensibilities of a carbon market with respect to its design.

Starting in September 2011, CarbonQuant is an ADT\textsuperscript{1} Inria.

See also the web page \url{http://carbonvalue.gforge.inria.fr}.

- Version: 0.1
5. Software

5.1. Identification of biological systems

We are currently considering the possibility to implement our Matlab algorithms into the Matlab toolbox _Contsid_, developed by the System Identification team of the CRAN (http://www.iris.cran.uhp-nancy.fr/contsid/).
5. Software

5.1. Spiking neural networks simulation
Participants: Mohamed-Ghaïth Kaabi, Dominique Martinez.
A spiking neuron is usually modeled as a differential equation describing the evolution over time of its membrane potential. Each time the voltage reaches a given threshold, a spike is sent to other neurons depending on the connectivity. A spiking neural network is then described as a system of coupled differential equations. For the simulation of such a network we have written two simulation engines: (i) Mvaspike based on an event-driven approach and (ii) sirene based on a time-driven approach.

- Mvaspike: The event-driven simulation engine was developed in C++ and is available on http://mvaspike.gforge.inria.fr. Mvaspike is a general event-driven purpose tool aimed at modeling and simulating large, complex networks of biological neural networks. It allows to achieve good performance in the simulation phase while maintaining a high level of flexibility and programmability in the modeling phase. A large class of spiking neurons can be used ranging from standard leaky integrate-and-fire neurons to more abstract neurons, e.g. defined as complex finite state machines.

- Sirene: The time-driven simulator engine was written in C and is available on http://sirene.gforge.inria.fr. It has been developed for the simulation of biologically detailed models of neurons—such as conductance-based neurons—and synapses. Its high flexibility allows the user to implement easily any type of neuronal or synaptic model and use the appropriate numerical integration routine (e.g. Runge-Kutta at given order).

5.2. DANA: Implementation of computational neuroscience mechanisms
Participants: Nicolas Rougier, Mathieu Lefort, Wahiba Taouali.
Computational neuroscience is a vast domain of research going from the very precise modeling of a single spiking neuron, taking into account ion channels and/or dendrites spatial geometry up to the modeling of very large assemblies of simplified neurons that are able to give account of complex cognitive functions. DANA attempts to address this latter modeling activity by offering a Python computing framework for the design of very large assemblies of neurons using numerical and distributed computations. However, there does not exist something as a unified model of neuron: if the formal neuron has been established some sixty years ago, there exists today a myriad of different neuron models that can be used within an architecture. Some of them are very close to the original definition while some others tend to refine it by providing extra parameters or variables to the model in order to take into account the great variability of biological neurons. DANA makes the assumption that a neuron is essentially a set of numerical values that can vary over time due to the influence of other neurons and learning. DANA aims at providing a constrained and consistent Python framework that guarantee this definition to be enforced anywhere in the model, i.e., no symbol, no homunculus, no central executive.

5.3. ENAS: Event Neural Assembly Simulation
Participants: Frédéric Alexandre, Axel Hutt, Nicolas Rougier, Thierry Viéville.
EnaS (that stands for “Event Neural Assembly Simulation”) is a middleware implementing our last numerical and theoretical developments, allowing to simulate and analyze so called “event neural assemblies”. The recent achievements include (in collaboration with the Neuromathcomp EPI): spike trains statistical analysis via Gibbs distributions, spiking network programing for exact event’s sequence restitution, discrete neural field parameters algorithmic adjustments and time-constrained event-based network simulation reconciling clock and event based simulation methods. It has been designed as plug-in for our simulators (e.g. DANA or Mvaspik) as other existing simulators (via the NeuralEnsemble meta-simulation platform) and additional modules for computations with neural unit assembly on standard platforms (e.g. Python or the Scilab platform).

5.4. OpenViBE

Participants: Laurent Bougrain, Baptiste Payan.

OpenViBE is a C++ open-source software devoted to the design, test and use of Brain-Computer Interfaces. The OpenViBE platform consists of a set of software modules that can be integrated easily and efficiently to design BCI applications. Key features of the platform are its modularity, its high-performance, its portability, its multiple-users facilities and its connection with high-end/Virtual Reality displays. The “designer” of the platform enables to build complete scenarios based on existing software modules using a dedicated graphical language and a simple Graphical User Interface (GUI). This software is available on the INRIA Forge under the terms of the LGPL-V2 license. The development of OpenVibe is done in association with the INRIA research team BUNRAKU for the national INRIA project: ADT LOIC (cf. § 7.2).

5.5. CLONES: Closed-Loop Neural Simulations

Participant: Thomas Voegtlin.

The goal of this work is to provide an easy-to-use framework for closed-loop simulations, where interactions between the brain and body of an agent are simulated.

We developed an interface between the Sofa physics engine, (http://www.sofa-framework.org) and the Brian neural simulator (http://www.briansimulator.org). The interface consists in a Sofa plugin and a Python module for Brian. Sofa and Brian use different system processes, and communicate via shared memory. Synchronization between processes is achieved through semaphores.

As a demonstration of this interface, a physical model of undulatory locomotion in the nematode c. elegans was implemented, based on the PhD work of Jordan H. Boyle.

CLONES was presented at the Python in Neuroscience Workshop [18].

5.6. GINNet-DynNet: Decision-making platform

Participants: Laurent Bougrain, Marie Tonnelier.

GINNet (Graphical Interface for Neural Networks) is a decision-aid platform written in Java, intended to make neural network teaching, use and evaluation easier, by offering various parametrizations and several data pre-treatments. GINNet is based upon a local library for dynamic neural network developments called DynNet. DynNet (Dynamic Networks) is an object-oriented library, written in Java and containing base elements to build neural networks with dynamic architecture such as Optimal Cell Damage and Growing Neural Gas. Classical models are also already available (multi-layer Perceptron, Kohonen self-organizing maps, ...). Variable selection methods and aggregation methods (bagging, boosting, arcing) are implemented too.

The characteristics of GINNet are the following: Portable (100% Java), accessible (model creation in few clicks), complete platform (data importation and pre-treatments, parametrization of every models, result and performance visualization). The characteristics of DynNet are the following: Portable (100% Java), extensible (generic), independent from GINNet, persistent (results are saved in HML), rich (several models are already implemented), documented.
This platform is composed of several parts:

1. Data manipulation: Selection (variables, patterns), descriptive analysis (stat., PCA..), detection of missing, redundant data.

2. Corpus manipulation: Variable recoding, permutation, splitting (learning, validation, test sets).


6. Results: Error curves, confusion matrix, confidence interval.

DynNet and GINNet are free softwares, registrated to the APP and distributed under CeCILL license, Java 1.4 compatible (http://ginnet.gforge.inria.fr). GINNet is available as an applet. For further information, see http://gforge.inria.fr/projects/ginnet (news, documentations, forums, bug tracking, feature requests, new releases...).
MASAIE Project-Team (section vide)
ALGORILLE Project-Team

5. Software

5.1. parXXL

Participants: Jens Gustedt, Stéphane Vialle.

parXXL is a library for large scale computation and communication that executes fine grained algorithms (computation and communication are of the same order of magnitude) on coarse grained architectures (clusters, grids, mainframes). Historically, parXXL is the result of a collaboration between INRIA and SUPÉLEC. This library fulfills the requirements of our model PRO, i.e., it uses an alternation of computation and communication steps. It realizes an abstraction layer between the algorithm as it was designed and its realization on different architectures and different modes of communication. The current version of this library has been registered at the APP and is available at http://parxxl.gforge.inria.fr/. It integrates a layer for message passing with MPI, a layer for shared memory with POSIX threads, a layer for out-of-core management with file mapping (system call mmap).

All three different realizations of the communication layers are quite efficient. They let us execute programs that are otherwise unchanged within the three different contexts. Usually, they reach the performance of programs that are directly written for a given context. Generally they outperform programs that are executed in a different context than they were written for, such as MPI programs that are executed on a shared memory mainframe, or such as multi-threaded programs that are executed on a distributed shared memory machine.

5.2. Distem

Participants: Tomasz Buchert, Emmanuel Jeanvoie, Lucas Nussbaum, Luc Sarzyniec.

Distem is a distributed systems emulator. In the context of research on Cloud, P2P, High Performance Computing or Grid systems, it can be used to transform an homogeneous cluster (composed of identical nodes) into an experimental platform where nodes have different performance, and are connected together through a complex network topology, thus facilitating the evaluation or benchmarking of applications targeting such environments.

Distem relies on modern Linux features (LXC, cgroups, cpufreq, iptables, traffic control) to steal resources from applications. At the node level, it provides the ability to introduce heterogeneity by splitting a multi-core node into several several virtual nodes of varying number of cores and CPU frequency. At the network level, it allows the user to describe and build virtual network topologies where each link has a given latency, and bandwidth limit.

Distem is controlled through a REST API to ease its integration into experiment scripts, but also provides a Ruby library and a command-line interface.

It has been registered with the APP, and is freely available under the GNU GPL.

More information is available from http://distem.gforge.inria.fr/.

5.3. Wrekavoc

Participants: Jens Gustedt, Lucas Nussbaum, Tomasz Buchert.

Wrekavoc addresses the problem of controlling the heterogeneity of a cluster to provide a configurable environment that allows for reproducible experiments on large sets of configurations using real applications with no emulation of the code. Work on Wrekavoc has stopped: current works are based on the Distem emulator.
5.4. SimGrid

**Participants:** Pierre-Nicolas Clauss, El Mehdi Fekari, Martin Quinson, Lucas Nussbaum, Cristian Rosa, Christophe Thiéry.

The SimGrid framework aims at being a scientific instrument to the evaluation of algorithmic solutions for large-scale distributed experiments. It is the result of a collaboration with Henri Casanova (Univ. of Hawaii, Manoa) and Arnaud Legrand (MESCAL team, INRIA Grenoble-Rhône-Alpes, France). Simulation is a common answer to the grid specific challenges such as scale and heterogeneity. SimGrid is one of the major simulators in the Grid community.

The main strong point of this is its carefully assessed **model validity**. To this end, the simulation kernel relies on a blend of analytical models and coarse-grain discrete event simulation. It proves several orders of magnitude faster than usual packet-level simulators used in the networking community (such as ns2 or GTNetS) while providing a good level of accuracy [43].

The SimGrid framework is currently extremely **fast**. Independent authors demonstrated its superior scalability over its main concurrence [36], [38]. In addition to the efficiency of the simulation models, this **scalability** is ensured by a layered architecture, with a simulation kernel computing the time taken by actions which need to consume resources to complete. Another layer of abstraction introduces the notion of processes and network routing between hosts. On top of this come the user interfaces aiming at providing the syntactic sugar easing the tool usage.

Several such user interfaces exist, ensuring the **versatility** of the SimGrid framework by adapting to the user goal: **MSG** helps the study of distributed heuristics. This is the historical interface of SimGrid, and remains the most used one. **SMPI** is a new interface which allows the simulation of MPI programs designed for multi-processor systems on a single computer [4]. **SimDag** eases the study of scheduling heuristics for DAGs of (parallel) tasks, which helps the work on parallel task scheduling. **GRAS** (Grid Reality And Simulation) eases the development of Grid services and infrastructures [39] through a specific interface implemented twice: once on top of the simulator for the comfort of development, and once using regular sockets for live deployments.

SimGrid can be freely downloaded from its web page and its user base is rapidly growing. Over the last decade, it grounded the experimental section of more than hundred scientific publications, not counting the ones being co-authored by members of the development team.

5.5. ORWL

**Participant:** Jens Gustedt.

ORWL is a reference implementation of the Ordered Read-Write Lock tools as described in [3]. It implements interfaces for locking and data management that easily allow to have an overlap between communication and computation. The main tool here is the introduction of a “handle” on a local or remote resource that can be used to trigger asynchronous prefetching of control and/or data. Also it implements a second layer of abstraction for the seamless programming of iterative tasks. With that layer iterative algorithms can be implemented that have guarantees for equity and deadlock-freeness.

ORWL is a standalone library that works on shared memory and in distributed settings. The implementation is uniquely based on C99 and POSIX. ORWL has already been registered at the APP. Final tests and benchmarks are on the way to ensure the quality of the implementation before it will be made publicly available.

5.6. P99

**Participant:** Jens Gustedt.

P99 is a toolbox of header files designated to ease programming in C, in particular modern C99. Originally, these macro definitions and tools for programming in C99 have been implemented for ORWL, but now they are separated out into a separate toolbox.
This toolbox allows e.g. the simplified use of variable length argument list for macros and functions, default arguments of functions, compile time code unrolling, scope bound resource management, transparent allocation and initialization. It has been registered at the APP and is available at http://p99.gforge.inria.fr/.
5. Software

5.1. Voip bots

Participants: Mohamed Nassar [contact], Olivier Festor.

VoIPbot is a VoIP security tool created as a demonstrator of how attacks can be launched against VoIP/SIP services and users in a remotely and distributed manner. The environment contains bots that can be remotely managed over an Internet Relay Chat (IRC) channel from a central manager. Our bots are currently able to perform the following tasks:

- send SPAM over IP Telephony (SPIT),
- distributed denial of service through intensive generation of invite messages to a target device,
- active scanning of users through incremental options messages issuance to servers and response analysis,
- cracking through brute-force testing of passwords against an identified user account,
- simple device scanning and fingerprinting,
- target aware device fuzzing.

The tool is developed using the Java programming language. It uses the JAIN-SIP, JMF and PIRCBOT libraries. The tool is distributed under a GPL2 Open Source license. Reports show its use mainly in the testing business so far.

5.2. SecSIP

Participants: Abdelkader Lahmadi [contact], Olivier Festor.

SecSip\(^2\) is developed by the team to defend SIP-based (The Session Initiation Protocol) services from known vulnerabilities. It presents a proactive point of defense between a SIP-based network of devices (servers, proxies, user agents) and the open Internet. Therefore, all SIP traffic is inspected and analyzed against authored Veto specification before it is forwarded to these devices. When initializing, the SecSIP runtime starts loading and parsing authored VeTo blocks to identify different variables, event patterns, operations and actions from each rule. It implements an input and output layer, to capture, inject, send and receive SIP packets from and to the network. Intercepted packets are moved to the SIP Packet parser module. The main function of this module is to extract different fields within a SIP message and trigger events specified within the definition blocks. During each execution cycle when a SIP message arrives, the SecSIP runtime uses a data flow acyclic graph network to find definition matching rules and triggers defined events. The paired events in each operator node are propagated over the graph until a pattern is satisfied. When the pattern is satisfied, the respective rule is fired and the set of actions is executed.

SecSIP is freely available on the Internet and has been demonstrated in various High Security Labs exhibits in 2011.

5.3. NDPMon

Participants: Isabelle Chrisment, Olivier Festor [contact].

\(^2\) [http://secsip.gforge.inria.fr/doku.php](http://secsip.gforge.inria.fr/doku.php)
The Neighbor Discovery Protocol Monitor (NDPMon) is an IPv6 implementation of the well-known ArpWatch tool. NDPMon monitors the pairing between IPv6 and Ethernet addresses (NDP activities: new station, changed Ethernet address, flip flop...). NDPMon also detects attacks on the NDP protocol, as defined in RFC 3756 (bogon, fake Router Advertisements...). New attacks based on the Neighbor Discovery Protocol and Address Auto-configuration (RFC 2461 and RFC 2462) have been identified and integrated in the tool. An XML file describes the default behavior of the network, with the authorized routers and prefixes, and a second XML document containing the neighbors database is used. This second file can be filled during a learning phase. All NDP activities are logged in the syslog utility, and so the attacks, but these ones are also reported by mail to the administrator. Finally, NDPMon can detect stack vulnerabilities, like the assignment of an Ethernet broadcast address on an interface.

NDPMon comes along with a WEB interface acting as a GUI to display the informations gathered by the tool, and give an overview of all alerts and reports. Thanks to color codes, the WEB interface makes possible for the administrator to have an history of what happened on his network and identify quickly problems. All the XML files used or produced by the daemon (neighbor cache, configuration file and alerts list) are translated in HTML via XSL for better readability. A statistic module is also integrated and gives informations about the discovery of the nodes and their type (MAC manufacturer distribution ...).

The software package and its source code is freely distributed under an opensource license (LGPL). It is implemented in C, and is available through a SourceForge project at http://ndpmon.sf.net. An open source community is now established for the tool which has distributions for several Operating Systems (Linux, FreeBSD, OpenBSD, NetBSD and Mac OS X). It is also integrated in FreeBSD ports at http://www.freebsd.org/cgi/cvsweb.cgi/ports/net-mgmt/ndpmon/. Binary distributions are also available for .deb and .rpm based Linux flavors.

5.4. AA4MM

Participants: Laurent Ciarletta, Julien Siebert [main developer].

This work has been undertaken in a joint PhD Thesis between the Madynes and MAIA Teams. Vincent Chevrier (MAIA team, LORIA) has been the co-advisor of this PhD and correspondent for this software.

AA4MM (Agents and Artefacts for Multi-modeling and Multi-simulation) is a framework for coupling existing and heterogeneous models and simulators in order to model and simulate complex systems. This is the first implementation of the AA4MM meta-model proposed in Julien Siebert’s PhD. It is written in Java and relies upon Java Messaging Services (JMS) for its distributed version.

AA4MM can be downloaded at http://www.loria.fr/~siebertj/aa4mm/index.html.

5.5. MASDYNE

Participants: Laurent Ciarletta, Julien Siebert [main developer].

This work is undertaken in a joint PhD Thesis between the Madynes and MAIA Teams. Vincent Chevrier (MAIA team, LORIA) has been co-advisor of this PhD and correspondent for this software.

Other contributors to this software are: Tom Leclerc (Madynes), Francois Klein, Christophe Torin, Marcel Lamenu, Guillaume Favre and Amir Toly.

MASDYNE (Multi-Agent Simulator of DYnamic Networks usErS) is a multi-agent simulator for modeling and simulating users behaviors in mobile ad hoc network. This software is part of joint work with MAIA team, on modeling and simulation of ubiquitous networks.

It has been notably coupled with a network simulator (JANE : Java Adhoc Network Development Environment) to advanced behavior capabilities to standard network simulations.
5. Software

5.1. QualiPSo Factory: Next Generation Forge

Participants: Gérald Oster [contact], Jérôme Blanchard, Christophe Bouthier.

The QualiPSo Factory ¹ is a next generation forge based on Service Oriented Architecture developed within the QualiPSo european project ². Forges transform foreigners into collaborators, sometimes into developers. Forges are online services that allow instantiation, composition and management of collaborative services. Traditionally, provided collaborative services are version control systems, issue trackers, forums, mailing lists or wikis. In the framework of the european QualiPSo project, we are designing and implementing the next generation of forges. The QualiPSo factory framework aims to ease collaboration between forge users and integration of new collaborative services by developers. Our proposal relies on a software oriented architecture (SOA) and thereby allows composition of services. The current architecture provides core services such as security, notification, indexation, composition and naming which are externalized to other collaborative services. The Factory has been delivered as an outcome of the Qualipso project. Its future needs to be clarified.

¹ http://qualipso.gforge.inria.fr/
² http://www.qualipso.org
5. Software

5.1. Graphite

Participants: Phuong Ho, Bruno Lévy, David Lopez, Romain Merland, Vincent Nivoliers, Jeanne Pellerin, Nicolas Ray.

Graphite is a research platform for computer graphics, 3D modeling and numerical geometry. It comprises all the main research results of our “geometry processing” group. Data structures for cellular complexes, parameterization, multi-resolution analysis and numerical optimization are the main features of the software. Graphite is publicly available since October 2003. It is hosted by Inria GForge since September 2008 (1000 downloads in two months). Graphite is one of the common software platforms used in the frame of the European Network of Excellence AIMShape.

5.2. MicroMegas

Participant: Samuel Hornus.

Micromegas is a 3D modeler, developed as a plugin of Graphite, dedicated to molecular biology. Micromegas is developed in cooperation with the Fourmentin Guilbert foundation. Biologists need simple spatial modeling tools to help in understanding the role of objects’ relative position in the functioning of the cell. In this context, we offer a tool for easy DNA modeling. The tool generates DNA along a Bézier curve, open or closed, allows fine-tuning of atoms’ position and, most importantly, exports to PDB.

5.3. OpenNL - Open Numerical Library

Participants: Thomas Jost, Bruno Lévy, Nicolas Ray, Rhaleb Zayer.

OpenNL is a standalone library for numerical optimization, especially well-suited to mesh processing. The API is inspired by the graphics API OpenGL, this makes the learning curve easy for computer graphics practitioners. The included demo program implements our LSCM \[7\] mesh unwrapping method. It was integrated in Blender by Brecht Van Lommel and others to create automatic texture mapping methods. OpenNL is extended with two specialized modules:

- **CGAL parameterization package**: this software library, developed in cooperation with Pierre Alliez and Laurent Saborel, is a CGAL package for mesh parameterization.
- **Concurrent Number Cruncher**: this software library extends OpenNL with parallel computing on the GPU, implemented using the CUDA API.

5.4. Intersurf

Participants: Xavier Cavin, Nicolas Ray.

Intersurf is a plugin of the VMD (Visual Molecular Dynamics) software. VMD is developed by the Theoretical and Computational Biophysics Group at the Beckmann Institute at University of Illinois. The Intersurf plugin is released with the official version of VMD since the 1.8.3 release. It provides surfaces representing the interaction between two groups of atoms, and colors can be added to represent interaction forces between these groups of atoms. We plan to include in this package the new results obtained this year in molecular surface visualization by Matthieu Chavent.

5.5. Gocad

Participants: Guillaume Caumon, Nicolas Cherpeau, Bruno Lévy, Romain Merland, Jeanne Pellerin.
Gocad is a 3D modeler dedicated to geosciences. It was developed by a consortium headed by Jean-Laurent Mallet, in the Nancy School of Geology. Gocad is now commercialized by Earth Decision Sciences (formerly T-Surf), a company which was initially a start-up company of the project-team. Gocad is used by all major oil companies (Total-Fina-Elf, ChevronTexaco, Petrobras, etc.), and has become a de facto standard in geomodeling. Luc Buatois’s work on GPU-based numerical solvers is now integrated in Gocad’s grid generation software SKUA.

5.6. LibSL

Participants: Anass Lasram, Sylvain Lefebvre.

LibSL is a Simple library for graphics. Sylvain Lefebvre continued development of the LibSL graphics library (under CeCill-C licence, filed at the APP). LibSL is a toolbox for rapid prototyping of computer graphics algorithms, under both OpenGL, DirectX 9/10, Windows and Linux. The library is actively used in both the REVES / INRIA Sophia-Antipolis and the Alice / INRIA Nancy Grand-Est teams.
MAGRIT Project-Team

5. Software

5.1. Software

Our software efforts are integrated in a library called RAlib which contains our research development on image processing, registration (2D and 3D) and visualization. This library is licensed by the APP (French agency for software protection).

The visualization module is called QGLSG: it enables the visualization of images, 2D and 3D objects under a consistent perspective projection. It is based on Qt\(^1\) and OpenScenegraph\(^2\) libraries. The QGLSG library integrates innovative features such as online camera distortion correction, and invisible objects that can be incorporated in a scene so that virtual objects can cast shadows on real objects, and occlusion between virtual and real objects are easier to handle. The library was also ported to Mac OS and Windows and a full doxygen documentation was written.

\(^{1}\) http://www.trolltech.com
\(^{2}\) http://www.openscenegraph.org/projects/osg
MAIA Project-Team

4. Software

4.1. FF

**Participant:** Jörg Hoffmann [correspondant].

FF is an automatic planning system, taking as input a high-level description of the planning task in the PDDL language (planning domain definition language), and returning a plan for the task. FF was continuously developed by Jörg Hoffmann over a time span of several years (ca. 1999 – 2006), before joining INRIA. FF has convincingly won the international planning competition in the year 2000, and has been one of the most widely used and cited planning systems (around 1000 citations up to now) ever since then. It still is competitive with the state of the art today. There are several different versions, for deterministic planning with Boolean state variables, for deterministic planning with numeric state variables, for non-deterministic planning with no probabilities (all outcomes are assumed to be equally likely), and finally a version tackling a particular variant of probabilistic planning.

4.2. TorchLight

**Participant:** Jörg Hoffmann [correspondant].

TorchLight is a system for automatic domain analysis in planning. It automatically infers properties of the search space surface under a particular heuristic function, called $h^+$, that underlies most current state of the art planning systems (including FF). TorchLight examines certain structural properties of the PDDL input, and exploits a number of connections between this structure and the search space surface under $h^+$. For example, one of its outputs provides an estimate of the fraction of states that lie on local minima.

4.3. AA4MM

**Participants:** Vincent Chevrier [correspondant], Julien Siebert.

*This work is undertaken in a joint PhD Thesis between MAIA and Madynes Team. Laurent Ciarletta (Madynes team, LORIA) is co-advisor of this PhD and correspondent for this software.*

AA4MM (Agents and Artefacts for Multi-modeling and Multi-simulation) is a framework for coupling existing and heterogeneous models and simulators in order to model and simulate complex systems. This is the first implementation of the AA4MM meta-model proposed in Julien Siebert’s PhD. It is written in Java and relies upon Java Messaging Services (JMS) for its distributed version.

4.4. MASDYNE

**Participants:** Vincent Chevrier [correspondant], Julien Siebert.

*This work is undertaken in a joint PhD Thesis between MAIA and Madynes Team. Laurent Ciarletta (Madynes team, LORIA) is co-advisor of this PhD and correspondent for this software.*

Other contributors to this software are: Tom Leclerc, François Klein, Christophe Torin, Marcel Lamenu, Guillaume Favre and Amir Toly.

MASDYNE (Multi-Agent Simulator of DYnamic Networks usErs) is a multi-agent simulator for modeling and simulating users behaviors in mobile ad hoc network. This software is part of joint work with MADYNES team, on modeling and simulation of ubiquitous networks.
5. Software

5.1. Generic Symbolic KDD Systems

5.1.1. The Coron Platform

Participants: Mehdi Kaytoue [contact person], Amedeo Napoli, Yannick Toussaint.

The Coron platform [118], [95] is a KDD toolkit organized around three main components: (1) Coron-base, (2) AssRuleX, and (3) pre- and post-processing modules. The software was registered at the "Agence pour la Protection des Programmes" (APP) and is freely available. The Coron-base component includes a complete collection of data mining algorithms for extracting itemsets such as frequent itemsets, frequent closed itemsets, frequent generators. In this collection we can find APriori, Close, Pascal, Eclat, Charm, and, as well, original algorithms such as ZART, Snow, Touch, and Talky-G. The Coron-base component contains also algorithms for extracting rare itemsets and rare association rules, e.g. APriori-rare, MRG-EXP, ARIMA, and BTB. AssRuleX generates different sets of association rules (from itemsets), such as minimal non-redundant association rules, generic basis, and informative basis. In addition, the Coron system supports the whole life-cycle of a data mining task and proposes modules for cleaning the input dataset, and for reducing its size if necessary. The Coron toolkit is developed in Java, is operational, and was already used in several research projects.

5.1.2. Orion: Skycube Computation Software

Participant: Chedy Raïssi [contact person].

This program implements the algorithms described in a research paper published last year at VLDB 2010 [113]. The software provides a list of four algorithms discussed in the paper in order to compute skycubes. This is the most efficient—in term of space usage and runtime—implementation for skycube computation (see https://github.com/leander256/Orion).

5.2. Stochastic systems for knowledge discovery and simulation

5.2.1. The CarottAge system

Participants: Florence Le Ber, Jean-François Mari [contact person].

CarottAge is a data mining system, freely available (GPL license) and based on Hidden Markov Models of second order. It provides a synthetic representation of temporal and spatial data. CarottAge is currently used by INRA researchers interested in mining the changes in territories related to the loss of biodiversity (projects ANR BiodivAgrim and ACI Ecoger) and/or water contamination.

In these practical applications, the system aims at building a partition—called the hidden partition—in which the inherent noise of the data is withdrawn as much as possible. The CarottAge system takes into account: (i) the various shapes of the territories that are not represented by square matrices of pixels, (ii) the use of pixels of different size with composite attributes representing the agricultural pieces and their attributes, (iii) the irregular neighborhood relation between those pixels, (iv) the use of shape files to facilitate the interaction with GIS (geographical information system).

CarottAge has been used for mining hydromorphological data. Actually a comparison was performed with three other algorithms classically used for the delineation of river continuaums and CarottAge proved to give very interesting results for that purpose [73].
5.2.2. The ARPEnTAge system

Participants: Florence Le Ber, Jean-François Mari [contact person].

ARPEnTAge 3 (for Analyse de Régularités dans les Paysages: Environnement, Territoires, Agronomie is a software based on stochastic models (HMM2 and Markov Field) for analyzing spatiotemporal data-bases [73]. ARPEnTAge is built on top of the CarottAge system to fully take into account the spatial dimension of input sequences. It takes as input an array of discrete data in which the columns contain the annual land-uses and the rows are regularly spaced locations of the studied landscape. Displaying tools and the generation of shape files have also been defined.

We model the spatial structure of the landscape by a Markov Random Field (MRF) whose sites are random Land Uses (LUS) located in the parcels. The dynamics of these LUS are modelled by a temporal HMM2. This leads to the definition of a MRF where the underlying mean field is approximated by a HMM2 that processes a Hilbert-Peano fractal curve spanning the image. This MRF is used to segment the landscape into patches, each of them being characterized by a temporal HMM2. The patch labels, together with the geographic coordinates, determine a clustered image of the landscape that can be coded within an ESRI shapefile.

ARPEnTAge is freely available (GPL license pending) and is currently used by INRA researchers interested in mining the changes in territories related to the loss of biodiversity (projects ANR BiodivAgrim and ACI Ecoger) and/or water contamination.

5.2.3. GenExp-LandSiTes: KDD and simulation

Participants: Sébastien Da Silva, Florence Le Ber [contact person], Jean-François Mari.

In the framework of the project “Impact des OGM” initiated by the French ministry of research, we have developed a software called GenExp-LandSiTes for simulating bidimensional random landscapes, and then studying the dissemination of vegetable transgenes. The GenExp-LandSiTes system is linked to the CarottAge system, and is based on computational geometry and spatial statistics. The simulated landscapes are given as input for programs such as “Mapod-Maïs” or “GeneSys-Colza” for studying the transgene diffusion. Other landscape models based on tessellation methods are under studies. The last version of GenExp allows an interaction with R and deals with several geographical data formats.

This work is now part of an INRA-INRIA project about landscape modeling, PAYOTE (2009–2011), that gathers eleven research teams of agronomists, ecologists, statisticians, and computer scientists. The PAYOTE project is now focusing on the comparison of various methods for analyzing and building temporal and spatial landscape structures. Sébastien da Silva is preparing his PhD thesis within this framework and is conducted both by Claire Lavigne (DR in ecology, INRA Avignon) and Florence Le Ber [62]. Florence Le Ber is also involved within a new INRA project on virtual landscape modelling.

5.3. KDD in Systems Biology

5.3.1. IntelliGO online

The IntelliGO measure computes semantic similarity between terms from a structured vocabulary (Gene Ontology: GO) and uses these values for computing functional similarity between genes annotated by sets of GO terms [82]. The IntelliGO measure is made available online (http://plateforme-mbi.loria.fr/intelligo/) to be used by members of the community for exploitation and evaluation purposes. It is possible to compute the functional similarity between two genes, the intra-set similarity value in a given set of genes, and the inter-set similarity value for two given sets of genes.

5.3.2. WAFOBI : KNIME nodes for relational mining of biological data

3 http://www.loria.fr/~jfmari/App/
KNIME (for “Konstanz Information Miner”) is an open-source visual programming environment for data integration, processing, and analysis. KNIME has been developed using rigorous software engineering practices and is used by professionals in both industry and academia. The KNIME environment includes a rich library of data manipulation tools (import, export) and several mining algorithms which operate on a single data matrix (decision trees, clustering, frequent itemsets, association rules...). The KNIME platform aims at facilitating the data mining experiment settings as many tests are required for tuning the mining algorithms. The evaluation of the mining results is also an important issue and its configuration is made easier.

A position of engineer (“Ingénieur Jeune Diplomé INRIA”) was granted to the Orpailleur team to develop some extra KNIME nodes for relational data mining using the ALEPH program (http://www.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/aleph.pl). The developed KNIME nodes include a data preparation node for defining a set of first-order predicates from a set of relation schemas and then a set of facts from the corresponding data tables (learning set). A specific node allows to configure and run the ALEPH program to build a set of rules. Subsequent nodes allow to test the first-order rules on a test set and to perform configurable cross validations. An INRIA APP procedure is currently pending.

5.3.3. **MOdel-driven Data Integration for Mining (MODIM)**

**Participants:** Marie-Dominique Devignes [contact person], Birama Ndiayé, Malika Smaïl-Tabbone.

The MODIM software (MOdel-driven Data Integration for Mining) is a user-friendly data integration tool which can be summarized along three functions: (i) building a data model taking into account mining requirements and existing resources; (ii) specifying a workflow for collecting data, leading to the specification of wrappers for populating a target database; (iii) defining views on the data model for identified mining scenarios. A steady-version of the software has been deposited through INRIA APP procedure in December, 2010.

Although MODIM is domain independent, it was used so far for biological data integration in various internal research studies. A poster was presented at the last JOBIM conference (Paris, June 2011). Recently, MODIM was used by colleagues from the LIFL for organizing data about non ribosomal peptide syntheses. Feedback from users led to extensions of the software. The sources can be downloaded at https://gforge.inria.fr/projects/modim/.

5.4. **Knowledge-Based Systems and Semantic Web Systems**

5.4.1. **The Kasimir System for Decision Knowledge Management**

**Participants:** Nicolas Jay, Jean Lieber [contact person], Amedeo Napoli, Thomas Meilender.

The objective of the Kasimir system is decision support and knowledge management for the treatment of cancer. A number of modules have been developed within the Kasimir system for editing of treatment protocols, visualization, and maintenance. Kasimir is developed within a semantic portal, based on OWL. KatexOWL (Kasimir Toolkit for Exploiting OWL Ontologies, http://katexowl.loria.fr) has been developed in a generic way and is applied to Kasimir. In particular, the user interface EdHibou of KatexOWL is used for querying the protocols represented within the Kasimir system.

The software CabamakA (case base mining for adaptation knowledge acquisition) is a module of the Kasimir system. This system performs case base mining for adaptation knowledge acquisition and provides information units to be used for building adaptation rules [123]. Actually, the mining process in CabamakA is implemented thanks to a frequent close itemset extraction module of the Coron platform (see § 5.1.1). A semantic wiki for the collaborative edition of decision protocols was developed and is going to be deployed.

5.4.2. **Taaable: a system for retrieving and creating new cooking recipes by adaptation**

**Participants:** Julien Cojan, Valmi Dufour-Lussier, Inaki Fernandez, Emmanuelle Gaillard, Laura Infante-Blanco, Florence Le Ber, Jean Lieber, Amedeo Napoli, Emmanuel Nauer [contact person], Yannick Toussaint.
Taaable is a system whose objectives are to retrieve textual cooking recipes and to adapt these retrieved recipes whenever needed. Suppose that someone is looking for a “leek pie” but has only an “onion pie” recipe: how can the onion pie recipe be adapted?

The Taaable system combines principles, methods, and technologies of knowledge engineering, namely case-based reasoning (CBR), ontology engineering, text mining, text annotation, knowledge representation, and hierarchical classification. Ontologies for representing knowledge about the cooking domain, and a terminological base for binding texts and ontology concepts, have been built from textual web resources. These resources are used by an annotation process for building a formal representation of textual recipes. A CBR engine considers each recipe as a case, and uses domain knowledge for reasoning, especially for adapting an existing recipe w.r.t. constraints provided by the user, holding on ingredients and dish types.

The Taaable system is available on line at http://taaable.fr. After being ranked twice second, in the 2008 and 2009 “Computer Cooking Contests” organized during the ICCBR conference, Taaable won the first price and the adaptation challenge, in 2010. In 2011, no contest was organized but the system has, however, been extended by two new features, both concerning knowledge acquisition using FCA [42]. The first feature uses FCA in order to enrich the domain ontology (especially the ingredient hierarchy), making the case retrieval more progressive and more precise [45]. The second feature uses FCA for extracting adaptation knowledge, in order to be able to better adapt a recipe to given constraints [47]. Current ongoing work on the Taaable project also includes formal representation of preparations [63].
5. Software

5.1. WinSnoori

contact : Yves Laprie (Yves.Laprie@loria.fr)

WinSnoori is a speech analysis software that we have been developing for 15 years. It is intended to facilitate the work of the scientist in automatic speech recognition, phonetics or speech signal processing. Basic functions of Snorri enable several types of spectrograms to be calculated and the fine edition of speech signals (cut, paste, and a number of filters) as the spectrogram allows the acoustical consequences of all the modifications to be evaluated. Beside this set of basic functions, there are various functionalities to annotate phonetically or orthographically speech files, to extract fundamental frequency, to pilot the Klatt synthesizer and to utilize PSOLA resynthesis.

The main improvement concerns automatic formant tracking which is now available with other tools for copy synthesis. It is now possible to determine parameters for the formant synthesizer of Klatt quite automatically. The first step is formant tracking, then the determination of F0 parameters and finally the adjustment of formant amplitudes for the parallel branch of the Klatt synthesizer enable a synthetic speech signal to be generated. The automatic formant tracking that has been implemented is an improved version of the concurrent curve formant tracking [60]. One key point of this tracking algorithm is the construction of initial rough estimates of formant trajectories. The previous algorithm used a mobile average applied onto LPC roots. The window is sufficiently large (200 ms) to remove fast varying variations due to the detection of spurious roots. The counterpart of this long duration is that the mobile average prevents formants fairly far from the mobile average to be kept. This is particularly sensitive in the case of F2 which presents low frequency values for back vowels. A simple algorithm to detect back vowels from the overall spectral shape and particularly energy levels has been added in order to keep extreme values of F2 which are relevant.

Together with other improvements reported during the last years, formant tracking enables copy synthesis. The current version of WinSnoori is available on [http://www.winsnoori.fr](http://www.winsnoori.fr).

5.2. SUBWEB

contacts : David Langlois (langlois@loria.fr) and Kamel Smaïli (smaili@loria.fr).

We published in 2007 a method which allows to align sub-titles comparable copora [61]. In 2009, we proposed an alignment web tool based on the developed algorithm. It allows to: upload a source and a target files, obtain an alignment at a sub-title level with a verbose option, and a graphical representation of the course of the algorithm. This work has been supported by CPER/TALC/SUBWEB ^2.

5.3. SELORIA

contact : Odile Mella (Odile.Mella@loria.fr).

SELORIA is a toolbox for speaker diarization.

---

The system contains the following steps:

- **Speaker change detection**: to find points in the audio stream which are candidates for speaker change points, a distance is computed between two Gaussian modeling data of two adjacent given-length windows. By sliding both windows on the whole audio stream, a distance curve is obtained. A peak in this curve is thus considered as a speaker change point.

- **Segment recombination**: too many speaker turn points detected during the previous step results in a lot of false alarms. A segment recombination using BIC is needed to recombine adjacent segments uttered by the same speaker.

- **Speaker clustering**: in this step, speech segments of the same speaker are clustered. Top-down clustering techniques or bottom-up hierarchical clustering techniques using BIC can be used.

- **Viterbi re-segmentation**: the previous clustering step provides enough data for every speaker to estimate multi-gaussian speaker models. These models are used by a Viterbi algorithm to refine the boundaries between speakers.

- **Second speaker clustering step (called cluster recombination)**: This step uses Universal Background Models (UBM) and the Normalized Cross Likelihood Ratio (NCLR) measure.

This toolbox is derived from mClust designed by LIUM.

### 5.4. ANTS

Contact: Dominique Fohr (fohr@loria.fr).

The aim of the Automatic News Transcription System (ANTS) is to transcribe radio broadcast news. ANTS is composed of five stages: broad-band/narrow-band speech segmentation, speech/music classification, speaker segmentation and clustering, detection of silences/breathing segments and large vocabulary speech recognition. The three first stages split the audio stream into homogeneous segments with a manageable size and allow the use of specific algorithms or models according to the nature of the segment.

Speech recognition is based on the Julius engine and operates in two passes: in the first pass, a frame-synchronous beam search algorithm is applied on a tree-structured lexicon assigned with bigram language model probabilities. The output of this pass is a word-lattice. In the second pass, a stack decoding algorithm using a trigram language model gives the N-best recognition sentences.

A real time version of ANTS has been developed. The transcription is done in real time on a quad-core PC.

### 5.5. JSafran

Contact: Christophe Cerisara (Christophe.Cerisara@loria.fr).

J-Safran is the “Java Syntaxico-semantic French Analyser”. Its development has started in June 2009 from the collaboration between Parole and Talaris in the context of the RAPSODIS project. It is an open-source dependency parsing platform that is dedicated to oral speech. Its main interesting features, as compared to other similar software, are:

- It is designed for both manual and semi-automatic edition of dependency graphs, as well as for fully automatic parsing. To this end, it integrates two of the best state-of-the-art automatic parsers of the litterature, the Malt Parser and the MATE parser, as well as a third experimental Maximum Entropy Markov Model-based parser developed from November 2011 in the team. It further integrates three automatic Part-of-speech taggers: the TreeTagger, the OpenNLP and MATE taggers.

- It is smoothly interfaced with the JTrans platform, thus enabling the user to directly listen to the aligned speech segments when annotating, which is an important added value to help disambiguation. The interface between both software goes well beyond simple method calls, as they both share for instance parts of the tokenization process and access a common immutable text source from the disk or on the Web.
• It supports multi-layer annotations, such as dependency relations, semantic role labeling, named entities and coreference links for instance, as well as inter-layer projection facilities.
• It offers a powerful rule-based search and tree manipulation language to transform for instance the annotation schema of a large corpus with a few commands only.
• As it is written in pure Java, it can run on any modern computer, either as a standalone application or embedded in a web page.

A description of JSafran is published in [16]. JSafran is distributed under the Cecill-C licence, and can be downloaded at http://synalp.loria.fr/?n=Research.Software

5.6. JTrans

Contact : Christophe Cerisara (Christophe.Cerisara@loria.fr).

JTrans is an open-source software for semi-automatic alignment of speech and textual corpus. It is written 100% in JAVA and exploits libraries developed since several years in our team. Two algorithms are available for automatic alignment: a block-viterbi and standard forced-alignment Viterbi. The latter is used when manual anchors are defined, while the former is used for long audio files that do not fit in memory. It is designed to be intuitive and easy to use, with a focus on GUI design. The rationale behind JTrans is to let the user control and check on-the-fly the automatic alignment algorithms. It is bundled for now with a French phonetic lexicon and French models.

Recent improvements include its integration within the JSafran platform and its release as a Java applet that can be demonstrated on web pages. During the last three months, JTrans has been downloaded about 120 times and seven users of JTrans, outside LORIA, have directly contacted the team for requests about JTrans.

JTrans is developed in the context of the CPER MISN TALC project, in collaboration between the Parole and Talaris INRIA teams, and CNRS researchers from the ATILF laboratory. It is distributed under the Cecill-C licence, and can be downloaded at http://synalp.loria.fr/?n=Research.Software

5.7. STARAP

contact : Dominique Fohr (fohr@loria.fr).

STARAP (Sous-Titrage Aidé par la Reconnaissance Automatique de la Parole) is a toolkit to help the making of sub-titles for TV shows. This toolkit performs:

• Parameterization of speech data;
• Clustering of parameterized data;
• Gaussian Mixture Models (GMM) training;
• Viterbi recognition.

This toolkit was realised in the framework of the STORECO contract and the formats of the input and output files are compatible with HTK toolkit.

5.8. TTS SoJA

contact : Vincent Colotte (Vincent.Colotte@loria.fr).

TTS SoJA (Speech synthesis platform in Java) is a software of text-to-speech synthesis system. The aim of this software is to provide a toolkit to test some steps of natural language processing and to provide a whole system of TTS based on non uniform unit selection algorithm. The software performs all steps from text to the speech signal. Moreover, it provides a set of tools to elaborate a corpus for a TTS system (transcription alignment, ...). Currently, the corpus contains 1800 sentences (about 3 hours of speech) recorded by a female speaker.
Most of the modules are developed in Java. Some modules are in C. The platform is designed to make easy the addition of new modules. The software runs under Windows and Linux (tested on Mandriva, Ubuntu). It can be launched with a graphical user interface or directly integrated in a Java code or by following the client-server paradigm.

The software license should easily allow associations of impaired people to use the software. A demo web site has been built: http://soja-tts.loria.fr

5.9. Corpus Recorder

Contact: Vincent Colotte (Vincent.Colotte@loria.fr).

Corpus Recorder is a software for the recording of audio corpora. It provides an easy tool to record with a microphone. The gain of the audio input is controlled during the recording. From a list of sentences, the output is a set of wav files automatically renamed with textual information given in input (nationality, speaker language, gender...). An easy syntactic tagging allows to display a textual context of the sentence to pronounce. This software is suitable for recording sentences with information to guide the speaker.

The software is developed in Tcl/Tk (tested under Windows and Linux). It was used for the recording of sentences for the TTS system SOJA and during the Intonale Project (Prosody Modeling).
5. Software

5.1. LEOPAR

Participants: Bruno Guillaume [correspondant], Guy Perrier, Mathieu Morey, Paul Masson.

5.1.1. Software description

LEOPAR is a parser for natural languages which is based on the formalism of Interaction Grammars [35]. It uses a parsing principle, called "electrostatic parsing" which consists in neutralizing opposite polarities. A positive polarity corresponds to an available linguistic feature and a negative one to an expected feature.

Parsing a sentence with an Interaction Grammar consists in first selecting a lexical entry for each of its words. A lexical entry is an underspecified syntactic tree, a tree description in other words. Then, all selected tree descriptions are combined by partial superposition guided by the aim of neutralizing polarities: two opposite polarities are neutralized by merging their support nodes. Parsing succeeds if the process ends with a minimal and neutral tree. As IGs are based on polarities and under-specified trees, LEOPAR uses some specific and non-trivial data-structures and algorithms.

The electrostatic principle has been intensively considered in LEOPAR. The theoretical problem of parsing IGs is NP-complete; the nondeterminism usually associated to NP-completeness is present at two levels: when a description for each word is selected from the lexicon, and when a choice of which nodes to merge is made. Polarities have shown their efficiency in pruning the search tree:

- In the first step (tagging the words of the sentence with tree descriptions), we forget the structure of descriptions, and only keep the bag of their features. In this case, parsing inside the formalism is greatly simplified because composition rules reduce to the neutralization of a negative feature-value pair \( f \leftarrow v \) by a dual positive feature-value pair \( f \rightarrow v \). As a consequence, parsing reduces to a counting of positive and negative polarities present in the selected tagging for every pair \((f, v)\): every positive occurrence counts for +1 and every negative occurrence for −1, the sum must be 0.

- Again in the tagging step, original methods were developed to filter out bad taggings. Each unsaturated polarity \( p \) in the grammar induces constraints on the set of contexts in which it can be used: the unsaturated polarity \( p \) must find a companion (i.e. a tree description able to saturated it); and the set of companions for the polarity \( p \) can be computed statically from the grammar. Each lexical selection which contains an unsaturated polarity without one of its companions can be safely removed.

- In the next step (node-merging phase), polarities are used to cut off parsing branches when their trees contain too many non neutral polarities.

5.1.2. Current state of the implementation

LEOPAR is presented and documented at http://leopar.loria.fr; an online demonstration page can be found at http://leopar.loria.fr/demo.

It is open-source (under the CECILL License http://www.cecill.info) and it is developed using the InriaGforge platform (http://gforge.inria.fr/projects/semagramme/).

The main features of current software are:

- automatic parsing of a sentence or a set of sentences,
- dependency and parse-tree representation of sentences,
- interactive parsing (the user chooses the couple of nodes to merge),
- visualization of grammars produced by XMG or of sets of description trees associated to some word in the linguistic resources,
During 2011, with the help of an engineer, the LEOPAR software was improved in several ways:

- A new graphical interface (using GTK) was designed
- New algorithms for the super-tagging step of the parsing process were implemented. These algorithms are described in [9].

5.2. ACG Development Toolkit

In order to support the theoretical work on ACG, we have been developing a support system. The objectives of such a system are twofold:

1. to make possible to implement and experiment grammars the modeling of linguistic phenomena;
2. to make possible to implement and experiment results related to the ACG formalisms. Such results can concern parsing algorithms, type extensions, language extensions, etc.

The current version of the ACG development toolkit prototype issues from a first release published in October 2008. Further releases have been published before the ESSLLI 2009 course on ACG. It focuses on providing facilities to develop grammars. To this end, the type system currently implemented is the linear core system plus the (non-linear) intuitionistic implication, and a special attention has been paid to type error management. As a major limitation, this version only considers transformation from abstract terms to object terms, and not the other way around.

Enabling transformation from the object terms to the abstract terms is the first step of future development for the ACG support system. A parsing algorithm based on [37]’s methods is being implemented for second-order ACGs. It is based on a translation of ACG grammars into Datalog programs and is well-suited to fine-grained optimization. A summer internship from ENS Cachan, Clovis Eberhart (L3) has been implementing the translation from the higher-order signatures and terms data structures to the Datalog clauses data structures. It still remains to be integrated to the main branch.

In order to allow for a larger character set as input, another extension implemented this summer by another internship from École des Mines de Nancy, Grégoire Brenon (M1) was to extend the lexer and the parser for the data files with UTF-8 capabilities (OCaml lacks such a built-in capability).

However, since we’re interested not only by recognizability (hence whether some fact is provable) but also by the parsing structure (hence the proof), the Datalog solver requires further adaptations. Note however that in the general case, the decidability of translating an object term to an abstract one is still an open problem.

5.3. GREW

Participants: Bruno Guillaume [correspondant], Guy Perrier, Mathieu Morey, Paul Masson.

Grew is a Graph Rewriting tools dedicated to applications in NLP. It was developed as a support tool during the PhD thesis of Mathieu Morey.

It is freely-available (from the page http://wikilligramme.loria.fr/doku.php?id=grew:grew ) and it is developed using the InriaGforge platform ( http://gforge.inria.fr/projects/semagramme/ )

We list below some of the major specificities of the GREW software.

- Graph structures can use a build-in notion of feature structures.
- The left-hand side of a rule is described by a graph called a pattern; injective graph morphisms are used in the pattern matching algorithm.
- Negative pattern can be used for a finer control on the left-hand side of rules.
- The right-hand side or rules is described by a sequence of atomic commands that describe how the graph should be modified during the rule application.

1Available at http://acg.gforge.inria.fr with a CeCILL license.
• Subset of rules are grouped in modules; the full rewriting process being a sequence of module applications.

• The GREW software has support both for confluent and non-confluent modules; when a non-confluent modules is used, all normal forms are returned and then ambiguity is handled in a natural way.

• GREW can be used on Corpus mode with statistics about rules usage or with an a Graphical User Interface which can show all intermediate graphs used during the rewriting process (useful either to debug rewriting system of for demonstrations).

During the last 18 months, the GREW software were used for several kind of applications manipulating syntactic and/or semantic graph representations:

• to build DMRS semantic representation from syntactic dependency trees ([26], [14], [9]);
• to enrich surface syntactic structures ([13], [9]);
• to detect annotation errors in the French Treebank.

5.4. Other developments

Participants: Bruno Guillaume [correspondant], Paul Masson.

Other peripheral developments of the team are available either as web service of as downloadable code:

• A concordancer named CONDOR. The main features of this tool are:
  – It is usable online: http://condor.loria.fr;
  – It is possible to search for all inflexions (given by a lexicon) of some words;
  – It is possible to combine two searches and to search for a couple of words to find collocations.

• A program (named DEP2PICT) to build graphical representations of dependency structures.
  – it is presented and documented at: http://dep2pict.loria.fr;
  – it is usable online at http://dep2pict.loria.fr/demo;
  – it can produce PNG, SVG and PDF output formats;
  – it can be used to represented dependency structures with chunks;
  – it support CONLL input format.
5. Software

5.1. GenI

**Participants:** Claire Gardent [correspondent], Eric Kow [developer], Carlos Areces [developer].

GenI is a surface realiser that generates sentences from first order logical formulae. It is implemented in Haskell and uses the Glasgow Haskell compiler to obtain executable code for Windows, Solaris, Linux and Mac OS X. GENI is compatible with both a grammar for French (SEMTAG) and for English (SEMXTAG), both grammars being produced using the XMG MetaGrammar Compiler. SEMTAG covers the basic syntactic structures of French as described in Anne Abeillé’s book “An Electronic Grammar for French”. SEMXTAG has a coverage similar to that of XTAG, the TAG grammar for English developed by the University of Pennsylvania. GenI is under GPL License. See also the web page http://talc.loria.fr/GenI-un-realisateur-de-surface.html.

- Version: 0.20.1

5.2. Web Service for the Multilingual-Assisted Chat Interface

**Participant:** Samuel Cruz-Lara [correspondent].

The Web Service for the Multilingual-Assisted Chat Interface program (WSMACI) is a linguistic assistant for virtual worlds. Its first version is dedicated to English assistance in such worlds. It has been developed in the context of the Metaverse1 project. It provides the end-users with MLIF-based provision of sentence analysis and word information (synonyms, definitions, translations) based on Google Translate, WordNet and the Brown Corpus.

- Version: 0.2

5.3. Emotion detection from textual information

**Participant:** Samuel Cruz-Lara [correspondent].

The 4 Layers Emotion Detection program (4LED) is an emotion detection tool. The emotions are extracted from texts in particular, from chat interfaces in virtual worlds. It has been developed in the context of the Metaverse1 project. The emotion detection process is based on SMILEY detection using WordNet-Domains and Tree-Tagger-based rules, WordNet-Affect, and keywords. http://talc.loria.fr/~metaverse/web_test/emotions/filterDetection/corpusCreation.php.

- Version: 0.2

5.4. Second Life Magic Carpet

**Participant:** Samuel Cruz-Lara [correspondent].

The Second Life Magic Carpet program (SLMC) is an assistant whose role is to guide people through virtual worlds with textual instructions. It has been developed in the context of the Metaverse1 project. It analyses the instructions of the visitors in order to find where they want to go, using web services for the analysis, for synonyms retrieving and for path finding.

- Version: 0.2

5.5. WikiAnalyzer

**Participant:** Alexandre Denis [correspondent].
The WikiAnalyzer is a tool developed in the CCCP-Prosodie project that aims to describe participants of Wikipedia projects. It provides a range of linguistic and structural analyses of Wikipedia discussion pages. The tool performs pages retrieval and automatic annotation of markers to build interactive profiles of participants. These profiles include information such as their level of expertise in the domain at hand, the use of subjective elements in their contributions, the connotation of the terms they use and enable to describe participants relative to their degree of conflictuality in the discussion. The structural analyses are parallel analyses on the structure of messages, enabling to categorize participants with regards to the type of contribution (starting a thread, participants they answer to, etc.). The tool has been developed in Java, and will be released as an online web application to the other members of the CCCP-Prosodie project.

- Version: 0.8

5.6. Emospeech Dialogue Toolkit

Participant: Lina-Maria Rojas Barahona [correspondent].

The Emospeech Dialogue Toolkit is a multi agent architecture for developing man/machine dialog systems in the context of a video game. It includes the following agents:

- Midiki Dialogue Manager: We extended and improved the open source MIDIKI (MITRE Dialogue Toolkit) software to support the multi-agent architecture and the configuration from a relational database.

- Wizard of Oz: We implemented two Wizard of OZ interfaces which allow a human to interact with other agents in the dialogue architecture. The free-wizard acts as a dialogue manager and permits a chat between two humans the player and the Wizard while simultaneously storing all interactions in a database. In contrast, the semi-automatic wizard connects the Wizard with Midiki, whereby the Wizard interprets and adjusts Midiki generation.

- Answer Selection: We trained a classifier with Conditional Random Fields that chooses the most plausible response to a player utterance.

In addition, we trained a Logistic Regression Classifier for the interpretation agent that communicates with MIDIKI.

The dialogue agents communicate with the Game Agent, Speech Recognition and/or Chatbox agents developed by the Parole team. The Wizard of Oz, in which a human simulates a dialogue system, is used to collect dialogue data which can be used for training the interpreter and/or the Answer Selection Classifier. Moreover, a Dialogue Configuration Tool has been implemented for the configuration of several dialogues for different game scenarios by configuring the characters and goals in the game and the goals to be discussed in each dialogue. (See http://talc.loria.fr:8081/EmoDial).

- Version: 1.0

5.7. IGNG-Fv2

Participant: Jean-Charles Lamirel [correspondent].

The IGNG-Fv2 program implements a new incremental clustering algorithm whose main domain of application is the statistical analysis of continuous flow of evolving textual data, as well as the one of static textual data. It has been developed in the context of the CPER TALC (McFiID action). It is based on a generic adaptation of the classical neural-based clustering approaches relying on gas of neurons with free topology. The IGNG-Fv2 approach exploits a combination of distance based and cluster data feature maximization criteria. This approach has been proved more efficient than the usual techniques for the analysis all kinds of static textual datasets. Considering its incremental character, it can also provide the information analysts with precise online detection of topic changes in the curse of a textual information flow.
5.8. **C-Quality**

**Participant:** Jean-Charles Lamirel [correspondent].

The C-Quality toolkit provides method-independent clustering quality measures and cluster labeling techniques specifically adapted to the interpretation of data analysis performed on textual data. The toolkit relies on an evaluation approach based on the exploitation of the maximized features of the data associated to each cluster after the clustering process without prior consideration of clusters profiles. The toolkit basic role is to act as an overall clustering quality evaluation tool. In a complementary way toolkit’s clusters labeling functionalities can be used altogether for visualizing or synthesizing clustering results, for optimizing learning of a clustering method, for validating cluster content and act as efficient variable selection methods in the framework of supervised or semi-supervised learning tasks.

5.9. **tl_dv2_ladl, a subcategorisation lexicon for French verbs.**

**Participant:** Ingrid Falk [correspondent].

tl_dv2_ladl is a subcategorisation lexicon for French verbs produced by merging three lexicons which were built or validated manually: Dicovalence (version 2), TreeLex and the LADL tables. tl_dv2_ladl lists subcategorisation frames for 5918 French verbs. An entry in the lexicon consists of a verb and an associated subcategorisation frame whereby each subcategorisation frame describes a set of syntactic arguments with each argument being described by a grammatical function and a syntactic category. Each entry also gives the original lexical resource the information was extracted from. tl_dv2_ladl can be downloaded from [http://talc.loria.fr/tl_dv2_ladl-a-subcategorisation.html](http://talc.loria.fr/tl_dv2_ladl-a-subcategorisation.html).

- Version: 0.1