RESEARCH CENTER

FIELD

Activity Report 2013

Section Software

Edition: 2014-03-19

ALGORITHMICS, PROGRAMMING, SOFTWARE AND ARCHITECTURE

APPLIED MATHEMATICS, COMPUTATION AND SIMULATION

COMPUTATIONAL SCIENCES FOR BIOLOGY, MEDICINE AND THE ENVIRONMENT

DIGITAL HEALTH, BIOLOGY AND EARTH

NETWORKS, SYSTEMS AND SERVICES, DISTRIBUTED COMPUTING

PERCEPTION, COGNITION AND INTERACTION

PERCEPTION, COGNITION, INTERACTION

AXIS Project-Team

5. Software and Platforms

5.1. Introduction

From its creation, AxIS has proposed new methods and software validated experimentally on various applications: Data Mining, Web Usage Mining, Information Retrieval, Activity Modeling. See Sections from 5.3 until 5.6 and our 2013 results.

In the context of the CPER Télius contract (2010-2013), AxIS has proposed to provide a Focus platform (renamed FocusLab) aiming the community of Living Labs or any researcher/actor involving in experimental project with users.

5.2. FocusLab Platform

Participant: Brigitte Trousse [co-correspondent].

Between 2010-2012 in the context of CPER Télius (cf. Section 7.1.1), we bought various hardware (eye-trackers, physiologic sensors, equilibrium platform, tablets, Arduino components, etc.) and software (Sphinx for questionnaires, Story Board for usage scenarios, Interface prototyping tools such as JustInMind, etc.) in order to observe and analyse user behaviours in supporting the design and evaluation of ICT-based services or products within a living lab approach and also to mine data. FocusLab hardware and software (under licences) were used since 2011 by Inria teams and external collaborators with success, supporting experiments or training.

Our goal was also to provide a web-based application for reserving FocusLab material (hardware, software and documentation) and to prepare the access/download of some software issued from Inria research. We started with AxIS software as a first step of the mutualised software part of the platform.

The development process of the web-based FocusLab platform started slowly in 2011, after finding some ways to fund human resources. We started by transforming some AxIS KDD methods into web services. Such a work was pursued this year (cf. Section 6.6) linked to Elliot purposes. This platform (http://focuslab.inria.fr) is based on a Service oriented Architecture.

5.3. Data Mining

5.3.1. Classification and Clustering Methods

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

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

Supervised methods

- a Java library (Somlib) that provides efficient implementations of several SOM(Self-Organizing Map) variants [44], [43], [69], [68], [73], especially those that can handle dissimilarity data (available on Inria's Gforge server (public access) Somlib, developed by AxIS Rocquencourt and Brieuc Conan-Guez from Université de Metz.
- a functional Multi-Layer Perceptron library, called FNET, that implements in C++ supervised classification of functional data [64], [67], [66], [65] (developed by AxIS Rocquencourt).

Unsupervised methods : partitioning methods

- Two partitioning clustering methods on the dissimilarity tables issued from a collaboration between AxIS Rocquencourt team and Recife University, Brazil: CDis and CCClust [77]. Both are written in C++ and use the “Symbolic Object Language” (SOL) developed for SODAS. And one partitioning method on interval data (Div).
- Two standalone versions improved from SODAS modules, SCluster and DIVCLUS-T [41] (AxIS Rocquencourt).

Unsupervised methods : agglomerative methods

- a Java implementation of the 2-3 AHC (developed by AxIS Sophia Antipolis). The software is available as a Java applet which runs the hierarchies visualization toolbox called HCT for Hierarchical Clustering Toolbox (see [3] and [42]).

A Web interface developed in C++ and running on our Apache internal Web server .is available for the following methods: SCluster, Div, Cdis, CCClust.
Previous versions of the above software have been integrated in the SODAS 2 Software [61] which was the result of the european project ASSO [6] (2001-2004). SODAS 2 supports the analysis of multidimensional complex data (numerical and non numerical) coming from databases mainly in statistical offices and administration using Symbolic Data Analysis [39]. This software is registrated at APP (Agence de la Protection des Programmes). For the latest version of the SODAS 2 software, see [60], [79].

In 2013, a new release of MND (Dynamic Clustering Method for Mixed Data) algorithm has been done based on [80] (cf. section 6.2.5) and used on clustering the user profiles and analysing user behaviour change (cf. Section 6.5.4).

5.3.2. *Extracting Sequential Patterns with Low Support*

Participant: Brigitte Trousse [correspondent].

Two methods for extracting sequential patterns with low support have been developed by D. Tanasa in his thesis (see Chapter 3 in [72] for more details) in collaboration with F. Masseglia and B. Trousse :

- **Cluster & Divide**,
- and **Divide & Discover** [8].

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

5.3.3. *Mining Data Streams*

Participant: Brigitte Trousse [correspondent].

In Marascu’s thesis (2009) [57], a collection of software have been developed for knowledge discovery and security in data streams. Three **clustering methods for mining sequential patterns (Java) in data streams** method have been developed in Java:

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

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

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

---

[6] ASSO: Analysis System of Symbolic Official data.

It has been tested on the following data:

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

In 2012 within the context of the ELLIOT contract, SCDS has been integrated as a Web service (Java version) in the first version of FOCUSLAB platform: a demonstration was made on San Rafaelle Hospital media use case at the first ELLIOT review at Brussels. We applied SCDS web service on data issued from two other use cases in Logistics (BIBA) and Green Services (Inria) [38].

The three C++ codes done for the CRE (Orange Labs) have been deposit at APP. The java code will be deposit in 2014 at APP.

5.4. Web Usage Mining

5.4.1. AWLH for Pre-processing Web Logs

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

AWLH (AxIS Web Log House) for Web Usage Mining (WUM) is issued from AxISLogMiner software which implements the mult-site log preprocessing methodology and extraction of sequential pattern with low support developed by D. Tanasa in his thesis [72] , [15] for Web Usage Mining (WUM). In the context of the Eiffel project (2008-2009), we isolated and redesigned the core of AxISlogMiner preprocessing tool (we called it AWLH) composed of a set of tools for pre-processing web log files. The web log files are cleaned before to be used by data mining methods, as they contain many noisy entries (for example, robots requests). The data are stored within a database whose model has been improved.

So AWLH offers:

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

5.4.2. ATWUEDA for Analysing Evolving Web Usage Data

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

ATWUEDA for Web Usage Evolving Data Analysis [52] [4] was developed by A. Da Silva in her thesis [52] under the supervision of Y. Lechevallier. This tool was developed in Java and uses the JRI library in order to allow the application of R which is a programming language and software environment for statistical computing functions in the Java environment.

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

Version 2. of ATWUEDA (September 2009) is available at Inria's gforge website.

The efficiency of ATWUEDA [46] has been demonstrated by applying it on real case studies such as on condition monitoring data streams of an electric power plant provided by EDF.

ATWUEDA is used by Telecom Paris Tech and EDF [4].

5.5. Information Retrieval

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

Participant: Brigitte Trousse [correspondent].

CBR*Tools [53], [54] is an object-oriented framework [55], [50] for Case-Based Reasoning which is specified with the UMT notation (Rational Rose) and written in Java. It offers a set of abstract classes to model the main concepts necessary to develop applications integrating case-based reasoning techniques: case, case base, index, measurements of similarity, reasoning control. It also offers a set of concrete classes which implements many traditional methods (closest neighbours indexing, Kd-tree indexing, neuronal approach based indexing, standards similarities measurements). CBR*Tools currently contains more than 240 classes divided in two main categories: the core package for basic functionality and the time package for the specific management of the behavioural situations. The programming of a new application is done by specialization of existing classes, objects aggregation or by using the parameters of the existing classes.

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

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

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

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

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

Participant: Brigitte Trousse [correspondent].

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

Based on Jaczynski's thesis [53], different recommender systems have been developed for supporting Web browsing, but also browsing inside a Web-based information system or for query formulation in the context of a meta search engine.

5.6. Activity Modeling

5.6.1. *K-MADe for Describing Human Operator or User Activities*

Participant: Dominique Scapin [correspondent].

K-MADe tool (Kernel of Model for Human Activity Description Environment). The K-MADe is intended for people wishing to describe, analyze and formalize the activities of human operators, of users, in environments (computerized or not), in real or simulated situation, in the field, or in the laboratory. Although all kinds of profiles of people are possible, this environment is particularly intended for ergonomics and HCI (Human Computer Interaction) specialists. It has been developed through collaboration between ENSMA (LISI XSlaboratory) and Inria.

This year we participated in the AFIHM Working Group on Task Models (http://www.gt-mdt.fr/fr/) "Groupe de Travail de l'AFIHM sur les Modèles de Tà¢ches"). Since the early work on MAD, domain modeling task is the subject of much research (particularly in the French-speaking community), in particular the definition of formalisms and tool construction, three of which are now operational and maintained: K-MADe, eCOMM and hAMSTERS, posing an alternative to CTT. Many teams use these formalisms in a variety of goals and task models occupy a place in the field of Model Driven Engineering, and support the teaching of HCI. The WG goals are to serve as a forum between research approaches, development teams and potential users, especially for non-IT users; fostering collaboration to validate approaches; encourage feedback in teaching task models; provide the French-speaking community and eventually the international community a set of centralized, shared resources about the notion of modeling tasks.

GALAAD Project-Team

5. Software and Platforms

5.1. Mathemagix, a free computer algebra environment

Participant: Bernard Mourrain.

http://www.mathemagix.org/

algebra, univariate polynomial, multivariate polynomial, matrices, series, fast algorithm, interpreter, compiler, hybrid software.

MATHEMAGIX is a free computer algebra system which consists of a general purpose interpreter, which can be used for non-mathematical tasks as well, and efficient modules on algebraic objects. It includes the development of standard libraries for basic arithmetic on dense and sparse objects (numbers, univariate and multivariate polynomials, power series, matrices, etc., based on FFT and other fast algorithms). These developments, based on C++, offer generic programming without losing effectiveness, via the parameterization of the code (*template*) and the control of their instantiations.

The language of the interpreter is imperative, strongly typed and high level. A compiler of this language is available. A special effort has been put on the embeding of existing libraries written in other languages like C or C++. An interesting feature is that this extension mechanism supports template types, which automatically induce generic types inside Mathemagix. Connections with GMP, MPFR for extended arithmetic, LAPACK for numerical linear algebra are currently available in this framework.

The project aims at building a bridge between symbolic computation and numerical analysis. It is structured by collaborative software developments of different groups in the domain of algebraic and symbolic-numeric computation.

In this framework, we are working more specifically on the following components:

- REALROOT: a set of solvers using subdivision methods to isolate the roots of polynomial equations in one or several variables; continued fraction expansion of roots of univariate polynomials; Bernstein basis representation of univariate and multivariate polynomials and related algorithms; exact computation with real algebraic numbers, sign evaluation, comparison, certified numerical approximation.
- SHAPE: tools to manipulate curves and surfaces of different types including parameterized, implicit with different type of coefficients; algorithms to compute their topology, intersection points or curves, self-intersection locus, singularities, ...

These packages are integrated from the former library SYNAPS (SYmbolic Numeric APplicationS) dedicated to symbolic and numerical computations. There are also used in the algebraic-geometric modeler AXEL.

Collaborators: Grégoire Lecerf, Joris van der Hoeven and Philippe Trébuchet.

5.2. Axel, a geometric modeler for algebraic objects

Participants: Nicolas Douillet, Anaïs Ducoffe [contact], Valentin Michelet, Bernard Mourrain, Hung Nguyen, Meriadeg Perrinel.

http://axel.inria.fr.

computational algebraic geometry, curve, implicit equation, intersection, parameterization, resolution, surface, singularity, topology

We are developing a software called AXEL (Algebraic Software-Components for gEometric modeLing) dedicated to algebraic methods for curves and surfaces. Many algorithms in geometric modeling require a combination of geometric and algebraic tools. Aiming at the development of reliable and efficient implementations, AXEL provides a framework for such combination of tools, involving symbolic and numeric computations.

The software contains data structures and functionalities related to algebraic models used in geometric modeling, such as polynomial parameterizations, B-splines, implicit curves and surfaces. It provides algorithms for the treatment of such geometric objects, such as tools for computing intersection points of curves or surfaces, for detecting and computing self-intersection points of parameterized surfaces, for implicitization, for computing the topology of implicit curves, for meshing implicit (singular) surfaces, etc.

The developments related to isogeometric analysis have been integrated as dedicated plugins. Optimization techniques and solvers for partial differential equations developed by R. Duvigneau (OPALE) have been connected.

The new version of the algebraic-geometric modelers based on the DTK platform is still developed in order to provide a better modularity and a better interface to existing computation facilities and geometric rendering interface. This software is intended to be multi-platform, and jobs are running nightly on the Continous Integration platform `ci.inria.fr` of Inria, performing builds and tests on Virtual Machines of different OS such as Fedora, Ubuntu, Windows.

AXEL is written in C++ and thanks to a wrapping system using SWIG, its data structures and algorithms can be integrated into C# programs, as well as Python and Java programs. This wrapper was used to integrate AXEL into the CAD software TopSolid, developed by Missler Company and written in C#. But it also enables AXEL to embed a Python interpreter.

Other functionalities were also added or improved: the scientific visualization was improved and it is now possible to create dynamic geometric model in AXEL.

The software is distributed as a source package, as well as binary packages for Linux, MacOSX and Windows. It is hosted at http://dtk.inria.fr/axel with some of its plugins developed on Inria's gforge server (http://gforge.inria.fr) The first version of the software has been downloaded more than 15000 times, since it is available. A new version, AXEL 2.3.1, was released at the end of this year.

Collaboration with Gang Xu (Hangzhou Dianzi University, China), Julien Wintz (Dream), Elisa Berrini (MyCFD, Sophia), Angelos Mantzaflaris (GISMO library, Linz, Austria) and Laura Saini (Post-Doc GALAAD/Missler, TopSolid).

ABSTRACTION Project-Team

5. Software and Platforms

5.1. The Apron Numerical Abstract Domain Library

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

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

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

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

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

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

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

Additionally, APRON is used internally by the team to assist the research on numeric domains and static analyses by enabling the development of fast prototypes. In 2013, APRON has been used to design a sufficient-condition generator prototype (6.3.2), the FUNCTION prototype analyzer for termination (5.6 ,6.12), a constraint solver based on numeric abstract domains (6.5), a prototype implementation and extension of the Two Variables Per Inequality abstract domain [27].

5.2. The Astrée Static Analyzer of Synchronous Software

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

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

ASTRÉE is a static analyzer for sequential programs based on abstract interpretation [41], [31], [42], [33].

The ASTRÉE static analyzer [30], [46][1] www.astree.ens.fr aims at proving the absence of runtime errors in programs written in the C programming language.

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

ASTRÉE discovers all runtime errors including:

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

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

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

ASTRÉE has achieved the following two unprecedented results:

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

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

5.3. The AstréeA Static Analyzer of Asynchronous Software

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

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

ASTRÉEA is a static analyzer prototype for parallel software based on abstract interpretation [43], [44], [35]. It started with support from THÉSÉE ANR project (2006–2010) and is continuing within the ASTRÉEA project (2012–2015).

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

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

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

ASTRÉEA is currently being applied to analyze a large industrial avionic software: 1.6 MLines of C and 15 threads, completed with a 2,500-line model of the ARINC 653 OS developed for the analysis. The analysis currently takes a few tens of hours on a 2.9 GHz 64-bit intel server using one core and generates around 1,050 alarms. The low computation time (only a few times larger than the analysis time by ASTRÉE of synchronous programs of a similar size and structure) shows the scalability of the approach (in particular, we avoid the usual combinatorial explosion associated to thread interleavings). Precision-wise, the result, while not as impressive as that of ASTRÉE, is quite encouraging. The development of AstréeA continues within the scope of the ASTRÉEA ANR project (8.1.1.2).

5.4. The MemCAD static analyzer

Participants: Xavier Rival [correspondent], Antoine Toubhans.

Keywords: Shape analysis. MemCAD is a static analyzer that focuses on memory abstraction. It takes as input C programs, and computes invariants on the data structures manipulated by the programs. It can also verify memory safety. It comprises several memory abstract domains (flat representation, graph abstraction with summaries based on inductive definitions of data-structures, such as lists) and combination operators for memory abstract domains (hierarchical abstraction, reduced product). The current implementation comes with over 200 small size test cases that are used as regression tests.

5.5. A New Tactic Engine for Coq

Participant: Arnaud Spiwack [Correspondent].

Keywords: Coq, Proof assistant, Dependent types, Tactics, Proof search.

Coq is a proof assistant based on dependent type theory developed chiefly at Inria. This project addresses longstanding usability issues when developing proofs interactively: proofs, in Coq, are typically sequences of instructions – called tactics – which transform the proof into further proof obligations. The expressiveness of tactic affects the kind of proofs which can be written realistically in Coq. Two issues have been addressed. First, providing more backtracking primitives: in a typical automated procedure – which a Coq user could write to discharge proof obligations without human effort – there is some amount of non-determinism. It is hence important to be able to devise strategies, and Coq suffered from limited options on that front. The new tactics support further primitives loosely inspired by Prolog, in particular a backtracking choice (like disjunction in Prolog), and a primitive "once" which is akind to Prolog's soft-cut control primitive.

The second issue is more fundamental: since Coq is based on dependent types, a proof can appear in the statement of a proof obligation. As a result, tactics should be able to handle so-called dependent subgoals (where several proof obligations are left to be discharged, and the proof of one of them is mentioned into the statement of another). This was not historically the case in Coq, which had a direct influence on some users.

Both of the backtracking primitive and the dependent subgoals are part of the development version of Coq and will be part of the next release.

5.6. FuncTion: An Abstract Domain Functor for Termination

Participant: Caterina Urban.

Keywords: Conditional termination, Ranking functions, Static analysis.

FUNCTION is a research prototype static analyzer to analyze the termination of programs written in a small non-deterministic imperative language: it can infer sufficient conditions so that all executions terminate (conditional definitive termination). Following the general framework to analyze termination by abstract interpretation proposed in [40], FUNCTION infers ranking functions using piecewise-defined abstract domains. A first version of FUNCTION implemented a domain of linear ranking functions partitioned by variable bounds [24], [23]. It has been generalized in [22] and [29] to support ordinal-valued ranking functions of the form $\sum_{i=1}^{n} \omega^i f_i$ where n is a constant and each f_i is a natural-valued linear function of the variables.

The analyzer is written in OCaml and implemented on top of the APRON library (5.1). It can be used on-line through a web interface: http://www.di.ens.fr/~urban/FuncTion.html.

FUNCTION entered the 3rd Competition on Software Verification (SV-COMP 2014) in the termination category (demonstration section, no ranking).

5.7. The OpenKappa Modeling Plateform

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

Keywords: Causal traces, Model reduction, Rule-based modeling, Simulation, Static analysis.

OPENKAPPA is a collection of tools to build, debug and run models of biological pathways. It contains a compiler for the Kappa Language [53], a static analyzer [52] (for debugging models), a simulator [51], a compression tool for causal traces [50], [48], and a model reduction tool [4], [49], [58].

OPENKAPPA is developed since 2007 and, the OCaml version currently consists of 46 000 lines of OCaml. Software are available in OCaml and in Java. Moreover, an Eclipse pluggin is available.A compiler from CellDesigner into Kappa has been released in 2013.

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

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

5.8. Translation Validation

Participant: Xavier Rival [correspondent].

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

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

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

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

5.9. Zarith

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

Keywords: Arbitrary precision integers, Arithmetic, OCaml.

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

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

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

ALF Project-Team

5. Software and Platforms

5.1. Panorama

The ALF team is developing several software prototypes for research purposes: compilers, architectural simulators, programming environments, ...

Among the many prototypes developed in the project, we describe here **ATMI**, a microarchitecture temperature model for processor simulation, **STiMuL**, a temperature model for steady state studies, **ATC**, an address trace compressor, **HAVEGE**, an unpredictable random number generator, **tiptop**, a user-level Linux utility that collects data from hardware performance counters for running tasks, and **Padrone**, a platform for dynamic binary analysis and optimization.

5.2. ATMI

Participant: Pierre Michaud.

Microarchitecture temperature model

Status : Registered with APP Number IDDN.FR.001.250021.000.S.P.2006.000.10600, Available under GNU General Public License

Research on temperature-aware computer architecture requires a chip temperature model. General purpose models based on classical numerical methods like finite differences or finite elements are not appropriate for such research, because they are generally too slow for modeling the time-varying thermal behavior of a processing chip.

We have developed an ad hoc temperature model, ATMI (Analytical model of Temperature in MIcroprocessors), for studying thermal behaviors over a time scale ranging from microseconds to several minutes. ATMI is based on an explicit solution to the heat equation and on the principle of superposition. ATMI can model any power density map that can be described as a superposition of rectangle sources, which is appropriate for modeling the microarchitectural units of a microprocessor.

Visit http://www.irisa.fr/alf/ATMI or contact Pierre Michaud.

5.3. STiMuL

Participant: Pierre Michaud.

Microarchitecture temperature modeling

Status: Registered with APP Number IDDN.FR.001.220013.000.S.P.2010.000.31235, Available under GNU General Public License

Some recent research has started investigating the microarchitectural implications of 3D circuits, for which the thermal constraint is stronger than for conventional 2D circuits.

STiMuL can be used to model steady-state temperature in 3D circuits consisting of several layers of different materials. STiMuL is based on a rigorous solution to the Laplace equation.The number and characteristics of layers can be defined by the user. The boundary conditions can also be defined by the user. In particular, STiMuL can be used along with thermal imaging to obtain the power density inside an integrated circuit. This power density could be used for instance in a dynamic simulation oriented temperature modeling such as ATMI.

STiMuL is written in C and uses the FFTW library for discrete Fourier transforms computations.

Visit http://www.irisa.fr/alf/stimul or contact Pierre Michaud.

5.4. ATC

Participant: Pierre Michaud.

Address trace compression

Status: registered with APP number IDDN.FR.001.160031.000.S.P.2009.000.10800, available under GNU LGPL License.

Trace-driven simulation is an important tool in the computer architect's toolbox. However, one drawback of trace-driven simulation is the large amount of storage that may be necessary to store traces. Trace compression techniques are useful for decreasing the storage space requirement. But general-purpose compression techniques are generally not optimal for compressing traces because they do not take advantage of certain characteristics of traces. By specializing the compression method and taking advantages of known trace characterics, it is possible to obtain a better tradeoff between the compression ratio, the memory consumption and the compression and decompression speed.

ATC is a utility and a C library for compressing/decompressing address traces. It implements a new lossless transformation, Bytesort, that exploits spatial locality in address traces. ATC leverages existing general-purpose compressors such as gzip and bzip2. ATC also provides a lossy compression mode that yields higher compression ratios while preserving certain important characteristics of the original trace.

Visit http://www.irisa.fr/alf/atc or contact Pierre Michaud.

5.5. HAVEGE

Participant: André Seznec.

Unpredictable random number generator

Contact : André Seznec

Status : Registered with APP Number IDDN.FR.001.500017.001.S.P.2001.000.10000. Available under the LGPL license.

An unpredictable random number generator is a practical approximation of a truly random number generator. Such unpredictable random number generators are needed for cryptography. HAVEGE (HArdware Volatile Entropy Gathering and Expansion) is a user-level software unpredictable random number generator for general-purpose computers that exploits the continuous modifications of the internal volatile hardware states in the processor as a source of uncertainty [9]. HAVEGE combines on-the-fly hardware volatile entropy gathering with pseudo-random number generation.

The internal state of HAVEGE includes thousands of internal volatile hardware states and is merely unmonitorable. HAVEGE can reach an unprecedented throughput for a software unpredictable random number generator: several hundreds of megabits per second on current workstations and PCs.

The throughput of HAVEGE favorably competes with usual pseudo-random number generators such as `rand()` or `random()`. While HAVEGE was initially designed for cryptology-like applications, this high throughput makes HAVEGE usable for all application domains demanding high performance and high quality random number generators, e.g., Monte Carlo simulations.

Visit http://www.irisa.fr/alf/index.php?option=com_content&view=article&id=5/havege&catid=3/projects&Itemid=3&lang=fr or contact André Seznec.

5.6. Tiptop

Participant: Erven Rohou.

Performance, hardware counters, analysis tool.

Status: Registered with APP (Agence de Protection des Programmes). Available under GNU General Public License v2.

Tiptop is a new simple and flexible user-level tool that collects hardware counter data on Linux platforms (version 2.6.31+). The goal is to make the collection of performance and bottleneck data as simple as possible, including simple installation and usage. In particular, we stress the following points.

- Installation is only a matter of compiling the source code. No patching of the Linux kernel is needed, and no special-purpose module needs to be loaded.
- No privilege is required, any user can run *tiptop* — non-privileged users can only watch processes they own, ability to monitor anybody's process opens the door to side-channel attacks.
- The usage is similar to *top*. There is no need for the source code of the applications of interest, making it possible to monitor proprietary applications or libraries. And since there is no probe to insert in the application, understanding of the structure and implementation of complex algorithms and code bases is not required.
- Applications do not need to be restarted, and monitoring can start at any time (obviously, only events that occur after the start of *tiptop* are observed).
- Events can be counted per thread, or per process.
- Any expression can be computed, using the basic arithmetic operators, constants, and counter values.
- A configuration file lets users define their prefered setup, as well as custom expressions.

Tiptop is written in C. It can take advantage of libncurses when available for pseudo-graphic display.

Tiptop version 2.2 was released in March 2013.

For more information, please contact Erven Rohou and/or visit http://tiptop.gforge.inria.fr.

5.7. Padrone

Participants: Erven Rohou, Emmanuel Riou.

Performance, profiling, dynamic optimization

Status: Ongoing development, early prototype.

Padrone is new platform for dynamic binary analysis and optimization. It provides an API to help clients design and develop analysis and optimization tools for binary executables. Padrone attaches to running applications, only needing the executable binary in memory. No source code or debug information is needed. No application restart is needed either. This is specially interesting for legacy or commercial applications, but also in the context of cloud deployment, where actual hardware is unknown, and other applications competing for hardware resources can vary. The profiling overhead is minimum.

Padrone is written in C.

For more information, please contact Erven Rohou.

AOSTE Project-Team

5. Software and Platforms

5.1. TimeSquare

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

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

TimeSquare offers four main functionalities:

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

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

5.2. K-Passa

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

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

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

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

5.3. SynDEx

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

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

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

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

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

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

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

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

5.4. Lopht

Participants: Thomas Carle, Manel Djemal, Zhen Zhang, Dumitru Potop Butucaru [correspondant].

The Lopht (Logical to Physical Time Compiler) has been designed as an implementation of the AAA methodology. Lopht is similar to SynDEx by relying on off-line allocation and scheduling techniques to allow real-time implementation of dataflow synchronous specifications onto multiprocessor systems. But it has two significant originality points: a stronger focus on efficiency (but without compromising correctness), and a focus on novel target architectures (many-core chips and time-triggered embedded systems).

Improved efficiency is attained through the use of classical and novel data structures and optimization algorithms pertaining to 3 fields: synchronous language compilation, classical compiler theory, and real-time scheduling. A finer representation of execution conditions allows us to make a better use of double resource reservation and thus improve latency and throughput. The use of software pipelining allows the improvement of computation throughput. The use of post-scheduling optimisations allows a reduction in the number of preemptions. The focus on novel architectures means that architecture descriptions need to define novel communication media such as the networks-on-chips (NoCs), and that real-time characteristics must include those specific to a time-triggered execution model, such as the Major Time Frame (MTF).

Significant contributions to the Lopht tool have been brought by T. Carle (the extensions concerning time-triggered platforms), M. Djemal (the extensions concerning many-core platforms), and Zhen Zhang under the supervision of D. Potop Butucaru. The tool has been used and extended during the PARSEC project. It is currently used in the IRT SystemX/FSF project, in the collaboration with Astrium Space Transportation (Airbus Defence and Space), and in the collaboration with Kalray SA. It has been developed in OCaml.

5.5. SAS

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

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

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

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

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

ARIC Project-Team

5. Software and Platforms

5.1. Overview

AriC software and hardware realizations are accessible from the web page http://www.ens-lyon.fr/LIP/AriC/ware.html. We describe below only those which progressed in 2013.

5.2. FloPoCo

Participants: Florent de Dinechin [correspondant], Matei Istoan.

The purpose of the FloPoCo project is to explore the many ways in which the flexibility of the FPGA target can be exploited in the arithmetic realm [32]. FloPoCo is a generator of operators written in C++ and outputting synthesizable VHDL automatically pipelined to an arbitrary frequency.

2013 saw more work on the *bit-heap* framework [28], [18]. In addition, several new operators were added, in particular for fixed-point sine, cosine [21] and arctangent.

Version 2.5.0 was released in 2013.

Among the known users of FloPoCo are U. Cape Town, U.T. Cluj-Napoca, Imperial College, U. Essex, U. Madrid, U. P. Milano, T.U. Muenchen, T. U. Kaiserslautern, U. Paderborn, CalTech, U. Pernambuco, U. Perpignan, U. Tohoku, U. Tokyo, Virginia Tech U. and several companies.

URL: http://flopoco.gforge.inria.fr/

- Version: 2.5.0 (June 2013)
- APP: IDDN.FR.001.400014.000.S.C.2010.000.20600 (version 2.0.0)
- License: pending, should be GPL-like.
- Type of human computer interaction: command-line interface, synthesizable VHDL output.
- OS/Middleware: Linux, Windows/Cygwin.
- Required library or software: MPFR, flex, Sollya.
- Programming language: C++.
- Documentation: online and command-line help, API in doxygen format, articles.

5.3. GNU MPFR

Participants: Vincent Lefèvre [correspondant], Paul Zimmermann [Caramel, Inria Nancy - Grand Est].

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

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

GNU MPFR 3.1.2 was released on 13 March 2013.

The main work done in the AriC project-team:

- Bug fixes and improved portability.
- Complete revision of the behavior on special values (signed zeros and infinities) and consistency with standards (IEEE 754-2008, ISO C, POSIX) checked. Thanks to this work, several problems in MPFR and the POSIX specification have been detected and the MPFR manual has been completed: https://sympa.inria.fr/sympa/arc/mpfr/2013-12/msg00001.html

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

GNU MPFR is now on the Ohloh community platform for free and open source software: https://www.ohloh.net/p/gnu-mpfr

- ACM: D.2.2 (Software libraries), G.1.0 (Multiple precision arithmetic), G.4 (Mathematical software).
- AMS: 26-04 Real Numbers, Explicit machine computation and programs.
- APP: no longer applicable (copyright transferred to the Free Software Foundation).
- License: LGPL version 3 or later.
- Type of human computer interaction: C library, callable from C or other languages via third-party interfaces.
- OS/Middleware: any OS, as long as a C compiler is available.
- Required library or software: GMP.
- Programming language: C.
- Documentation: API in texinfo format (and other formats via conversion); algorithms are also described in a separate document.

5.4. Exhaustive Tests for the Correct Rounding of Mathematical Functions

Participant: Vincent Lefèvre.

The search for the worst cases for the correct rounding (hardest-to-round cases) of mathematical functions ($\exp$, $\log$, $\sin$, $\cos$, etc.) in a fixed precision (mainly double precision) using Lefèvre's algorithm is implemented by a set of utilities written in Perl, with calls to Maple/intpakX for computations on intervals and with C code generation for fast computations. It also includes a client-server system for the distribution of intervals to be tested and for tracking the status of intervals (fully tested, being tested, aborted).

The Perl scripts have been improved and some minor bugs have been fixed.

5.5. FPLLL: A Lattice Reduction Library

Participant: Damien Stehlé [correspondant].

fplll contains several algorithms on lattices that rely on floating-point computations. This includes implementations of the floating-point LLL reduction algorithm, offering different speed/guarantees ratios. It contains a "wrapper" choosing the estimated best sequence of variants in order to provide a guaranteed output as fast as possible. In the case of the wrapper, the succession of variants is oblivious to the user. It also includes a rigorous floating-point implementation of the Kannan-Fincke-Pohst algorithm that finds a shortest non-zero lattice vector, and the BKZ reduction algorithm.

The fplll library is used or has been adapted to be integrated within several mathematical computation systems such as Magma, Sage, and PariGP. It is also used for cryptanalytic purposes, to test the resistance of cryptographic primitives.

Versions 4.0.4 was released in 2013, fixing a number of user-interface bugs.

URL: http://perso.ens-lyon.fr/damien.stehle/fplll/

- ACM: D.2.2 (Software libraries), G.4 (Mathematical software)
- APP: Procedure started
- License: LGPL v2.1
- Type of human computer interaction: C++ library callable, from any C++ program.
- OS/Middleware: any, as long as a C++ compiler is available.
- Required library or software: MPFR and GMP.
- Programming language: C++.
- Documentation: available in html format on **URL:**http://perso.ens-lyon.fr/damien.stehle/fplll/fplll-doc.html

5.6. Sipe

Participant: Vincent Lefèvre.

Sipe is a mini-library in the form of a C header file, to perform radix-2 floating-point computations in very low precisions with correct rounding, either to nearest or toward zero. The goal of such a tool is to do proofs of algorithms/properties or computations of tight error bounds in these precisions by exhaustive tests, in order to try to generalize them to higher precisions. The currently supported operations are addition, subtraction, multiplication (possibly with the error term), fused multiply-add/subtract (FMA/FMS), and miscellaneous comparisons and conversions. Sipe provides two implementations of these operations, with the same API and the same behavior: one based on integer arithmetic, and a new one based on floating-point arithmetic; see [25], [39].

New in 2013:

- the floating-point implementation;
- rounding toward zero (only with the integer implementation).
- ACM: D.2.2 (Software libraries), G.4 (Mathematical software).
- AMS: 26-04 Real Numbers, Explicit machine computation and programs.
- License: LGPL version 2.1 or later.
- Type of human computer interaction: C header file.
- OS/Middleware: any OS.
- Required library or software: GCC compiler.
- Programming language: C.
- Documentation: comment at the beginning of the code and Research report Inria RR-7832.
- URL: https://www.vinc17.net/research/sipe/

ATEAMS Project-Team

4. Software and Platforms

4.1. MicroMachinations

Participant: Riemer Van Rozen [correspondent].

Characterization: A-2-up3, SO-4, SM-2-up3, EM-3, SDL-3-up4, OC-DA-3-CD-3-MS-3-TPM-3.

WWW:

Objective: To create an integrated, live environment for modeling and evolving game economies. This will allow game designers to experiment with different strategies to realize game mechanics. The environment integrates with the SPIN model checker to prove properties (reachability, liveness). A runtime system for executing game economies allows MicroMachinations models to be embedded in actual games.

Users: Game designers

Impact: One of the important problems in game software development is the distance between game design and implementation in software. MicroMachinations has the potential to bridge this gap by providing live design tools that directly modify or create the desired software behaviors.

Competition: None.

Engineering: The front-end of MicroMachinations is built using the Rascal language workbench, including visualization, model checking, debugging and standard IDE features. The runtime library is implemented in C++ and will be evaluated in the context of industrial game design.

Publications: [28]

4.1.1. Novelties

- Development on MMLib was started which allows the execution of game economies directly within games.

4.2. Derric

Participants: Tijs Van Der Storm, Jeroen Van Den Bos [correspondent].

Characterization: A-2-up3, SO-4, SM-2-up3, EM-3, SDL-3-up4, OC-DA-3-CD-3-MS-3-TPM-3.

WWW: http://www.derric-lang.org

Objective: Encapsulate all the variability in the construction of so-called “carving” algorithms, then generate the fastest and most accurate implementations. Carving algorithms recover information that has been deleted or otherwise scrambled on digital media such as hard-disks, usb sticks and mobile phones.

Users: Digital forensic investigation specialists

Impact: Derric has the potential of revolutionizing the carving area. It does in 1500 lines of code what other systems need tens of thousands of lines for with the same accuracy. Derric will be an enabler for faster, more specialized and more successful location of important evidence material.

Competition: Derric competes in a small market of specialized open-source and commercial carving tools.

Engineering: Derric is a Rascal program of 1.5 kloc designed by two persons.

Publications: [35], [34][14], [16], [15]

4.2.1. Novelties

- Construction of a 1TB benchmark based on Wikipedia images.
- The Derric DSL for digital forensics now features Trinity, a runtime IDE to debug file format descriptions [35].

4.3. Rascal

Participants: Paul Klint, Jurgen Vinju [correspondent], Tijs Van Der Storm, Jeroen Van Den Bos, Mark Hills, Bert Lisser, Atze Van Der Ploeg, Vadim Zaytsev, Anastasia Izmaylova, Michael Steindorfer, Ali Afroozeh, Ashim Shahi.

Characterization: A5, SO-4, SM-4, EM-4, SDL-4-up5, OC-DA-3-CD-3-MS-3-TPM-3.

WWW: http://www.rascal-mpl.org

Objective: Provide a completely integrated programming language parametric meta programming language for the construction of any kind of meta program for any kind of programming language: analysis, transformation, generation, visualization.

Users: Researchers in model driven engineering, programming languages, software engineering, software analysis, as well as practitioners that need specialized tools.

Impact: Rascal is making the mechanics of meta programming into a non-issue. We can now focus on the interesting details of the particular fact extraction, model, source analysis, domain analysis as opposed to being distracted by the engineering details. Simple things are easy in Rascal and complex things are manageable, due to the integration, the general type system and high-level programming features.

Competition: There is a plethora of meta programming toolboxes and frameworks available, ranging from plain parser generators to fully integrated environments. Rascal is distinguished because it is a programming language rather than a specification formalism and because it completely integrates different technical domains (syntax definition, term rewriting, relational calculus). For simple tools, Rascal competes with scripting languages and for complex tools it competes context-free general parser generators, with query engines based on relational calculus and with term rewriting and strategic programming languages.

Engineering: Rascal is about 100 kLOC of Java code, designed by a core team of three and with a team of around 8 phd students and post-docs contributing to its design, implementation and maintenance. The goal is to work towards more bootstrapping and less Java code as the project continues.

Publications: [7], [6], [8], [5], [6]

4.3.1. Novelties

- A new language-parametric model to represent software projects, called M3 [38].
- Performance improvements of the Rascal interpreter throughout.
- Initial version of a compiler for Rascal, based on new language construct guarded coroutines.
- Origin tracking for values and expressions of type string.
- A library for accessing and analyzing Excel and Word documents.
- Improvements to the Rascal IDE: better output handling, hyper linked source code locations in the console, dedicated project explorer view.
- Content completion for DSLs implemented in Rascal.
- Significant improvements to the Rascal static type checker.
- Experiments with improved GLL parsing (Iguana).
- Several new example DSL implementations to illustrate Rascal as a language workbench: Marvol, a DSL for controlling NAO robots, and two implementations of a DSL for questionnaires (DemoQLes and QL-R-Kemi).

4.4. IDE Meta-tooling Platform

Participants: Jurgen Vinju [correspondent], Michael Steindorfer.

IMP, the IDE meta tooling platform is an Eclipse plugin developed mainly by the team of Robert M. Fuhrer at IBM TJ Watson Research institute. It is both an abstract layer for Eclipse, allowing rapid development of Eclipse based IDEs for programming languages, and a collection of meta programming tools for generating source code analysis and transformation tools.

Characterization: A5, SO-3, SM4-up5, EM-4, SDL-5, DA-2-CD-2-MS-2-TPM-2

WWW: https://github.com/impulse-org/

Objective: The IDE Meta Tooling Platform (IMP) provides a high-level abstraction over the Eclipse API such that programmers can extend Eclipse with new programming languages or domain specific languages in a few simple steps. IMP also provides a number of standard meta tools such as a parser generator and a domain specific language for formal specifications of configuration parameters.

Users: Designers and implementers of IDEs for programming languages and domain specific languages. Also, designers and implementers of meta programming tools.

Impact: IMP is popular among meta programmers especially for it provides the right level of abstraction.

Competition: IMP competes with other Eclipse plugins for meta programming (such as Model Driven Engineering tools), but its API is more general and more flexible. IMP is a programmers framework rather than a set of generators.

Engineering: IMP is a long-lived project of many contributors, which is managed as an Eclipse incubation project at `eclipse.org`. Currently we are moving the project to Github to explore more different ways of collaboration.

Publications: [2]

4.4.1. Novelties

- The IMP program database (PDB) was completely redesigned.

4.5. Ensō

Participant: Tijs Van Der Storm [correspondent].

Characterization: A5, SO-4, SM-3-up-4, EM-2-up-4, SDL-4, OC-DA-4-CD-4-MS-4-TPM-4

WWW: http://www.enso-lang.org

Objective: Together with Prof. Dr. William R. Cook of the University of Texas at Austin, and Alex Loh, Tijs van der Storm has been designing and implementing a new programming system, called Ensō. Ensō is theoretically sound and practical reformulation of model-based development. It is based on model-interpretation as opposed to model transformation and code generation. Currently, the system already supports models for schemas (data models), web applications, context-free grammars, diagram editors and security.

Users: All programmers.

Impact: Ensō has the potential to revolutionize the activity of programming. By looking at model driven engineering from a completely fresh perspective, with as key ingredients interpreters and partial evaluation, it may make higher level (domain level) program construction and maintenance as effective as normal programming.

Competition: Ensō competes as a programming paradigm with model driven engineering tools and generic programming and languages that provide syntax macros and language extensions.

Engineering: Ensō is a completely self-hosted system in 7000 lines of code.

Publications: [12], [17], [11]

4.5.1. Novelties

- A compiler for a dedicated Ensō language, which targets JavaScript.
- Added a demo based on the LWC'13 questionnaire language assignment.

4.6. LiveQL

Participant: Tijs Van Der Storm [correspondent].

Characterization: A1, SO-3, SM-1, EM-2, SDL-4, OC-DA-4-CD-4-MS-4-TPM-4

WWW: https://github.com/cwi-swat/liveql

Objective: Experimenting with live programming concepts and techniques in the context of domain specific languages (DSLs).

Users: End-user programmers.

Impact: LiveQL is an experiment in making a DSL "live", i.e. any change to the DSL program is immediately reflected in the running program. This has the potential to widen the audience of DSL users to include end-user programmers.

Competition: The end-goal is to provide live end-user programming environments with domain-specific checking and optimization. The most similar tools are spreadsheet applications. However, these are still quite general.

Engineering: LiveQL is built in Java, using the ANTLR parser generator.

Publications: [36]

4.7. QL-R-Kemi

Participant: Tijs Van Der Storm [correspondent].

Characterization: A1, SO-3, SM-1, EM-2, SDL-4, OC-DA-4-CD-4-MS-4-TPM-4

WWW: https://github.com/cwi-swat/QL-R-Kemi

Objective: Demonstrate the language workbench features of the Rascal meta programming language and environment. Investigate domain specific language application in the domain of questionnaires and surveys.

Users: Students, scientists.

Impact: Questionnaires are common in social science, tax administration and statistics. Discovering the right abstractions for describing questionnaires has the potential to significantly improve the practice of constructing questionnaire software.

Competition: Traditional survey tools are often wizard-based, lack computational capabilities and lack a formal foundation. The same language is built in a number of different language workbenches which served as a benchmark to compare such tools [24].

Engineering: Uses all features of the Rascal language workbench.

Publications: [24]

CAIRN Project-Team

5. Software and Platforms

5.1. Panorama

With the ever raising complexity of embedded applications and platforms, the need for efficient and customizable compilation flows is stronger than ever. This need of flexibility is even stronger when it comes to research compiler infrastructures that are necessary to gather quantitative evidence of the performance/energy or cost benefits obtained through the use of reconfigurable platforms. From a compiler point of view, the challenges exposed by these complex reconfigurable platforms are quite significant, since they require the compiler to extract and to expose an important amount of coarse and/or fine grain parallelism, to take complex resource constraints into consideration while providing efficient memory hierarchy and power management.

Because they are geared toward industrial use, production compiler infrastructures do not offer the level of flexibility and productivity that is required for compiler and CAD tool prototyping. To address this issue, we have designed an extensible source-to-source compiler infrastructure that takes advantage of leading edge model-driven object-oriented software engineering principles and technologies.

`../../../../projets/cairn/IMG/CairnToolFlow.png`

Figure 2. CAIRN*'s general software development framework.*

Figure 2 shows the global framework that is being developed in the group. Our compiler flow mixes several types of intermediate representations. The baseline representation is a simple tree-based model enriched with control flow information. This model is mainly used to support our source-to-source flow, and serves as the backbone for the infrastructure. We use the extensibility of the framework to provide more advanced representations along with their corresponding optimizations and code generation plug-ins. For example, for our pattern selection and accuracy estimation tools, we use a data dependence graph model in all basic blocks instead of the tree model. Similarly, to enable polyhedral based program transformations and analysis, we introduced a specific representation for affine control loops that we use to derive a Polyhedral Reduced Dependence Graph (PRDG). Our current flow assumes that the application is specified as a system level hierarchy of communicating tasks, where each task is expressed using C (or Scilab in the short future), and where the system level representation and the target platform model are defined using Domain Specific Languages (DSL).

Gecos (Generic Compiler Suite) is the main backbone of CAIRN's flow. It is an open source Eclipse-based flexible compiler infrastructure developed for fast prototyping of complex compiler passes. Gecos is a 100% Java based implementation and is based on modern software engineering practices such as Eclipse plugin or model-driven software engineering with EMF (Eclipse Modeling Framework). As of today, our flow offers the following features:

- An automatic floating-point to fixed-point conversion flow (for HLS and embedded processors). **ID.Fix** is an infrastructure for the automatic transformation of software code aiming at the conversion of floating-point data types into a fixed-point representation. http://idfix.gforge.inria.fr.
- A polyhedral-based loop transformation and parallelization engine (mostly targeted at HLS). http://gecos.gforge.inria.fr. It was used for source-to-source transformations in the context of Nano2012 projects in collaboration with STMicroelectronics.
- A custom instruction extraction flow (for ASIP and dynamically reconfigurable architectures). **Durase** and **UPaK** are developed for the compilation and the synthesis targeting reconfigurable platforms and the automatic synthesis of application specific processor extensions. They use advanced technologies, such as graph matching and graph merging together with constraint programming methods.
- Several back-ends to enable the generation of VHDL for specialized or reconfigurable IPs, and SystemC for simulation purposes (e.g. fixed-point simulations).

5.2. Gecos

Participants: Steven Derrien [corresponding author], Nicolas Simon, Maxime Naullet, Antoine Morvan.

Keywords: source-to-source compiler, model-driven software engineering, retargetable compilation.

The Gecos (Generic Compiler Suite) project is a source-to-source compiler infrastructure developed in the Cairn group since 2004. It was designed to enable fast prototyping of program analysis and transformation for hardware synthesis and retargetable compilation domains.

Gecos is 100% Java based and takes advantage of modern model driven software engineering practices. It uses the Eclipse Modeling Framework (EMF) as an underlying infrastructure and takes benefits of its features to make it easily extensible. Gecos is open-source and is hosted on the Inria gforge at http://gecos.gforge.inria.fr.

The Gecos infrastructure is still under very active development, and serves as a backbone infrastructure to projects of the group. Part of the framework is jointly developed with Colorado State University and since 2012 it is used in the context of the ALMA European project.

Developments in Gecos in 2013 have focused on polyhedral loop transformations and efficient SIMD code generation for fixed point arithmetic data-types as a part of the ALMA project. Significant efforts were also been put to provide a coarse-grain parallelization engine targeting the data-flow actor model in the context of the COMPA ANR project. An article describing the design choice and the main features of the framework was presented at the international workshop on Source Code Analysis and Manipulation in september 2013 [46].

5.3. ID.Fix: Infrastructure for the Design of Fixed-point Systems

Participants: Olivier Sentieys [corresponding author], Romuald Rocher, Nicolas Simon.

Keywords: fixed-point arithmetic, source-to-source code transformation, accuracy optimization, dynamic range evaluation

The different techniques proposed by the team for fixed-point conversion are implemented on the ID.Fix infrastructure. The application is described with a C code using floating-point data types and different pragmas, used to specify parameters (dynamic, input/output word-length, delay operations) for the fixed-point conversion. This tool determines and optimizes the fixed-point specification and then, generates a C code using fixed-point data types (ac_fixed) from Mentor Graphics. The infrastructure is made-up of two main modules corresponding to the fixed-point conversion (ID.Fix-Conv) and the accuracy evaluation (ID.Fix-Eval)

The different developments carried-out in 2013 allowed us to obtain a fixed-point conversion tool handling functions, conditional structures and repetitive structures having a fixed number of iterations during time. The frontend has been modified to reduce limitations due to syntax of C langage. A new data type (sc_fixed) is now able to be generated from the back-end. In the context of the DEFIS ANR project, the ID.Fix tool has been reorganized to be integrated in the DEFIS toolflow.

In 2013, ID.Fix has been demonstrated during University Booth at IEEE/ACM DATE and IEEE/ACM DAC. See http://www.youtube.com/watch?v=nKYA4hezplQ

5.4. UPaK: Abstract Unified Pattern-Based Synthesis Kernel for Hardware and Software Systems

Participants: Christophe Wolinski [corresponding author], François Charot, Antoine Floc'H [former member].

Keywords: compilation for reconfigurable systems, pattern extraction, constraint-based programming.

We are developing (with strong collaboration of Lund University, Sweden and Queensland University, Australia) UPaK *Abstract Unified Pattern Based Synthesis Kernel for Hardware and Software Systems* [123]. The preliminary experimental results obtained by the UPak system show that the methods employed in the systems enable a high coverage of application graphs with small quantities of patterns. Moreover, high application execution speed-ups are ensured, both for sequential and parallel application execution with processor extensions implementing the selected patterns. UPaK is one of the basis for our research on compilation and synthesis for reconfigurable platforms. It is based on the HCDG representation of the Polychrony software designed at Inria-Rennes in the project-team Espresso.

5.5. DURASE: Automatic Synthesis of Application-Specific Processor Extensions

Participants: Christophe Wolinski [corresponding author], François Charot, Antoine Floc'H.

Keywords: compilation for reconfigurable systems, instruction-set extension, pattern extraction, graph covering, constraint-based programming.

We are developing a framework enabling the automatic synthesis of application specific processor extensions. It uses advanced technologies, such as algorithms for graph matching and graph merging together with constraints programming methods. The framework is organized around several modules.

- CoSaP: Constraint Satisfaction Problem. The goal of CoSaP is to decouple the statement of a constraint satisfaction problem from the solver used to solve it. The CoSaP model is an Eclipse plugin described using EMF to take advantage of the automatic code generation and of various EMF tools.

- HCDG: Hierarchical Conditional Dependency Graph. HCDG is an intermediate representation mixing control and data flow in a single acyclic representation. The control flow is represented as hierarchical guards specifying the execution or the definition conditions of nodes. It can be used in the Gecos compilation framework via a specific pass which translates a CDFG representation into an HCDG.
- Patterns: Flexible tools for identification of computational pattern in a graph and graph covering. These tools model the concept of pattern in a graph and provide generic algorithms for the identification of pattern and the covering of a graph. The following sub-problems are addressed: (sub)-graphs isomorphism, patterns generation under constraints, covering of a graph using a library of patterns. Most of the implemented algorithms use constraints programming and rely on the CoSaP module to solve the optimization problem.

5.6. PowWow: Power Optimized Hardware and Software FrameWork for Wireless Motes (AP-L-10-01)

Participants: Olivier Sentieys [corresponding author], Olivier Berder, Arnaud Carer, Steven Derrien.

Keywords: Wireless Sensor Networks, Low Power, Preamble Sampling MAC Protocol, Hardware and Software Platform

PowWow is an open-source hardware and software platform designed to handle wireless sensor network (WSN) protocols and related applications. Based on an optimized preamble sampling medium access (MAC) protocol, geographical routing and `protothread` library, PowWow requires a lighter hardware system than Zigbee [85] to be processed (memory usage including application is less than 10kb). Therefore, network lifetime is increased and price per node is significantly decreased.

CAIRN's hardware platform (see Figure 3) is composed of:

- The motherboard, designed to reduce power consumption of sensor nodes, embeds an MSP430 microcontroller and all needed components to process PowWow protocol except radio chip. JTAG, RS232, and I2C interfaces are available on this board.
- The radio chip daughter board is currently based on a TI CC2420.
- The coprocessing daughter board includes a low-power FPGA which allows for hardware acceleration for some PowWow features and also includes dynamic voltage scaling features to increase power efficiency. The current version of PowWow integrates an Actel IGLOO AGL250 FPGA and a programmable DC-DC converter. We have shown that gains in energy of up to 700 can be obtained by using FPGA acceleration on functions like CRC-32 or error detection with regards to a software implementation on the MSP430.
- Finally, a last daughter board is dedicated to energy harvesting techniques. Based on the energy management component LTC3108 from Linear Technologies, the board can be configured with several types of stored energy (batteries, micro-batteries, super-capacitors) and several types of energy sources (a small solar panel to recover photovoltaic energy, a piezoelectric sensor for mechanical energy and a Peltier thermal energy sensor).

PowWow distribution also includes a generic software architecture using event-driven programming and organized into protocol layers (PHY, MAC, LINK, NET and APP). The software is based on Contiki [101], and more precisely on the `Protothread` library which provides a sequential control flow without complex state machines or full multi-threading.

To optimize the network regarding a particular application and to define a global strategy to reduce energy, PowWow offers the following extra tools: over-the-air reprogramming (and soon reconfiguration), analytical power estimation based on software profiling and power measurements, a dedicated network analyzer to probe and fix transmissions errors in the network. More information can be found at http://powwow.gforge.inria.fr.

../../../../projets/cairn/IMG/PowWowRcup.png

Figure 3. CAIRN*'s PowWow motherboard with radio and energy-harvesting boards connected*

5.7. Ziggie: a Platform for Wireless Body Sensor Networks

Participants: Olivier Sentieys, Olivier Berder, Arnaud Carer, Antoine Courtay [corresponding author], Robin Bonamy.

Keywords: Wireless Body Sensor Networks, Low Power, Gesture Recognition, Localization, Hardware and Software Platform

The Zyggie sensor node has been developed in the team to create an autonomous Wireless Body Sensor Network (WBSN) with the capabilities of monitoring body movements. The Zyggie platform is part of the BoWI project funded by CominLabs. Zyggie is composed of:

- An ATMEGA128RFA1 microcontroller,
- An MPU9150 Inertial Measurement Unit (IMU),
- An RF AS193 switch with two antennas,
- An LSP331AP barometer,
- A DC/DC voltage regulator with a battery charge controller,
- A wireless inductive battery charge controller and
- Some switches and control LEDs.

../../../../projets/cairn/IMG/Ziggie.png

Figure 4. CAIRN*'s Ziggie platform for WBSN*

The IMU is composed of a 3-axis accelerometer, a 3-axis gyrometer and a 3-axis magnetometer. The IMU is communicating its data to the embedded microcontroller via an I2C protocol. We also developed our own MAC protocol for synchronization and data exchanges between nodes. The Zyggie platform is used in many PhD works for evaluating data fusion algorithms (RSSI + IMU data) (Zhongwei Zheng, UR1 and Alexis Aulery, UBS/UR1), low power computing algorithms (Alexis Aulery, UBS/UR1), wireless protocols (Viet Hoa Nguyen, UR1) and body channel characterization (Rizwan Masood, TB).

5.8. SoCLib: Open Platform for Virtual Prototyping of Multi-Processors System on Chip

Participants: François Charot [corresponding author], Laurent Perraudeau [external collaborator].

Keywords: SoC modeling, SystemC simulation model

SoCLib is an open platform for virtual prototyping of multi-processors system on chip (MP-SoC) developed in the framework of the SoCLib ANR project. The core of the platform is a library of SystemC simulation models for virtual components (IP cores), with a guaranteed path to silicon. All simulation models are written in SystemC, and can be simulated with the standard SystemC simulation environment distributed by the OSCI organization. Two types of models are available for each IP-core: CABA (Cycle Accurate / Bit Accurate), and TLM-DT (Transaction Level Modeling with Distributed Time). All simulation models are distributed as free software. We have developed the simulation model of the NIOSII processor, of the Altera Avalon interconnect, and of the TMS320C62 DSP processor from Texas Instruments. Find more information on its dedicated web page: http://www.soclib.fr.

CAMUS Team

5. Software and Platforms

5.1. PolyLib

Participant: Vincent Loechner.

PolyLib [1] 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 (Parkas Projet, Inria Paris - Rocquencourt).

5.2. ZPolyTrans

Participant: Vincent Loechner.

ZPolyTrans [2] 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 [29].

It allows for example to compute the number of solutions of a Presburger formula by eliminating existential 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

Participants: Alain Ketterlin, Philippe Clauss.

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.

---

[1] http://icps.u-strasbg.fr/PolyLib

[2] http://ZPolyTrans.gforge.inria.fr

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. In particular, it has been used by Andres Charif-Rubial for his PhD work (Université de Versailles Saint-Quentin en Yvelines), and is now included in a released tool called MAQAO (http://www.maqao.org). Our code is also used by Jean-Thomas ACQUAVIVA, at Commissariat à l'Énergie Atomique, for work on compressing instruction traces. These colleagues have slightly modified the code we gave them. We plan to release a stable version incorporating most of their changes in the near future. We also plan to change the license to avoid such drifts in the future.

5.4. Binary files decompiler

Participant: Alain Ketterlin.

Our research on efficient memory profiling has led 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 [27]. 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.

5.5. Parwiz: a dynamic dependency analyser

Participant: Alain Ketterlin.

We have developed 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.

5.6. APOLLO software and LLVM

Participants: Aravind Sukumaran-Rajam, Juan Manuel Martinez Caamaño, Willy Wolff, Alexandra Jimborean, Jean-François Dollinger, Philippe Clauss.

We are developing a new framework called APOLLO (Automatic speculative POLyhedral Loop Optimizer) whose main concepts are based on our previous framework VMAD. However, several important implementation issues are now handled differently in order to improve the performance and usability of the framework, and also to open its evolution to new interesting perspectives. Thus VMAD played the role of a prototype which is now being re-written as a sustainable tool named APOLLO. As VMAD, APOLLO is dedicated to automatic, dynamic and speculative parallelization of loop nests that cannot be handled efficiently at compile-time. It is composed of a static part consisting of specific passes in the LLVM compiler suite, plus a modified Clang frontend, and a dynamic part consisting of a runtime system. It is described in more details in subsection 6.1 .

Aravind Sukumaran-Rajam (PhD student), Juan Manuel Martinez Caamaño (PhD student), Jean-François Dollinger (PhD student), Willy Wolff (Master student) and Philippe Clauss are the main contributors of APOLLO. It will soon be distributed.

5.7. IBB source-to-source compiler

Participants: Imen Fassi, Philippe Clauss.

We have developed a multifor-compiler called IBB for Iterate-But-Better. IBB translates any C program containing multifor-loops into an equivalent C program in which all multifor-loops are replaced with equivalent for-loops. The resulting source code can then be compiled using any C compiler to produce executable code. IBB will soon be 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 [28] as a framework. This tool is written in the specification language of Coq.

5.9. CLooG

Participant: Cédric Bastoul.

CLooG [3] is a free software and library to generate code (or an abstract syntax tree of a code) for scanning Z-polyhedra. That is, it finds a code (e.g. in C, FORTRAN...) that reaches each integral point of one or more parameterized polyhedra. CLooG has been originally written to solve the code generation problem for optimizing compilers based on the polyhedral model. Nevertheless it is used now in various area e.g. to build control automata for high-level synthesis or to find the best polynomial approximation of a function. CLooG may help in any situation where scanning polyhedra matters. While the user has full control on generated code quality, CLooG is designed to avoid control overhead and to produce a very effective code. CLooG is widely used (including by GCC and LLVM compilers), disseminated (it is installed by default by the main Linux distributions) and considered as the state of the art in polyhedral code generation.

5.10. OpenScop

Participant: Cédric Bastoul.

OpenScop [4] is an open specification that defines a file format and a set of data structures to represent a static control part (SCoP for short), i.e., a program part that can be represented in the polyhedral model. The goal of OpenScop is to provide a common interface to the different polyhedral compilation tools in order to simplify their interaction. To help the tool developers to adopt this specification, OpenScop comes with an example library (under 3-clause BSD license) that provides an implementation of the most important functionalities necessary to work with OpenScop.

5.11. Clan

Participant: Cédric Bastoul.

Clan [5] is a free software and library which translates some particular parts of high level programs written in C, C++, C# or Java into a polyhedral representation called OpenScop. This representation may be manipulated by other tools to, e.g., achieve complex analyses or program restructurations (for optimization, parallelization or any other kind of manipulation). It has been created to avoid tedious and error-prone input file writing for polyhedral tools (such as CLooG, LeTSeE, Candl etc.). Using Clan, the user has to deal with source codes based on C grammar only (as C, C++, C# or Java). Clan is notably the frontend of the two major high-level compilers Pluto and PoCC.

5.12. Candl

Participant: Cédric Bastoul.

Candl [6] is a free software and a library devoted to data dependences computation. It has been developed to be a basic bloc of our optimizing compilation tool chain in the polyhedral model. From a polyhedral representation of a static control part of a program, it is able to compute exactly the set of statement instances in dependence relation. Hence, its output is useful to build program transformations respecting the original program semantics. This tool has been designed to be robust and precise. It implements some usual techniques for data dependence removal, as array privatization or array expansion, offers simplified abstractions like dependence vectors and performs violation dependence analysis. Candl is notably the dependence analyzer of the two major high-level compilers Pluto and PoCC.

5.13. Clay

Participant: Cédric Bastoul.

---

[3] http://www.cloog.org

[4] http://icps.u-strasbg.fr/~bastoul/development/openscop

[5] http://icps.u-strasbg.fr/~bastoul/development/clan

[6] http://icps.u-strasbg.fr/~bastoul/development/candl

Clay [7] is a free software and library devoted to semi-automatic optimization using the polyhedral model. It can input a high-level program or its polyhedral representation and transform it according to a transformation script. Classic loop transformations primitives are provided. Clay is able to check for the legality of the complete sequence of transformation and to suggest corrections to the user if the original semantics is not preserved. Clay is still experimental at this report redaction time but is already used during advanced compilation labs at Paris-Sud University and is one of the foundations of our ongoing work on simplifying code manipulation by programmers.

---

[7] http://icps.u-strasbg.fr/~bastoul/development/clay

CARAMEL Project-Team

5. Software and Platforms

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 (now AriC, 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 Joldeş 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 development in 2013 is the release of version 3.1.2 (the "canard à l'orange" release) in March. This version fixes a few bugs from previous version.

5.3. GNU MPC

Participant: Paul Zimmermann [contact].

GNU 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 GNU MPC library provides correct rounding on both the real part x and the imaginary part y of any result. GNU 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 and Sage computational number theory systems.

No new version of GNU MPC was released in 2013, which confirms the status of mature library.

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.

In February 2013, a new version 6.4.4 was released. Apart from bug fixes, this new release provides some improvements (better integration of the GPU code, of the new *-batch* option, ...).

In September 2013, a new record prime of 83 digits was found by R. Propper using GMP-ECM.

5.5. Finite Fields

Participants: Pierrick Gaudry, Emmanuel Thomé [contact], Luc Sanselme.

$\mathtt{mp}\mathbb{F}_q$ is (yet another) library for computing in finite fields. The purpose of $\mathtt{mp}\mathbb{F}_q$ 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*. $\mathtt{mp}\mathbb{F}_q$ can adapt to finite fields of any characteristic and any extension degree. However, one of the targets being the use in cryptology, $\mathtt{mp}\mathbb{F}_q$ somehow focuses on prime fields and on fields of characteristic two.

When it was first written in 2007, $\mathtt{mp}\mathbb{F}_q$ established 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. A stream of academic works followed the idea behind $\mathtt{mp}\mathbb{F}_q$ and improved over such timings, notably by Scott, Aranha, Longa, Bos, Hisil, Costello.

The library's purpose being the *generation* of code rather than its execution, the working core of $\mathtt{mp}\mathbb{F}_q$ consists of roughly 18,000 lines of Perl code, which generate most of the `C` code. $\mathtt{mp}\mathbb{F}_q$ is distributed at http://mpfq.gforge.inria.fr/.

In 2013, version 1.1 of $\mathtt{mp}\mathbb{F}_q$ has been released. This new release includes new assembly code by Luc Sanselme providing optimized arithmetic over fields whose characteristic fits in a number of bits which fit within half-word boundaries.

In 2013, Hamza Jeljeli collaborated with Bastien Vialla from LIRMM, Montpellier to integrate experimental code based on RNS arithmetic (Residue Number System), intending to provide back-end functionality for the linear algebra code in CADO-NFS. This feature set is still experimental.

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).

The current version of GF2X is 1.1, released in May 2012 under the GNU GPL. Since 2009, GF2X can be used as an auxiliary package for the widespread software library NTL, as of version 5.5.

In 2013, the development version of GF2X has been updated to incorporate detection of Intel Haswell microprocessors, which provide much improved performance for the PCLMULQDQ instruction (this instruction if of utmost importance for GF2X).

An LGPL-licensed portion of GF2X is also part of the CADO-NFS software package.

5.7. CADO-NFS

Participants: Cyril Bouvier, Jérémie Detrey, Alain Filbois, Pierrick Gaudry, Alexander Kruppa, Emmanuel Thomé [contact], Paul Zimmermann.

CADO-NFS is a program to factor integers using the Number Field Sieve algorithm (NFS), originally developped 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, and is referenced from the software page at http://cado-nfs.gforge.inria.fr/. A major new release, CADO-NFS 2.0, was published in November 2013. The client/server framework was completely rewritten to allow the use of CADO-NFS routinely on clusters of 100 to 1000 nodes.

More and more people use CADO-NFS to perform medium to large factorizations. Also in 2013 some researchers in the field wrote some papers where they study the implementation and default parameters of CADO-NFS. This is very useful feedback from the scientific community.

5.8. Belenios

Participants: Pierrick Gaudry, Stéphane Glondu [contact].

In collaboration with the CASSIS team, we develop an open-source private and verifiable electronic voting protocol, named BELENIOS. Our system is an evolution of an existing system, Helios, developed by Ben Adida, and used e.g. by UCL and the IACR association in real elections. The main differences with Helios are the following ones:

- In Helios, the ballot box publishes the encrypted ballots together with their corresponding voters. This raises a privacy issue in the sense that whether someone voted or not shall not necessarily be publicized on the web. Publishing this information is in particylar forbidden by CNIL's recommendation. BELENIOS no longer publishes voters' identities, still guaranteeing correctness of the tally.
- Helios is verifiable except that one has to trust that the ballot box will not add ballots. The addition of ballots is particularly hard to detect as soon as the list of voters is not public. We have therefore introduced an additional authority that provides cretentials that the ballot box can verify but bot forge.

This new version has been implemented by Stéphane Glondu [1]. and has been tested in July 2013 in a mock election in the teams CASSIS and CARAMEL.

---

[1] http://belenios.gforge.inria.fr/

CARTE Project-Team

5. Software and Platforms

5.1. Morphus/MMDEX

MMDEX is a virus detector based on morphological analysis. It is composed of our own disassembler tool, on a graph transformer and a specific tree-automaton implementation. The tool is used in the EU-Fiware project and by some other partners (e.g. DAVFI project).
Written in C, 20k lines.
APP License, IDDN.FR.001.300033.000.R.P.2009.000.10000, 2009.

5.2. TraceSurfer

TraceSurfer is a self-modifying code analyzer coming with an IDA add-on. It works as a wave-builder. In the analysis of self-modifying programs, one basic task is indeed to separate parts of the code which are self-modifying into successive layers, called waves. TraceSurfer extracts waves from traces of program executions. Doing so drastically simplifies program verification.
Written in C, 5k lines.
http://code.google.com/p/tartetatintools/

5.3. CROCUS

CROCUS is a program interpretation synthetizer. Given a first order program (possibly written in OCAML), it outputs a quasi-interpretation based on max, addition and product. It is based on a random algorithm. The interpretation is actually a certificate for the program's complexity. Users are non academics (some artists).
Written in Java, 5k lines.

CASCADE Project-Team (section vide)

CASSIS Project-Team

5. Software and Platforms

5.1. Protocol Verification Tools

Participants: Stéphane Glondu, Pierre-Cyrille Héam, Olga Kouchnarenko, Steve Kremer, Michaël Rusinowitch, Mathieu Turuani, Laurent Vigneron.

5.1.1. AVISPA

Cassis has been 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 [1] 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 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 [77]) allows *CL-AtSe* to be correct and complete. Each protocol step is represented by a constraint on the protocol state, used to check for reachability of the next state. *CL-AtSe* includes a proper handling of sets, choice points, specification of any attack states through a language for expressing e.g. secrecy, authentication, fairness, or 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).

CL-AtSe has been successfully used [65] 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 [82] for tool details and precise benchmarks).

Cl-Atse has been enhanced in various ways. In particular, the tool fully supports Aslan semantics introduced in [63], including Horn Clauses (for intruder-independent deductions, e.g. for credential management), and LTL-based security properties. Also, bug information and correction are processed through a bugzilla server, and online analysis and orchestration are available on our team server (https://cassis.loria.fr). Cl-Atse supports negative constraints on the intruder's knowledge [66]. This extension of Cl-Atse allows us to reduce drastically the orchestrator's processing times. It has also been used to model e.g. separation of duties and non-disclosure policies. We have also extended the syntax and semantics of ASLan to better model lists of undefined length, directly inside messages. Cl-AtSe tool now supports membership predicates, deletion operators ans so on for managing these lists, and offers a first reference implementation for other tools in Avantssar. In particular, the ASLan translator has been updated by our partners.

---

[1] http://www.avispa-project.org

5.1.3. Akiss

We develop the *Akiss* (Active Knowledge in Security Protocols) tool for verifying indistinguishability properties in cryptographic protocols. Indistinguishability properties are essential in formal verification of cryptographic protocols. They are needed to model anonymity properties, strong versions of confidentiality and resistance against offline guessing attacks, which can be conveniently modeled using process equivalences. *Akiss* implements a procedure to verify equivalence properties for a bounded number of sessions of cryptographic protocols. As in the applied pi-calculus, the protocol specification language is parametrized by a first-order sorted term signature and an equational theory which allows formalization of algebraic properties of cryptographic primitives. *Akiss* is able to verify trace equivalence for determinate cryptographic protocols. On determinate protocols, trace equivalence coincides with observational equivalence which can therefore be automatically verified for such processes. When protocols are not determinate *Akiss* can be used for both under- and over-approximations of trace equivalence, which proved successful on several examples. The procedure can handle a large set of cryptographic primitives, namely those that can be modeled by an optimally reducing convergent rewrite system.

The underlying procedure is based on a fully abstract modelling of the traces of a bounded number of sessions of the protocols into first-order Horn clauses on which a dedicated resolution procedure is used to decide equivalence properties. Although termination of the resolution procedure has not been proved, the procedure has been effectively tested on examples, some of which are outside the scope of other existing tools, including checking anonymity in several electronic voting protocols.

Recent developments include the possibility for checking everlasting indistinguishability properties. This feature was added when analyzing everlasting privacy properties in electronic voting protocols. We are currently working on a generalization of the procedure to allow associative-commutative operators and in particular a re-design of the resolution procedure for allowing analysis of protocols that use exclusive or. Expected case studies for this development include unlinkability in RFID protocols.

The *Akiss* tool is freely available at https://github.com/ciobaca/akiss.

5.1.4. Belenios

In collaboration with the Caramel team, we develop an open-source private and verifiable electronic voting protocol, named *Belenios*. Our system is an evolution of an existing system, Helios, developed by Ben Adida, and used e.g. by UCL and the IACR association in real elections. The main differences with Helios are the following ones:

- In Helios, the ballot box publishes the encrypted ballots together with their corresponding voters. This raises a privacy issue in the sense that whether someone voted or not shall not necessarily be publicized on the web. Publishing this information is in particular forbidden by the CNIL's recommendations. Belenios no longer publishes voters' identities, still guaranteeing the correctness of the tally.
- Helios is verifiable except that one has to trust that the ballot box will not add ballots. The addition of ballots is particularly hard to detect as soon as the list of voters is not public. We have therefore introduced an additional authority that provides credentials that the ballot box can verify but not forge.

This new version has been implemented by Stéphane Glondu and has been tested in July 2013 in a mock election in the teams Cassis and Caramel.

In a first step, *Belenios* has been implemented as an extension of existing Helios system. However, the existing software development of Helios is large and its security becomes difficult to assess. We have therefore re-implemented entirely the code of the bulletin box, yielding a now independent software [2].

---

[2] http://belenios.gforge.inria.fr/

In Helios as well as *Belenios*, votes are encrypted using the public key of the election. To ensure privacy, the corresponding decryption key is not known to anyone. Instead, several authorities detain a share of it. For robustness reasons (and as recommended by the CNIL), it is important to be able to decrypt even if some of the authorities are missing. We have implemented the threshold decryption scheme that we have proposed [40]. This implementation is currently available only within the Helios system and we plan to integrate it to *Belenios* in the next months.

5.2. Testing Tools

Participants: Fabrice Bouquet, Frédéric Dadeau, Kalou Cabrera.

5.2.1. Hydra

Hydra is an 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 transferringthe existing work on test generation from B abstract machines, JML, and statecharts using constraint solving techniques.

5.2.2. jMuHLPSL

jMuHLPSL [9] is a mutant generator tool that takes as input a verified HLPSL protocol, and computes mutants of this protocol by applying systematic mutation operators on its contents. The mutated protocol then has to be analyzed by a dedicated protocol analysis tool (here, the AVISPA tool-set). Three verdicts may then arise. The protocol can still be *safe*, after the mutation, this means that the protocol is not sensitive to the realistic “fault” represented by the considered mutation. This information can be used to inform the protocol designers of the robustness of the protocol w.r.t. potential implementation choices, etc. The protocol can also become *incoherent*, meaning that the mutation introduced a functional failure that prevents the protocol from being executed entirely (one of the participants remains blocked in a given non-final state). The protocol can finally become *unsafe* when the mutation introduces a security flaw that can be exploited by an attacker. In this case, the AVISPA tool-set is able to compute an attack-trace, that represents a test case for the implementation of the protocol. If the attack can be replayed entirely, then the protocol is not safe. If the attack can not be replayed then the implementation does not contain the error introduced in the original protocol.

The tool is written in Java, and it is freely available at: http://members.femto-st.fr/sites/femto-st.fr.frederic-dadeau/files/content/pub/jMuHLPSL.jar.

5.3. Collaborative Tools

Participant: Abdessamad Imine.

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.

5.3.1. P2PEdit

This prototype is implemented in Java and supports the collaborative editing of HTML pages and it is deployed on P2P JXTA platform [3]. 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.

5.3.2. 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). This application aims to make a collaborative calendar where users can simultaneously modify events (or appointements) 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.

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). We have also 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.

---

[3] http://www.sun.com/software/jxta/

CELTIQUE Project-Team

4. Software and Platforms

4.1. Javalib

Participants: Frédéric Besson [correspondant], David Pichardie, Pierre Vittet, Laurent Guillo.

Javalib is an efficient library to parse Java .class files into OCaml data structures, thus enabling the OCaml programmer to extract information from class files, to manipulate and to generate valid .class files.
See also the web page http://sawja.inria.fr/.

- Version: 2.3
- Programming language: Ocaml

4.2. SAWJA

Participants: Frédéric Besson [correspondant], David Pichardie, Pierre Vittet, Laurent Guillo.

Sawja is a library written in OCaml, relying on Javalib to provide a high level representation of Java bytecode programs. It name comes from Static Analysis Workshop for JAva. Whereas Javalib is dedicated to isolated classes, Sawja handles bytecode programs with their class hierarchy and with control flow algorithms.

Moreover, Sawja provides some stackless intermediate representations of code, called JBir and A3Bir. The transformation algorithm, common to these representations, has been formalized and proved to be semantics-preserving.
See also the web page http://sawja.inria.fr/.

- Version: 1.5
- Programming language: Ocaml

4.3. Jacal

Participants: Frédéric Besson [correspondant], Thomas Jensen, David Pichardie, Delphine Demange, Pierre Vittet.

Static program analysis, Javacard, Certification, AFSCM

`Jacal` is a JAvaCard AnaLyseur developed on top of the SAWJA (see Section 4.2) platform. This proprietary software verifies automatically that Javacard programs conform with the security guidelines issued by the AFSCM (Association Française du Sans Contact Mobile). `Jacal` is based on the theory of abstract interpretation and combines several object-oriented and numeric analyses to automatically infer sophisticated invariants about the program behaviour. The result of the analysis is thereafter harvest to check that it is sufficient to ensure the desired security properties.

4.4. Timbuk

Participant: Thomas Genet [correspondant].

Timbuk is a library of OCAML functions for manipulating tree automata. More precisely, Timbuk deals with finite bottom-up tree automata (deterministic or not). This library provides the classical operations over tree automata (intersection, union, complement, emptiness decision) as well as exact or approximated sets of terms reachable by a given term rewriting system. This last operation can be certified using a checker extracted from a Coq specification. The checker is now part of the Timbuk distribution. Timbuk distribution now also provides a CounterExample Guided Abstraction Refinement (CEGAR) tool for tree automata completion. The CEGAR part is based on the Buddy BDD library. Timbuk also provides an implementation of Lattice Tree Automata to (efficiently) represent built-in values such as integers, strings, etc. in recognized tree languages. See also the web page http://www.irisa.fr/celtique/genet/timbuk/.

- Version: 3.1
- Programming language: Ocaml

4.5. JSCert

Participants: Martin Bodin, Alan Schmitt.

The JSCert project aims to really understand JavaScript. JSCert itself is a mechanised specification of JavaScript, written in the Coq proof assistant, which closely follows the ECMAScript 5 English standard. JSRef is a reference interpreter for JavaScript in OCAML, which has been proved correct with respect to JSCert and tested with the Test 262 test suite.

We plan to build other verification and analysis projects on top of JSCert and JSRef, in particular the certification of derivations in program logics or static analyses.

This project is an ongoing collaboration between Inria and Imperial College. More information is available at http://jscert.org/.

COMETE Project-Team

5. Software and Platforms

5.1. Location Guard

Participants: Konstantinos Chatzikokolakis [correspondant], Marco Stronati.

The purpose of *Location Guard* is to implement obfuscation techniques for achieving location privacy, in a an easy and intuitive way that makes them available to the general public. Various modern applications, running either on smartphones or on the web, allow third parties to obtain the user's location. A smartphone application can obtain this information from the operating system using a system call, while web application obtain it from the browser using a javascript call.

Although both mobile operating systems and browsers require the user's permission to disclose location information, the user faces an "all-or-nothing" choice: either disclose his exact location and give up his privacy, or stop using the application. This forces many users to disclose their location, although ideally they would like to enjoy some privacy.

The API level of a browser or an operating system would be an ideal place for integrating a location obfuscation technique, in a way that is easy to understand for the average user, and readily available to all applications. When an application asks for the user's location, the browser or operating system can ask the user's permission, but including the option to provide an obfuscated location instead of the real one! Different levels of obfuscation can be also offered, so that the user can chose to provide more accurate location to applications that really need it, and more noisy location to those that don't.

A prototype of Location Guard has been already implemented for Google Chrome. In the future we plan to extend it to other desktop and mobile browsers (Firefox, Internet Explorer, etc), as well as to implement it in modern mobile operating systems, primarily on Android.

https://github.com/chatziko/location-guard

5.2. PRISM model generator

Participants: Konstantinos Chatzikokolakis [correspondant], Catuscia Palamidessi.

This software generates PRISM models for the Dining Cryptographers and Crowds protocols. It can also use PRISM to calculate the capacity of the corresponding channels. More information can be found in [29] and in the file README file width instructions at the URL http://www.lix.polytechnique.fr/comete/software/README-anonmodels.html.

The software can be download at http://www.lix.polytechnique.fr/comete/software/anonmodels.tar.gz. These scripts require Perl to run and have been tested in Linux. The GUI of the corners tool also requires the Perl/TK library. Finally some parts of the model generator tool require PRISM and gnuplot to be installed.

5.3. Calculating the set of corner points of a channel

Participants: Konstantinos Chatzikokolakis [correspondant], Catuscia Palamidessi.

The corner points can be used to compute the maximum probability of error and to improve the Hellman-Raviv and Santhi-Vardy bounds. More information can be found in [30] and in the file README file width instructions at the URL http://www.lix.polytechnique.fr/comete/software/README-corners.html.

The software can be download at http://www.lix.polytechnique.fr/comete/software/corners.tar.gz. These scripts require Perl to run and have been tested in Linux. The GUI of the corners tool also requires the Perl/TK library. Finally some parts of the model generator tool require PRISM and gnuplot to be installed.

5.4. MMCsp, a compiler for the π-calculus

Participant: Catuscia Palamidessi [correspondant].

MMCsp is a compiler from a simple probabilistic π-calculus to PRISM. models. It is built on XSB, a tabled logic programming system, and generates the symbolic semantic representation of a probabilistic pi-calculus term in text. A separate Java program then translates this semantic representation into a probabilistic model for PRISM.

The tool was developed by Peng Wu during his postdoc period in Comète in 2005-2007, in the context of the collaboration between the teams Comète and PRISM under the Inria/ARC Project ProNoBis. It is based on the papers [32] and [31].

The source code is free and can be download from http://www.cs.ucl.ac.uk/staff/p.wu/mmc_sp_manual.html.

COMPSYS Project-Team

5. Software and Platforms

5.1. Introduction

This section lists and briefly describes the software developments conducted within Compsys. Most are tools that we extend and maintain over the years. They mainly concern three activities: a) the development of research tools, in general available on demand, linked to polyhedra and loop/array transformations, b) the development of tools linked to the start-up Zettice, in general not available, c) the development of algorithms within the back-end compilers of STMicroelectronics and/or Kalray.

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

The other activity concerns the developments within the compilers of industrial partners such as STMicroelectronics and Kalray. These are not stand-alone tools, which could be used externally, but algorithms and data structures implemented inside the LAO back-end compiler or other compiler branches, year after year, with the help of STMicroelectronics or Kalray colleagues. They are also completed by important efforts for integration and evaluation within the complete compiler toolchains. They concern exact (ILP-based) methods, algorithms for aggressive optimizations, techniques for just-in-time compilation, code representations, and for improving the design of the compiler.

More recently, an important development activity has been started in the context of the Zettice start-up project (see Section 7.3). An important effort of applied research and software development has been achieved since, which results, in particular, in two major software developments: Dcc (DPN C Compiler) and IceGEN. These tools are outlined in Sections 5.8 and 5.9 .

5.2. Pip

Participants: Cédric Bastoul [professor, Strasbourg University and Inria/CAMUS], Paul Feautrier.

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

5.3. Syntol

Participant: Paul Feautrier.

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

- The new code generator, based on the ideas of Boulet and Feautrier [16], generates a counter automaton that can be presented as a C program, as a rudimentary VHDL program at the RTL level, as an automaton in the Aspic input format, or as a drawing specification for the DOT tool.
- Hardware synthesis can only be applied to bounded parallelism programs. Our present aim is to construct threads with the objective of minimizing communications and simplifying synchronization. The distribution of operations among threads is specified using a placement function, which is found using techniques of linear algebra and combinatorial optimization.

5.4. Cl@k

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

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

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

5.5. PoCo

Participant: Christophe Alias.

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

- A C front-end extracting the polyhedral representation of the input program. The parser itself is based on EDG (*via* Rose), an industrial C/C++ parser from Edison group used in Intel compilers.
- An extended language of pragmas to feed the source code with compilation directives (a schedule, for example).
- A symbolic layer on polyhedral libraries Polylib (set operations on polyhedra) and Piplib (parameterized ILP, see Section 5.2). This feature simplifies drastically the developer task.
- Some dependence analysis (polyhedral dependence graph, array dataflow analysis), array region analysis, array liveness analysis.
- A C and VHDL code generation based on the ideas of P. Boulet and P. Feautrier [16].

The array dataflow analysis (ADA) of PoCo has been extended to a FADA (Fuzzy ADA) by M. Belaoucha, former PhD student at Université de Versailles. FADALib is available at https://bitbucket.org/mbelaoucha/fadalib. PoCo has been developed by Christophe Alias. It represents more than 19000 lines of C++ code. The tools Bee, Chuba, and RanK presented thereafter make an extensive use of PoCo abstractions.

5.6. Bee

Participants: Christophe Alias, Alain Darte.

Bee is a source-to-source optimizer that contracts the temporary arrays of a program under scheduling constraints. Bee bridges the gap between the mathematical optimization framework described in [17] and implemented in Cl@k (Section 5.4), and effective source-to-source array contraction. Bee applies a precise lifetime analysis for arrays to build the mathematical input of Cl@k. Then, Bee derives the array allocations from the basis found by Cl@k and generates the C code accordingly. Bee is – to our knowledge – the only complete array contraction tool.

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

- Bee was made available to STMicroelectronics as a binary.
- Bee has been transferred to the (incubated) start-up Zettice, initiated by Alexandru Plesco.
- Bee has been used as an external tool by the compiler Gecos developed in the Cairn team at Irisa.

Bee has been implemented by Christophe Alias, using the compiler infrastructure PoCo (see Section 5.5). It represents more than 2400 lines of C++ code.

5.7. Chuba

Participants: Christophe Alias, Alain Darte, Alexandru Plesco [Compsys/Zettice].

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

So far, the back-end of Chuba is specific to the HLS tool C2H but the analysis is quite general and adapting Chuba to other HLS tools should be possible. Besides, it is interesting to mention that the program analysis and optimizations implemented in Chuba address a problem that is also very relevant in the context of GPGPUs. The underlying theory and corresponding experiments are described in [4].

Chuba has been implemented by Christophe Alias, using the compiler infrastructure PoCo (see Section 5.5). It represents more than 900 lines of C++. The reduced size of Chuba is mainly due to the high-level abstractions provided by PoCo.

5.8. Dcc

Participants: Christophe Alias, Alexandru Plesco [Compsys/Zettice].

Dcc (DPN C Compiler) is the *front-end* of the HLS tool transferred to the start-up Zettice (see Section 7.3). Dcc takes as input a C program annotated with pragmas and produces an optimized data-aware process network (DPN). A DPN is a regular process network that makes explicit the I/O transfers and the synchronizations. Dcc features throughput optimization, communication vectorization, and automatic parallelization. Furthermore, Dcc applies analysis to build the DPN circuitry: multiplexing, channels sizing and allocation, FSM generation. To do so, Dcc uses extensively the analysis implemented in PoCo (Section 5.5), in particular dataflow analysis and control generation, and Bee (Section 5.6 for buffer sizing. The DPN specific analysis of Dcc is currently under patent deposit.

Dcc represents more than 3000 lines of C++ code.

5.9. IceGEN

Participants: Christophe Alias, Alexandru Plesco [Compsys/Zettice].

IceGEN (Integrated Circuit Generator) is the *back-end* of the HLS tool transferred to the start-up Zettice (see Section 7.3). IceGEN takes as input the DPN produced by Dcc (see Section 5.8) and generates:

- a SystemC description relevant for fast and accurate circuit simulation.
- a VHDL description of the circuit, which can be mapped efficiently to an FPGA.

IceGEN makes an extensive use of the pipelined arithmetic operators of the tool FloPoCo [18] developed by Florent De Dinechin, formerly from Inria ARIC team.

IceGEN represents more than 6000 lines of C++ code.

5.10. C2fsm

Participant: Paul Feautrier.

C2fsm is a general tool that converts an arbitrary C program into a counter automaton. This tool reuses the parser and pre-processor of Syntol (see Section 5.3), which has been greatly extended to handle `while` and `do while` loops, `goto`, `break`, and `continue` statements. C2fsm reuses also part of the code generator of Syntol and has several output formats, including FAST (the input format of Aspic, see Section 5.11), a rudimentary VHDL generator, and a DOT generator which draws the output automaton. C2fsm is also able to do elementary transformations on the automaton, such as eliminating useless states, transitions and variables, simplifying guards, or selecting cut-points, i.e., program points on loops that can be used by RanK (see Section 5.12) to prove program termination.

5.11. Aspic

Participant: Laure Gonnord.

Aspic is an invariant generator for general counter automata. Used with C2fsm (see Section 5.10), it can be used to derivate invariant for numerical C programs, and also prove safety. It is also part of the WTC toolsuite (see http://compsys-tools.ens-lyon.fr/wtc/index.html), a set of examples to demonstrate the capability of the RanK tool (see Section 5.12) for evaluating worse-case time complexity (number of transitions when executing an automaton).

Aspic implements the theoretical results of Laure Gonnord's PhD thesis on acceleration techniques and has been maintained since 2007.

5.12. RanK

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

RanK is a software tool that can prove the termination of a program (in some cases) by computing a *ranking function*, i.e., a mapping from the operations of the program to a well-founded set that *decreases* as the computation advances. In case of success, RanK can also provide an upper bound of the worst-case time complexity of the program as a symbolic affine expression involving the input variables of the program (parameters), when it exists. In case of failure, RanK tries to prove the non-termination of the program and then to exhibit a counter-example input. This last feature is of great help for program understanding and debugging, and has already been experimented. The theory underlying RanK was presented at SAS'10 [14].

The input of RanK is an integer automaton, computed by C2fsm (see Section 5.10), representing the control structure of the program to be analyzed. RanK uses the Aspic tool (see Section 5.11), developed by Laure Gonnord during her PhD thesis, to compute automaton invariants. RanK has been used to discover successfully the worst-case time complexity of many benchmarks programs of the community (see the WTC benchmark suite http://compsys-tools.ens-lyon.fr/wtc/index.html). It uses the libraries Piplib (Section 5.2) and Polylib.

RanK has been implemented by Christophe Alias, using the compiler infrastructure PoCo (Section 5.5). It represents more than 3000 lines of C++. The tool has been presented at the CSTVA'13 workshop [11].

5.13. SToP

Participants: Christophe Alias, Guillaume Andrieu [University of Lille], Laure Gonnord [Compsys].

SToP (Scalable Termination of Programs) is the implementation of the modular termination technique presented at the TAPAS'12 workshop [15]. It takes as input a large irregular C program and conservatively checks its termination. To do so, SToP generates a set of small programs whose termination implies the termination of the whole input program. Then, the termination of each small program is checked thanks to RanK (see Section 5.12). In case of success, SToP infers a ranking (schedule) for the whole program. This schedule can be used in a subsequent analysis to optimize the program.

SToP represents more than 2000 lines of C++.

5.14. Simplifiers

Participant: Paul Feautrier.

The aim of the `simple` library is to simplify Boolean formulas on affine inequalities. It works by detecting redundant inequalities in the representation of the subject formula as an ordered binary decision diagram (OBDD), see details in [22]. It uses PIP (see Section 5.2) for testing the feasibility – or unfeasibility – of a conjunction of affine inequalities.

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

5.15. LAO Developments in Aggressive Compilation

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

Our past aggressive optimization techniques are all implemented in stand-alone experimental tools (as for example for register coalescing algorithms) or within LAO, the back-end compiler of STMicroelectronics, or both. They concern SSA construction and destruction, instruction-cache optimizations, register allocation. Here, we report only our activities related to register allocation.

Our developments on register allocation within the STMicroelectronics compiler started when Cédric Vincent (bachelor degree, under Alain Darte supervision) developed a complete register allocator in LAO, the assembly-code optimizer of STMicroelectronics. This was the first time a complete implementation was done with success, outside the MCDT (now CEC) team, in their optimizer. This continued with developments made during the master internships and PhD theses of Florent Bouchez, Benoit Boissinot, and Quentin Colombet, and post-doctoral works of Sebastian Hack and Florian Brandner. In 2009, Quentin Colombet started to develop and integrate into the main trunk of LAO a full implementation of a two-phases register allocation. This implementation now includes two different decoupled spilling phases, the first one as described in Sebastian Hack's PhD thesis and a second ILP-based solution. It also includes an up-to-date graph-based register coalescing. Finally, since all these optimizations take place under SSA form, it includes also a mechanism for going out of colored-SSA (register-allocated SSA) form that can handle critical edges and does further optimizations. See details in the "new results" presented in previous Compsys activity reports.

5.16. LAO Developments in JIT Compilation

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

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

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

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

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

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

Detailed information, generic core tools, and LLVM IR based generator for MinIR are available at http://www.assembla.com/spaces/minir-dev/wiki. MinIR was presented at WIR'11 [29].

CONTRAINTES Project-Team

5. Software and Platforms

5.1. BIOCHAM, biochemical abstract machine

Participants: François Fages, François-Marie Floch, Steven Gay, Sylvain Soliman.

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

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

Based on this formal framework, BIOCHAM features:

- Boolean and numerical simulators (Rosenbrock's method for the differential semantics, Gillespie's algorithm with tau lipping for the stochastic semantics);
- a temporal logic language (CTL for qualitative models and LTL(R_{lin}) with numerical constraints for quantitative models) for formalizing biological properties such as reachability, checkpoints, oscillations or stability, and checking them automatically with model-checking techniques;
- automatic search procedures to infer parameter values, initial conditions and even reaction rules from temporal logic properties;
- automatic detection of invariants, through constraint-based analysis of the underlying Petri net;
- automatic model reduction and comparison, through the use of subgraph epimorphisms [8];
- an SBGN-compatible reaction graph editor;
- an event handler allowing the encoding of hybrid models and formalisms.

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

5.2. Nicotine

Participant: Sylvain Soliman.

Nicotine is a Prolog framework dedicated to the analysis of Petri nets. It was originally built for the computation of invariants using GNU Prolog's CLP(FD) solver but has been further extended to allow import/export of various Petri nets formats. In 2013 it was ported to SWI Prolog, in order to use its more general FD solver for the satisfaction of the min-plus constraints coming from the tropical equilibration problem [13].

5.3. STSE (Spatio-Temporal Simulation Environment)

Participant: Szymon Stoma.

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

5.4. YeastImageToolkit

Participants: Szymon Stoma, Grégory Batt, Pascal Hersen, Artémis Llamosi.

Yeast Image Toolkit (YIT) is a set of tools facilitating segmentation and tracking of yeast cells in brightfield images. Toolkit consists of novel segmentation and tracking algorithm (CellStar), benchmark images and software allowing allowing to asses performance of the tracking. The software is currently under development and is designed to be a CellProfiler plugin. Project webpage is: http://yeast-image-toolkit.biosim.eu/.

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

Participants: François Fages, Thierry Martinez.

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

5.6. Rules2CP

Participants: François Fages, Raphaël Martin, Thierry Martinez.

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

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

5.7. SiLCC, linear concurrent constraint programming

Participant: Thierry Martinez.

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

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

5.8. EMoP, existential modules for Prolog

Participant: Thierry Martinez.

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

5.9. CHRat, CHR with ask and tell

Participant: Thierry Martinez.

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

5.10. CLPGUI, constraint logic programming graphical user interface

Participant: François Fages.

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

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

CONVECS Project-Team

5. Software and Platforms

5.1. The CADP Toolbox

Participants: Hubert Garavel [correspondent], Frédéric Lang, Radu Mateescu, Wendelin Serwe.

We maintain and enhance CADP (*Construction and Analysis of Distributed Processes* – formerly known as *CAESAR/ALDEBARAN Development Package*) [4], a toolbox for protocols and distributed systems engineering [1]. In this toolbox, we develop and maintain the following tools:

- CAESAR.ADT [41] is a compiler that translates LOTOS abstract data types into C types and C functions. The translation involves pattern-matching compiling techniques and automatic recognition of usual types (integers, enumerations, tuples, etc.), which are implemented optimally.
- CAESAR [47], [46] is a compiler that translates LOTOS processes into either C code (for rapid prototyping and testing purposes) or finite graphs (for verification purposes). The translation is done using several intermediate steps, among which the construction of a Petri net extended with typed variables, data handling features, and atomic transitions.
- OPEN/CAESAR [42] is a generic software environment for developing tools that explore graphs on the fly (for instance, simulation, verification, and test generation tools). Such tools can be developed independently of any particular high level language. In this respect, OPEN/CAESAR plays a central role in CADP by connecting language-oriented tools with model-oriented tools. OPEN/CAESAR consists of a set of 16 code libraries with their programming interfaces, such as:
  - CAESAR_GRAPH, which provides the programming interface for graph exploration,
  - CAESAR_HASH, which contains several hash functions,
  - CAESAR_SOLVE, which resolves Boolean equation systems on the fly,
  - CAESAR_STACK, which implements stacks for depth-first search exploration, and
  - CAESAR_TABLE, which handles tables of states, transitions, labels, etc.

  A number of on-the-fly analysis tools have been developed within the OPEN/CAESAR environment, among which:
  - BISIMULATOR, which checks bisimulation equivalences and preorders,
  - CUNCTATOR, which performs steady-state simulation of continuous-time Markov chains,
  - DETERMINATOR, which eliminates stochastic nondeterminism in normal, probabilistic, or stochastic systems,
  - DISTRIBUTOR, which generates the graph of reachable states using several machines,
  - EVALUATOR, which evaluates MCL formulas,
  - EXECUTOR, which performs random execution,
  - EXHIBITOR, which searches for execution sequences matching a given regular expression,
  - GENERATOR, which constructs the graph of reachable states,
  - PROJECTOR, which computes abstractions of communicating systems,
  - REDUCTOR, which constructs and minimizes the graph of reachable states modulo various equivalence relations,

---

[1] http://cadp.inria.fr

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

    For instance, one can define recursive functions on sets of states, which allow evaluation and diagnostic generation fixed point algorithms for usual temporal logics (such as HML [51], CTL [36], ACTL [37], etc.) to be defined in XTL.
- PBG (*Partitioned BCG Graph*) is a file format implementing the theoretical concept of *Partitioned LTS* [45] and providing a unified access to a graph partitioned in fragments distributed over a set of remote machines, possibly located in different countries. The PBG format is supported by several tools, such as:
  - PBG_CP, PBG_MV, and PBG_RM, which facilitate standard operations (copying, moving, and removing) on PBG files, maintaining consistency during these operations,
  - PBG_MERGE (formerly known as BCG_MERGE), which transforms a distributed graph into a monolithic one represented in BCG format,
  - PBG_INFO, which displays various statistical information about a distributed graph.
- The connection between explicit models (such as BCG graphs) and implicit models (explored on the fly) is ensured by OPEN/CAESAR-compliant compilers, e.g.:
  - BCG_OPEN, for models represented as BCG graphs,
  - CAESAR.OPEN, for models expressed as LOTOS descriptions,
  - EXP.OPEN, for models expressed as communicating automata,
  - FSP.OPEN, for models expressed as FSP [57] descriptions,
  - LNT.OPEN, for models expressed as LNT descriptions, and
  - SEQ.OPEN, for models represented as sets of execution traces.

The CADP toolbox also includes TGV (*Test Generation based on Verification*), which has been developed by the VERIMAG laboratory (Grenoble) and the VERTECS project-team at Inria Rennes – Bretagne-Atlantique.

The CADP tools are well-integrated and can be accessed easily using either the EUCALYPTUS graphical interface or the SVL [43] scripting language. Both EUCALYPTUS and SVL provide users with an easy and uniform access to the CADP tools by performing file format conversions automatically whenever needed and by supplying appropriate command-line options as the tools are invoked.

5.2. The TRAIAN Compiler

Participants: Hubert Garavel [correspondent], Frédéric Lang, Wendelin Serwe.

We develop a compiler named TRAIAN for translating LOTOS NT descriptions into C programs, which will be used for simulation, rapid prototyping, verification, and testing.

The current version of TRAIAN, which handles LOTOS NT types and functions only, has useful applications in compiler construction [44], being used in all recent compilers developed by CONVECS.

The TRAIAN compiler can be freely downloaded from the CONVECS Web site [2].

5.3. The PIC2LNT Translator

Participants: Radu Mateescu, Gwen Salaün [correspondent].

We develop a translator named PIC2LNT from an applied π-calculus (see § 6.1) to LNT, which enables the analysis of concurrent value-passing mobile systems using CADP.

PIC2LNT is developed by using the SYNTAX tool (developed at Inria Paris-Rocquencourt) for lexical and syntactic analysis together with LOTOS NT for semantical aspects, in particular the definition, construction, and traversal of abstract trees.

The PIC2LNT translator can be freely downloaded from the CONVECS Web site [3].

5.4. The PMC Partial Model Checker

Participants: Radu Mateescu, Frédéric Lang.

We develop a tool named PMC (*Partial Model Checker*, see § 6.4), which performs the compositional model checking of dataless MCL formulas on networks of communicating automata described in the EXP language.

PMC can be freely downloaded from the CONVECS Web site [4].

---

[2] http://convecs.inria.fr/software/traian

[3] http://convecs.inria.fr/software/pic2lnt

[4] http://convecs.inria.fr/software/pmc

CRYPT Team (section vide)

DEDUCTEAM Exploratory Action

5. Software and Platforms

5.1. Dedukti

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

Dedukti's core is based on the standard algorithm [42] for type-checking semi-full pure type systems and implements a state-of-the-art reduction machine inspired from Matita's [40] and modified to deal with rewrite rules.

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

Dedukti has been developed by Mathieu Boespflug, Olivier Hermant, Quentin Carbonneaux, and Ronan Saillard. It is composed of about 1000 lines of OCaml.

5.2. Coqine, Holide and Focalide

Dedukti comes with three companion tools: Holide, an embedding of HOL proofs through the OpenTheory format [51], Coqine, an embedding of Coq proofs, and Focalide, an embedding of FoCaLiZe certified programs. All of the OpenTheory standard library and a part of Coq's and FoCaLiZe's libraries are checked by Dedukti.

A preliminary version of Coqine supports the following features of Coq: the raw Calculus of Constructions, inductive types, and fixpoint definitions. Coqine is currently being rewritten to support universes. Coqine has been developed by Mathieu Boespflug, Guillaume Burel, and Ali Assaf.

Holide supports all the features of HOL, including ML-polymorphism, constant definitions, and type definitions. It is able to translate all of the OpenTheory standard theory library. Holide has been developed by Ali Assaf.

Focalide supports the object-oriented features of FoCaLiZe, including inheritance, late-binding, redefinition and class parameters, and functional programming features of FoCaLiZe. It has been updated to work with the last version of FoCaLiZe. Focalide has been developed by Raphaël Cauderlier.

5.3. iProver Modulo

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

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

iProver Modulo and autotheo have been developed by Guillaume Burel. iProver Modulo is released under a GPL license.

iProver Modulo entered CASC-24, the competition of Automated Theorem Provers held during the 24th CADE conference, in the first-order theorem division, using autotheo to orient the axioms of the problems.

5.4. Super Zenon and Zenon Modulo

Several extensions of the *Zenon* automated theorem prover (developed by Damien Doligez at *Inria* in the *Gallium* team) to Deduction modulo have been studied. These extensions intend to be applied in the context of the automatic verification of proof rules and obligations coming from industrial applications formalized using the *B* method.

The first extension, developed by Mélanie Jacquel and David Delahaye, is called *Super Zenon* and is an extension of *Zenon* to superdeduction, which can be seen as a variant of Deduction modulo. This extension is a generalization of previous experiments [10] together with Catherine Dubois and Karim Berkani (*Siemens*), where *Zenon* has been used and extended to superdeduction to deal with the *B* set theory and automatically prove proof rules of *Atelier B*. This generalization consists in allowing us to apply the extension of *Zenon* to superdeduction to any first order theory by means of a heuristic that automatically transforms axioms of the theory into rewrite rules. This work is described in [5], which also proposes a study of the possibility of recovering intuition from automated proofs using superdeduction.

The second extension, developed by Pierre Halmagrand, David Delahaye, Damien Doligez, and Olivier Hermant, is called *Zenon Modulo* and is an extension of *Zenon* to Deduction modulo. Compared to *Super Zenon*, this extension allows us to deal with rewrite rules both over propositions and terms. Like *Super Zenon*, *Zenon Modulo* is able to deal with any first order theory by means of a similar heuristic. To assess the approach of *Zenon Modulo*, we have applied this extension to the first order problems coming from the TPTP library. An increase of the number of proved problems has been observed, with in particular a significant increase in the category of set theory. This result in the set category allows us to be quite optimistic for the use of *Zenon Modulo* in the framework of the *BWare* project, since the *B* method is actually based on a set theory modeling technique. Over these problems of the TPTP library, we have also observed a significant proof size reduction, which confirms this aspect of Deduction modulo. These results are gathered into two publications [22], [23].

5.5. Zipperposition and Logtk

Zipperposition is an implementation of the superposition method. It experiments theory handling using its extensibility features. Current development includes splitting it into a generic library for representing logic data structures and algorithms, and a prover that uses this library. The library is called LOGTK("logic tool kit"). Zipperposition itself, in its development version, can deal with polymorphic logic, and integer and rational arithmetic. Theoretical work on an efficient inference system for arithmetic is ongoing. It entered CASC-24, the competition of Automated Theorem Provers held during the 24th CADE conference, in the first-order theorem division.

Zipperposition is developed by Simon Cruanes.

Logtk is in active development, in parallel with Zipperposition, and an unstable version is released here. The library, among other things, provides first-order terms, with polymorphic types and some type inference, first-order formulas, unification, term ordering, term rewriting, reduction of formulas to CNF, congruence closure, and an optional implementation of the meta-prover Simon Cruanes and Guillaume Burel have published about [21]. Logtk focuses on efficiency and generality of its constructs. Several term indexing structures usable for rewriting, resolution, or subsumption checking expose a functorial interface that allows to associate any data with indexed terms.

Logtk also relies on some smaller OCaml developments by Simon Cruanes, especially a full-fledged implementation of Datalog and efficient iterators.

5.6. CoLoR

CoLoR is a Coq library on rewriting theory and termination of more than 83,000 lines of code [2]. It provides definitions and theorems for:

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

CoLoR is distributed under the CeCILL license. It is currently developed by Frédéric Blanqui and Kim-Quyen Ly, but various people participated to its development since 2006 (see the website for more information).

5.7. HOT

HOT is an automated termination prover for higher-order rewrite systems based on the notion of computability closure and size annotation [44]. It won the 2012 competition in the category “higher-order rewriting union beta”. The sources are not public. It is developed by Frédéric Blanqui.

5.8. Moca

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

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

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

Moca supports two kinds of relations:

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

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

Moca is distributed under QPL. It is developed by Frédéric Blanqui, Pierre Weis (EPI Pomdapi) and Richard Bonichon (CEA).

5.9. Rainbow

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

Rainbow is distributed under the CeCILL license. It is currently developed by Frédéric Blanqui and Kim-Quyen Ly. See the website for more information.

DICE Team

5. Software and Platforms

5.1. GPeer: a peer-to-peer javascript communication library

Our software development has been oriented towards systems working in browsers, with the support of an Inria ADT project in cooperation with the ASAP team. To answer our technological objectives, we are working on decentralized architectures, browser to browser, developed in javascript/HTML5. We rely on the WebRTC JavaScript protocol proposed by Google to develop a communication layer between peers. Many peer-to-peer protocols share common elements, that we group in a generic library for developing peer-to-peer systems. The joint library developed with the ASAP team handles any gossip based communication overlay. We design peer messages, tracker management and resilient behavior. The library is a standard bridge between complex browser to browser applications and low level networking layers such as WebRTC. With the use of our library, we can reproduce systems such as bitTorrent, but also provide new applications without the need of either native applications or identified servers.

5.2. Fluxion: a software plugin for flows in AngularJS

The joint project with Worldline aims at managing mobile code in complex Web architectures. Load variation in data-centers is currently poorly resolved. Most of the time, systems overestimate resource consumption in order to absorb burst usage. These consumption overestimation has a cost both in terms of the SLA negotiated with the client and the non-availability of reserved resources. With Wordline we focus on code mobility for high performance Web architectures and design a fast and reactive framework, transparently moving functions between running systems. The Fluxion model is our approach to design mobile application modules that are a mix of functional programming and flow based reactive systems.

5.3. BitBallot: a decentralized voting protocol

The BitBallot voting protocol is designed to target large scale communities. The protocol allows users to share only restricted amounts of their data and computation with central platforms as well as other peers. Convinced by the need of new election mechanisms, to support emerging forms of more continuous democracy, we are developing BitBallot, to allow elections to be organized independently of any central authority. The protocol guarantees the following properties, anonymity of the data sources, non interruptible run-time, global access to results, and non predictability of results through partial communication spying.

DREAMPAL Team

5. Software and Platforms

5.1. Software and Platforms

Download page : https://gforge.inria.fr/frs/?group_id=3646

5.1.1. HoMade

HoMade V4 is available and was used by 140 students this year on a Xilinx Nexys3 board. The Xilinx Virtex6 and Virtex7 are also supporting this new release. All the design is in VHDL except some ISE schematic specifications.

The main novelty of this release concerns :

- three stage pipe-lining of the HoMade core,
- new execution stack to improve frequency,
- instruction memory loading via UART port,
- MSPMD support
- reflexive features: Write-In-Program-Memory (WIM) instruction
- development of new IPs.

A test with 56 HoMade Slaves on a ring topology was running on a Virtex6 on a parallel matrix-vector multiplication example.

A low-level stack-based assembler supports binary generation from a Forth-like post-fixed syntax. It is written in Forth and can automatically generate binary code for the UART Port loading. This assembler will be merged with the JHomade software.

5.1.2. JHomade

JHomade is a software suite including a compiler for the HoMade processor. It allows us to compile HiHope programs (or HoMade assembly) and load the binary on the FPGA board. Its first release was in december 2013.

5.1.3. $\mathbb{K}$-based language-independent symbolic execution and verification tool

The results in [9], [14] were implemented in the $\mathbb{K}$ framework and are distributed with it. The implementation allows users to symbolically execute programs in arbitrary languages defined in $\mathbb{K}$, with the only restriction that data cannot become code (and reciprocally). It also allows users to formally verify programs against specifications written in Reachability Logic, a specification formalism that can be seen as a language-independent Hoare logic. These language-independent tools will be specialized to the languages of interest in the project (HiHope, HoMade assembly and machine code).

ESPRESSO Project-Team

5. Software and Platforms

5.1. The Polychrony toolset and its hypertext source documentation

Participants: Loïc Besnard, Thierry Gautier, Paul Le Guernic.

The Polychrony toolset is an Open Source development environment for critical/embedded systems. It is based on Signal, a real-time polychronous dataflow language. It provides a unified model-driven environment to perform design exploration by using top-down and bottom-up design methodologies formally supported by design model transformations from specification to implementation and from synchrony to asynchrony. It can be included in heterogeneous design systems with various input formalisms and output languages.

The Polychrony toolset provides a formal framework:

- to validate a design at different levels, by the way of formal verification and/or simulation,
- to refine descriptions in a top-down approach,
- to abstract properties needed for black-box composition,
- to assemble heterogeneous predefined components (bottom-up with COTS),
- to generate executable code for various architectures.

The Polychrony toolset contains three main components and an experimental interface to GNU Compiler Collection (GCC):

- The Signal toolbox, a batch compiler for the Signal language, and a structured API that provides a set of program transformations. The Signal toolbox can be installed without other components. The Signal toolbox is distributed under GPL V2 license.
- The Signal GUI, a Graphical User Interface to the Signal toolbox (editor + interactive access to compiling functionalities). The Signal GUI is distributed under GPL V2 license.
- The SME/SSME platform, a front-end to the Signal toolbox in the Eclipse environment. The SME/SSME platform is distributed under EPL license.
- GCCst, a back-end to GCC that generates Signal programs (not yet available for download).

In 2013, to be able to use the Signal GUI both as a specific tool and as a graphical view under Eclipse, the code of the Signal GUI has been restructured in three parts: a common part used by both tools (28 classes), a specific part for the Signal GUI (2 classes), a specific part for Eclipse (2 classes). Such a structuration facilitates the maintenance of the products.

The Polychrony toolset also provides:

- libraries of Signal programs,
- a set of Signal program examples,
- user oriented and implementation documentations,
- facilities to generate new versions.

The building of the Signal toolbox is managed using the CMake utility (http://www.cmake.org). It is used to ensure the portability on the different operating systems (CMake generates native makefiles). For the same reason, in 2013, we have integrated the management of the tests of the Signal batch compiler under CMake (using CTest).

../../../../projets/espresso/IMG/PolychronyToolsetMapSSME.png

Figure 7. The Polychrony toolset high-level architecture

The Polychrony toolset can be freely downloaded on the following web sites:

- The Polychrony toolset public web site: http://www.irisa.fr/espresso/Polychrony/index.php. This site, intended for users and for developers, contains downloadable executable and source versions of the software for differents platforms, user documentation, examples, libraries, scientific publications and implementation documentation. In particular, this is the site for the new open-source distribution of Polychrony.
- The Inria GForge: https://gforge.inria.fr. This site, intended for internal developers, contains the whole sources of the environment and their documentation.

In 2012, during the OPEES project, we have integrated Polychrony on the Polarsys Experimental Eclipse platform, a new industry collaboration to build open source tools for safety-critical software development. In 2013, the integration to the actual Eclipse Polarsys platform (http://www.polarsys.org) has been started. The proposal project is available at http://www.eclipse.org/proposals/polarsys.polychrony. The creation of a new Polarsys project (as defined in the Eclipse Development Process) is decomposed into several steps (before the first release): project proposal, community review, trademark review, creation review, provisioning, initial contribution. For trademark reasons, Polychrony is called POP as Polarsys Project (POP, a polychronous modeling environment on Polarsys). At the end of 2013, we are at the "provisioning" step. It will be provisioned as soon as the employer consent (Inria, CNRS) form will be signed.

The Polychrony toolset currently runs on Linux, MacOS and Windows systems.

Dassault Systèmes, supplies a commercial implementation of Polychrony, called RT-Builder, used for industrial scale projects.

As part of its open-source release, the Polychrony toolset not only comprises source code libraries but also an important corpus of structured documentation, whose aim is not only to document each functionality and service, but also to help a potential developer to package a subset of these functionalities and services, and adapt them to developing a new application-specific tool: a new language front-end, a new back-end compiler. This multi-scale, multi-purpose documentation aims to provide different views of the software, from a high-level structural view to low-level descriptions of basic modules. It supports a distribution of the software "by apartment" (a functionality or a set of functionalities) intended for developers who would only be interested by part of the services of the toolset.

A high-level architectural view of the Polychrony toolset is given in Figure 7 .

5.2. The Eclipse interface

Participant: Loïc Besnard.

Meta-modeling, Eclipse, Ecore, Signal, Model transformation

We have developed a meta-model and interactive editor of Polychrony in Eclipse. Signal-Meta is the meta-model of the Signal language implemented with Eclipse/Ecore. It describes all syntactic elements specified in [33]: all Signal operators (e.g. arithmetic, clock synchronization), model (e.g. process frame, module), and construction (e.g. iteration, type declaration).

The meta-model primarily aims at making the language and services of the Polychrony environment available to inter-operation and composition with other components (e.g. AADL, Simulink, GeneAuto) within an Eclipse-based development toolchain. Polychrony now comprises the capability to directly import and export Ecore models instead of textual Signal programs, in order to facilitate interaction between components within such a toolchain.

It also provides a graphical modeling framework allowing to design applications using a component-based approach. Application architectures can be easily described by just selecting components via drag and drop, creating some connections between them and specifying their parameters as component attributes. Using the modeling facilities provided with the Topcased framework, we have created a graphical environment for Polychrony called SME (Signal-Meta under Eclipse). To highlight the different parts of the modeling in Signal, we split the modeling of a Signal process in three diagrams: one to model the interface of the process, one to model the computation (or dataflow) part, and one to model all explicit clock relations and dependences. The SME environment is available through the ESPRESSO update site [24]. Note that a new meta-model of Signal, called SSME (Syntactic Signal-Meta under Eclipse), closer to the Signal abstract syntax, has been defined and integrated in the Polychrony toolset.

It should be noted that the Eclipse Foudation does not host code under GPL license. So, the Signal toolbox useful to compile Signal code from Eclipse is hosted on our web server. For this reason, the building of the Signal toolbox, previously managed under Eclipse, has now been exported. The interface of the Signal toolbox for Eclipse is now managed using the CMake tool like the Signal toolbox and the Signal GUI.

5.3. Integrated Modular Avionics design using Polychrony

Participants: Loïc Besnard, Thierry Gautier, Paul Le Guernic, Jean-Pierre Talpin.

The Apex interface, defined in the ARINC standard [25], provides an avionics application software with the set of basic services to access the operating-system and other system-specific resources. Its definition relies on the Integrated Modular Avionics approach (IMA [26]). A main feature in an IMA architecture is that several avionics applications (possibly with different critical levels) can be hosted on a single, shared computer system. Of course, a critical issue is to ensure safe allocation of shared computer resources in order to prevent fault propagations from one hosted application to another. This is addressed through a functional partitioning of the applications with respect to available time and memory resources. The allocation unit that results from this decomposition is the *partition*.

A partition is composed of *processes* which represent the executive units (an ARINC partition/process is akin to a Unix process/task). When a partition is activated, its owned processes run concurrently to perform the functions associated with the partition. The process scheduling policy is priority preemptive. Each partition is allocated to a processor for a fixed time window within a major time frame maintained by the operating system. Suitable mechanisms and devices are provided for communication and synchronization between processes (e.g. *buffer*, *event*, *semaphore*) and partitions (e.g. *ports* and *channels*). The specification of the ARINC 651-653 services in Signal [5] is now part of the Polychrony distribution and offers a complete implementation of the Apex communication, synchronization, process management and partitioning services. Its Signal implementation consists of a library of generic, parameterizable Signal modules.

5.4. Safety-Critical Java Level 1 Code generation from Dataflow Graph Specifications

Participants: Adnan Bouakaz, Jean-Pierre Talpin.

We have proposed a dataflow design model [19] of SCJ/L1 applications [43] in which handlers (periodic and aperiodic actors) communicate only through lock-free channels. Hence, each mission is modeled as a dataflow graph. The presented dataflow design model comes with a development tool integrated in the Eclipse IDE for easing the development of SCJ/L1 applications and enforcing the restrictions imposed by the design model. It consists of a GMF editor where applications are designed graphically and timing and buffering parameters can be synthesized. Indeed, abstract affine scheduling is first applied on the dataflow subgraph, that consists only of periodic actors, to compute timeless scheduling constraints (e.g. relation between the speeds of two actors) and buffering parameters. Then, symbolic fixed-priority schedulability analysis (i.e., synthesis of timing and scheduling parameters of actors) considers both periodic and aperiodic actors.

Through a model-to-text transformation, using Acceleo, the SCJ code for missions, interfaces of handlers, and the mission sequencer is automatically generated in addition to the annotations needed by the memory checker. Channels are implemented as cyclic arrays or cyclical asynchronous buffers; and a fixed amount of memory is hence reused to store the infinite streams of tokens. The user must provide the SCJ code of all the `handleAsyncEvent()` methods. We have integrated the SCJ memory checker [52] in our tool so that potential dangling pointers can be highlighted at compile-time. To enhance functional determinism, we would like to develop an ownership type system to ensure that actors are strongly isolated and communicate only through buffers.

FORMES Team

5. Software and Platforms

5.1. CoLoR

Participants: Frédéric Blanqui, Kim-Quyen Ly.

CoLoR is a Coq library on rewriting theory and termination of more than 83,000 lines of code [4]. It provides definitions and theorems for:

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

CoLoR is distributed under the CeCILL license. It is currently developed by Frédéric Blanqui and Kim-Quyen Ly, but various people participated to its development since 2006.

5.2. HOT

Participant: Frédéric Blanqui.

HOT is an automated termination prover for higher-order rewrite systems based on the notion of computability closure and size annotation [24]. It won the 2012 competition in the category “higher-order rewriting union beta”. The sources are not public.

5.3. Moca

Participant: Frédéric Blanqui.

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

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

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

Moca supports two kinds of relations:

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

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

Moca is distributed under QPL. It is developed by Frédéric Blanqui, Pierre Weis (EPI Pomdapi) and Richard Bonichon (CEA).

5.4. Rainbow

Participants: Frédéric Blanqui, Kim-Quyen Ly.

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

Rainbow is distributed under the CeCILL license. It is currently developed by Frédéric Blanqui and Kim-Quyen Ly. See the web site for more information.

5.5. CoqMT

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

The proof-assistant Coq is based on a complex type theory, which resulted from various extensions of the Calculus of Constructions studied independently from each other. With the collaboration of Bruno Barras, we decided to address the challenge of proving the real type theory underlying Coq, and even, indeed, of its recent extension CoqMT developed in FORMES by Pierre-Yves Strub. To this end, we have studied formally the theory CoqMTU, which extends the pure Calculus of Constructions by inductive types, a predicative hierarchy of universes, and a decidable theory T for some first-order inductive types. Recently, we were able to announce the complete certification of CoqMTU in Coq augmented with appropriate intuitionistic set-theoretic axioms in order to fight Gödel's incompleteness theorem~[16]. As a consequence, Coq and CoqMTU are the first proof assistants, of which consistency (relative to intuitionistic set theory IZF augmented with the afore-mentioned axioms) is formally entirely proved (in Coq). While previous formal proofs for Coq and other proof assistants all assumed strong normalization, the present one *proves* strong normalization thanks to the new notion of *strongly-normalizing* model introduced by Bruno Barras. While consistency is done already, decidability of type-checking in CoqMTU remains to be done. This is a straightforward consequence for Coq, but a non-trivial task for CoqMTU because of the interaction between inductive types and the first-order theory T. It should however be done by the summer of 2014. We consider this work as a major scientific achievement of the team.

5.6. SimSoC

Participants: Vania Joloboff [correspondant], Antoine Rouquette, Shenpeng Wang.

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

SimSoC is based on SystemC standard and uses Transaction Level Modeling for interactions between the simulation models. The current version is based on the open source libraries from the OSCI Consortium: SystemC version 2.3 and TLM 2.0.1 [39], [21]. Hardware components are modeled as TLM models, and since TLM is itself based on SystemC, the simulation is driven by the SystemC kernel. We use standard, unmodified, SystemC, hence the simulator has a single simulation loop.

The third open source version of **SimSoC**, release 0.8.0, has been released in September 2013. It contains a full simulator for ARM (V5 and V6) and PowerPC both running at an average speed of about 100 Millions instructions per second in, and a deprecated simulator for the MIPS architecture. **SimSoC** is distributed under LGPL on Inria Gforge web site.

5.7. SimSoC-Cert

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

Simulators such as **SimSoC** make it possible to reduce development time and development cost, allowing for the software engineers to run fast iterative cycles without requiring a hardware development board. Then a critical issue is: *does the simulator actually simulate the real hardware?*

Considering only one module in **SimSoC**, namely the ARM simulator, it somehow encodes the 1138 pages of the ARM reference manual in C++. The whole simulator, which simulates ARM and PowerPC architecture, includes about 60,000 lines of manually coded C++ code. Then, mistakes in the hand written code are unavoidable and difficult to find due to the complexity. From the experiments performed on **SimSoC**, bugs bringing a wrong behavior were observed from time to time but it was hard to reveal where they were. Using intensive tests can cover most of the instructions, but still left some untested rare cases of instructions, which lead to potential problems.

Therefore, a better approach is required to gain confidence in the correctness of the simulator. Our proposal has been to certify the ARM CPU simulator from **SimSoC** using formal methods. We aimed at proving a significant part of the correctness of **SimSoC** in order to support the claim that the implementation of the simulator and the real hardware system will exhibit the same behavior.

In addition, we developed tools that can automatically generate in various C the core simulator, including the decoding functions and the instruction set of the ARMv6 architecture manual [18] (implemented by the ARM11 processor family). The input of SimSoC-Cert is the ARMv6 architecture manual itself.

In order to get the required flexibility and accuracy, we wanted to experiment a direct approach based on a general proof assistant such as Coq. Fortunately, an operational semantics formalized in Coq of a large enough subset of the C language is available from the **CompCert** project. We then decided to base our correctness proofs on this technology. Up to our knowledge, this is the first development of formal correctness proofs based on operational semantics, at least at this scale.

Based on this, we first developed *simlight* (8000 generated lines of C, plus 1500 hand-written lines of C), a simulator for ARMv6 programs using no peripheral and no coprocessor. Next, we developed *simlight2*, a fast ARMv6 simulator integrated inside a SystemC/TLM module, now part of SimSoC v0.8.

We can also generate similar programs for SH4 [20] but this is still experimental (work done by Frédéric Tuong in 2011).

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

GALLIUM Project-Team

5. Software and Platforms

5.1. OCaml

Participants: Damien Doligez [correspondant], Alain Frisch [LexiFi], Jacques Garrigue [Nagoya University], Fabrice Le Fessant, Xavier Leroy, Luc Maranget.

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

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

5.2. CompCert C

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

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

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

5.3. The diy tool suite

Participants: Luc Maranget [correspondant], Jade Alglave [University College London], Susmit Sarkar [University of St Andrews], Peter Sewell [University of Cambridge].

The **diy** suite provides a set of tools for testing shared memory models: the **litmus** tool for running tests on hardware, various generators for producing tests from concise specifications, and **herd**, a memory model simulator. Tests are small programs written in x86, Power or ARM assembler that can thus be generated from concise specification, run on hardware, or simulated on top of memory models. Test results can be handled and compared using additional tools.

The tool suite and a comprehensive documentation are available from http://diy.inria.fr/.

5.4. Zenon

Participant: Damien Doligez.

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

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

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

5.5. JoCaml

Participant: Luc Maranget.

JoCaml is an experimental extension of OCaml that adds support for concurrent and distributed programming, following the programming model of the join-calculus.

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

GEOMETRICA Project-Team

5. Software and Platforms

5.1. CGAL, the Computational Geometry Algorithms Library

Participants: Jean-Daniel Boissonnat, Olivier Devillers, Monique Teillaud, Mariette Yvinec.

With the collaboration of Pierre Alliez, Hervé Brönnimann, Manuel Caroli, Pedro Machado Manhães de Castro, Frédéric Cazals, Frank Da, Christophe Delage, Andreas Fabri, Julia Flötotto, Philippe Guigue, Michael Hemmer, Samuel Hornus, Clément Jamin, Menelaos Karavelas, Sébastien Loriot, Abdelkrim Mebarki, Naceur Meskini, Andreas Meyer, Sylvain Pion, Marc Pouget, François Rebufat, Laurent Rineau, Laurent Saboret, Stéphane Tayeb, Jane Tournois, Radu Ursu, and Camille Wormser http://www.cgal.org

CGAL is a C++ library of geometric algorithms and data structures. Its development has been initially funded and further supported by several European projects (CGAL, GALIA, ECG, ACS, AIM@SHAPE) since 1996. The long term partners of the project are research teams from the following institutes: Inria Sophia Antipolis - Méditerranée, Max-Planck Institut Saarbrücken, ETH Zürich, Tel Aviv University, together with several others. In 2003, CGAL became an Open Source project (under the LGPL and QPL licenses).

The transfer and diffusion of CGAL in industry is achieved through the company GEOMETRY FACTORY (http://www.geometryfactory.com). GEOMETRY FACTORY is a *Born of Inria* company, founded by Andreas Fabri in January 2003. The goal of this company is to pursue the development of the library and to offer services in connection with CGAL (maintenance, support, teaching, advice). GEOMETRY FACTORY is a link between the researchers from the computational geometry community and the industrial users.

The aim of the CGAL project is to create a platform for geometric computing supporting usage in both industry and academia. The main design goals are genericity, numerical robustness, efficiency and ease of use. These goals are enforced by a review of all submissions managed by an editorial board. As the focus is on fundamental geometric algorithms and data structures, the target application domains are numerous: from geological modeling to medical images, from antenna placement to geographic information systems, etc.

The CGAL library consists of a kernel, a list of algorithmic packages, and a support library. The kernel is made of classes that represent elementary geometric objects (points, vectors, lines, segments, planes, simplices, isothetic boxes, circles, spheres, circular arcs...), as well as affine transformations and a number of predicates and geometric constructions over these objects. These classes exist in dimensions 2 and 3 (static dimension) and d (dynamic dimension). Using the template mechanism, each class can be instantiated following several representation modes: one can choose between Cartesian or homogeneous coordinates, use different number types to store the coordinates, and use reference counting or not. The kernel also provides some robustness features using some specifically-devised arithmetic (interval arithmetic, multi-precision arithmetic, static filters...).

A number of packages provide geometric data structures as well as algorithms. The data structures are polygons, polyhedra, triangulations, planar maps, arrangements and various search structures (segment trees, d-dimensional trees...). Algorithms are provided to compute convex hulls, Voronoi diagrams, Boolean operations on polygons, solve certain optimization problems (linear, quadratic, generalized of linear type). Through class and function templates, these algorithms can be used either with the kernel objects or with user-defined geometric classes provided they match a documented interface.

Finally, the support library provides random generators, and interfacing code with other libraries, tools, or file formats (ASCII files, QT or LEDA Windows, OpenGL, Open Inventor, Postscript, Geomview...). Partial interfaces with Python, SCILAB and the Ipe drawing editor are now also available.

GEOMETRICA is particularly involved in general maintenance, in the arithmetic issues that arise in the treatment of robustness issues, in the kernel, in triangulation packages and their close applications such as alpha shapes, in mesh generation and related packages. Two researchers of GEOMETRICA are members of the CGAL Editorial Board, whose main responsibilities are the control of the quality of CGAL, making decisions about technical matters, coordinating communication and promotion of CGAL.

CGAL is about 700,000 lines of code and supports various platforms: GCC (Linux, Mac OS X, Cygwin...), Visual C++ (Windows), Intel C++. A new version of CGAL is released twice a year, and it is downloaded about 10000 times a year. Moreover, CGAL is directly available as packages for the Debian, Ubuntu and Fedora Linux distributions.

More numbers about CGAL: there are now 12 editors in the editorial board, with approximately 20 additional developers. The user discussion mailing-list has more than 1000 subscribers with a relatively high traffic of 5-10 mails a day. The announcement mailing-list has more than 3000 subscribers.

GRACE Project-Team

5. Software and Platforms

5.1. ECPP

François Morain has been continually improving his primality proving algorithm, ECPP, originally developed in the early 1990s. Proving the primality of a 512-bit number requires less than a second on an average PC. Morain's personal record is around 25000 decimal digits, using the FASTECPP variant that he started developing in 2003. The code is written in C, and based on publicly available packages (GMP, MPFR, MPC, MPFRCX).

5.2. SEA

Together with E. Schost and L. De Feo, François Morain has developed a new implementation of the SEA algorithm for computing the cardinality of elliptic curves over large prime and binary finite fields. This program is a `gforge` project, based on the NTL library. The large prime case is relevant to cryptographic needs; the binary case, while not useful in contemporary cryptography, is a good testbed for De Feo's FAAST package.

5.3. TIFA

TIFA (Tools for Integer FActorization), initially developed in 2006, has been continuously improved during the last few years. TIFA includes a base library written in C99 using the GMP library, stand-alone factorization programs, and a basic benchmarking framework. Available online at http://www.lix.polytechnique.fr/Labo/Jerome.Milan/tifa/tifa.xhtml, TIFA is distributed under the Lesser General Public License (version 2.1 or later).

5.4. Quintix

Guillaume Quintin's Quintix library implements efficient arithmetic in Galois rings and their unramified extensions, the root-finding algorithms presented in [7], basic functions for the manipulation of Reed–Solomon codes, and the complete Sudan list-decoding algorithm. Part of the Mathemagix computer algebra system (http://www.mathemagix.org/), the source code is distributed under the General Public License (version 2 or higher).

5.5. finitefieldz

Within the Mathemagix CAS (http://www.mathemagix.org/), Guillaume Quintin wrote the finitefieldz package, which provides arithmetic for finite fields and towers of finite fields, as well as univariate polynomial root finding and factorization over finite fields.

5.6. Decoding

DECODING is a standalone C library licensed under the GPLv2. Its primary goal is to implement Guruswami–Sudan list decoding-related algorithms as efficiently as possible. Its secondary goal is to give an efficient tool for the implementation and benchmarking of general decoding algorithms. As of 2012/12/13, DECODING provides a working list decoding algorithm, but there is no unique decoding algorithm (though this can be emulated by list-decoding up to half the minimum distance). The library is being still under development, and more algorithms will be added. DECODING was presented at the 2012 International Symposium on Symbolic and Algebraic Computation.

Hycomes Team

4. Software and Platforms

4.1. Mica: A Modal Interface Compositional Analysis Toolbox

Participant: Benoît Caillaud.

http://www.irisa.fr/s4/tools/mica/

Mica is an Ocaml library developed by Benoît Caillaud implementing the Modal Interface algebra published in [5], [4]. The purpose of Modal Interfaces is to provide a formal support to contract based design methods in the field of system engineering. Modal Interfaces enable compositional reasoning methods on I/O reactive systems.

In Mica, systems and interfaces are represented by extension. However, a careful design of the state and event heap enables the definition, composition and analysis of reasonably large systems and interfaces. The heap stores states and events in a hash table and ensures structural equality (there is no duplication). Therefore complex data-structures for states and events induce a very low overhead, as checking equality is done in constant time.

Thanks to the Inter module and the mica interactive environment, users can define complex systems and interfaces using Ocaml syntax. It is even possible to define parameterized components as Ocaml functions.

Mica is available as an open-source distribution, under the CeCILL-C Free Software License Agreement (http://www.cecill.info/licences/Licence_CeCILL-C_V1-en.html).

4.2. Flipflop: A Test and Flip Net Synthesis Tool for Maintenance and Surgical Process Mining

Participant: Benoît Caillaud.

http://tinyurl.com/oql6f3y

Flipflop is a Test and Flip net synthesis tool implementing a linear algebraic polynomial time algorithm. Computations are done in the $\mathbb{Z}/2\mathbb{Z}$ ring. Test and Flip nets extend Elementary Net Systems by allowing test to zero, test to one and flip arcs. The effect of flip arcs is to complement the marking of the place. While the net synthesis problem has been proved to be NP hard for Elementary Net Systems, thanks to flip arcs, the synthesis of Test and Flip nets can be done in polynomial time. Test and flip nets have the required expressivity to give concise and accurate representations of surgical processes (models of types of surgical operations). Test and Flip nets can express causality and conflict relations. The tool takes as input either standard XES log files (a standard XML file format for process mining tools) or a specific XML file format for surgical applications. The output is a Test and Flip net, solution of the following synthesis problem: Given a finite input language (log file), compute a net, which language is the least language in the class of Test and Flip net languages, containing the input language.

This software has been designed in the context of the S3PM project (see Section 6.1).

INDES Project-Team

5. Software and Platforms

5.1. Introduction

Most INDES software packages, even the older stable ones that are not described in the following sections are freely available on the Web. In particular, some are available directly from the Inria Web site:

http://www.inria.fr/valorisation/logiciels/langages.fr.html

Most other software packages can be downloaded from the INDES Web site:

http://www-sop.inria.fr/teams/indes

5.2. Functional programming

Participants: Cyprien Nicolas, Bernard Serpette, Manuel Serrano [correspondant].

5.2.1. The Bigloo compiler

The programming environment for the Bigloo compiler [7] is available on the Inria Web site at the following URL: http://www-sop.inria.fr/teams/indes/fp/Bigloo. The distribution contains an optimizing compiler that delivers native code, JVM bytecode, and .NET CLR bytecode. It contains a debugger, a profiler, and various Bigloo development tools. The distribution also contains several user libraries that enable the implementation of realistic applications.

BIGLOO was initially designed for implementing compact stand-alone applications under Unix. Nowadays, it runs harmoniously under Linux and MacOSX. The effort initiated in 2002 for porting it to Microsoft Windows is pursued by external contributors. In addition to the native back-ends, the BIGLOO JVM back-end has enabled a new set of applications: Web services, Web browser plug-ins, cross platform development, etc. The new BIGLOO .NET CLR back-end that is fully operational since release 2.6e enables a smooth integration of Bigloo programs under the Microsoft .NET environment.

5.3. Language-based Security

Participants: Tamara Rezk [correspondant], José Santos.

5.3.1. IFJS compiler

The IFJS compiler is applied to JavaScript code. The compiler generates JavaScript code instrumented with checks to secure code. The compiler takes into account special features of JavaScript such as implicit type coercions and programs that actively try to bypass the inlined enforcement mechanisms. The compiler guarantees that third-party programs cannot (1) access the compiler internal state by randomizing the names of the resources through which it is accessed and (2) change the behaviour of native functions that are used by the enforcement mechanisms inlined in the compiled code.

The compiler is written in JavaScript and can be found at http://www-sop.inria.fr/indes/ifJS.

5.4. Web programming

Participants: Gérard Berry, Cyprien Nicolas, Manuel Serrano [correspondant].

5.4.1. The HOP web programming environment

HOP is a higher-order language designed for programming interactive web applications such as web agendas, web galleries, music players, etc. It exposes a programming model based on two computation levels. The first one is in charge of executing the logic of an application while the second one is in charge of executing the graphical user interface. HOP separates the logic and the graphical user interface but it packages them together and it supports strong collaboration between the two engines. The two execution flows communicate through function calls and event loops. Both ends can initiate communications.

The HOP programming environment consists in a web *broker* that intuitively combines in a single architecture a web server and a web proxy. The broker embeds a HOP interpreter for executing server-side code and a HOP client-side compiler for generating the code that will get executed by the client.

An important effort is devoted to providing HOP with a realistic and efficient implementation. The HOP implementation is *validated* against web applications that are used on a daily-basis. In particular, we have developed HOP applications for authoring and projecting slides, editing calendars, reading RSS streams, or managing blogs.

HOP has won the software *open source contest* organized by the ACM Multimedia Conference 2007. It is released under the GPL license. It is available at http://hop.inria.fr.

5.5. Old software

5.5.1. Camloo

Camloo is a caml-light to bigloo compiler, which was developed a few years ago to target bigloo 1.6c. New major releases 0.4.x of camloo have been done to support bigloo 3.4 and bigloo 3.5. Camloo make it possible for the user to develop seamlessly a multi-language project, where some files are written in caml-light, in C, and in bigloo. Unlike the previous versions of camloo, 0.4.x versions do not need a modified bigloo compiler to obtain good performance. Currently, the only supported backend for camloo is bigloo/C. We are currently rewriting the runtime of camloo in bigloo to get more portability and to be able to use HOP and camloo together.

5.5.2. Skribe

SKRIBE is a functional programming language designed for authoring documents, such as Web pages or technical reports. It is built on top of the SCHEME programming language. Its concrete syntax is simple and looks familiar to anyone used to markup languages. Authoring a document with SKRIBE is as simple as with HTML or LaTeX. It is even possible to use it without noticing that it is a programming language because of the conciseness of its original syntax: the ratio *tag/text* is smaller than with the other markup systems we have tested.

Executing a SKRIBE program with a SKRIBE evaluator produces a target document. It can be HTML files for Web browsers, a LaTeX file for high-quality printed documents, or a set of *info* pages for on-line documentation.

5.5.3. Scheme2JS

Scm2JS is a Scheme to JavaScript compiler distributed under the GPL license. Even though much effort has been spent on being as close as possible to R5RS, we concentrated mainly on efficiency and interoperability. Usually Scm2JS produces JavaScript code that is comparable (in speed) to hand-written code. In order to achieve this performance, Scm2JS is not completely R5RS compliant. In particular it lacks exact numbers.

Interoperability with existing JavaScript code is ensured by a JavaScript-like dot-notation to access JavaScript objects and by a flexible symbol-resolution implementation.

Scm2JS is used on a daily basis within HOP, where it generates the code which is sent to the clients (web-browsers). Scm2JS can be found at http://www-sop.inria.fr/indes/scheme2js.

5.5.4. The FunLoft language

FunLoft (described in http://www-sop.inria.fr/teams/indes/rp/FunLoft) is a programming language in which the focus is put on safety and multicore.

FunLoft is built on the model of FairThreads which makes concurrent programming simpler than usual preemptive-based techniques by providing a framework with a clear and sound semantics. FunLoft is designed with the following objectives:

- provide a safe language, in which, for example, data-races are impossible.
- control the use of resources (CPU and memory), for example, memory leaks cannot occur in FunLoft programs, which always react in finite time.
- have an efficient implementation which can deal with large numbers of concurrent components.
- benefit from the real parallelism offered by multicore machines.

A first experimental version of the compiler is available on the Reactive Programming site http://www-sop.inria.fr/teams/indes/rp. Several benchmarks are given, including cellular automata and simulation of colliding particles.

5.5.5. CFlow

The prototype compiler “CFlow” takes as input code annotated with information flow security labels for integrity and confidentiality and compiles to F# code that implements cryptography and protocols that satisfy the given security specification.

Cflow has been coded in F#, developed mainly on Linux using mono (as a substitute to .NET), and partially tested under Windows (relying on .NET and Cygwin). The code is distributed under the terms of the CeCILL-B license.

5.5.6. FHE type-checker

We have developed a type checker for programs that feature modern cryptographic primitives such as fully homomorphic encryption. The type checker is thought as an extension of the “CFlow” compiler developed last year on the same project. It is implemented in F#. The code is distributed under the terms of the CeCILL-B license.

5.5.7. Mashic compiler

The Mashic compiler is applied to mashups with untrusted scripts. The compiler generates mashups with sandboxed scripts, secured by the same origin policy of the browsers. The compiler is written in Bigloo and can be found at http://www-sop.inria.fr/indes/mashic/.

LFANT Project-Team

5. Software and Platforms

5.1. Pari/Gp

Participants: Karim Belabas [correspondent], Bill Allombert, Henri Cohen, Andreas Enge.

http://pari.math.u-bordeaux.fr/

PARI/GP is a widely used computer algebra system designed for fast computations in number theory (factorisation, algebraic number theory, elliptic curves, ...), but it also contains a large number of other useful functions to compute with mathematical entities such as matrices, polynomials, power series, algebraic numbers, etc., and many transcendental functions.

- PARI is a C library, allowing fast computations.
- GP is an easy-to-use interactive shell giving access to the PARI functions.
- `gp2c`, the GP-to-C compiler, combines the best of both worlds by compiling GP scripts to the C language and transparently loading the resulting functions into GP; scripts compiled by `gp2c` will typically run three to four times faster.
- Version of PARI/GP: 2.5.5
- Version of `gp2c`: 0.0.8
- License: GPL v2+
- Programming language: C

5.2. GNU MPC

Participants: Andreas Enge [correspondent], Mickaël Gastineau [CNRS], Philippe Théveny [INRIA project-team ARIC], Paul Zimmermann [INRIA project-team CARAMEL].

http://mpc.multiprecision.org/.

GNUMPC is a C library for the arithmetic of complex numbers with arbitrarily high precision and correct rounding of the result. It is built upon and follows the same principles as GNUMPFR.

It is a prerequisite for the GNU compiler collection GCC since version 4.5, where it is used in the C and Fortran front ends for constant folding, the evaluation of constant mathematical expressions during the compilation of a program. Since 2011, it is an official GNU project.

2012 has seen the first release of the major version 1.0.

- Version: 1.0.1 *Fagus silvatica*
- License: LGPL v3+
- ACM: G.1.0 (Multiple precision arithmetic)
- AMS: 30.04 Explicit machine computation and programs
- APP: Dépôt APP le 2003-02-05 sous le numéro IDDN FR 001 060029 000 R P 2003 000 10000
- Programming language: C

5.3. MPFRCX

Participant: Andreas Enge.

http://mpfrcx.multiprecision.org/

MPFRCX is a library for the arithmetic of univariate polynomials over arbitrary precision real (MPFR) or complex (MPC) numbers, without control on the rounding. For the time being, only the few functions needed to implement the floating point approach to complex multiplication are implemented. On the other hand, these comprise asymptotically fast multiplication routines such as Toom-Cook and the FFT.

- Version: 0.4.2 *Cassava*
- License: LGPL v2.1+
- Programming language: C

5.4. CM

Participant: Andreas Enge.

http://cm.multiprecision.org/

The CM software implements the construction of ring class fields of imaginary quadratic number fields and of elliptic curves with complex multiplication via floating point approximations. It consists of libraries that can be called from within a C program and of executable command line applications. For the implemented algorithms, see [8].

- Version: 0.2 *Blindhühnchen*
- License: GPL v2+
- Programming language: C

5.5. AVIsogenies

Participants: Damien Robert [correspondent], Gaëtan Bisson, Romain Cosset [INRIA project-team CARAMEL].

http://avisogenies.gforge.inria.fr/.

AVISOGENIES (Abelian Varieties and Isogenies) is a MAGMA package for working with abelian varieties, with a particular emphasis on explicit isogeny computation.

Its prominent feature is the computation of (ℓ, ℓ)-isogenies between Jacobian varieties of genus-two hyperelliptic curves over finite fields of characteristic coprime to ℓ; practical runs have used values of ℓ in the hundreds.

It can also be used to compute endomorphism rings of abelian surfaces, and find complete addition laws on them.

- Version: 0.6
- License: LGPL v2.1+
- Programming language: Magma

5.6. APIP

Participant: Jérôme Milan.

http://www.lix.polytechnique.fr/~milanj/apip/apip.xhtml

APIP, Another Pairing Implementation in PARI, is a library for computing standard and optimised variants of most cryptographic pairings.

The following pairings are available: Weil, Tate, ate and twisted ate, optimised versions (à la Vercauteren–Hess) of ate and twisted ate for selected curve families.

The following methods to compute the Miller part are implemented: standard Miller double-and-add method, standard Miller using a non-adjacent form, Boxall et al. version, Boxall et al. version using a non-adjacent form.

The final exponentiation part can be computed using one of the following variants: naive exponentiation, interleaved method, Avanzi-Mihailescu's method, Kato et al.'s method, Scott et al.'s method.

- Version: 2012-10-17
- License: GPL v2+
- Programming language: C with libpari

5.7. CMH

Participants: Andreas Enge, Emmanuel Thomé [INRIA project-team CARAMEL].

http://cmh.gforge.inria.fr/

CMH computes Igusa class polynomials, parameterising two-dimensional abelian varieties (or, equivalently, Jacobians of hyperelliptic curves of genus 2) with given complex multiplication.

- Version: development snapshot
- License: GPL v3+
- Programming language: C

5.8. Cubic

Participant: Karim Belabas.

http://www.math.u-bordeaux1.fr/~belabas/research/software/cubic-1.2.tgz

CUBIC is a stand-alone program that prints out generating equations for cubic fields of either signature and bounded discriminant. It depends on the PARI library. The algorithm has quasi-linear time complexity in the size of the output.

- Version: 1.2
- License: GPL v2+
- Programming language: C

5.9. Euclid

Participant: Pierre Lezowski.

http://www.math.u-bordeaux1.fr/~plezowsk/euclid/index.php.

Euclid is a program to compute the Euclidean minimum of a number field. It is the practical implementation of the algorithm described in [41]. Some corresponding tables built with the algorithm are also available. Euclid is a stand-alone program depending on the PARI library.

- Version: 1.0
- License: LGPL v2+
- Programming language: C

5.10. KleinianGroups

Participant: Aurel Page.

http://www.normalesup.org/~page/Recherche/Logiciels/logiciels.html

KLEINIANGROUPS is a Magma package that computes fundamental domains of arithmetic Kleinian groups.

- Version: 1.0
- License: GPL v3+
- Programming language: Magma

MARELLE Project-Team

5. Software and Platforms

5.1. Tralics

Participant: José Grimm [correspondant].

Tralics is a Latex-to-XML translator available at http://www-sop.inria.fr/marelle/tralics. Version 2.15 has been released in 2012.

5.2. Semantics

Participant: Yves Bertot [correspondant].

This is a library for the Coq system, where the description of a toy programming language is presented. The value of this library is that it can be re-used in classrooms to teach programming language semantics or the Coq system. The topics covered include introductory notions to domain theory, pre and post-conditions, abstract interpretation, and the proofs of consistency between all these point of views on the same programming language. Standalone tools for the object programming language can be derived from this development.
See also the web page http://coq.inria.fr/pylons/pylons/contribs/view/Semantics/v8.4.

- ACM: F3.2 F4.1
- AMS: 68N30
- Programming language: Coq

5.3. Easycrypt

Participants: Gilles Barthe [IMDEA Software Institute], François Dupressoir [IMDEA Software Institute], Benjamin Grégoire [correspondant], César Kunz [IMDEA Software Institute], Benedikt Schmid [IMDEA Software Institute], Pierre-Yves Strub [IMDEA Software Institute].

EasyCrypt is a toolset for reasoning about relational properties of probabilistic computations with adversarial code. Its main application is the construction and verification of game-based cryptographic proofs. EasyCrypt can also be used for reasoning about differential privacy.

ZooCrypt is an automated tool for analyzing the security of padding-based public-key encryption schemes (i.e. schemes built from trapdoor permutations and hash functions). ZooCrypt includes an experimental mechanism to generate EasyCrypt proofs of security of analyzed schemes.

5.4. CoqEAL

Participants: Maxime Dénès, Yves Bertot [correspondant].

CoqEAL is a library of certified algorithms for linear algebra to be used in Coq. It provides a collection of algorithms to compute efficiently on matrices and polynomials. These algorithms are designed to run efficiently directly in the Coq system and take the best advantage of the internal execution capabilities of the this system (virtual machine execution of native code execution after compilation).

5.5. CoqApprox

Participants: Nicolas Brisebarre [CNRS], Mioara Joldes, Érik Martin-Dorel, Micaela Mayero [Iut de Villetaneuse], Jean-Michel Muller, Ioana Paşca [Iut de Nimes], Laurence Rideau, Laurent Théry.

We develop a formalization of rigorous polynomial approximation using Taylor models inside the Coq proof assistant, with a special focus on genericity and efficiency for the computations.

5.6. CoqHensel

Participants: Érik Martin-Dorel, Laurent Théry, Micaela Mayero [Iut de Villetaneuse], Guillaume Hanrot [ENS Lyon].

The CoqHensel library provides a Coq formalization of Hensel's lemma for both univariate and bivariate cases, with some effective and modular certificate checkers for the univariate small integral roots problem, the bivariate small integral roots problem, as well as the integer small value problem (ISValP), with the ultimate goal to provide a fully formally verified chain for solving the Table Maker's Dilemma.

MEXICO Project-Team

5. Software and Platforms

5.1. Software and Platform

5.1.1. Software

5.1.1.1. libalf: the Automata Learning Framework

Participant: Benedikt Bollig [correspondant].

`libalf` is a comprehensive, open-source library for learning finite-state automata covering various well-known learning techniques (such as, Angluin s L^*, Biermann, and RPNI, as well as a novel learning algorithm for NFA. libalf is highly flexible and allows for facilely interchanging learning algorithms and combining domain-specific features in a plug-and-play fashion. Its modular design and its implementation in C++ make it a flexible platform for adding and engineering further, efficient learning algorithms for new target models (e.g., Büchi automata).

Details on `libalf` can be found at http://libalf.informatik.rwth-aachen.de/

5.1.1.2. Mole/Cunf: unfolders for Petri Nets

Participants: Stefan Schwoon [correspondant], César Rodríguez.

Mole computes, given a safe Petri net, a finite prefix of its unfolding. It is designed to be compatible with other tools, such as PEP and the Model-Checking Kit, which are using the resulting unfolding for reachability checking and other analyses. The tool Mole arose out of earlier work on Petri nets. Details on Mole can be found at http://www.lsv.ens-cachan.fr/~schwoon/tools/mole/. Mole served as an experimentation platform for several of our papers this year, notably [38] and [46].

In the context of MExICo, we have created a new tool called Cunf [47], which is able to handle contextual nets, i.e. Petri nets with read arcs [12].While in principle every contextual net can be transformed into an equivalent Petri net and then unfolded using Mole, Cunf can take advantage of their special features to do the job faster and produce a smaller unfolding. Cunf has recently been extended with a verification component that takes advantage of these features; More details can be found at http://www.lsv.ens-cachan.fr/~rodrigue/tools/cunf/. Moreover, Cunf has been integrated into the CosyVerif environment (see section 5.1.2.1). Cunf has also participated in the Model Checking Contest held at the Petri Nets conference in 2013.

5.1.1.3. COSMOS : a Statistical Model Checker for the Hybrid Automata Stochastic Logic

Participant: Benoît Barbot [correspondant].

COSMOS is a statistical model checker for the Hybrid Automata Stochastic Logic (HASL). HASL employs Linear Hybrid Automata (LHA), a generalization of Deterministic Timed Automata (DTA), to describe accepting execution paths of a Discrete Event Stochastic Process (DESP), a class of stochastic models which includes, but is not limited to, Markov chains. As a result HASL verification turns out to be a unifying framework where sophisticated temporal reasoning is naturally blended with elaborate reward-based analysis. COSMOS takes as input a DESP (described in terms of a Generalized Stochastic Petri Net), an LHA and an expression Z representing the quantity to be estimated. It returns a confidence interval estimation of Z; recently, it has been equipped with functionalities for rare event analysis. COSMOS is written in C++ and is freely available to the research community.

Details on COSMOS can be found at http://www.lsv.ens-cachan.fr/~barbot/cosmos/

5.1.2. Platforms

5.1.2.1. CosyVerif

Participants: Serge Haddad, Alban Linard [correspondant], Benoît Barbot.

CosyVerif (http://www.cosyverif.org/) is a platform dedicated to the formal specification and verification of dynamic systems. It allows to specify systems in a graphical editor, using several formalisms (such as automata and Petri nets) and to run verification tools on these models in a dedicated execution server. These tools are mainly developed by researchers of the MeFoSyLoMa group (a Parisian verification group, http://www.mefosyloma.fr/).

The platform is available as installable bundles, that contain both the client, the server, and the tools. It is also usable through two public servers: one with the latest release, one with the development version.

CosyVerif does not only handle several formalisms, but also allows to easily define new ones and integrate them within the platform. To the best of our knowledge, no other verification framework presents such a feature.

It has different kinds of users:

- Tool developers, that are usually researchers, can use the platform to distribute their tools, and have a demonstration version easily available.
- Students can use this platform in modeling and verification courses.
- Industrial case studies are also a target of the CosyVerif platform, in order to promote the practice of formal verification in industry.

The platform is managed by a steering committee consisting of researchers and engineers. This committee decides strategic orientations as well as technical choices.

This year, we have improved the platform in several ways.

- Tools: the platform handles two families of formalisms: automata and Petri nets, both with extensions. It currently integrates 10 tools with 4 new ones this year. Some of them perform structural analyses like invariant computations, while other tools perform behavioural analyses: symbolic reachability graph building, unfolding, stochastic simulations, etc.
- Server: the execution server has been enhanced with asynchronous executions, that allow to disconnect and reconnect the client in long executions. It has also been improved by the ability to communicate between servers to share their available tools.
- Client: a new command line client has been developed for scripting the executions.
- Usability: the client and server are new distributed as one bundle that can be installed easily on all platforms. The server and its tools are embedded within a virtual machine to achieve this portability.

All the developed software are open source and free software tools.

Two engineers have worked this year on CosyVerif:

- Francis Hulin-Hubard, part-time (CNRS engineer);
- Alban Linard, full-time (Inria engineer).

CosyVerif has been the subject of two international communications [28], [29]. It has been presented at the french-speaking PhD school ETR'2013 in Toulouse. and used for teaching in the master SAR of University Pierre et Marie Curie.

MUTANT Project-Team

5. Software and Platforms

5.1. Antescofo

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

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

`../../../../projets/mutant/IMG/AntescofoVMachines.png`

Figure 5. General scheme of Antescofo virtual machine

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

A complete new version of Antescofo has been released on November 2013 on Ircam Forumnet. This version is the result of one year of intensive effort by MuTant team members and associate artists.

This release include major improvements on the reactive language: richer set of synchronization strategies and control structures, dynamic continuous actions, high order function and processes, historicized variables and dynamic expressions everywhere (delays or periods as variables or expressions).

The new internal architecture unifies completely the handling of external (musical) events and the handling of internal (logical) events in a framework able to manage multiple time frames (relative, absolute or computed).

The new version targets the Max and PureData (Pd) environments on Mac, but also on Linux (Pd version) and offers also a standalone offline version. The standalone version is used to simulate a performance in Ascograph.

An important enhancement has been made by proposing a richer set of synchronization strategies between the event recognized by the listening machine and the action to be performed by the reactive engines. Theses new strategies include *anticipative* strategies that exhibits a smoother musical behavior. These strategies are now tested in various musical situations like accompaniments and in the creation of new pieces.

Some new results including a behavioral semantics of the static kernel of the Antescofo reactive engine [17] and tools for formal verification and conformance testing of the system [24], [35] are presented below.

5.2. Ascograph: Antescofo Visual Editor

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

The Antescofo programming language can be extended to visual programing to better integrate existing scores and to allow users to construct complex and embedded temporal structures that are not easily integrated into text. This project is held since October 2012 thanks to Inria ADT Support.

AscoGraph, the new Antescofo graphical score editor has been released this year. It provides a autonomous Integrated Development Environment (IDE) for the authoring of Antescofo scores. Antescofo listening machine, when going forward in the score during recognition, uses the message passing paradigm to perform tasks such as automatic accompaniment, spatialization, etc. The Antescofo score is a text file containing notes (chord, notes, trills, ...) to follow, synchronization strategies on how to trigger actions, and electronic actions (the reactive language). This editor shares the same score parsing routines with Antescofo core, so the validity of the score is checked on saving while editing in AscoGraph, with proper parsing errors handling. Graphically, the application is divided in two parts (see Figure 3). On the left side, a graphical representation of the score, using a timeline with tracks view. On the right side, a text editor with syntax coloring of the score is displayed. Both views can be edited and are synchronized on saving. Special objects such as "curves", are graphically editable: they are used to provide high-level variable automation facilities like breakpoints functions (BPF) with more than 30 interpolations possible types between points, graphically editable.

One other really important feature is the score import from MusicXML or MIDI files, which make the complete workflow of the composition of a musical piece much easier than before.

AscoGraph is strongly connected with Antescofo core object (using OSC over UDP): when a score is edited and modified it is automatically reloaded in Antescofo, and on the other hand, when Antescofo follows a score (during a concert or rehearsal) both graphical and textual view of the score will scroll and show the current position of Antescofo.

AscoGraph is released under Open-Source MIT license and has been released publicly along with new Antescofo architecture during IRCAM Forum 2013.

PAREO Project-Team

5. Software and Platforms

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.

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, Christophe Calvès, Horatiu Cirstea, Pierre-Etienne Moreau [correspondant].

Since 2002, we have developed a new system called *Tom* [31], presented in [17], [18]. 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 experiments on the efficient compilation of rule-based systems [29]. 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 algorithms and rewriting rules for term-graphs.

Tom is documented, maintained, and available at http://tom.loria.fr as well as at http://gforge.inria.fr/projects/tom.

PARKAS Project-Team

5. Software and Platforms

5.1. Lucid Synchrone

Participant: Marc Pouzet [contact].

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

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

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

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

Three major release of the language has been done and the current version is V3 (dev. in 2006). As of 2013, the language is still used for teaching and in our research but we do not develop it anymore. Nonetheless, we have integrated several features from Lucid Synchrone in new research prototypes described below. The Heptagon language and compiler are a direct descendent of it. The new language Zélus for hybrid systems modeling borrows many features originaly introduced in Lucid Synchrone.

5.2. ReactiveML

Participants: Guillaume Baudart, Louis Mandel [contact], Cédric Pasteur.

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

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

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

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

The language was mainly used for the simulation of mobile ad hoc networks at the Pierre and Marie Curie University and for the simulation of sensor networks at France Telecom and Verimag (CNRS, Grenoble). A new application to mixed music programming has been developed.

In 2013, a new web site has been developed. New programming constructs have been added. The runtime system has been cleanup. Moreover, a new implementation based on the PhD of Cédric Pasteur has also been provided http://reactiveml.org/these_pasteur.

5.3. Heptagon

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

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

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

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

Transfer activities based on our experience in Heptagon are taking place through the "Fiabilité and Sûreté de Fonctionnement" project at IRT SystemX, led by Alstom Transport, since 2013.

Heptagon is jointly developed with Gwenael Delaval and Alain Girault from the Inria POP ART team (Grenoble). Gwenael Delaval is developing the controller synthesis tool BZR (http://bzr.inria.fr/) above Heptagon. Both software are distributed under a GPL licence.

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

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

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

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

In 2013, a complete re-implemtantion has been started. This new version will take into account the new features developed during the PhD of Adrien Guatto. Parallel code generation for this new version also involves compilation and runtime system research in collaboration with Nhat Minh Lê and Robin Morisset.

5.5. ML-Sundials

Participants: Timothy Bourke, Jun Inoue, Marc Pouzet [contact].

The ML-Sundials bindings allow the use of the state-of-the-art Sundials numerical simulation library from OCaml programs (like, for instance, the Zélus runtime). The Sundials packages includes three main components: CVODE, IDA, and KINSOL.

This year we redesigned and reimplemented the interface to CVODE to fix a problem with memory leaks between OCaml and C heaps. We have submitted an APP request for this code. The CVODE component is an important part of our work on the Zélus programming language.

We also developed a new interface for the IDA component, which we have started to use in our experiments with DAEs (Modelica).

We plan to develop an interface for the remaining KINSOL component over the next three months and then to release the entire library under an open-source license.

5.6. Zélus

Participants: Timothy Bourke, Marc Pouzet [contact].

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

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

This year we reimplemented several basic numeric solver algorithms after a careful analysis of the Simulink versions together with the binding to SUNDIALS CVODE. This was necessary to enable detailed comparsions between our tool and Simulink (the de facto industrial standard in this domain). We also improved the algorithm for zero-crossing detection, simplified and streamlined the back-end interface.

We developed several new examples to aid in the development, debugging, and dissemination of our work together with various talks and demonstrations. These included a simple backhoe model (which served as a introducing example in the HSCC paper [12]), an adaptive control example from Astrom and Wittenmark's text, and a model of Zeno behaviour based on a zig-zagging object (presented at Synchron).

Zélus has been released officially in 2013 with several complete documented examples on http://zelus.di.ens.fr. An important software development has been done in the compiler internals during year 2013: a new *causality analysis* has been designed and implemented and a new back-end to generate efficient sequential code for both the discrete step and the continuous step.

5.7. GCC

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

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

Licence: GPLv3+ and LGPLv3+

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

PARKAS contributes to the polyhedral compilation framework, also known as Graphite. We also distribute an experimental branch for a stream-programming extension of OpenMP called OpenStream (used in numerous research activites and grants). This effort borrows key design elements to synchronous data-flow languages.

Tobias Grosser is one of main contributors of the Graphite optimization pass of GCC.

5.8. isl

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

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

Licence: MIT

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

isl has become the de-facto standard for every recent polyhedral compilation project. Thanks to a license change from LGPL to MIT, its adoption is also picking up in industry.

5.9. ppcg

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

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

Licence: MIT

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

5.10. Tool support for the working semanticist

Participant: Francesco Zappa Nardelli [contact].

Languages, semantics, tool support, theorem prouvers.

We are working on tools to support large scale semantic definitions, for programming languages and architecture specifications. For that we develop two complementary tools, Ott and Lem.

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

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

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

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

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

In collaboration with Peter Sewell (Cambridge University) and Scott Owens (University of Kent).

The current version of Ott is about 30000 lines of OCaml. The tool is available from http://moscova.inria.fr/~zappa/software/ott (BSD licence). It is widely used in the scientific community. In 2013 we implemented several bug-fixes, we kept the theorem prouver backends up-to date with the prover evolution, and we have been working toward a closer integration with the Lem tool.

The development version of Lem is available from http://www.cs.kent.ac.uk/people/staff/sao/lem/.

5.11. Cmmtest: a tool for hunting concurrency compiler bugs

Participants: Francesco Zappa Nardelli [contact], Robin Morisset, Pankaj More, Anirudh Kumar, Pankaj Prateek Kewalramani, Pejman Attar.

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

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

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

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

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

PARSIFAL Project-Team

5. Software and Platforms

5.1. Abella

Participants: Kaustuv Chaudhuri [correspondant], Matteo Cimini, Dale Miller, Olivier Savary-Bélanger, Yuting Wang.

Main web-site: http://abella-prover.org.

Abella is an interactive theorem prover based on the two-level logic approach. It consists of a sophisticated reasoning logic that supports induction, co-induction, and generic reasoning, and a specification logic that is based on logic programming. Abella was initially designed to reason about simple second-order Lambda Prolog programs, which is sufficient for the computational specifications.

During 2013, as part of the RAPT Associated Team, Chaudhuri and Yuting Wang (former intern from Univ. Minnesota) released version 2.0 of Abella, a culmination of nearly two years of work and a significant improvement in its expressivity. Specifically,

> The Abella specification logic now supports the full higher-order hereditary Harrop logic of λProlog. This logic allows for very natural specifications of higher-order relations, and leads to cleaner and simpler proofs.
>
> The Abella reasoning logic was extended with support for arbitrary dynamic contexts and incremental backchaining. The design is based on fundamental insights from *focusing*, a core strength of the team.
>
> A number of illustrative examples of the use of higher-order reasoning were added to the Abella examples library, including a novel new characterization of marked β-reduction in the λ-calculus in terms of a simple higher-order inductive definition of λ-paths.
>
> These results were published in PPDP 2013 [26].

Abella continues to evolve as part of RAPT. In 2013, we hosted an intern from McGill University, Olivier Savary-Bélanger (supervised by Chaudhuri), who investigated extensions of Abella with *regular context schemas*. Among his contributions:

> Abella's reasoning level has been augmented with a *plugin* system that both extends the syntax of Abella theories and adds new tactics.
>
> The main plugin for context schemas allows definitions of regular contexts and context relations, with entirely automatic proofs of the main administrative lemmas.
>
> Experimentally, this extension can be used to eliminate up to 40% of the proof text, including nearly 100% of the administrative lemmas on contexts, from typical examples from the meta-theory of the λ-calculus.

We expect this extension to become part of the 2.1 release of Abella, scheduled for later in 2014.

One important application of Abella emerged in 2013: the formalization of bisimulation-up-to techniques for process calculi such as CCS and the π-calculus. Chaudhuri, Cimini, and Miller have formulated the correctness proof of a number of prominent up-to-techniques using the co-inductive and higher-order facilities of Abella. This work indicates an important emerging direction for Abella: modular reasoning.

In terms of development, we have welcomed Savary-Bélanger into the development team, and added a number of collaborators into the management team for the Abella web-site.

5.2. Bedwyr

Participants: Quentin Heath, Dale Miller [correspondant].

Main web-site: http://slimmer.gforge.inria.fr/bedwyr/.

During the first half of 2013, Quentin Heath was working as an engineer on the team, supported by the BATT ADJ project funded by Inria. During that time, we worked exclusively on making improvements to Bedwyr. In particular, he made extensive and important changes to the tabling mechanism of Bedwyr, a feature of model checking systems that is capable of remembering past successful proofs (it can even support a finite failure as a successful proof of a negation). These extension allow lemmas to be used to greatly extend the scope of what can be inferred from a table. For example, if we are attempting to show that there is a winning strategy for a given board position, we would certainly like to make use of a lemma that allows one to infer that winning strategies are preserved under symmetries of the board. There are a number of design issues that go along with the design of such a tabling mechanism: for example, should one use such lemmas in a forward-chaining or backward-chaining fashion. Quentin has tested both of these options in order to collect information as to what the trade-offs would be.

We should note that Quentin Heath is now a PhD student on the team and is addressing a number of theoretical questions related to his research on Bedwyr.

5.3. Profound

Participant: Kaustuv Chaudhuri [correspondant].

Profound is a new interactive theorem proving and proof-exploration tool based on the idea of building formal proofs *without* the use of formal proof languages. The core concepts are a generalization of *deep inference* for the underlying logical formalism, and *proof-by-pointing* for the user-interaction metaphors.

A user proves a theorem in *Profound* by using the keyboard and mouse to select subformulas of the theorem and dragging them to their suitable "destination". For instance, the formula $(A \to C) \to (A \wedge B \to C)$ is proved by dragging the two As and the two Cs to each other. This kind of *direct manipulation* is nevertheless constrained by the system to be both correct—meaning that no manipulation is logically unsound—and complete—meaning that every provable theorem can be proved using these metaphors.

The system is still in its early stages, but it currently supports first-order classical linear logic. It has been documented in a paper at ITP 2013 [18].

We are in the process of extending the system to intuitionistic logics, and adding a back-end exporter for more traditional proof systems with formal proof languages such as Coq and Isabelle.

5.4. Psyche

Participants: Mahfuza Farooque, Stéphane Graham-Lengrand [correspondant].

Psyche (*Proof-Search factorY for Collaborative HEuristics*) is a modular proof-search engine whose first version, 1.0, was released in 2012:
http://www.lix.polytechnique.fr/~lengrand/Psyche/

Its motivation is twofold:

On the one hand, prove some mathematics of the broadest range while making the most of problem-specific techniques; On the other hand, gain high confidence about the correctness of the proofs produced without having to rely on a proof-checker.

Psyche's proof-search mechanism is simply the incremental construction of proof-trees in the polarized and focused sequent calculus. Its architecture organizes an interaction between a trusted universal kernel and smart plugins that are meant be efficient at solving certain kinds of problems:

The kernel contains the mechanisms for exploring the proof-search space in a sound and complete way, taking into account branching and backtracking. The output of Psyche comes from the (trusted) kernel and is therefore correct by construction. The plugins then drive the kernel by specifying how the branches of the search space should be explored, depending on the kind of problem that is being treated. The quality of the plugin is then measured by how fast it drives the kernel towards the final answer.

In 2013, major developments were achieved in Psyche, which now handles classical propositional logic *modulo a theory* such as linear arithmetic, equality with uninterpreted symbols, arrays, etc. It therefore works in the same logic as Sat-Modulo-Theories (SMT) solvers and the architecture to handle such theories is the main contribution of 2013, in particular with the integration of the *simplex* algorithm.

Thanks to a plugin that simulates the behavior of a SAT-solver (DPLL) [21], the new version of Psyche can now simulate the behavior of SMT-solvers.

A lot of features inspired by SAT-solvers have now been lifted to proof-search in general, such as a *memoization table* to record and re-use known proofs, the technique of *2-watched literals* to efficiently propagate direct consequences of new hypotheses, machine learning techniques for *restart policies*, etc.

Psyche has been the topic of the 2013 publication [23].

PI.R2 Project-Team

4. Software and Platforms

4.1. COQ (http://coq.inria.fr)

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

4.1.1. Version 8.5

Version 8.5 is expected to be released after the summer of 2014. It will be a major release of the Coq proof assistant, including 6 major new features:

- Parallel development and compilation, inside files and across files, by Enrico Tassi (Inria SpecFun), a result of the Paral-ITP ANR project.
- New proof engine of Arnaud Spiwack (formerly pi.r2 postdoc), more expressive and with clearer semantics.
- Native compilation by Maxime Dénès and Benjamin Grégoire (Inria Marelle, M. Dénès is now at the University of Pennsylvania). A compilation scheme from Coq to OCaml to native code, considerably improving on the previous virtual machine implementation by B. Grégoire.
- A Universe Polymorphic extension by Matthieu Sozeau that allows universe-generic developments, as required by the Homotopy Type Theory library for example.
- Primitive projections for records by Matthieu Sozeau.
- A new document generation system by F. Ripault and Yann Régis-Gianas.

A more detailed description of all the new features will be made in next year's report, but some elements can already be found below.

4.1.2. Evaluation algorithms

Pierre Boutillier has worked on the practical implementation of the unfolding algorithm for global constants that he proposed last year, so that it could become the default process to simplify terms by tactics. A formal presentation has been written in his PhD, to be defended in February 2014.

4.1.3. Internal representation of projections

The change of representation for record projections implemented by Matthieu Sozeau will be part of the 8.5 release. It provides not only exponential gains in performance (type-checking and comparison time and space usage) but also a better basis to work with canonical structures in the unification algorithm, allowing to improve for example the ssreflect inference mechanism significantly. Benchmarks on the HoTT library and a groupoid model construction confirm the exponential gain in performance.

4.1.4. Universes

Matthieu Sozeau followed up his work on universe polymorphism and uncovered important theoretical problems regarding conversion and unification of universe polymorphic constants in the presence of cumulativity and the Prop $\leq$ Type rule. After a careful study of the alternative solutions, he designed a practical correction for the issue and developed a paper proof of conservativity of the complete new system over the original theory of Coq. A paper describing this work has been submitted. The universe polymorphic system, already in use by the HoTT community, will be part of the upcoming 8.5 release.

4.1.5. The Equations plugin

Matthieu Sozeau continued work on the Equations plugin and fixed the remaining bugs preventing full automation of a middle-size example of formalization of the normalization proof of a simply-typed lambda calculus. Wojciech Jedynak, a student of Dariusz Biernacki and Małgorzata Biernacka at the University of Wroclaw, is working under his supervision to do the remaining work before an official release of the plugin can be done. Wojciech Jedynak applied for a 4-month internship supervised by Matthieu Sozeau on an extension of Equations, to start in March 2014.

4.1.6. Internal architecture of the Coq software

With the help of many others, Pierre Letouzey organized in November 2013 the migration of the official Coq source repository from subversion to git. The native use of this decentralized version control system eases the exchange of code amongst Coq developpers (either from the Coq dev team or from external contributors).

Pierre Letouzey, Pierre-Marie Pédrot and Xavier Clerc have continued to work at improving the quality of the OCaml code which composes Coq :

- Many modules have been revised, in particular with cleaner naming convention.
- Almost all uses of the generic OCaml comparison has been chased and transformed into specific code. This avoided many potential bugs with advanced structures, while improving performances at the same time.
- The codes handling OCaml exceptions have been reworked to avoid undue interceptions of critical exceptions.
- Issues involving exceptions are now quite simpler to debug, thanks to easy-to-obtain backtraces.

4.1.7. Efficiency

Pierre-Marie Pédrot has been working on the overall optimization of Coq, by tracking hotspots in the code. Coq trunk is currently much more efficient than its v8.4 counterpart, and is about as quick as v8.3, while having been expanded with a lot of additional features.

Pierre Letouzey has improved the representation of Coq binary files : these files are now smaller (thanks to more sharing), and are reloaded quickly by Coq (thanks to deferred loading of opaque proof terms, which are large and almost never accessed by the user).

4.1.8. Documentation generation

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

4.1.9. General maintenance

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

4.1.10. Modules in Coq

In 2013, Pierre Letouzey has proposed an important rework of the code implementing the module system of Coq. This code was inherited from the successive works of several PhD students, and was in pretty poor shape. While being equivalent in terms of features for the user, the new code should be quite more readable and robust, as well as more efficient: the memory sharing of modular structures should be better, leading to reduced memory footprint for Coq as well as smaller Coq compiled files.

4.1.11. The Coq extraction

Pierre Letouzey has collaborated with colleagues with the aim of extending the extraction tool to additional target languages:

- `C++` with Gabriel Dos Reis and his student Robert Schumacher
- `F#` with David Monniaux

These experiments have been quite promising. In the case of `C++`, an article has been written, it should be re-submitted soon for publication.

4.1.12. Formalisation in Coq

Hugo Herbelin's type-theoretic construction of semi-simplicial sets [22] has been formalised in Coq.

Matthieu Sozeau and Nicolas Tabareau formalised a groupoid model in Coq http://github.com/mattam82/groupoid.

Jaime Gaspar has verified in Coq the correctness of Jean-Louis Krivine's proof that Zermelo-Fraenkel set theory without choice ZF is contained in a variant ZF_ε (useful for Krivine's classical realisability).

4.2. Other software developments

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

Yann Régis-Gianas started the development of the "Hacking Dojo", a web platform to automatically grade programming exercises.

In a different vein, Jaime Gaspar showed that is not always possible to get cross-references right in LaTeX by presenting a LaTeX file where a cross-reference is always wrong (no matter how many times we compile the file).

POLSYS Project-Team

5. Software and Platforms

5.1. FGb

Participant: Jean-Charles Faugère [contact].

FGb is a powerful software for computing Groebner bases.It includes the new generation of algorihms for computing Gröbner bases polynomial systems (mainly the F4, F5 and FGLM algorithms). It is implemented in C/C++ (approximately 250000 lines), standalone servers are available on demand. Since 2006, FGb is dynamically linked with Maple software (version 11 and higher) and is part of the official distribution of this software.
See also the web page http://www-polsys.lip6.fr/~jcf/Software/FGb/index.html.

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

5.2. RAGlib

Participant: Mohab Safey El Din [contact].

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

5.3. Epsilon

Participant: Dongming Wang [contact].

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

PRIVATICS Team

4. Software and Platforms

4.1. Mobilitics

Mobilitics is a joint project, started in 2012 between Inria and CNIL, which targets privacy issues on smartphones. The goal is to analyze the behavior of smartphones applications and their operating system regarding users private data, that is, the time they are accessed or sent to third party companies usually neither with user's awareness nor consent.

In the presence of a wide range of different smartphones available in terms of operating systems and hardware architecture, Mobilitics project focuses actually its study on the two more widespread mobile platforms which are IOS(Iphone) and Android.

Indeed, both versions of Mobilitics software should provide these common requirements: Be able to capture any event about private data access such as User location, Device Unique Identifier, Address Book… Store these events in a local database on the phone for offline analysis Send this local database to Mobilitics server for privacy leakage statistics

A Mobilitics prototype for Iphone has been developed since January 2012 at Privatics. It has already embedded the features listed above and much more. However, a separate prototype for Android has been also developed since September 2012 fulfilling the same equirements listed above because IOS and Android are different in either software or hardware level.

Indeed, some live experiments have been conducted by CNIL with Mobilitics prototype for IOS with the help of volunteers equipped with iphones which they have used for a period of four(4) months(September 2012-January 2013). As a result, some visualization tools have been developed for the data collected in order to showcase private data leakage by the apps which the participants of the experiment have used. Therefore, a press conference has been held by CNIL in Paris in April 2013 during which Mobilitics results for Iphone have been published onto several French newspapers (see Section 8.3)

Likewise, some live experiments will be conducted on Android this year in February 2014 for at least three(3) months with volunteers equipped with Galaxy Nexus smartphones on which Mobilitics will be deployed. As a consequence, a press release by CNIL will be scheduled for the publication of the results obtained for Android with a perspective of comparing Google privacy policy to Apple one.

PROSECCO Project-Team

5. Software and Platforms

5.1. ProVerif

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

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

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

The PROVERIF verifier can prove the following properties:

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

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

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

5.2. CryptoVerif

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

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

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

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

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

5.3. Cryptosense Analyzer

Participants: Graham Steel [correspondant], Romain Bardou.

See also the web page http://cryptosense.com.

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

In June 2013 we submitted a patent application (13 55374) on the reverse engineering process. We also concluded a license agreement between Inria PROSECCO and the nascent spin-off company Cryptosense to commercialize the tool.

5.4. miTLS

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

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

A paper describing the miTLS library was published at IEEE S&P 2013, and two updates to the software were released in 2013. The software and associated research materials are available from http://mitls.rocq.inria.fr.

5.5. WebSpi

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

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

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

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

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

5.6. Defensive JavaScript

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

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

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

SECRET Project-Team (section vide)

SECSI Project-Team

5. Software and Platforms

5.1. Orchids

Participants: Jean Goubault-Larrecq [correspondant], Pierre-Arnaud Sentucq.

The ORCHIDS real-time intrusion detection system was created in 2003-04 at SECSI. Orchids is at the core of a contract between Inria and DGA, started in April 2013, for three years.

Progress in 2013 included:

- Creation of a collection of VirtualBox virtual machines with a pre-installed instance of Orchids, for easy testing and/or installation.
- A collection of scripts, allowing one to rebuild the above cited virtual machines automatically from the sources, as a nightly build (in progress).
- A new algorithm for evaluating the worst-case thread complexity of detection by Orchids, whose first principles were laid out by Jean Goubault-Larrecq, and with two prototype implementations done by Jean-Philippe Lachance, a young L2 intern from Université Laval, Québec. The purpose is to warn users of the complexity of the tasks they delegate to Orchids, and to avert denial of service attacks on Orchids itself.

Objectives for 2014 include:

- Simplifying the Orchids installation process, which has gotten complicated over the years.
- Implementing a frontend tool incorporating the full-fledged version of the worst-case thread complexity algorithm mentioned above, plus some other checks.

SPADES Team

5. Software and Platforms

5.1. Implementations of Synchronous Programs

Participant: Alain Girault.

We have been cooperating for several years with the INRIA team AOSTE (INRIA Sophia-Antipolis and Rocquencourt) on the topic of fault tolerance and reliability of safety critical embedded systems. In particular, we have implemented several new heuristics for fault tolerance and reliability within SYNDEX[4]. Our first scheduling heuristic produces static multiprocessor schedules tolerant to a specified number of processor and communication link failures [62]. The basic principles upon which we rely to make the schedules fault tolerant are, on the one hand, the active replication of the operations [63], and on the other hand, the active replication of communications for point-to-point communication links, or their passive replication coupled with data fragmentation for multi-point communication media (*i.e.*, buses) [64]. Our second scheduling heuristic is multi-criteria: it produces a static multiprocessor schedule such that the reliability is maximized, the power consumption is minimized, and the execution time is minimized [12][4] [37], [38]. Our results on fault tolerance are summarized in a web page [5].

5.2. Apron and BddApron Libraries

Participant: Bertrand Jeannet.

5.2.1. Principles

The APRON library [6] is dedicated to the static analysis of the numerical variables of a program by abstract interpretation [51]. Many abstract domains have been designed and implemented for analysing the possible values of numerical variables during the execution of a program (see Figure 1). However, their API diverge largely (datatypes, signatures, ...), and this does not ease their diffusion and experimental comparison *w.r.t.* efficiency and precision aspects.

The APRON library provides:

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

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

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

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

The BDDAPRON library [7] aims at a similar goal, by adding finite-types variables and expressions to the concrete semantics of APRON domains. It is built upon the APRON library and provides abstract domains for the combination of finite-type variables (booleans, enumerated types, bit vectors) and numerical variables (integers, rationals, floating-point numbers). It first allows the manipulation of expressions that freely mix, using BDDs and MTBDDs, finite-type and numerical APRON expressions and conditions. It then provides abstract domains that combine BDDs and APRON abstract values for representing invariants holding on both finite-type variables and numerical variables.

---

[4] http://www-rocq.inria.fr/syndex

[5] http://pop-art.inrialpes.fr/~girault/Projets/FT

[6] http://apron.cri.ensmp.fr/library/

[7] http://pop-art.inrialpes.fr/~bjeannet/bjeannet-forge/bddapron/index.html

`../../../../projets/spades/IMG/bertrand1.png`

Figure 1. Typical static analyser and examples of abstract domains

5.2.2. Implementation and Distribution

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

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

`../../../../projets/spades/IMG/bertrand2.png`

Figure 2. Organisation of the Apron *library*

5.3. ReaVer

Participant: Bertrand Jeannet.

ReaVer (REActive VERifier [8]) is a tool framework for the safety verification of discrete and hybrid systems specified by logico-numerical data-flow languages, like Lustre, LucidSynchrone or Zelus. It provides time-unbounded analysis based on abstract interpretation techniques.

It features partitioning techniques and several logico-numerical analysis methods based on Kleene iteration with widening and descending iterations, abstract acceleration, max-strategy iteration, and relational abstractions; logico-numerical product and power domains (based on the APRON and BddApron domain libraries)

---

[8] http://members.ktvam.at/schrammel/research/reaver

with convex polyhedra, octagons, intervals, and template polyhedra; and front-ends for the hybrid NBAC format, LUSTRE via lus2nbac, and ZELUS/LUCIDSYNCHRONE. Compared to NBAC, it is connected to higher-level, more recent synchronous and hybrid languages, and provides many more options regarding analysis techniques.

It has been used for several experimental comparisons published in papers. It integrates all the methods developed by Peter Schrammel in his PhD.

5.4. Prototypes

5.4.1. Logical Causality

Participant: Gregor Goessler.

We are developing LOCA, a prototype tool written in Scala that implements the analysis of logical causality described in 6.1.1 . LOCA currently supports causality analysis in BIP. The core analysis engine is implemented as an abstract class, such that support for other models of computation (MOC) can be added by instantiating the class with the basic operations of the MOC.

5.4.2. Cosyma

Participant: Gregor Goessler.

We have developed COSYMA, a tool for automatic controller synthesis for incrementally stable switched systems based on multi-scale discrete abstractions. The tool accepts a description of a switched system represented by a set of differential equations and the sampling parameters used to define an approximation of the state-space on which discrete abstractions are computed. The tool generates a controller — if it exists — for the system that enforces a given safety or time-bounded reachability specification.

5.4.3. Automatic Controller Generation

Participant: Alain Girault.

We have developed a software tool chain to allow the specification of models, controller synthesis, and the execution or simulation of the results. It is based on existing synchronous tools, and thus consists primarily in the use and integration of SIGALI[9] and Mode Automata [10]. It is the result of a collaboration with Emil Dumitrescu (INSA Lyon) and Eric Rutten from the CTRL-A Inria team.

Useful component templates and relevant properties can be materialized, on one hand, by libraries of task models, and, on the other hand, by properties and synthesis objectives.

5.4.4. The Interproc family of static analyzers

Participant: Bertrand Jeannet [contact person].

These analyzers and libraries are of general use for people working in the static analysis and abstract interpretation community.

- FIXPOINT[11]: a generic fix-point engine written in OCAML. It allows the user to solve systems of fix-point equations on a lattice, using a parameterized strategy for the iteration order and the application of widening. It also implements recent techniques for improving the precision of analysis by alternating post-fixpoint computation with widening and descending iterations in a sound way [66].
- INTERPROC[12]: a simple interprocedural static analyzer that infers properties on the numerical variables of programs in a toy language. It is aimed at demonstrating the use of the previous library and the above-described APRON library, and more generally at disseminating the knowledge in abstract interpretation. It is also deployed through a web-interface [13].
- CONCURINTERPROC extends INTERPROC with concurrency, for the analysis of multithreaded programs interacting via shared global variables. It is also deployed through a web-interface [14].

---

[9] http://www.irisa.fr/vertecs/Logiciels/sigali.html

[10] http://www-verimag.imag.fr

[11] http://http://pop-art.inrialpes.fr/people/bjeannet/bjeannet-forge/fixpoint

[12] http://pop-art.inrialpes.fr/people/bjeannet/bjeannet-forge/interproc

[13] http://pop-art.inrialpes.fr/interproc/interprocweb.cgi

- PINTERPROC extends INTERPROC with pointers to local variables. It is also deployed through a web-interface [15].

5.4.5. The SIAAM virtual machine

Participants: Quentin Sabah, Jean-Bernard Stefani [contact person].

The SIAAM abstract machine is an object-based realization of the Actor model of concurrent computation. Actors can exchange arbitrary object graphs in messages while still enjoying a strong isolation property. It guarantees that each actor can only directly access objects in its own local heap, and that information between actors can only flow via message exchange [10]. The SIAAM machine has been implemented for Java as a modified Jikes virtual machine. The resulting SIAAM software comprises:

- A modified Jikes RVM that implements actors and actor isolation as specified by the SIAAM machine.
- A set of static analyses build using the Soot Java optimization framework for optimizing the execution of the SIAAM/Jikes virtual machine, and for helping programmers diagnose potential performance issues.
- A formal proof using the Coq proof assistant of the SIAAM isolation property.

---

[14] http://pop-art.inrialpes.fr/interproc/concurinterprocweb.cgi

[15] http://pop-art.inrialpes.fr/interproc/pinterprocweb.cgi

Specfun Team

5. Software and Platforms

5.1. Mgfun

(1994–): Maple package for symbolic summation, integration, and other closure properties of multivariate special functions.

Now distributed as part of Algolib, a collection of packages for combinatorics and manipulations of special functions, available at http://algo.inria.fr/libraries/.

5.2. DDMF

(2007–): Web site consisting of interactive tables of mathematical formulas on elementary and special functions. The formulas are automatically generated by OCaml and computer-algebra routines. Users can ask for more terms of the expansions, more digits of the numerical values, proofs of some of the formulas, etc. See http://ddmf.msr-inria.inria.fr/.

5.3. DynaMoW

(2007–): Programming tool for controlling the generation of mathematical websites that embed dynamical mathematical contents generated by computer-algebra calculations. Written in OCaml. See http://ddmf.msr-inria.inria.fr/DynaMoW/.

5.4. Ring

(2004–): Coq normalization tool and decision procedure for expressions in commutative ring theories. Written in Coq and OCaml. Integrated in the standard distribution of the Coq proof assistant since 2005.

5.5. SSReflect

(2006–): Extension of the language of the Coq system. Originally written by G. Gonthier for his formal proof of the Four-Color Theorem. A. Mahboubi and E. Tassi participate to its development, maintenance, distribution, user support and have written its user manual. See http://www.msr-inria.fr/projects/mathematical-components/.

5.6. Coqfinitgroup

(2006–): Coq libraries that cover the mechanization of the proof of the Odd Order Theorem. Stable libraries are distributed with the SSReflect extension. A. Mahboubi is one of the main contributors to the code and its documentation. E. Tassi contributed to the design of core data structures and to parts of the formalization. A formal proof was completed in September 2012, and the content of the libraries, under continued improvements in view of potential reuse, is available online at http://www.msr-inria.fr/projects/mathematical-components/.

SUMO Team

5. Software and Platforms

5.1. Sigali

Participants: Hervé Marchand, Nicolas Berthier.

Sigali is a model-checking tool that operates on ILTS (Implicit Labeled Transition Systems, an equational representation of an automaton), an intermediate model for discrete event systems. It offers functionalities for verification of reactive systems and discrete controller synthesis. It is developed jointly by the Espresso/TEA and SUMO teams. The techniques used consist in manipulating the system of equations instead of the set of solutions, which avoids the enumeration of the state space. Each set of states is uniquely characterized by a predicate and the operations on sets can be equivalently performed on the associated predicates. Therefore, a wide spectrum of properties, such as liveness, invariance, reachability and attractivity, can be checked. Algorithms for the computation of predicates on states are also available. Sigali is connected with the Polychrony environment (Espresso/Tea project-team) as well as the Matou environment (VERIMAG), thus allowing the modeling of reactive systems by means of Signal Specification or Mode Automata and the visualization of the synthesized controller by an interactive simulation of the controlled system. Sigali is registered at APP under the identification number IDDN.FR.001.370006.S.P.1999.000.10600.

Sigali is also integrated as part of the compiler of the language BZR (web site).

We are currently developing a new version of Sigali that will be able to handle numerical variables.

5.2. Tipex

Participants: Thierry Jéron, Hervé Marchand, Srinivas Pinisetty.

We are implementing a prototype tool named Tipex (TImed Properties Enforcement during eXecution) for the enforcement of timed properties, in collaboration with Ylies Falcone (LIG, Grenoble). Tipex is based on the theory and algorithms that we develop for the synthesis of enforcement monitors for properties specified by timed automata (TA) [45] (see Subsection 6.2.3). The prototype is developped in python, and uses the PyUPPAAL and DBMpyuppaal libraries of the UPPAAL tool. It is currently restricted to safety and co-safety timed property. The property provided as input to the tool is a TA that can be specified using the UPPAAL tool, and is stored in XML format. The tool synthesizes an enforcement monitor from this TA, which can then be used to enforce a sequence of timed events to satisfy the property. Experiments have been conducted on a set of case studies. This allowed to validate the architecture and feasibility of enforcement monitoring in a timed setting and to have a first assessment of performance (and to what extent the overhead induced by monitoring is negligible).

5.3. SOFAT

Participants: Loïc Hélouët, Rouwaida Abdallah.

SOFAT is the acronym for Scenario Oracle and Formal Analysis Toolbox. As this name suggests it is a formal analysis toolbox for scenarios. Scenarios are informal descriptions of behaviors of distributed systems. SOFAT allows the edition and analysis of distributed systems specifications described using Message Sequence Charts, a scenario language standardized by the ITU [54]. The main functionalities proposed by SOFAT are the textual edition of Message Sequence Charts, their graphical visualization, the analysis of their formal properties, and their simulation. The analysis of the formal properties of a Message Sequence Chart specification determines if a description is regular, local choice, or globally cooperative. Satisfaction of these properties allows respectively for model-checking of logical formulae in temporal logic, implementation, or comparison of specifications. All these applications are either undecidable problems or unfeasible if the Message Sequence

Chart description does not satisfy the corresponding property. The SOFAT toolbox implements most of the theoretical results obtained on Message Sequence Charts this last decade. It is regularly updated and re-distributed. The purpose of this software is twofold: provide a scenario based specification tool for developers of distributed applications; serve as a platform for theoretical results on scenarios and partial orders SOFAT provides several functionalities, that are: syntactical analysis of scenario descriptions, formal analysis of scenario properties, interactive simulation of scenarios when possible, and diagnosis. See also the web page.

This year, SOFAT has been extended with model transformation techniques that allow to transform non-implementable HMSCs into implementable ones [49].

APP: IDDN.FR.001.080027.000.S.P.2003.00.10600
Programming language: Java

5.4. DAXML

Participant: Loïc Hélouët.

DAXML is an implementation of Distributed Active Documents, a formalism for data centric design of Web Services proposed by Serge Abiteboul. This implementation is based on a REST framework, and can run on a network of machines connected to internet and equipped with JAVA. This implementation was realized during the post doc of Benoit Masson in 2011. A demo of the software is available at this web page. We plan to maintain this prototype as a demonstrator for our Web Services activities, and to distribute the sources.

TASC Project-Team

5. Software and Platforms

5.1. CHOCO

Participants: Nicolas Beldiceanu, Alexis de Clerq, Jean-Guillaume Fages [main developer], Narendra Jussien [correspondant], Arnaud Letort, Xavier Lorca [correspondant], Thierry Petit, Charles Prud'Homme [main developer], Remi Douence.

CHOCO is a Java discrete constraints library integrating within a same system *explanations*, *soft constraints* and *global constraints* (90000 lines of source code). This year developments were focussing on the following aspects:

1. Since September 2011, we are working on a new version of the CHOCO solver. This implies a total refactoring of the source code in order to make it simpler to use and maintain. We introduce a new propagation engine framework that directly handle state-of-the-art techniques, such as *advisors*, *propagator groups*, *activity-based search* and *explanations*, to ensure a good level of efficiency, and plug a MiniZinc modeling language parser. An alpha release will be available by the beginning of 2013.
2. In the context of the new version of the CHOCO solver we design an *adaptive propagation engine* to enhance performance as well as a *solver independent language to write strategies* for controlling the new adaptive propagation engine. The adaptive propagation engine can both deal with variable-oriented propagation engines and constraint-oriented propagation engines. It is usually accepted that there is no best approach in general and modern constraint solvers therefore implement only one.
3. New scalable global constraints were provides both in the context of *graph constraints* (with also graph variables) and in the context of *scheduling constraints*. These constraints respectively allow to handle sparse graphs with up to 10000 vertices, and resource scheduling problems with up to one million tasks.
4. A new global constraint called *focus* for concentrating high cost values motivated by several concrete examples, such as resource constrained scheduling problems with machine rentals, was introduced.
5. The work on providing probability-based constraints to get light propagation filtering algorithm has been pursued. A particular focus has been put on calculating the probabilistic indicator for the bound-consistency propagator of an *alldifferent* constraint.
6. A stable version of Choco, tagged 3.1.0, is available since September 2nd 2013. This version won two silver medals in the MiniZinc Challenge 2013. It has been downloaded more than 4000 times between September and December 2013.

The link to the system and documentation is http://choco.emn.fr.

5.2. IBEX

Participants: Ignacio Salas Donoso, Anthony Baire, Gilles Chabert [correspondant], Rémi Douence, Bertrand Neveu, Gilles Trombettoni.

IBEX (Interval-Based EXplorer) is a C++ library for solving nonlinear constraints over real numbers (25000 lines of source code). The main feature of Ibex is its ability to build solver/paver strategies declaratively through the contractor programming paradigm.

In 2013 the work on IBEX has focussed on the following points.

- Continuing last year work on the redesign of the architecture, the IBEX library has been augmented with new features. First, *affine forms* (with the help of Jordan Ninin) have been introduced in core calculations as an alternative to interval arithmetic. Five different implementations are under testing. A symbolic differentiation module has been developed and used for applying first-order conditions in global optimization (an interval variant of Khun-Tucker conditions). We have started a redesign of the global optimizer to integrate properly first-order conditions and exact equality constraints (not relaxed to inequalities). This work will be pursued in early 2014.
- In deterministic continuous constrained global optimization, upper bounding the objective function generally resorts to local minimization at several nodes of the branch and bound. We have proposed an alternative approach when the constraints are inequalities and the feasible space has a non-null volume. First, we extract an inner region, i.e., an entirely feasible convex polyhedron or box in which all points satisfy the constraints. Second, we select a point inside the extracted inner region and update the upper bound with its cost. We have implemented this principle with two inner region extraction algorithms, once being based on the algorithm published in CP'10 by G. Chabert & N. Beldiceanu for sweeping with continuous domains. The corresponding paper is *Upper Bounding in Inner Regions for Global Optimization under Inequality Constraints* that has been accepted for publication in JOGO (Journal of Global Optimization).
- The packaging of IBEX has also been considerably enhanced with the integration of a 3rd interval library (Filib++) and a 2nd LP solver (CPlex) and by making the library compatible with 64-bits platforms. The writing of documentation and tutorials has continued. A document of specifications has also been written for an automatic benchmarking tool.

5.3. CHOCO-IBEX

Participants: Gilles Chabert [correspondant], Jean-Guillaume Fages [correspondant], Charles Prud'Homme [correspondant].

Work has been done to provide an interface for connecting the CHOCO and the IBEX libraries in order to handle problems where we both have continuous and discrete variables. This interface allows to filter continuous domains from CHOCO with the IBEX engine as well as to check for unsatisfiability or entailment. It also manages reification variables. The "Choco-Ibex" interface, initially designed for filtering only, has been augmented with *inflators*, a generic service on which the hybrid *geost* sweep algorithm is based. This gives a basis for a possible implementation of a future hybrid packing solver (objects with curved shapes), the target application of the NetWMS2 project. The interface is available in Choco-3.1.0.

5.4. Artificial Intelligence Using Randomness

Participant: Florian Richoux [correspondant].

AIUR (Artificial Intelligence Using Randomness) is an AI for $StarCraft: BroodWar^{tm}$.

The main idea is to be unpredictable by making some stochastic choices. The AI starts a game with a "mood" randomly picked up among 5 moods, dictating some behaviors (aggressive, fast expand, macro-game, ...). In addition, some other choices (productions, timing attacks, early aggressions, ...) are also taken under random conditions.

Learning is an essential part of AIUR. For this, it uses persistent I/O files system to record which moods are efficient against a given opponent, in order to modify the probability distribution for the mood selection.

AIUR is an open source program under GNU GPL V3 licence, written in C++ (18.000 lines of code). Source and documentations are available at http://code.google.com/p/aiurproject/. AIUR finished 3rd to $StarCraft^{tm}$ AI competitions organized at the conferences AIIDE 2013 and CIG 2013.

5.5. Global Constraint Catalog

Participants: Nicolas Beldiceanu [correspondant], Mats Carlsson, Helmut Simonis.

The global constraint catalog presents and classifies global constraints and describes different aspects with meta data. It consist of

1. a pdf version that can be downloaded from http://www.emn.fr/z-info/sdemasse/gccat/ (at item *last working version*) containing 423 constraints, 3936 pages and 900 figures,
2. an on line version accessible from the previous address,
3. meta data describing the constraints (buton *PL* for each constraint, e.g., alldifferent.pl),
4. an online service (i.e, a *constraint seeker*) which provides a web interface to search for global constraints, given positive and negative ground examples.

This year developments were focussing on:

1. maintaining the catalogue,
2. making the *core global constraints* (10 constraints) more accessible to a wider audience:
   - for this purpose examples with their corresponding pictures have been systematically provided for showing all solutions for an example of each core global constraint.
   - in addition a set of about 55 exercises with their corrections have been done for half of the core global constraints.
3. a redesign of all the 900 figures of the catalog has been undertaken in autumn 2012 using TikZ (in December 2013 750 figures were redesigned).
4. adding counting information related to the number of solutions of a constraint (integer sequences and visualization).
5. adding constraints related to sequences that we found relevant for learning constraints from electricity production curves.

TOCCATA Team

5. Software and Platforms

5.1. The CiME rewrite toolbox

Participants: Évelyne Contejean [contact], Claude Marché, Andrei Paskevich.

CiME is a rewriting toolbox. Distributed since 1996 as open source, at URL http://cime.lri.fr. Beyond a few dozens of users, CiME is used as back-end for other tools such as the TALP tool developed by Enno Ohlebusch at Bielefeld university for termination of logic programs; the MU-TERM tool (http://www.dsic.upv.es/~slucas/csr/termination/muterm/) for termination of context-sensitive rewriting; the CARIBOO tool (developed at Inria Nancy Grand-Est) for termination of rewriting under strategies; and the MTT tool (http://www.lcc.uma.es/~duran/MTT/) for termination of Maude programs. CiME2 is no longer maintained, and the currently developed version is CiME3, available at http://a3pat.ensiie.fr/pub. The main new feature of CiME3 is the production of traces for *Coq*. CiME3 is also developed by the participants of the A3PAT project at the CNAM, and is distributed under the Cecill-C license.

5.2. The Why platform

Participants: Claude Marché [contact], Jean-Christophe Filliâtre, Guillaume Melquiond, Andrei Paskevich.

Criteria for Software Self-Assessment [5]: A-3, SO-4, SM-3, EM-2, SDL-5-down, OC-4.

The *Why* platform is a set of tools for deductive verification of Java and C source code. In both cases, the requirements are specified as annotations in the source, in a special style of comments. For Java (and Java Card), these specifications are given in JML and are interpreted by the *Krakatoa* tool. Analysis of C code must be done using the external *Frama-C* environment, and its Jessie plugin which is distributed in *Why*.

The platform is distributed as open source, under GPL license, at http://why.lri.fr/. The internal VC generator and the translators to external provers are no longer under active development, as superseded by the *Why3*system described below.

The *Krakatoa* and *Jessie* front-ends are still maintained, although using now by default the *Why3* VC generator. These front-ends are described in a specific web page http://krakatoa.lri.fr/. They are used for teaching (University of Evry, École Polytechnique, etc.), used by several research groups in the world, e.g at Fraunhofer Institute in Berlin [93], at Universidade do Minho in Portugal [54], at Moscow State University, Russia (http://journal.ub.tu-berlin.de/eceasst/article/view/255).

5.3. The Why3 system

Participants: Jean-Christophe Filliâtre [contact], Claude Marché, Guillaume Melquiond, Andrei Paskevich.

Criteria for Software Self-Assessment: A-3-up, SO-4, SM-4, EM-4, SDL-5, OC-4.

Why3 is the next generation of *Why*. *Why3* clearly separates the purely logical specification part from generation of verification conditions for programs. It features a rich library of proof task transformations that can be chained to produce a suitable input for a large set of theorem provers, including SMT solvers, TPTP provers, as well as interactive proof assistants.

It is distributed as open source, under GPL license, at http://why3.lri.fr/.

---

[5] self-evaluation following the guidelines (http://www.inria.fr/content/download/11783/409665/version/4/file/SoftwareCriteria-V2-CE.pdf) of the Software Working Group of Inria Evaluation Committee(http://www.inria.fr/institut/organisation/instances/commission-d-evaluation)

Why3 is used as back-end of our own tools *Krakatoa* and *Jessie*, but also as back-end of the GNATprove tool (Adacore company), and in a near future of the WP plugin of Frama-C. *Why3* has been used to develop and prove a significant part of the programs of our team gallery http://proval.lri.fr/gallery/index.en.html, and used for teaching (Master Parisien de Recherche en Informatique).

Why3 is used by other academic research groups, e.g. within the CertiCrypt/EasyCrypt project (http://easycrypt.gforge.inria.fr/) for certifying cryptographic programs.

5.4. The Alt-Ergo theorem prover

Participants: Sylvain Conchon [contact], Évelyne Contejean, Alain Mebsout, Mohamed Iguernelala.

Criteria for Software Self-Assessment: A-3-up, SO-4, SM-4-up, EM-4, SDL-5, OC-4.

Alt-Ergo is an automatic, little engine of proof dedicated to program verification, whose development started in 2006. It is fully integrated in the program verification tool chain developed in our team. It solves goals that are directly written in the *Why*'s annotation language; this means that *Alt-Ergo* fully supports first order polymorphic logic with quantifiers. *Alt-Ergo* also supports the standard [113] defined by the SMT-lib initiative.

It is currently used in our team to prove correctness of C and Java programs as part of the *Why* platform and the new *Why3* system. *Alt-Ergo* is also called as an external prover by the Pangolin tool developed by Y. Regis Gianas, Inria project-team Gallium http://code.google.com/p/pangolin-programming-language/. Alt-Ergo is usable as a back-end prover in the SPARK verifier for ADA programs, since Oct 2010. It is planed to be integrated in next generation of Airbus development process.

Alt-Ergo is distributed as open source, under the CeCILL-C license, at URL http://alt-ergo.lri.fr/.

5.5. The Cubicle model checker modulo theories

Participants: Sylvain Conchon [contact], Alain Mebsout.

Partners: A. Goel, S. Krstić (Intel Strategic Cad Labs in Hillsboro, OR, USA), F. Zaïdi (LRI, Université Paris-sud)

Cubicle is an open source model checker for verifying safety properties of array-based systems. This is a syntactically restricted class of parametrized transition systems with states represented as arrays indexed by an arbitrary number of processes. Cache coherence protocols and mutual exclusion algorithms are typical examples of such systems.

Cubicle model-checks by a symbolic backward reachability analysis on infinite sets of states represented by specific simple formulas, called cubes. Cubicle is based on ideas introduced by MCMT (http://users.mat.unimi.it/users/ghilardi/mcmt/) from which, in addition to revealing the implementation details, it differs in a more friendly input language and a concurrent architecture. Cubicle is written in OCaml. Its SMT solver is a tightly integrated, lightweight and enhanced version of *Alt-Ergo*; and its parallel implementation relies on the Functory library.

5.6. Bibtex2html

Participants: Jean-Christophe Filliâtre [contact], Claude Marché.

Criteria for Software Self-Assessment: A-5, SO-3, SM-3, EM-3, SDL-5, OC-4.

Bibtex2html is a generator of HTML pages of bibliographic references. Distributed as open source since 1997, under the GPL license, at http://www.lri.fr/~filliatr/bibtex2html/. We estimate that between 10000 and 100000 web pages have been generated using Bibtex2html.

Bibtex2html is also distributed as a package in most Linux distributions. Package popularity contests show that it is among the 20% most often installed packages.

5.7. OCamlgraph

Participants: Jean-Christophe Filliâtre [contact], Sylvain Conchon.

OCamlgraph is a graph library for *OCaml*. It features many graph data structures, together with many graph algorithms. Data structures and algorithms are provided independently of each other, thanks to *OCaml* module system. OCamlgraph is distributed as open source, under the LGPL license, at http://OCamlgraph.lri.fr/. It is also distributed as a package in several Linux distributions. OCamlgraph is now widely spread among the community of *OCaml* developers.

5.8. Mlpost

Participant: Jean-Christophe Filliâtre [contact].

Mlpost is a tool to draw scientific figures to be integrated in LaTeX documents. Contrary to other tools such as TikZ or MetaPost, it does not introduce a new programming language; it is instead designed as a library of an existing programming language, namely *OCaml*. Yet it is based on MetaPost internally and thus provides high-quality PostScript figures and powerful features such as intersection points or clipping. Mlpost is distributed as open source, under the LGPL license, at http://mlpost.lri.fr/. Mlpost was presented at JFLA'09 [56].

5.9. Functory

Participant: Jean-Christophe Filliâtre [contact].

Functory is a distributed computing library for *OCaml*. The main features of this library include (1) a polymorphic API, (2) several implementations to adapt to different deployment scenarios such as sequential, multi-core or network, and (3) a reliable fault-tolerance mechanism. Functory was presented at JFLA 2011 [92] and at TFP 2011 [91].

5.10. The Pff library

Participant: Sylvie Boldo [contact].

Criteria for Software Self-Assessment: A-2, SO-3, SM-3, EM-3, SDL-5, OC-4.

The Pff library for the *Coq* proof assistant is a formalization of floating-point arithmetic with high-level definitions and high-level properties [64].

It is distributed as open source, under a LGPL license, at http://lipforge.ens-lyon.fr/www/pff/, and is packaged in Debian and Ubuntu as “coq-float”.

It was initiated by M. Daumas, L. Rideau and L. Théry in 2001, and then developed and maintained by S. Boldo since 2004. It is now only maintained by S. Boldo. The development has ended as this library is now subsumed by the Flocq library (see below).

5.11. The Flocq library

Participants: Sylvie Boldo [contact], Guillaume Melquiond.

Criteria for Software Self-Assessment: A-2, SO-3, SM-3, EM-3, SDL-5, OC-4.

The Flocq library for the *Coq* proof assistant is a comprehensive formalization of floating-point arithmetic: core definitions, axiomatic and computational rounding operations, high-level properties [6]. It provides a framework for developers to formally certify numerical applications.

It is distributed as open source, under a LGPL license, at http://flocq.gforge.inria.fr/. It was first released in 2010.

5.12. The Gappa tool

Participant: Guillaume Melquiond [contact].

Criteria for Software Self-Assessment: A-3, SO-4, SM-4, EM-3, SDL-5, OC-4.

Given a logical property involving interval enclosures of mathematical expressions, Gappa tries to verify this property and generates a formal proof of its validity. This formal proof can be machine-checked by an independent tool like the *Coq* proof-checker, so as to reach a high level of confidence in the certification [83] [119].

Since these mathematical expressions can contain rounding operators in addition to usual arithmetic operators, Gappa is especially well suited to prove properties that arise when certifying a numerical application, be it floating-point or fixed-point. Gappa makes it easy to compute ranges of variables and bounds on absolute or relative roundoff errors.

Gappa is being used to certify parts of the mathematical libraries of several projects, including CRlibm, FLIP, and CGAL. It is distributed as open source, under a Cecill-B / GPL dual-license, at http://gappa.gforge.inria.fr/. Part of the work on this tool was done while in the Arénaire team (Inria Rhône-Alpes), until 2008.

5.13. The Interval package for Coq

Participant: Guillaume Melquiond [contact].

Criteria for Software Self-Assessment: A-3, SO-4, SM-3, EM-3, SDL-4, OC-4.

The Interval package provides several tactics for helping a *Coq* user to prove theorems on enclosures of real-valued expressions. The proofs are performed by an interval kernel which relies on a computable formalization of floating-point arithmetic in *Coq*.

It is distributed as open source, under a LGPL license, at http://www.lri.fr/~melquion/soft/coq-interval/. Part of the work on this library was done while in the Mathematical Components team (Microsoft Research–Inria Joint Research Center).

5.14. The Alea library for randomized algorithms

Participant: Christine Paulin-Mohring [contact].

Criteria for Software Self-Assessment: A-2, SO-3, SM-2, EM-3, SDL-4, OC-4.

The ALEA library is a *Coq* development for modeling randomized functional programs as distributions using a monadic transformation. It contains an axiomatisation of the real interval $[0, 1]$ and its extension to positive real numbers. It introduces definition of distributions and general rules for approximating the probability that a program satisfies a given property.

It is distributed as open source, at http://www.lri.fr/~paulin/ALEA. It is used as a basis of the Certicrypt environment (MSR-Inria joint research center, Imdea Madrid, Inria Sophia-Antipolis) for formal proofs for computational cryptography [59]. It is also experimented in LABRI as a basis to study formal proofs of probabilistic distributed algorithms. ALEA version 8 distributed in May 2013 includes a module to reason with random variables with values in positive real numbers.

5.15. The Coccinelle library for term rewriting

Participant: Évelyne Contejean [contact].

Coccinelle is a *Coq* library for term rewriting. Besides the usual definitions and theorems of term algebras, term rewriting and term ordering, it also models some of the algorithms implemented in the CiME toolbox, such a matching, matching modulo associativity-commutativity, computation of the one-step reducts of a term, RPO comparison between two terms, etc. The RPO algorithm can effectively be run inside *Coq*, and is used in the Color developement (http://color.inria.fr/) as well as for certifying Spike implicit induction theorems in *Coq* (Sorin Stratulat).

Coccinelle is available at http://www.lri.fr/~contejea/Coccinelle, and is distributed under the Cecill-C license.

5.16. The Coquelicot library for real analysis

Participants: Sylvie Boldo [contact], Catherine Lelay, Guillaume Melquiond.

Criteria for Software Self-Assessment: A-3, SO-4, SM-2, EM-3, SDL-4, OC-4.

Coquelicot is a *Coq* library dedicated to real analysis: differentiation, integration, and so on. It is a conservative extension of the standard library of *Coq*, but with a strong focus on usability.

Coquelicot is available at http://coquelicot.saclay.inria.fr/.

5.17. CFML

Participant: Arthur Charguéraud [contact].

Criteria for Software Self-Assessment: A-2, SO-4, SM-2, EM-3, SDL-1, OC-4. The *CFML* tool supports the verification of *OCaml* programs through interactive *Coq* proofs. The tool is made of two parts: on the one hand, a characteristic formula generator implemented as an *OCaml* program that parses *OCaml* code and produces *Coq* formulae; and, on the other hand, a *Coq* library that provides notation and tactics for manipulating characteristic formulae interactively in *Coq*.

CFML is distributed under the LGPL license, and is available at http://arthur.chargueraud.org/softs/cfml/. The tool has been initially developed by A. Charguéraud in 2010, and has been maintained and improved since by the author.

TRIO Team

5. Software and Platforms

5.1. ANR Open-PEOPLE platform

Participants: Anis Koubaa, 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.

In 2013, work in TRIO on this platform was minimal, because of the lack of development resources.

We performed software updates on the servers, ensuring the continuity of access (which is especially important since other partners of the former ANR Open-PEOPLE project still use this platform and actively develop on it). We fixed a few bugs on the platform.

In late 2013, we started working, in the context of Anis Koubaa's student project, on adding a new functionality to help develop energy combustion models, namely the automatic extraction of mathematical laws, derived from the measurements (cloud of points) coming from the experimental hardware platform.

5.2. VITRAIL

Participants: Pierre Caserta, Romarik Jodin, Olivier Zendra.

The aim of the VITRAIL operation is to provide tools for the advanced and immersive visualization of programs. Some of this work has been done with the University of Montréal, the University of Montpellier and to a lesser extent the 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.

In 2013, we were able to restart developments on the VITRAIL platform.

We first explored a Linux port, toward which we progressed but were stopped by the fact we relied on Ogre3D, a library that calls some Windows specific APIs. We have identified those, and we believe that an OpenGL port would be a sensible path to OS portability.

We also ported our VITRAIL Vizualizer software to a new type of display, namely 3 interactive whiteboards placed so as to form a 3-side box.

Finally, we also successfully implemented a first prototype of interaction through a head-tracking system relying on two cameras. First experiments gave promising results.

VEGAS Project-Team

4. Software and Platforms

4.1. 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 [7], [10] 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 [29] bundled with the GMP multi-precision integer arithmetic [28]. 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 [30].

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.

4.2. 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 [4] 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 ALCIX 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.

4.3. 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 [32] (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 [34]. We are currently developing a model of the *Bivariate Algebraic Kernel* based a new bivariate polynomial solver.

4.4. Fast_polynomial: fast polynomial evaluation software

The library *fast_polynomial*[1] provides fast evaluation and composition of polynomials over several types of data. It is interfaced for the computer algebra system *Sage* and its algorithms are documented[2]. This software is meant to be a first step toward a certified numerical software to compute the topology of algebraic curves and surfaces. It can also be useful as is and is submitted for integration in the computer algebra system *Sage*.

This software is focused on *fast online computation*, *multivariate evaluation*, *modularity*, and *efficiency*.

Fast online computation. The library is optimized for the evaluation of a polynomial on several point arguments given one after the other. The main motivation is numerical path tracking of algebraic curves, where a given polynomial criterion must be evaluated several thousands of times on different values arising along the path.

Multivariate evaluation. The library provides specialized fast evaluation of multivariate polynomials with several schemes, specialized for different types such as *mpz* big ints, *boost* intervals with hardware precision, *mpfi* intervals with any given precision, etc.

Modularity. The evaluation scheme can be easily changed and adapted to the user needs. Moreover, the code is designed to easily extend the library with specialization over new *C++* objects.

Efficiency. The library uses several tools and methods to provide high efficiency. First, the code uses templates, such that after the compilation of a polynomial for a specific type, the evaluation performance is equivalent to low-level evaluation. Locality is also taken into account: the memory footprint is minimized, such that an evaluation using the classical Hörner scheme will use $O(1)$ temporary objects and divide and conquer schemes will use $O(\log n)$ temporary objects, where n is the degree of the polynomial. Finally, divide and conquer schemes can be evaluated in parallel, using a number of threads provided by the user.

---

[1] http://trac.sagemath.org/sage_trac/ticket/13358

[2] http://arxiv.org/abs/1307.5655

VERIDIS Project-Team

5. Software and Platforms

5.1. The veriT solver

Participants: David Déharbe, Pablo Dobal, Haniel Barbosa, Pascal Fontaine [correspondent].

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 over 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 at the veriT Web site.

Efforts in 2013 have been focused on efficiency, and more specifically on arithmetic. A preliminary prototype integrating the solver Redlog for non-linear arithmetic has been stabilized. First results are encouraging; this prepares the ground for the starting ANR project SMArT (Satisfiability Modulo Arithmetic Theories), involving both sites of the VeriDis team (veriT being developed in Nancy and Redlog being designed in Saarbrücken), as well as Systerel as an industrial partner.

In late 2013, Haniel Barbosa joined the team as a PhD student. He will work on theoretical and practical aspects of handling quantifiers in SMT frameworks, which is currently an important challenge for SMT, and he will implement his techniques in veriT.

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 solver is available as a plugin for the Rodin platform for discharging proof obligations generated in Event-B [39]; on a large repository of industrial and academic cases, this SMT-based plugin decreased by 75% the number of proof obligations requiring human interactions, compared to the original B prover.

5.2. The TLA+ proof system

Participants: Bhargav Bhatt, Stephan Merz [correspondent], Hernán Vanzetto.

TLAPS, the TLA$^+$ proof system, is a platform for developing and mechanically verifying proofs about TLA$^+$ specifications. It is developed at the Joint MSR-Inria Centre. The TLA$^+$ proof language is hierarchical and explicit. 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.

The current version 1.2.1 of TLAPS was released in September 2013, it is distributed under a BSD-like license at http://tla.msr-inria.inria.fr/tlaps/content/Home.html. The prover currently handles the non-temporal part of TLA$^+$ and can 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, and a backend for interfacing with SMT solvers. The SMT backend, developed in Nancy, has been further improved in 2013 and is now considered by users as the most useful backend prover for system verification. During his internship in the summer of 2013, Bhargav Bhatt helped design and implement a standard library of TLA$^+$ theorems about functions, sequences, and finite sets that is now part of the TLAPS distribution. Development of support for temporal reasoning in TLAPS has started in late 2013.

ALEA Project-Team

5. Software and Platforms

5.1. BiiPS software

BiiPS is a general software, developped by Adrien Todeschini, for Bayesian inference with interacting particle systems, a.k.a. sequential Monte Carlo (SMC) methods. It aims at popularizing the use of these methods to non-statistician researchers and students, thanks to its automated “black box" inference engine.

It borrows from the BUGS/JAGS software, widely used in Bayesian statistics, the statistical modeling with graphical models and the language associated with their descriptions.

Unlike MCMC methods used by BUGS/JAGS, SMC methods are more adapted to dynamic problems (tracking, signal filtering, etc).

A beta version of the software can be downloaded from the website of the BiiPS project. This software has been presented at the international workshop BayesComp in Kyoto, the international conference ISBA in Tokyo, the conference on Premières Rencontres R in Bordeaux, and the international workshop on efficient simulation in finance in Paris.

Adrien Todeschini participated to a session focused on Recent Developments in Software for MCMC (and SMC) (MCMSki IV, Fifth IMS-ISBA joint meeting MCMSki IV, Chamonix Mont-Blanc, France).

This invited panel features four leading researchers working on software development for Bayesian computation. Each panelist will highlight their particular software, including its history, development, and relative strengths and weaknesses. Looking forward, panelists will discuss and debate the future of Bayesian computation and software development, including challenges, opportunities and bottlenecks. Emphasis throughout will be on simplifying and automating the implementation of Monte Carlo methods, with an eye towards scalability to larger and more complex models and data.

APICS Project-Team

5. Software and Platforms

5.1. RARL2

Participant: Martine Olivi [corresponding participant].

Status: Currently under development. A stable version is maintained.

This software is developed in collaboration with Jean-Paul Marmorat (Centre de mathématiques appliquées (CMA), École des Mines de Paris).

RARL2 (Réalisation interne et Approximation Rationnelle L2) is a software for rational approximation (see Section 3.3.2.2) http://www-sop.inria.fr/apics/RARL2/rarl2.html.

The software RARL2 computes, from a given matrix-valued function in $\overline{H}^{2^{m\times l}}$, a local best rational approximant in the L^2 norm, which is *stable and of prescribed McMillan degree* (see Section 3.3.2.2). It was initially developed in the context of linear (discrete-time) system theory and makes an heavy use of the classical concepts in this field. The matrix-valued function to be approximated can be viewed as the transfer function of a multivariable discrete-time stable system. RARL2 takes as input either:

- its internal realization,
- its first N Fourier coefficients,
- discretized (uniformly distributed) values on the circle. In this case, a least-square criterion is used instead of the L^2 norm.

It thus performs model reduction in case 1) and 2) and frequency data identification in case 3). In the case of band-limited frequency data, it could be necessary to infer the behavior of the system outside the bandwidth before performing rational approximation (see Section 3.2.2). An appropriate Möbius transformation allows to use the software for continuous-time systems as well.

The method is a steepest-descent algorithm. A parametrization of MIMO systems is used, which ensures that the stability constraint on the approximant is met. The implementation, in Matlab, is based on state-space representations.

The number of local minima can be rather high so that the choice of an initial point for the optimization can play a crucial role. Two methods can be used: 1) An initialization with a best Hankel approximant. 2) An iterative research strategy on the degree of the local minima, similar in principle to that of RARL2, increases the chance of obtaining the absolute minimum by generating, in a structured manner, several initial conditions.

RARL2 performs the rational approximation step in our applications to filter identification (see Section 4.5) as well as sources or cracks recovery (see Section 4.2). It was released to the universities of Delft, Maastricht, Cork and Brussels. The parametrization embodied in RARL2 was also used for a multi-objective control synthesis problem provided by ESTEC-ESA, The Netherlands. An extension of the software to the case of triple poles approximants is now available. It provides satisfactory results in the source recovery problem and it is used by FindSources3D (see Section 5.6).

5.2. RGC

Participant: Fabien Seyfert [corresponding participant].

Status: A stable version is maintained.

This software is developed in collaboration with Jean-Paul Marmorat (Centre de mathématiques appliquées (CMA), École des Mines de Paris).

The identification of filters modeled by an electrical circuit that was developed by the team (see Section 4.5) led us to compute the electrical parameters of the underlying filter. This means finding a particular realization (A, B, C, D) of the model given by the rational approximation step. This 4-tuple must satisfy constraints that come from the geometry of the equivalent electrical network and translate into some of the coefficients in (A, B, C, D) being zero. Among the different geometries of coupling, there is one called "the arrow form" [57] which is of particular interest since it is unique for a given transfer function and is easily computed. The computation of this realization is the first step of RGC. Subsequently, if the target realization is not in arrow form, one can nevertheless show that it can be deduced from the arrow-form by a complex- orthogonal change of basis. In this case, RGC starts a local optimization procedure that reduces the distance between the arrow form and the target, using successive orthogonal transformations. This optimization problem on the group of orthogonal matrices is non-convex and has many local and global minima. In fact, there is not even uniqueness of the filter realization for a given geometry. Moreover, it is often relevant to know all solutions of the problem, because the designer is not even sure, in many cases, which one is being handled. The assumptions on the reciprocal influence of the resonant modes may not be equally well satisfied for all such solutions, hence some of them should be preferred for the design. Today, apart from the particular case where the arrow form is the desired form (this happens frequently up to degree 6) the RGC software provides no guarantee to obtain a single realization that satisfies the prescribed constraints. The software Dedale-HF (see Section 5.4), which is the successor of RGC, solves with guarantees this constraint realization problem.

5.3. PRESTO-HF

Participant: Fabien Seyfert [corresponding participant].

Status: Currently under development. A stable version is maintained.

PRESTO-HF: a toolbox dedicated to lowpass parameter identification for microwave filters http://www-sop.inria.fr/apics/Presto-HF. In order to allow the industrial transfer of our methods, a Matlab-based toolbox has been developed, dedicated to the problem of identification of low-pass microwave filter parameters. It allows one to run the following algorithmic steps, either individually or in a single shot:

- determination of delay components caused by the access devices (automatic reference plane adjustment),
- automatic determination of an analytic completion, bounded in modulus for each channel,
- rational approximation of fixed McMillan degree,
- determination of a constrained realization.

For the matrix-valued rational approximation step, Presto-HF relies on RARL2 (see Section 5.1), a rational approximation engine developed within the team. Constrained realizations are computed by the RGC software. As a toolbox, Presto-HF has a modular structure, which allows one for example to include some building blocks in an already existing software.

The delay compensation algorithm is based on the following strong assumption: far off the passband, one can reasonably expect a good approximation of the rational components of S_{11} and S_{22} by the first few terms of their Taylor expansion at infinity, a small degree polynomial in $1/s$. Using this idea, a sequence of quadratic convex optimization problems are solved, in order to obtain appropriate compensations. In order to check the previous assumption, one has to measure the filter on a larger band, typically three times the pass band.

This toolbox is currently used by Thales Alenia Space in Toulouse, Thales airborn systems and a license agreement has been recently negotiated with TAS-Espagna. XLIM (University of Limoges) is a heavy user of Presto-HF among the academic filtering community and some free license agreements are currently being considered with the microwave department of the University of Erlangen (Germany) and the Royal Military College (Kingston, Canada).

5.4. Dedale-HF

Participant: Fabien Seyfert [corresponding participant].

Status: Currently under development. A stable version is maintained.

Dedale-HF is a software dedicated to solve exhaustively the coupling matrix synthesis problem in reasonable time for the users of the filtering community. For a given coupling topology, the coupling matrix synthesis problem (C.M. problem for short) consists in finding all possible electromagnetic coupling values between resonators that yield a realization of given filter characteristics. Solving the latter problem is crucial during the design step of a filter in order to derive its physical dimensions as well as during the tuning process where coupling values need to be extracted from frequency measurements (see Figure 3).

`../../../../projets/apics/IMG/Vue_ensemble.png`

Figure 3. Overall scheme of the design and tuning process of a microwave filter.

Dedale-HF consists in two parts: a database of coupling topologies as well as a dedicated predictor-corrector code. Roughly speaking each reference file of the database contains, for a given coupling topology, the complete solution to the C.M. problem associated to particular filtering characteristics. The latter is then used as a starting point for a predictor-corrector integration method that computes the solution to the C.M. problem of the user, *i.e.* the one corresponding to user-specified filter characteristics. The reference files are computed off-line using Groebner basis techniques or numerical techniques based on the exploration of a monodromy group. The use of such a continuation technique combined with an efficient implementation of the integrator produces a drastic reduction, by a factor of 20, of the computational time.

Access to the database and integrator code is done via the web on http://www-sop.inria.fr/apics/Dedale/WebPages. The software is free of charge for academic research purposes: a registration is however needed in order to access full functionality. Up to now 90 users have registered world wide (mainly: Europe, U.S.A, Canada and China) and 4000 reference files have been downloaded.

A license of this software has been sold end of 2011, to TAS-Espagna, in order for it to tune filters with topologies having multiple solutions. The use of Dedale-HF is here coupled with that of Presto-HF.

5.5. easyFF

Participant: Fabien Seyfert.

Status: A stable version is maintained.

This software has been developed by Vincent Lunot (Taiwan Univ.) during his PhD. He still continues to maintain it.

EasyFF is a software dedicated to the computation of complex, and in particular multi-band, filtering functions. The software takes as input, specifications on the modulus of the scattering matrix (transmission and rejection), the filter's order and the number of transmission zeros. The output is an "optimal" filtering characteristic in the sense that it is the solution of an associated min-max Zolotarev problem. Computations are based on a Remez-type algorithm (if transmission zeros are fixed) or on linear programming techniques if transmission zeros are part of the optimization [10].

5.6. FindSources3D

Participant: Juliette Leblond [corresponding participant].

Status: Currently under development. A stable version is maintained.

This software is developed in collaboration with Maureen Clerc and Théo Papadopoulo from the Athena Project-Team, and with Jean-Paul Marmorat (Centre de mathématiques appliquées - CMA, École des Mines de Paris).

FindSources3D [2] is a software dedicated to source recovery for the inverse EEG problem, in 3-layer spherical settings, from point-wise data (see http://www-sop.inria.fr/apics/FindSources3D/). Through the algorithm described in [8] and Section 4.2 , it makes use of the software RARL2 (Section 5.1) for the rational approximation step in plane sections. The data transmission preliminary step ("cortical mapping") is solved using boundary element methods through the software OpenMEEG (its CorticalMapping features) developed by the Athena Team (see http://www-sop.inria.fr/athena/software/OpenMEEG/). A new release of FindSources3D is now available, which is being demonstrated and distributed to the medical team we maintain contact with (hosp. la Timone, Marseille). A further release is currently under development, due to the strong interest for this software by the German firm BESA GmbH (see http://www.besa.de/), involved in EEG software for research and clinical applications, and a deeper collaboration with this company has been started this year. Figure 4 shows the good results of a two sources distribution recovered by FindSources3D from potential values at electrodes on a sphere (scalp) generated by BESA's simulator, and then back to a more realistic head geometry. There, the achieved localization error is small enough, and FindSources3D provides suitable initial guess to heavier dedicated recovery tools, along with an estimation of the number of sources which may be incorporated to the software as an additional functionality (at the moment, the user is still involved in this estimation). Taking into account several time instants will be considered next.

5.7. Sollya

Participant: Sylvain Chevillard [corresponding participant].

Status: Currently under development. A stable version is maintained.

This software is developed in collaboration with Christoph Lauter (LIP6) and Mioara Joldeş (LAAS).

---

[2] CeCILL license, APP version 2.0 (2012): IDDN.FR.001.45009.001.S.A.2009.000.10000

../../../../projets/apics/IMG/bilateral_temporal_solution2.png

Figure 4. Recovered 2 sources by FindSources3D (courtesy of BESA).

Sollya is an interactive tool where the developers of mathematical floating-point libraries (libm) can experiment before actually developing code. The environment is safe with respect to floating-point errors, *i.e.* the user precisely knows when rounding errors or approximation errors happen, and rigorous bounds are always provided for these errors.

Among other features, it offers a fast Remez algorithm for computing polynomial approximations of real functions and also an algorithm for finding good polynomial approximants with floating-point coefficients to any real function. It also provides algorithms for the certification of numerical codes, such as Taylor Models, interval arithmetic or certified supremum norms.

It is available as a free software under the CeCILL-C license at http://sollya.gforge.inria.fr/.

ASPI Project-Team (section vide)

BACCHUS Team

5. Software and Platforms

5.1. AeroSol

Participants: Dragan Amenga-Mbengoué [Bacchus], Simon Delmas [Cagire], Damien Genet [Bacchus], Maxime Mogé [Cagire], Yann Moguen [Cagire], Francois Pellegrini [Bacchus], Vincent Perrier [Corresponding member], Francois Rué [Bacchus], Mario Ricchiuto [Bacchus].

The `AeroSol` software is jointly developed in teams Bacchus and Cagire. It is a high order finite element library written in C++. The code has been designed so as to allow for efficient computations, with continuous and discontinuous finite elements methods on hybrid and possibly curvilinear meshes. The work of the team Bacchus is focused on continuous finite elements methods, while the team Cagire is focused on discontinuous Galerkin methods. However, everything is done for sharing the largest part of code we can. More precisely, classes concerning IO, finite elements, quadrature, geometry, time iteration, linear solver, models and interface with `PaMPA` are used by both of the teams. This modularity is achieved by mean of template abstraction for keeping good performances. The distribution of the unknowns is made with the software `PaMPA`, developed within the team Bacchus and the team Castor.

This year some important features were added including : definition of CMake options for optimization and for using different compilers (GNU gcc, Intel icc, and IBM xlc) ; new element classes (lagrange and hierarchical orthogonal finite element basis for pyramids, Gauss Lagrange elements) ; implicit time integrators (backward Euler, Crank-Nicolson, and BDF from 2nd to 6th order) ; anisotropic diffusion models and (compressible) Navier-Stokes models ; debuggin by looging at memory traces with an interfacing with the PAPI library (tests have also been performed with VTUNE and TAU) ; improvements in schemes robustness and efficiency (Galerkin discretization of advection optimized by stocking most of the geometrical functions and finite elements computations, explicit and implicit version of the DG discretization of diffusion problems, implementation of Taylor-Galerkin stabilization and simplified SUPG stabilization) ; boundary conditions (time dependent, periodic, non reflecting) ; low Mach numerical fluxes for DG ; development of steady and unsteady tests related to all these new features.

5.2. COCA

Participants: Mario Ricchiuto [corresponding member], Gérard Vignoles, Gregory Perrot.

`COCA`(CodeOxydationCompositesAutocicatrisants) is a `Fortran 90` code for the simulation of the oxidation process in self-healing composites `COCA` solves the discrete finite element equations relative to the oxidation (chemistry) and flow (potential) models. Time integration is performed with an implicit approach (Backward Euler or second order backward differencing). The linear algebraic systems arising in the discretization are solved with the `MUMPS`library.

5.3. RealfluiDS

Participants: Dante de Santis, Gianluca Geraci, Pietro Marco Congedo, Rémi Abgrall [corresponding member].

`RealfluiDS` is a software dedicated to the simulation of inert or reactive flows. It is also able to simulate multiphase, multimaterial, MHD flows and turbulent flows (using the SA model). There exist 2D and 3D dimensional versions. The 2D version is used to test new ideas that are later implemented in the 3D one. This software implements the more recent residual distribution schemes. The code has been parallelized with and without overlap of the domains. The uncertainty quantification library `RobUQ`has been coupled to the software. A partitioning tool exists in the package, which uses `Scotch`. Recently, the code has been developed for taking into account real-gas effects, in order to use a whatever complex equation of state. Further developments concerning multiphase effects are under way.

5.4. MMG3D

Participants: Cécile Dobrzynski [corresponding member], Algiane Froehly.

`MMG3D` is a tetrahedral fully automatic remesher. Starting from a tetrahedral mesh, it produces quasi-uniform meshes with respect to a metric tensor field. This tensor prescribes a length and a direction for the edges, so that the resulting meshes will be anisotropic. The software is based on local mesh modifications and an anisotropic version of Delaunay kernel is implemented to insert vertices in the mesh. Moreover, `MMG3D` allows one to deal with rigid body motion and moving meshes. When a displacement is prescribed on a part of the boundary, a final mesh is generated such that the surface points will be moved according this displacement. `MMG3D` is used in particular in GAMMA for their mesh adaptation developments, but also at EPFL (maths department), Dassault Aviation, Lemma (a french SME), etc. `MMG3D` can be used in `FreeFem++` (http://www.freefem.org), a free software which eases the solving of PDEs and in Gmsh (http://geuz.org/gmsh/). More details can be found on http://www.math.u-bordeaux1.fr/~dobj/logiciels/mmg3d.php.

A new version of `MMG3D` is under development. The big novelty of this version is the modification of the surface triangulation. A. Froehly, ingenieer in the FUI Rodin, is working on this new version.

5.5. ORComp

Participants: Pietro Marco Congedo [Corresponding member], Rémi Abgrall, Dante de Santis, Maria-Giovanna Rodio.

The `ORComp` platform is a simulation tool permitting to design an ORC cycle. Starting from the solar radiation, this plateform computes the cycle providing the best performance with optimal choices of the fluid and the operating conditions. It includes `RobUQ`, a simulation block of the ORC cycles, the `RealfluiDScode` for the simulation of the turbine and of the heat exchanger, the software `FluidProp` (developed at the University of Delft) for computing the fluid thermodynamic properties.

5.6. PaMPA

Participants: Cédric Lachat, François Pellegrini [Corresponding member], Cécile Dobrzynski, Hervé Guillard [PUMAS], Laurent Hascoët [Tropics].

`PaMPA` ("Parallel Mesh Partitioning and Adaptation") is a middleware library dedicated to the management of distributed meshes. Its purpose is to relieve solver writers from the tedious and error prone task of writing again and again service routines for mesh handling, data communication and exchange, remeshing, and data redistribution. It is based on a distributed data structure that represents meshes as a set of *entities* (elements, faces, edges, nodes, etc.), linked by *relations* (that is, computation dependencies).

`PaMPA` interfaces with `Scotch` for mesh redistribution, and with `MMG3D` for parallel remeshing of tetrahedral elements. Other sequential remeshers can be plugged-in, in order to handle other types of elements.

Following the PhD defense of Cédric Lachat last December, version `1.0` is about to be released publicly under the GPL license. This version allows users to declare distributed meshes, to declare values attached to the entities of the meshes (e.g. temperature attached to elements, pressures to the faces, etc.), to exchange values between overlapping entities located at the boundaries of subdomains assigned to different processors, to iterate over the relations of entities (e.g. iterate over the faces of elements), to remesh in parallel the areas of a mesh that need to be emeshed, and to redistribute evenly the remeshed mesh across the processors of the parallel architecture.

`PaMPA` is already used as the data structure manager for two solvers being developed at Inria: `Plato` (team PUMAS) and `AeroSol`(teams BACCHUS and CAGIRE).

5.7. RobUQ

Participants: Pietro Marco Congedo [Corresponding member], Rémi Abgrall, Gianluca Geraci, Maria Giovanna Rodio, Kunkun Tang, Julie Tryoen.

The `RobUQ` platform has been conceived to solve problems in uncertainty quantification and robust design. It includes the optimization code `ALGEN`, and the uncertainty quantification code `NISP`. It includes also some methods for the computation of high-order statistics, efficient strategies for robust optimization, the Simplex2 method. Some methods are developed in partnership with the Stanford University (in the framework of the associated team AQUARIUS). Other methods are developed in the context of ANR `UFO`.

5.8. Scotch

Participants: François Pellegrini [corresponding member], Sébastien Fourestier.

parallel graph partitioning, parallel static mapping, parallel sparse matrix block ordering, graph repartitioning, mesh partitioning.

`Scotch` (http://www.labri.fr/~pelegrin/scotch/) is a software package for parallel and sequential sparse matrix ordering, parallel and sequential graph partitioning, as well as sequential static mapping and remapping, without and with fixed vertices, and mesh and hypergraph partitioning.

The initial purpose of `Scotch` was to compute high-quality static mappings of valuated graphs representing parallel computations onto target architectures of arbitrary topologies. This allows the mapper to take into account the topology and heterogeneity of the target architecture in terms of processor speed and link bandwidth. This feature, which was meant for the NUMA machines of the 1980's, has not been widely used in the past because machines in the 1990's became UMA again thanks to hardware advances. Now, architectures become NUMA again, and these features are regaining popularity.

The `Scotch` package consists of two libraries: the sequential `Scotch` library, and the parallel `PT-Scotch` library (for "*Parallel Threaded Scotch*") that operates according to the distributed memory paradigm, using MPI. `Scotch` was the first full 64-bit implementation of a general purpose graph partitioner.

Version `6.0`, released on December 2012, corresponding to the 20th anniversary of `Scotch`, offers many new features: static mapping with fixed vertices, static remapping, and static remapping with fixed vertices. Several critical algorithms of the formerly strictly sequential `Scotch` library can now run in a multi-threaded way. All of these features, which exist only in the sequential version, will be available to the parallel `PT-Scotch` library in the upcoming release `6.1`.

`Scotch` has been integrated in numerous third-party software, which indirectly contribute to its diffusion. It is natively available in several Linux and Unix distributions, as well as on some vendors platforms (SGI, etc).

5.9. SLOWS

Participants: Mario Ricchiuto [corresponding member], Andrea Filippini.

`SLOWS` ("*Shallow-water fLOWS*") is a `C`-platform allowing the simulation of free surface shallow water flows with friction. Arbitrary bathymetries are allowed, defined either by some complex piecewise analytical expression, or by xyz data files, the classical Manning model for friction is used, and an Exner model is implemented for sediment transport. For non-hydrostatic propagation the enhanced Boussinesq equations of Madsen and Sorensen are used. The equations are discretized with a residual based approach which is an adaptation of the schemes developed for aeronautics applications. Due to the inherent unsteadiness of these flows, the time discretization plays an important role. Three different approaches are available, based on conditionally depth-positivity preserving implicit schemes, or on conditionally depth-positivity preserving genuinely explicit discretizations, or on an unconditionally depth-positivity preserving space-time approach. Newton and frozen Newton loops are used to solve the implicit nonlinear equations. The linear algebraic systems arising in the discretization are solved with the `MUMPS`library.

5.10. Nomesh

Participants: Cécile Dobrzynski [corresponding member], Algiane Froehly.

`Nomesh` is a software allowing the generation of third order curved simplicial meshes. Starting from a "classical" mesh with straight elements composed by triangles and/or tetrahedra, we are able to curve the boundary mesh. Starting from a mesh with some curved elements, we can verify if the mesh is valid, that means there is no crossing elements and only positive Jacobian. If the curved mesh is non valid, we modify it using linear elasticity equations until having a valid curved mesh.

BIPOP Project-Team

5. Software and Platforms

5.1. Nonsmooth dynamics: Siconos

Participants: Vincent Acary, Maurice Brémond, Olivier Huber, Franck Pérignon.

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

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

1. SICONOS/NUMERICS This library contains a set of numerical algorithms, already well identified, to solve non smooth dynamical systems. This library is written in low-level languages (C,F77) in order to ensure numerical efficiency and the use of standard libraries (Blas, Lapack, ...)
2. SICONOS/KERNEL This module is an object-oriented structure (C++) for the modeling and the simulation of abstract dynamical systems. It provides the users with a set of classes to describe their nonsmooth dynamical system (dynamical systems, intercations, nonsmooth laws, ...) and to perform a numerical time integration and solving.
3. SICONOS/FRONT-END. This module is mainly an auto-generated wrapper in Python which provides a user-friendly interface to the Siconos libraries. A scilab interface is also provided in the Front-End module.
4. SICONOS/CONTROL This part is devoted to the implementation of control strategies of non smooth dynamical systems.
5. SICONOS/MECHANICS. This part is dedicated to the modeling and the simulation of multi-body systems with 3D contacts, impacts and Coulomb's friction. It uses the Siconos/Kernel as simulation engine but relies on a industrial CAD library (OpenCascade and pythonOCC) to deal with complex body geometries and to compute the contact locations and distances between B-Rep description and on Bullet for contact detection between meshes.

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

5.2. Optimization

Participant: Claude Lemaréchal.

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

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

5.2.1. Code M1QN3

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

5.2.2. Code M2QN1

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

5.2.3. Code N1CV2

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

5.2.4. Modulopt

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

5.3. Simulation of fibrous materials subject to frictional contact

5.3.1. MECHE: Modeling Entangling within Contacting hair fibErs

Participants: Florence Bertails-Descoubes, Gilles Daviet, Alexandre Derouet-Jourdan, Romain Casati, Laurence Boissieux.

The software MECHE was essentially developed during the MECHE ADT (2009-2011, research engineer: Gilles Daviet), for simulating the dynamics of assemblies of thin rods (such as hair), subject to contact and friction. Currently, this software is extensively used by two PhD students (A. Derouet-Jourdan and R. Casati) and continues to be enriched with new rod models and inversion modules. This software combines a panel of well-accepted models for rods (ranging from reduced coordinates to maximal coordinates models, and including models recently developed by some members of the group) with classical as well as innovative schemes for solving the problem of frictional contact (incorporating the most recent results of the group, as well as the new contact solver we published in [8]). The aim of this software is twofold: first, to compare and analyze the performance of nonsmooth schemes for the frictional contact problem, in terms of realism (capture of dry friction, typically), robustness, and computational efficiency. A first study of this kind was conducted in 2010-2011 onto the different rod models that were available in the software. New studies are planned for evaluating further rod models. Second, we believe such a software will help us understand the behavior of a fibrous material (such as hair) through virtual experiments, thanks to which we hope to identify and understand some important emergent phenomena. A careful validation study against experiments started to be conducted in 2011 in collaboration with physicists from L'Oréal. Once this discrete elements model will be fully validated, our ultimate goal would be to build a continuous macroscopic model for the hair medium relying on nonsmooth laws. The core of this software was transferred to L'Oréal in 2011, and to AGT Digital in early 2013, by Gilles Daviet and Florence Bertails-Descoubes. It was also used for generating a number of simulations supporting at least 4 of our research publications.

5.3.2. Cloc: super-space clothoids

Participants: Romain Casati, Florence Bertails-Descoubes.

This software implements the super-space clothoid model published this year in [25]. This model consists of a new dynamic rod primitive relying upon high-order elements with a linear curvature (clothoidal arcs). The source code of this software is distributed from our webpages from December 2013, based on a dual licensing policy: a free GPLv.3 license, mainly dedicated to academics; and a commercial license, mainly dedicated to industry.

5.3.3. APPROCHE: APPROximate Curves with HElices

Participants: Alexandre Derouet-Jourdan, Florence Bertails-Descoubes.

APPROCHE is a software that implements the 3d floating tangents algorithm published. The algorithm takes as input a set of curves, either represented as splines or sequences of points, and fits each curve to a C^1-smooth piecewise helix. This software has been transferred to L'Oréal in December 2013 and some source code will be made freely available to academics under the GPLv.3 licence.

CAD Team (section vide)

CAGIRE Team

5. Software and Platforms

5.1. AeroSol

Participants: Dragan Amenga-Mbengoué [Bacchus], Simon Delmas [Cagire], Damien Genet [Bacchus], Maxime Mogé [Cagire], Yann Moguen [Cagire], Francois Pellegrini [Bacchus], Vincent Perrier [Cagire, correspondant], Francois Rué [Bacchus], Mario Ricchiuto [Bacchus].

The software AeroSol is jointly developed in the team Bacchus and the team Cagire. It is a high order finite element library written in C++. The code design has been carried for being able to perform efficient computations, with continuous and discontinuous finite elements methods on hybrid and possibly curvilinear meshes.

The work of the team Bacchus is focused on continuous finite elements methods, while the team Cagire is focused on discontinuous Galerkin methods. However, everything is done for sharing the largest part of code we can. More precisely, classes concerning IO, finite elements, quadrature, geometry, time iteration, linear solver, models and interface with PaMPAare used by both of the teams. This modularity is achieved by mean of template abstraction for keeping good performances.

The distribution of the unknowns is made with the software PaMPA, developed within the team Bacchus and the team Castor.

This year, Simon Delmas and Yann Moguen were recruited within the team Cagire. Their respective development, low Mach solver for compressible flows and turbulence injection boundary conditions are performed in the library Aerosol. At the end of 2012, Aerosol had the following features

- **development environment** use of CMake for compilation, CTest for automatic tests and memory checking, lcov and gcov for code coverage reports. Development of a CDash server for collecting the unitary tests and the memory checking. Beginning of the development of an interface for functional tests.
- **In/Out** link with the XML library for handling with parameter files. Reader for GMSH, and writer on the VTK-ASCII legacy format (cell and point centered). Parallel GMSH reader, XML paraview files on unstructured meshes (vtu) and parallel XML based files (pvtu).
- **Quadrature formula** up to 11th order for Lines, Quadrangles, Hexaedra, Pyramids, Prisms, up to 14th order for tetrahedron, up to 21st order for triangles. Gauss-Lobatto type quadrature formula for lines, triangles, quadrangles and hexaedra.
- **Finite elements** up to fourth degree for Lagrange finite elements and hierarchical orthogonal finite element basis (with Dubiner transform on simplices) on lines, triangles, quadrangles, tetrahedra, prisms and hexaedra. Finite element basis that are interpolation basis on Gauss-Legendre points for lines, quadrangles, and hexaedra.
- **Geometry** elementary geometrical functions for first order lines, triangles, quadrangles, prisms, tetrahedra and hexaedra.
- **Time iteration** explicit Runge-Kutta up to fourth order, explicit Strong Stability Preserving schemes up to third order. Optimized CFL time schemes: SSP(2,3) and SSP(3,4). CFL time stepping.
- **Linear Solvers** link with the external linear solver UMFPack, PETSc and MUMPS. Internal solver for diagonal matrices.
- **Memory handling** discontinuous and continuous, sequential and parallel discretizations based on PaMPA for generic meshes.

- **Models** Perfect gas Euler system, real gas Euler system (template based abstraction for a generic equation of state), scalar advection, Waves equation in first order formulation, generic interface for defining space-time models from space models.
- **Numerical schemes** continuous Galerkin method for the Laplace problem (up to fifth order) with non consistent time iteration or with direct matrix inversion. Discontinuous Galerkin methods for hyperbolic systems. SUPG and Residual disribution schemes.
- **Numerical fluxes** centered fluxes, exact Godunov' flux for linear hyperbolic systems, and Lax-Friedrich flux.
- **Parallel computing** Mesh redistribution, computation of Overlap with PaMPA. collective asynchronous communications (PaMPA based). Tests on the cluster Avakas from MCIA, and on Mésocentre de Marseille, and PlaFRIM.
- **C++/Fortran interface** Tests for binding fortran with C++.

This year, the following features were added

- **development environment** Definition of CMake options for optimization and for using different compilers. Currently, the following compilers have been tested: GNU gcc, Intel icc, and IBM xlc. Aerosol can now be linked with HDF5, PAPI, and can use different BLAS implementations like eigen or MKL.
- **In/Out** Point centered visualization for discontinuous approximations. XML binary output for Paraview was added. The link with HDF5 was added for parallel IO for defining XDMF format. A geometrical pre-partitioning was developped for reducing the size ofthe parallel graph in the parallel mesh reading.
- **Pyramids** Mesh reader, Lagrange and hierarchical orthogonal finite element basis were added for pyramids. Geometrical functions for linear pyramids were also added.
- **Finite element** Gauss Lagrange finite element basis (order 1 and 2) for triangles.
- **Time iteration** the following implicit integration schemes were added: backward Euler, Crank-Nicolson, and BDF from 2nd to 6th order.
- **Linear Solvers** Interface with PETSc was tested on a parallel environment. An in house block diagonal solver was developped.
- **Memory handling** Aerosol can now work on hybrid meshes.
- **Models** the generic model interface supports now diffusive models. Anisotropic diffusion and (compressible) Navier-Stokes models were added.
- **Instrumentation** Aerosol can give some traces on memory consumption/problems with an interfacing with the PAPI library. Tests have also been performed with VTUNE and TAU.
- **Parallel computing** Tests were performed on the clusters Pyrene (Université de Pau), poincaré (Maison de la Simulation), and on the Tier-1 cluster Turing (IDRIS).
- **Numerical schemes** The DG discretization of advection problems was optimized by stocking most of the geometrical functions and finite elements computations , and by using BLAS implementations for linear computations. Implicit versions of the DG discretization of advection problems. Development of explicit and implicit version of the DG discretization of diffusion problems. Time dependent boundary conditions, periodic boundary conditions, non reflecting boundary conditions. Development of low Mach numerical fluxes, and development of stationary and unstationary tests for this kind of problem.

CALVI Project-Team

5. Software and Platforms

5.1. SeLaLib

Participants: Aurore Back, Raphaël Blanchard, Edwin Chacon Golcher, Samuel de Santis, Aliou Diouf, Pierre Navaro, Morgane Bergot, Emmanuel Frénod, Virginie Grandgirard, Adnane Hamiaz, Philippe Helluy, Sever Hirstoaga, Michel Mehrenberger, Laurent Navoret, Nhung Pham, Eric Sonnendrücker.

Under the 'Fusion' large scale initiative, we have continued our work in the development of the ADT Selalib (the Semi-Lagrangian Library), now finishing its third year. This library provides building blocks for the development of numerical simulations for the solution of the fundamental equation of plasma physics: the Vlasov equation. In this context we have continued to add new modules improved interfaces and implemented 'continuous integration' software development techniques to improve code robustness and portability. Furthermore, we continue to involve other researchers within France and abroad to aid in the further development of this software product.

One of the aims of the ADT 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.

This year many developments have incorporated into Selalib: semi-Lagrangian solvers on curvilinear grids, new models, new fully-Eulerian solvers, new linear solvers for the Poisson or quasineutrality equation. More details are given in the corresponding sections, because we always try to test our new algorithms within Selalib.

Selalib is available on the Inria Forge http://selalib.gforge.inria.fr/

5.2. CLAC

Participants: Philippe Helluy, Michel Massaro, Thomas Strub.

CLAC is a generic Discontinuous Galerkin solver, written in C/C++, based on the OpenCL and MPI frameworks. CLAC means “Conservation Laws Approximation on many Cores”.

It is clear now that a future supercomputer will be made of a collection of thousands of interconnected multicore processors. Globally it appears as a classical distributed memory MIMD machine. But at a lower level, each of the multicore processors is itself made of a shared memory MIMD unit (a few classical CPU cores) and a SIMD unit (a GPU). When designing new algorithms, it is important to adapt them to this architecture. Our philosophy will be to program our algorithms in such a way that they can be run efficiently on this kind of computers. Practically, we will use the MPI library for managing the high level parallelism, while the OpenCL library will efficiently operate the low level parallelism.

We have invested for several years now into scientific computing on GPUs, using the open standard OpenCL (Open Computing Language). We were recently awarded a prize in the international AMD OpenCL innovation challenge thanks to an OpenCL two-dimensional Vlasov-Maxwell solver that fully runs on a GPU. OpenCL is a very interesting tool because it is an open standard now available on almost all brands of multicore processors and GPUs. The same parallel program can run on a GPU or a multicore processor without modification.

CLAC is also a joint project with a Strasbourg small company, AxesSim, which develops software for electromagnetic simulations.

Because of the envisaged applications of CLAC, which may be either academic or commercial, it is necessary to conceive a modular framework. The heart of the library is made of generic parallel algorithms for solving conservation laws. The parallelism can be both fine-grained (oriented towards GPUs and multicore processors) and coarse-grained (oriented towards GPU clusters). The separate modules allow managing the meshes and some specific applications. In this way, it is possible to isolate parts that should be protected for trade secret reasons. The open source part of CLAC will be made freely available on the web later on. We have made an APP deposit of the first version of CLAC in October 2012. The versioning of CLAC project is also registered in the Inria Forge http://clac.gforge.inria.fr.

CLASSIC Project-Team (section vide)

COMMANDS Project-Team

4. Software and Platforms

4.1. Bocop

Participants: Pierre Martinon [corresponding author], Daphné Giorgi, Joseph Frédéric Bonnans.

Web page: http://bocop.org

The Bocop project aims to develop an open-source toolbox for solving optimal control problems, with collaborations with industrial and academic partners. Optimal control (optimization of dynamical systems governed by differential equations) has numerous applications in transportation, energy, process optimization, and biology. The software reuses some packages from the COIN-OR library, in particular the well-known nonlinear programming solver Ipopt, features a user-friendly interface and can be deployed on Windows / Mac / Linux.

The project is supported by Inria with the recruitment of Vincent Grelard as developer in 2010-2012, and then Daphné Giorgi since October 2012. The first prototype was released at the end of 2011, Bocop is currently at version 1.1.4 and has been downloaded more than 700 times. The software was first succesfully tested on several academic problems, see [55] available on http://bocop.org. Starting in 2012, several research collaborations were initiated in fields such as bio-reactors for energy production ([30], [27]), swimming micro-robots ([39]), and quantum control for medical imaging ([25]). Bocop was also featured during our participation in the Imatch "Optimisation and Control" in october, which resulted in a contract with the startup Safety Line (aeronautics).

Bocop auto-assessment according to Inria notice: A3up4, SO3, SM3, EM3up4, SDL4up5

../../../../projets/commands/IMG/bocop-screen-init.png

../../../../projets/commands/IMG/bocop-screen-b

Figure 1. BOCOP

4.2. CollAv

Participants: Hasnaa Zidani [corresponding author], Olivier Bokanowski, Anna Désilles.

This software simulates the evolution of controlled dynamical systems (possibly under uncertainties). The numerical algorithm here is based on HJB or viability approaches, and allows the design of optimal planning strategies (according to a criterion determined by the user: time, energy, ...). It also provides conflict resolution and avoidance of collisions with fixed or moving obstacles. So far, the software is used in collaboration with DGA for avoidance collision of UaVs, and by Volkswagen in some studies related to collision avoidance of cars.

4.3. OCOPHyS

Participant: Hasnaa Zidani [corresponding author].

This is a software for optimisation-based controller design for operating in different regimes or modes of operation. The software can be used, for example, to determine the optimal management for hybrid vehicles or hybrid engines with multiple energy sources. However, the methods used in software are still quite general and can be used in many applications.

4.4. BiNoPe-HJ

Participants: Hasnaa Zidani [corresponding author], Olivier Bokanowski, Anna Désilles.

Web page: http://www.ensta-paristech.fr/ zidani/BiNoPe-HJ

This project aims at developping sequential and parallel MPI/openMP C++ solvers for the approximation of Hamilton-Jacobi-Bellman (HJB) equations in a d-dimensional space. The main goal is to provide an HJB solvers that can work in dimension d (limited by the machine's capacity). The solver outputs can be visualized with Matlab or Paraview (via VTK files).

The development of the HJB Solver has been initiated under a partnership between COMMANDS and the SME HPC-project in the period between December 2009 to November 2011. Currently, it is still maintained and improved by COMMANDS.

In 2012, two versions were released:

- HJB-SEQUENTIAL-REF: sequential version that can run on any machine
- HJB-PARALLEL-REF: parallel version that can run only on multi-core architectures.

4.5. Shoot

Participant: Pierre Martinon [corresponding author].

Shoot was designed for the resolution of optimal control problems via indirect methods (necessary conditions, Pontryagin's Maximum Principle). Such methods transform the original problem into finding a zero of a certain shooting function. The package offers several choices of integrators and solvers, and can handle control discontinuities. Features also include the use of variational equations to compute the Jacobian of the shooting function, as well as homotopy and grid shooting techniques for easier initialization. Shoot is an academic software, and was used during several research contracts with the CNES (french space agency).

CORIDA Project-Team

4. Software and Platforms

4.1. Simulation of viscous fluid-structure interactions

Participants: Takeo 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 developped by ALICE.

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

4.2. Fish locomotion in perfect fluids with potential flow

Participants: Alexandre Munnier [correspondant], Marc Fuentes, Bruno Pinçon.

SOLEIL is a Matlab suite to simulate the self-propelled swimming motion of a single 3D swimmer immersed in a potential flow. The swimmer is modeled as a shape-changing body whose deformations can be either prescribed as a function of time (simulation of the direct swimming problem) or computed in such a way that the swimmer reaches a prescribed location (control problem). For given deformations, the hydrodynamical forces exerted by the fluid on the swimmer are expressed as solutions of 2D integral equations on the swimmer's surface, numerically solved by means of a collocation method.

SOLEIL is free, distributed under licence GPL v3. More details are available on the project web page http://soleil.gforge.inria.fr/.

The next step of SOLEIL (under progress) is to take into account a fluid whose flow is governed by Stokes equations.

- Version: 0.1
- Programming language:Matlab/C++

4.3. SUSHI3D : SimUlations of Structures in Hydrodynamic Interactions

Participants: Marc Fuentes, Jean-François Scheid, Jérémy Sinoir, Takéo Takahashi, Rhaleb Zayer.

SUSHI3D is a 3D solver for numerical simulations of Fluid/Structures Interactions. The Navier-Stokes equations are coupled with the dynamics of immersed bodies which can be either rigid or deformable. The deformable body case is handled and designed for fish-swimming. The numerical method used to solve the full differential system is based on a Lagrange-Galerkin method with finite elements.

- Version: 1.0
- Programming language:Matlab/C++

CQFD Project-Team

5. Software and Platforms

5.1. Package edrGraphicalTools

This R package gives graphical tools for selecting the number of slices and the dimension of the model in SIR and SAVE approaches. It also provides the estimation of the reduction dimension subspace and the non parametric estimation of the link function using smoothing techniques. The package is available via the link http://cran.r-project.org/web/packages/edrGraphicalTools/index.html.

5.2. Package ClustOfVar

This R package is dedicated to cluster analysis of a set of variables. Variables can be quantitative, qualitative or a mixture of both. A new version 0.8 of the package is available since december 2013 via the link http://cran.r-project.org/web/packages/ClustOfVar/index.html. This version provides now a function to predict values of new observations on the synthetic variables of the clusters. This new function was used for supervised classification and variable selection in gene expressions data [42].

5.3. Package PCAmixdata

This package is dedicated to factorial analysis and rotation of quantitative data, qualitative data, or mixed data. The PCAMIX method, proposed in this package includes the ordinary principal component analysis (PCA) and multiple correspondence analysis (MCA) as special cases. Orthogonal varimax rotation of the principal components of PCAMIX is also implemented in this package.This year, a new method has been developped for Multiple Factorial Analysis in case of mixtures of quantitative and qualitative variables within groups. It was implemented in the package and presented to the 2èmes Rencontres R in Lyon [44], and to the 45èmes Journées de Statistique in Toulouse [41].

DEFI Project-Team

5. Software and Platforms

5.1. RODIN

Participant: Grégoire Allaire [correspondant].

In the framework of the RODIN project we continue to develop with our software partner ESI the codes Topolev and Geolev for topology and geometry shape optimization of mechanical structures using the level set method.

5.2. FreeFem++ Toolboxes

5.2.1. Shape optimization toolbox in FreeFem++

Participants: Grégoire Allaire, Olivier Pantz.

We propose several FreeFem++ routines which allow the users to optimize the thickness, the geometry or the topology of elastic structures. All examples are programmed in two space dimensions. These routines have been written by G. Allaire, B. Boutin, C. Dousset, O. Pantz. A web page of this toolbox is available at http://www.cmap.polytechnique.fr/~allaire/freefem_en.html.

We also have written a C++ code to solve the Hamilton Jacoby equation used in the Level-set shape optimization method. This code has been linked with FreeFem++ routines.

5.2.2. Eddy current problems

Participants: Zixian Jiang, Kamel Riahi.

We developed a FreeFem++ toolbox that solves direct and inverse problems for an axisymmetric and 3D eddy current problems related to non destructive testing of deposits on the shell side of PWR fuel tubes. For the 3-D version, one can refer to http://www.cmap.polytechnique.fr/~riahi

5.2.3. Contact managements

Participant: Olivier Pantz.

We have developed a toolbox running under Freefem++ in order to take into account the non-intersection constraints between several deformable bodies. This code has been used to treat contacts between red blood cells in our simulations, but also between genuine non linear elastic structure. It can handle both contacts and self-contacts.
Moreover, a toolbox based on the Penalization method has also been developed.

5.2.4. De-Homogenization

Participant: Olivier Pantz.

We have developed a code under Freefem++ that implements our De-Homogenization method. It has been used to solve the compliance minimization problem of the compliance of an elastic shape. In particular, it enables us to recover well known optimal Michell's trusses for shapes of low density.

5.3. Scilab and Matlab Toolboxes

5.3.1. Shape optimization toolbox in Scilab

Participant: Grégoire Allaire [correspondant].

Together with Georgios Michailidis, we improved a Scilab toolbox for 2-d shape and topology optimization by the level set method which was originally produced by Anton Karrman and myself. The routines, a short user's manual and several examples are available on the web page: http://www.cmap.polytechnique.fr/~allaire/levelset_en.html

5.3.2. *Conformal mapping method*

Participant: Houssem Haddar [correspondant].

This Scilab toolbox is dedicated to the resolution of inverse 2-D electrostatic problems using the conformal mapping method introduced by Akdumann, Kress and Haddar. The toolbox treats the cases of a simply connected obstacle with Dirichlet, Neumann or impedance boundary conditions or a simply connected inclusion with a constant conductivity. The latest development includes the extension of the method to the inverse scattering problem at low frequencies as introduced by Haddar-Kress (2012).

5.4. Sampling methods for inverse problems

5.4.1. *Samplings-2d*

Participant: Houssem Haddar [correspondant].

This software is written in Fortran 90 and is related to forward and inverse problems for the Helmholtz equation in 2-D. It includes three independent components. The first one solves to scattering problem using integral equation approach and supports piecewise-constant dielectrics and obstacles with impedance boundary conditions. The second one contains various samplings methods to solve the inverse scattering problem (LSM, RGLSM(s), Factorization, MuSiC) for near-field or far-field setting. The third component is a set of post processing functionalities to visualize the results
See also the web page http://sourceforge.net/projects/samplings-2d/.

- License: GPL
- Type of human computer interaction: sourceforge
- OS/Middelware: Linux
- Programming language: Fortran
- Documentation: fichier

5.4.2. *Samplings-3d*

Participant: Houssem Haddar [correspondant].

This software is written in Fortran 90 and is related to forward and inverse problems for the Helmholtz equation in 3-D. It contains equivalent functionalities to samplings-2d in a 3-D setting.

5.4.3. *Time domain samplings-2d*

Participant: Houssem Haddar [correspondant].

This software is written in Fortran 90 and is related to forward and inverse problems for the time dependent wave equation in 2-D. The forward solver is based on a FDTD method with PMLs. The inverse part is an implementation of the linear sampling method in a near field setting and the factorization method in a far field setting.

5.4.4. *Factorization Method for EIT*

Participant: Giovanni Migliorati.

We developed a numerical code that implements the Factorization Method applied to the Continuous Model, in the framework of Electrical Impedance Tomography featuring an inhomogeneous background. The numerical scheme relies on the approximation by the finite element method of the solution to the dipole-like Neumann boundary-value problem. Two regularization techniques are implemented, i.e. the Tikhonov regularization embedding Morozov principle, and the classical Picard Criterion. The code now supports the case of piecewise-wise constant by unknown background.

5.5. BlochTorreyPDESolver

Participants: Jing-Rebecca Li [correspondant], Dang Van Nguyen.

We developed two numerical codes to solve the multiple-compartments Bloch-Torrey partial differential equation in 2D and 3D to simulate the water proton magnetization of a sample under the influence of diffusion-encoding magnetic field gradient pulses.

We coupled the spatial discretization with an efficient time discretization adapted to diffusive problems called the (explicit) Runge-Kutta-Chebyshev method.

The version of the code using Finite Volume discretization on a Cartesian grid is complete (written by Jing-Rebecca Li). The version of the code using linear Finite Elements discretization is complete (written by Dang Van Nguyen and Jing-Rebecca Li).

See the web page http://www.cmap.polytechnique.fr/~jingrebeccali/ for more details.

DISCO Project-Team

5. Software and Platforms

5.1. OreModules

Participants: Alban Quadrat [correspondent], Daniel Robertz [Univ. Aachen], Frédéric Chyzak [Inria Rocquencourt, Algorithms Project].

The OREMODULES package [92], based on the commercial Maple package Ore_algebra [93], is dedicated to the study of linear multidimensional systems defined over certain Ore algebras of functional operators (e.g., ordinary or partial differential systems, time-delay systems, discrete systems) and their applications in mathematical systems theory, control theory and mathematical physics. OREMODULES is original because it combines the recent developments of the Gröbner bases over some noncommutative polynomial rings [100], [102] and new algorithms of algebraic analysis in order to effectively check classical properties of module theory (e.g., existence of a non-trivial torsion submodule, torsion-freeness, reflexiveness, projectiveness, stably freeness, freeness), it gives their system-theoretical interpretations (existence of autonomous elements or successive parametrizations, existence of minimal/injective parametrizations or Bézout equations) [109], [108], [91] and it computes important tools of homological algebra (e.g., (minimal) free resolutions, split exact sequences, extension functors, projective or Krull dimensions, Hilbert power series). The abstract language of homological algebra used in the algebraic analysis approach carries over to the implementations in OREMODULES: up to the choice of the domain of functional operators which occurs in a given system, all algorithms are stated and implemented in sufficient generality such that linear systems defined over the Ore algebras developed in the Ore_algebra package are covered at the same time. Applications of the OREMODULES package to mathematical systems theory, control theory and mathematical physics are illustrated in a large library of examples. The binary of the package is freely available at http://wwwb.math.rwth-aachen.de/OreModules/.

A Mathematica version of the OREMODULES package is in development. It is being developed by Maris Tõnso (Institute of Cybernetics, University of Tallinn), Thomas Cluzeau (ENSIL, University of Limoges) and A. Quadrat within the PHC Parrot project CASCAC. The Mathematica version of the OREMODULES package is based on the implementation of Gröbner bases over Ore algebras available in the Mathematica HolonomicFunctions package developed by Christoph Koutschan.

5.2. Stafford

Participants: Alban Quadrat [correspondent], Daniel Robertz [Univ. Aachen].

The STAFFORD package of OREMODULES [92] contains an implementation of two constructive versions of Stafford's famous but difficult theorem [124] stating that every ideal over the Weyl algebra $A_n(k)$ (resp., $B_n(k)$) of partial differential operators with polynomial (resp., rational) coefficients over a field k of characteristic 0 (e.g., $k=\mathbb{Q}$, $\mathbb{R}$) can be generated by two generators. Based on this implementation and algorithmic results developed in [119] by the authors of the package, two algorithms which compute bases of free modules over the Weyl algebras $A_n(\mathbb{Q})$ and $B_n(\mathbb{Q})$ have been implemented. The rest of Stafford's results developed in [124] have recently been made constructive in [121] (e.g., computation of unimodular elements, decomposition of modules, Serre's splitting-off theorem, Stafford's reduction, Bass' cancellation theorem, minimal number of generators) and implemented in the STAFFORD package. The development of the STAFFORD package was motivated by applications to linear systems of partial differential equations with polynomial or rational coefficients (e.g., computation of injective parametrization, Monge problem, differential flatness, the reduction and decomposition problems and Serre's reduction problem). To our knowledge, the STAFFORD package is the only implementation of Stafford's theorems nowadays available. The binary of the package is freely available at http://wwwb.math.rwth-aachen.de/OreModules/.

5.3. QuillenSuslin

Participants: Alban Quadrat [correspondent], Anna Fabiańska [Univ. Aachen].

The QUILLEN-SUSLIN package [96] contains an implementation of the famous Quillen-Suslin theorem [123], [125]. In particular, this implementation allows us to compute bases of free modules over a commutative polynomial ring with coefficients in a field (mainly $\mathbb{Q}$) and in a principal ideal domain (mainly $\mathbb{Z}$). The development of the QUILLEN-SUSLIN package was motivated by different constructive applications of the Quillen-Suslin theorem in multidimensional systems theory [96] (e.g., the Lin-Bose conjectures, the computation of (weakly) left/right/doubly coprime factorizations of rational transfer matrices, the computation of injective parametrizations of flat linear multidimensional systems with constant coefficients, the reduction and decomposition problems, Serre's reduction problem). To our knowledge, the QUILLEN-SUSLIN package is the only implementation of the Quillen-Suslin theorem nowadays available. The binary of the package is freely available at http://wwwb.math.rwth-aachen.de/QuillenSuslin.

5.4. OreMorphisms

Participants: Alban Quadrat [correspondent], Thomas Cluzeau [ENSIL, Univ. Limoges].

The OREMORPHISMS package [95] of OREMODULES [91] is dedicated to the implementation of homological algebraic tools such as the computations of homomorphisms between two finitely presented modules over certain noncommutative polynomial algebras (Ore algebras), of kernel, coimage, image and cokernel of homomorphisms, Galois transformations of linear multidimensional systems and idempotents of endomorphism rings. Using the packages STAFFORD and QUILLEN-SUSLIN, the factorization, reduction and decomposition problems can be constructively studied for different classes of linear multidimensional systems. Many linear systems studied in engineering sciences, mathematical physics and control theory have been factorized, reduced and decomposed by means of the OREMORPHISMS package. The binary of the package is freely available at http://www-sop.inria.fr/members/Alban.Quadrat/OreMorphisms/index.html.

A Mathematica version of the OREMORPHIMS package is in development. It is being developed by Maris Tõnso (Institute of Cybernetics, University of Tallinn), Thomas Cluzeau (ENSIL, University of Limoges) and Alban Quadrat within the PHC Parrot project CASCAC. The Mathematica version of the OREMORPHIMS package is based on the implementation of Gröbner bases over Ore algebras available in the Mathematica HolonomicFunctions package developed by Christoph Koutschan.

5.5. JanetMorphisms

Participants: Alban Quadrat [correspondent], Thomas Cluzeau [ENSIL, Univ. Limoges], Daniel Robertz [Univ. Aachen].

The JANETMORPHISMS package is dedicated to a new mathematic approach to quasilinear systems of partial differential equations (e.g., Burger's equation, shalow water equations, Euler equations of a compressible fluid) based on algebraic analysis and differential algebra techniques [90]. This package computes symmetries, first integrals of motion, conservation laws, study Riemann invariants... The JANETMORPHISMS package is based on the `Janet` package (http://wwwb.math.rwth-aachen.de/Janet/).

5.6. PurityFiltration

Participant: Alban Quadrat [correspondent].

The PURITYFILTRATION package, built upon the OREMODULES package, is an implementation of a new effective algorithm obtained in [30] which computes the purity/grade filtration [86], [87] of linear functional systems (e.g., partial differential systems, differential time-delay systems, difference systems) and equivalent block-triangular matrices. See Section 6.1 . This package is used to compute closed form solutions of over/underdetermined linear partial differential systems which cannot be integrated by the standard computer algebra systems such as Maple and Mathematica. This package will soon be available.

5.7. AbelianSystems

Participants: Alban Quadrat [correspondent], Mohamed Barakat [Univ. Kaiserslautern].

The ABELIANSYSTEMS package is an implementation of an algorithm developed in [30] for the computation of the purity/grade filtration [86], [87] in the powerful `homalg` package of GAP 4 dedicated to constructive homological algebra methods, and developed by Barakat (University of Kaiserslautern) and his collaborators (http://homalg.math.rwth-aachen.de/). This package both supersedes the existing PURITYFILTRATION package which uses the non-efficient Maple Gröbner basis computation (see Section 5.6), and the original `homalg` procedure which computes purity filtration by means of time-consuming spectral sequences. Using the `homalg` package philosophy, the ABELIANSYSTEMS package can be used for the computation of the purity filtration of objects in different constructive abelian categories such as coherent sheaves over projective schemes as demonstrated in the `homag` package called `Sheaves` (see http://homalg.math.rwth-aachen.de/).

5.8. SystemTheory

Participants: Alban Quadrat [correspondent], Thomas Cluzeau [ENSIL, Univ. Limoges], Markus Lange-Hegermann [Univ. Aachen], Mohamed Barakat [Univ. Kaiserslautern].

The SYSTEMTHEORY package is a `homalg` based package dedicated to mathematical systems. This package, still in development, will include the algorithms developed in the OREMODULES and OREMORPHISMS packages. It currently contains an implementation of the OREMORPHISMS procedures which handle the decomposition problem aiming at decomposing a module/system into direct sums of submodules/subsystems, and Serre's reduction problem aiming at finding an equivalent system defined by fewer unknowns and fewer equations.

5.9. YALTA

Participants: David Avanessoff, Catherine Bonnet [correspondent], Hugo Cavalera, André R. Fioravanti [UNICAMP], Jim Pioche.

The YALTA toolbox is dedicated to the study of classical and fractional systems with delay in the frequency-domain. Its objective is to provide basic but important information such as, for instance, the position of the neutral chains of poles and unstable poles, as well as the root locus with respect to the delay of the system. The corresponding algorithms are based on recent theoretical results (see, for instance, [88] and [97]) and on classical continuation methods exploiting the particularities of the problem [98], [99]. For classical delay systems, a Pade2 approximation scheme is available as well as a finite-dimensional approximation of the system.

Binaries are freely available at http://yalta-toolbox.gforge.inria.fr/.

DOLPHIN Project-Team

5. Software and Platforms

5.1. ParadisEO

Participants: Clive Canape, Laetitia Jourdan, Arnaud Liefooghe, Nouredine Melab, Alexandre Quemy, El-Ghazali Talbi [correspondent], Sébastien Verel.

ParadisEO (PARallel and DIStributed Evolving Objects) is a C++ white-box object-oriented framework dedicated to the flexible design of metaheuristics. See web site http://paradiseo.gforge.inria.fr. Based on EO, a template-based ANSI-C++ compliant evolutionary computation library, it is composed of five modules:

- Paradiseo-EO provides tools for the development of population-based metaheuristic (evolutionary and genetic algorithm, genetic programming, particle swarm optimization, etc.)
- Paradiseo-MO provides tools for the development of single solution-based metaheuristics (hill-climbing, tabu search, simulated annealing, iterative local search, variable neighborhood search, incremental evaluation, partial neighborhood, etc.)
- Paradiseo-MOEO provides tools for the design of multi-objective metaheuristics (MO fitness assignment, MO diversity preservation, elitism, performance indicators, easy-to-use state-of-the-art algorithms, etc)
- Paradiseo-PEO provides tools for the design of parallel and distributed metaheuristics (parallel evaluation, parallel evaluation function, island model)
- Paradiseo-SMP provides tools for the design of shared memory parallel metaheuristics (parallel evaluation, island model)

Furthermore, Paradiseo also introduces tools for the design of distributed, hybrid and cooperative models:

- High level hybrid metaheuristics: coevolutionary and relay models.
- Low level hybrid metaheuristics: coevolutionary and relay models.

The ParadisEO framework has been especially designed to best suit to the following objectives:

- Maximum design and code reuse: ParadisEO is based on a clear conceptual separation of the solution methods from the problems they are intended to solve. This separation confers to the user a maximum code and design reuse.
- Flexibility and adaptability: The fine-grained nature of the classes provided by the framework allows a higher flexibility compared to other frameworks.
- Utility: ParadisEO allows the user to cover a broad range of metaheuristics, problems, parallel distributed models, hybridization mechanisms, etc.
- Transparent and easy access to performance and robustness: As the optimization applications are often time-consuming the performance issue is crucial. Parallelism and distribution are two important ways to achieve high performance execution. ParadisEO is one of the rare frameworks that provide the most common parallel and distributed models. These models can be exploited in a transparent way, one has just to instantiate their associated provided classes.
- Portability: The implemented models are portable on distributed-memory machines as well as on shared-memory multiprocessors, as they use standard libraries such as MPI and std::threads. ParadisEO supports the most recent version of standard of the C++ programming, c++11.

This year a new module, Paradiseo-SMP, has been released. All the new features is managed via the Inria Gforge project http://paradiseo.gforge.inria.fr. The version 2.0 has been released in Septembre 2012.

5.1.1. Paradiseo-SMP: a new module for shared memory parallel

This year, we released a new module dedicated to shared memory parallel. This module improves the technical mechanisms of Paradiseo-PEO thanks a new software architecture and the new c+11 features.
Paradiseo-SMP implements parallel evaluation, dynamic heterogeneous island model, and their hybridization. The main features are:

- Dynamic Island Model: topology can be changed during the execution.
- Heterogeneous Islands: different kinds of population-based metaheuristics can communicate (evolutionary and genetic algorithm, particle swarm optimization, etc.).
- Island Model and master/slave model can be hybridized.

All these new features are developed in c++11.

5.1.2. New technical features

Regarding the technical aspects, the compatibility with dependencies taken into account is:

- c++11 features supporting.
- Checked compatibility with different operating systems.
- Reviewed and checked compatibility with new versions of the tools used (CMake, g++, clang, MinGW...).
- Unit and integration test of all additional components, and experiments on classical applications.

5.1.3. Contributions and documentations

Many investigations were made in this context in order to help users to manipulate the framework.

- New quick start guide is available.
- New tutorials:
  - Tutorials SMP.
  - Tutorials GPU.
- Updated implementation for classical problems.

Self-assessment of the team effort (software criteria: http://www.inria.fr/institut/organisation/instances/commission-d-evaluation)

(A-4-up5) Audience: 4 - Used in many universities for teaching and several companies.

(SO-4) Software Originality: 4 - ParadisEO aggregates the last results of the Dolphin team.

(SM-4) Software Maturity: 4 - Extensive documentation, strong software engineering and testing, regression testing, user feedback ...

(EM-2-up3) Evolution and Maintenance: 2 - Basic maintenance with persistent attention to users.

(SDL-4) Software Distribution and Licensing: 4 - CeCILL license, public source, Windows and Mac installer, Linux packages.

(OC) Own Contribution: (Design/Architecture) DA-4, (Coding/Debugging) CD-4, (Maintenance/Support) MS-4, (Team/Project Management) TPM-4

GAMMA3 Project-Team

3. Software and Platforms

3.1. BLGEOL-V1 software

Participants: Patrick Laug [correspondant], Houman Borouchaki.

BLGEOL-V1 software can generate hex-dominant meshes of geologic structures complying with different geometric constraints: surface topography (valleys, reliefs, rivers), geologic layers and underground workings. First, a reference 2D domain is obtained by projecting all the line constraints into a horizontal plane. Different size specifications are given for rivers, outcrop lines and workings. Using an adaptive methodology, the size variation is bounded by a specified threshold in order to obtain a high quality quad-dominant mesh. Secondly, a hex-dominant mesh of the geological medium is generated by a vertical extrusion, taking into account the surfaces found (interfaces between two layers, top or bottom faces of underground workings). The generation of volume elements follows a global order established on the whole set of surfaces to ensure the conformity of the resulting mesh.

GECO Project-Team

5. Software and Platforms

5.1. IRHD

We develop a software for reconstruction of corrupted and damaged images, named IRHD (for Image Reconstruction via Hypoelliptic Diffusion). One of the main features of the algorithm on which the software is based is that it does not require any information about the location and character of the corrupted places. Another important advantage is that this method is massively parallelizable; this allows to work with sufficiently large images. Theoretical background of the presented method is based on the model of geometry of vision due to Petitot, Citti and Sarti. The main step is numerical solution of the equation of 3D hypoelliptic diffusion. IRHD is based on Fortran.

GEOSTAT Project-Team

5. Software and Platforms

5.1. Fluex

Participants: Denis Arrivault [correspondant], Rémi Paties, Hussein Yahia, Joel Sudre.

- Denis Arrivault has joined the team for a complete refoundation, rewriting, generalization and diffusion of the FluidExponents software, now called Fluex. FluidExponents is a software implementation of the MMF, presently written in Java, in a cooperative development mode on the Inria GForge, deposited at APP in 2010. Denis Arrivault has delivered the first Fluex package in December 2013, consisting of a core implementation under Gforge of the Microcanonical Multiscale Formalism in the form of C++ classes, for 1D, 2D 3D and 3D+t general signals. Fluex is in the process of being deposited in 2014. The Fluex project is carried on in 2014 by Rémi Paties. Contact: denis.arrivault@inria.fr, remi.paties@inria.fr
- A matlab code for the speech GCI detection algorithm has been made publicly available on the GeoStat website.

I4S Project-Team

4. Software and Platforms

4.1. COSMAD

Participants: Michael Doehler, Laurent Mevel.

With the help of former engineers, I4S team has developed and maintained a Scilab toolbox devoted to modal analysis and vibration monitoring of structures or machines subjected to known or ambient (unknown) excitation. This software (COSMAD 3.64) has been registered at the APP under the number

IDDN.FR.001.210011.002.S.A.2003.000.20700

A list of test-cases (simulators, laboratory test-beds, real structures) for which COSMAD has been used is available on I4S website. The problem is to identify the eigenstructure (eigenvalues and observed components of the associated eigenvectors) of the state transition matrix of a linear dynamical system, using only the observation of some measured outputs summarized into a sequence of covariance matrices corresponding to successive time shifts. Other services are

- *Output-only and Input/Ouptut subspace-based identification*,
- *Automated on-line identification package*,
- *Subspace-based identification through moving sensors data fusion*,
- *Damage detection and monitoring*,
- *Damage localization*,

The modules have been tested by different partners, especially the French industrial partners, EADS, Dassault and Sopemea, within the FLITE2 project, by partners from the past CONSTRUCTIF project, and within the framework of bilateral contracts with SNECMA and SVS.

Based on intensive internal evaluation of the toolbox, on both simulated and real data sets, EADS Space Transportation and CNES have been investigating how to use the toolbox for the exploitation of the Ariane 5 flight data sets.

This Scilabtoolbox continues to play the role of a programming and development environment for all our newly designed algorithms. Moreover, offering a *maintained* Scilab platform turns out to be a crucial factor in convincing industrial partners to undertake joint investigations with us. Just recently, SNECMA funded development for the Cosmad toolbox in 2010.

4.2. PEGASE

Participants: Vincent Le Cam, Mathieu Le Pen, Laurent Mevel.

We have developed a generic wireless platform that can be considered as the a result of redundant needs in wireless monitoring especially applied to civil engineering monitoring applications. This platform includes software and hardware bricks and aims at being generic by its native implementation of sober components, the worldwide TCP/IP protocol (802.11g), a signal processor, a small GPS receiver, and a micro embedded operating system (uClinux).

Since 2009, this platform -named PEGASE - is subject of an industrial transfer that has generated some tens of indivudual sales. A set of pluggable boards (that integrate the application specific sensing operation) offers a ready-to-use panel of wireless sensing solutions for developing specific applications as well as they can be seen as prototyping boards for further electronic developments.

As PEGASE platform reached a mature level of dissemination, LCPC recent efforts are now leaded with the goal of improving its wireless capacities. Those works concern energy saving while keeping a high level of embedded processing, of sampling rate or time-synchronization.

As software layers are mainly written in standard C language under Linux OS, those pragmatic solutions could easily be re-used by even radically different systems. The focus will specifically be pointed on: an algorithm that allows PEGASE wireless boards to be synchronized up to some uS using a GPS technique while keeping the GPS receiver OFF most of the time; a description of how the use of an operating system such as uClinux allows a full and remotely update of wireless sensors; the hardware and software strategies that have been developed to make PEGASE fully autonomous using solar cells.

The main characteristics of PEGASE feature are the following:

- Use of TCP/IP/WiFi as the wireless protocol: reliable, low-cost, scalable (IP is the worldwide protocol). Turned OFF when PEGASE doesn't communicate.
- Use of the Analog Device low-power Blackfin BF537 as core processor (Digital Signal Processor): 16 bits processor able of complex operations.
- Implementation of a small and low-power GPS receiver to ensure localization and, first of all, absolute time synchronization up to few μS GMT.
- uClinux as the embedded operating system: allows high level of abstraction while PEGASE algorithms are then programmed using standard ANSI C language.

Since its first version on january 2008, PEGASE has been used in various configurations where its properties fitted specific needs. Since a third-party partner (A3IP company) has been licensed by LCPC, PEGASE has been sold in hundreds of specimens and implemented in various configurations. This dissemination proved the capacity of wireless systems to really answer a large spectrum of applications. Developments in progress have the goal to increase this panoply. Even if uClinux and WiFi integration could be considered as *heavy*, the result is a great ability for developers or customers to achieve their own applications. The genericity of C language and the worldwide IP protocol make them ubiquitous. A quite expert job has been leaded to develop specific embedded drivers under uClinux OS in order to get specific behaviors for time synchronization, quartz drift auto-training and correction. This specific and dynamic correction takes temperature effects into account and the result is an absolute time synchronisation better that 4 μS. Even if technologies evolve (components, processor, batteries...), generic principle could be extracted independently from technological choices. Those main principles are: daughter/mother boards, Linux integration, a ready to use c-object library, a boost circuit linked to a MPPT algorithm, GPS synchronization and quartz correction. Most of the improvements can be reused and applied to other wireless platforms even using drastically different electronic implementations.

m Porting of subspace modal analysis algorithms is currently under way on the PEGASE platform.

IPSO Project-Team (section vide)

MATHRISK Project-Team

5. Software and Platforms

5.1. PREMIA

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

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

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

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

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

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

- AMS: 91B28;65Cxx;65Fxx;65Lxx;65Pxx
- License: Licence Propriétaire (genuine license for the Consortium Premia)
- Type of human computer interaction: Console, interface in Nsp, Web interface
- OS/Middelware: Linux, Mac OS X, Windows
- APP: The development of Premia started in 1999 and 15 are released up to now and registered at the APP agency.
- Programming language: C/C++ librairie Gtk
- Documentation: the PNL library is interfaced via doxygen
- Size of the software: 280580 lines for the Src part only, that is 11 Mbyte of code, 130400 lines for PNL, 103 Mbyte of PDF files of documentation.
- interfaces : Nsp for Windows/Linux/Mac, Excel, binding Python, and a Web interface.
- Publications: [1], [68], [75], [83], [86], [55]

5.1.1. Content of Premia

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

1. **Equity derivatives:**

   The following models are considered:

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

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

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

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

2. **Interest rate derivatives**

   The following models are considered:

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

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

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

   Reduced form models and copula models are considered.

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

4. **Hybrid products**

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

5. **Energy derivatives: swing options**

   Mean reverting and jump models are considered.

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

5.1.2. *Premia design*

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

1. Development of the PNL. Here are the major 2013 contributions (by Jérôme Lelong):
   1. PNL relies on CMake for compiling.
   2. Add the sampling of new distributions: log-normal, inverse Gaussian, asymmetric double exponential distributions.
   3. Add the computation of eigenvalues and eigenvectors for complex matrices. Based on this new function, add the computation of the matrix logarithm for complex matrices.
   4. Add Newton's algorithm with Armijo line search.
   5. The top level PnlOjbect is modified to keep track of the number of references on an object to improve memory management in lists. This delicate change in the core of the library enabled us to speed codes based on lists by a great deal.
   6. Several other functions have also been added.
2. Premia

1. The compilation of Premia is now based on CMake which is a cross-platform building tool. It allows us to maintain a single building chain and to automatically generate Makefiles or a Visual project. This technology change significantly improves our ability to generate Windows versions.
2. Add support for PnlMatrix both in Premia VAR and in the Nsp toolbox.
3. A model size change in the Nsp GUI automatically propagates to all parameters thanks to the addition a *Return* callback in the GUI.
4. Some fixes in the core of Premia: several setters were broken.
5. Refactor the credit toolbox to simplify the number of products.
6. Scripts to generate new model templates have been significantly improved and reimplemented in Python.
7. Improve the generic functions `Get`, `FGet`, `Show`, `PrintVar` and `FScanVar` to enable all the models to use them. This led us to remove a lot of code.

5.1.3. Algorithms implemented in Premia in 2013

Premia 15 was delivered to the consortium members in March 2013. It contains the following new algorithms:

- **Interest Rate, Inflation, FX**
  - Inflation products with stochastic volatility and stochastic interest rates. S. Singor, L. Grzelak C.W.Oosterlee D.D.B. van Bragt.
  - On cross-currency models with stochastic volatility and correlated interest rates. L. Grzelak C.W.Oosterlee. *Applied Math. Finance, to appear.*
  - Repricing the Cross Smile: An Analytic Joint Density. P.Austing. *preprint 2011*
- **Energy and Commodities**
  - Efficient Pricing of Commodity Options with Early-Exercise under the Ornstein–Uhlenbeck process. C.W.Oosterlee. B.Zhang. *preprint 2011*
  - A finite dimensional approximation for pricing moving average options. M. Bernhart P.Tankov X. Warin, *to appear in SIAM Journal on Financial Mathematics.*
  - Pricing and hedging spread options. R. Carmona V.Durrleman. *SIAM Rev. 45 (2003), no. 4, 627—685.*
  - Closed form spread option valuation. P.Bjerksund G. Stensland *Quantitative Finance, 2011.*
  - A Fourier transform method for spread option pricing. T. R. Hurd and Z. Zhou. *SIAM Journal on Financial Mathematics. 1, 142-–157, 2010.*
  - Multi-asset spread option pricing and hedging *Quantitative Finance, Vol.10, No3, 305-324, 2010.*
  - Unspanned Stochastic Volatility and the Pricing of Commodity Derivatives. A.B.Trolle E.Schwartz. *Rev. Financ. Stud. (2009) 22(11): 4423-4461*
  - Pricing Commodity Swaptions in Multifactor Models. K.Larsson. *The Journal of Derivatives Winter 2011, Vol. 19, No. 2,32-44.*
- **Equity Derivatives**
  - Importance sampling and Statistical Romberg Method. M.B. Alaya A.Kebaier K.Hajji
  - New approximations in local volatility models. E. Gobet, A.Suleiman
  - Componentwise splitting methods for pricing American options under stochastic volatility. Ikonen, S.; Toivanen,J. *Int. J. Theor. Appl. Finance 10 (2007), no. 2, 331–361.*
  - ADI finite difference schemes for option pricing in the Heston model with correlation. K.J. in ’t Hout and S. Foulon. *Int. J. Numer. Anal. Mod. 7, 303–320 (2010).*

- ADI schemes with Ikonen-Toivanen splitting for pricing American put options in the Heston model. T. Haentjens, K. in 't Hout and K. Volders. In: Numerical Analysis and Applied Mathematics, eds. T. E. Simos et. al., AIP Conf. Proc. 1281, 231-234 (2010).
- Pricing of Timer Options. C. Bernard Z. Cui. *Journal of Computational Finance, to appear*
- Efficient Simulation of the Double Heston Model. P.Gauthier D.Possamai.
- Greedy methods method for basket options. T.Lelievre J.I.Acevedo
- Pricing higher-dimensional American options using the stochastic grid method. C.W.Oosterlee S. Jain
- Calibration in the Heston model. L. Abbas Turki

The software Premia 15 has been registered at the APP (Agence pour la Protection des Programmes) with the reference IDDN.FR.001.190010.012.S.C.2001.000.31000.

Maxplus Project-Team

5. Software and Platforms

5.1. Boîte à outil Maxplus de SCILAB/Maxplus toolbox of Scilab

Trois chercheurs du groupe (S. Gaubert, J.-P. Quadrat, et G. Cohen) ont développé (à partir d'une première version réalisée par M. Mc Gettrick) la *boîte à outils Maxplus* de Scilab, qui est téléchargeable librement parmi les contributions du site Scilab, et qui est maintenant intégrée par défaut dans Scicoslab. Cette boîte à outils implémente l'ensemble du calcul numérique linéaire max-plus, elle comprend en particulier le stockage creux des matrices, et des algorithmes efficaces pour le calcul de la valeur propre basées sur les itérations sur les politiques. Elle a été utilisées par plusieurs chercheurs, voir notamment [65], [135]. Il faut aussi noter que le groupe de L. Hardouin, du LISA/Istia, a complété la boîte à outils Maxplus en interfaçant leur propre librairie C++, qui permet le calcul des séries de transfert de graphes d'événements temporisés.

English version

Three researchers of the team (S. Gaubert, J.-P. Quadrat, and G. Cohen, building on a preliminary version of M. McGettrick) have developed and released the *Maxplus toolbox* of Scilab, which is freely available among the contributions on the Scilab web site, and which is now included by default in Scicoslab. It implements all basic linear algebra functionalities, with a special attention to large sparse matrices, including efficient algorithms for eigenvalue computation based on policy iteration. The software has been used by several researchers in their work, including [65], [135]. It should be noted that the team of L. Hardouin, from LISA/Istia, has completed the toolbox by interfacing their own C++ library computing the transfer series of a timed event graph.

5.2. Itérations sur les politiques pour les jeux stochastiques à somme nulle/Policy iterations for zero sum stochastic games

L'algorithme d'itérations sur les politiques pour les jeux stochastiques à somme nulle pour le cas de paiements ergodiques (gain moyen par unité de temps), et dégénérés de type "multi-chaîne" a été introduit dans [84]. Plusieurs stages ont permis l'implémentation partielle en Scilab, C ou C++, et le test de ce type d'algorithmes (voir le travail de Vishesh Dhingra [98]), ou de son couplage avec la résolution de systèmes linéaires par des méthodes multigrilles algébriques (stage de Shantanu Gangal en 2007). Le travail de thèse de Sylvie Detournay a permis le développement d'un programme complet. Le code écrit par Sylvie Detournay (en C) a été déposé sur InriaGForge. Pour le moment il n'est accessible qu'aux membres de l'équipe.

English version

The policy iteration algorithm for zero sum repeated games with ergodic payoff (i.e. mean payoff per time unit), and in degenerate "multichain" cases, has been introduced in [84]. Several internships allowed us to implement in Scilab, C or C++, and to test such algorithms (see the work of Vishesh Dhingra [98]), or its combinaison with the resolution of linear systems by algebraic multigrid methods (internship of Shantanu Gangal in 2007). The PhD thesis work of Sylvie Detournay allowed us to develop a complete program. The program written by Sylvie Detournay (in C language) has been posted on InriaGForge. For the moment it can only be seen by members of the team.

5.3. TPLib: bibliothèque pour la manipulation de polyèdres tropicaux/TPLib: tropical polyhedra library

TPLib est une bibliothèque écrite en OCaml qui permet de manipuler des polyèdres tropicaux. Elle est distribuée sous license LGPL https://gforge.inria.fr/projects/tplib.

Cette bibliothèque implémente notamment des algorithmes permettant de passer d'une représentation externe d'un polyèdre à une représentation interne, ou inversement (voir §6.2.1 pour plus de détails). Elle permet aussi de réaliser d'autres opérations fondamentales, comme le calcul du complexe polyédral associé à un polyèdre donné (au sens de Develin et Sturmfels [96]), ou le calcul de cônes tangents tropicaux. Enfin, elle fournit toutes les primitives permettant d'utiliser les polyèdres tropicaux en tant que domaine abstrait numérique, afin de déterminer des invariants de programmes ou systèmes faisant intervenir les opérations $\min$ et $\max$ (voir [63]).

TPLib est aujourd'hui utilisé dans le logiciel Polymake [116], développé à la Technische Universität Darmstadt (Allemagne). Ce dernier logiciel constitue une boite à outils permettant de manipuler des nombreux objets mathématiques (polytopes convexes, complexes polyédraux, graphes, matroïdes, polytopes tropicaux).

Le développement d'interfaces avec d'autres logiciels est désormais facilité grâce à la présence de *bindings* dans le langage C. Grâce à cela, un prototype d'interface a été réalisé entre TPLib et l'outil VerifyTAPN (https://launchpad.net/verifytapn), qui permet la vérification de réseaux de Pétri avec arcs temporisés (voir §6.5.4). De même, une interface à la bibliothèque de domaines abstraits numériques APRON [128] est également en cours de développement.

English version

TPLib is a library written in OCaml, which allows to manipulate tropical polyhedra. It is distributed under LGPL https://gforge.inria.fr/projects/tplib.

This library implements algorithms allowing to pass from an external representation of a polyhedron to an internal description, or inversely (see §6.2.1 for more details). Besides, the library allows to perform several fundamental operations over tropical polyhedra, such as computing the associated polyhedral complex (see Develin and Sturmfels [96]), or determining the tropical tangent cone at any point. Finally, it provides all the primitives allowing to use tropical polyhedra as an numerical abstract domain, in order to determine program/system invariants involving the operations $\min$ and $\max$ (see [63]).

TPLib is now used in the software Polymake [116], developed in Technische Universität Darmstadt (Germany). Polymake is a toolbox allowing to manipulate mathematic objects such as convex polytopes, polyhedral complexes, graphs, matroids, and tropical polytopes.

The development of further interfaces is now easier thanks to the distribution of bindings in C language. Using these bindings, a prototype of interface has been created between TPLib and the model-checker VerifyTAPN (https://launchpad.net/verifytapn), which allows the verification of timed-arc Petri Nets (see §6.5.4). An interface to the numerical abstract domain APRON [128] is also under development.

MC2 Project-Team

5. Software and Platforms

5.1. eLYSe

Participant: Olivier Saut.

eLYse is a numerical platform used for our computations in Biology (tumor growth), micro-fluidics and complex Newtonian fluid flows. The platform is divided in two libraries : one is devoted to the modelling equations and the other one includes the numerical solvers. For example, we are able to treat (in 2D and 3D) transport equations, diffusion equations, Navier-Stokes equations, Maxwell system and the interaction fluid-structure by level-set and penalization methods. The solvers are based on finite volume methods on cartesian grids and allow parallel computations. See also the web page http://www.math.u-bordeaux1.fr/~osaut/pages/eLYSe.html.

- Version: 0.7
- ACM: ACM J.2 J.3 G.1.8 G.1.10
- AMS: AMS65Z05 35Q92
- Keywords: Modélization and numerical simulations, Finite volume methods, Level Set approach, Penalization method
- APP: En cours
- Type of human computer interaction: console
- OS/Middelware: Platform developped on Mac OS X architecture.
- Required library or software: Petsc (http://www.mcs.anl.gov/petsc/petsc-as/) Vtk (http://www.vtk.org/) Blitz++ (http://c2.com/cgi/wiki?BlitzPlusPlus) (optionnel) Boost (http://www.boost.org/)
- Programming language: C++
- Documentation: doxygen.

5.2. Kesaco

Participant: Olivier Saut.

Kesaco is a set of libraries and programs aiming at applications of mathematical modeling in clinical oncology. It features:

- A library of specialized mathematical model describing the growth of different types of cancers (secondary tumors in the lung, gliomas).
- A set of programs useful to validate mathematical models (compute the various behavior they can produce) and to build databases of numerical simulations.
- Segmentation and registration routines to use medical images directly in our numerical codes.
- Calibration methods to recover the parameters of the models using sequences of medical images. Three techniques are implemented (a genetic algorithm, a technique based on reduced order models, a sensitivity technique).

All these routines are adapted to run on a MP architecture. The webpage may be found at http://www.math.u-bordeaux1.fr/~osaut/pages/kesaco.html.

- Version: 0.2
- Keywords: Modélization and numerical simulations
- APP: En cours
- Type of human computer interaction: console
- OS/Middelware: Platform developped on Mac OS X architecture.
- Required library or software: eLYSe, Insight Toolkit (http://www.itk.org)
- Programming language: C++
- Documentation: doxygen.

5.3. NaSCar

Participant: Michel Bergmann [correspondant].

This code is devoted to solve 3D-flows in around moving and deformable bodies. The incompressible Navier-Stokes equations are solved on fixed grids, and the bodies are taken into account thanks to penalization and/or immersed boundary methods. The interface between the fluid and the bodies is tracked with a level set function or in a Lagrangian way. The numerical code is fully second order (time and space). The numerical method is based on projection schemes of Chorin-Temam's type. The code is written in C language and use Petsc (http://www.mcs.anl.gov/petsc/petsc-as/) library for the resolution of large linear systems in parallel.

NaSCar can be used to simulate both hydrodynamic bio-locomation as fish like swimming and aerodynamic flows such wake generated by a wind turbine.

- Version: 1
- Keywords: numerical analyse, fluid mechanics, langage C, PETSc
- Software benefit : simulate a flow around a deformable obstacle, moving into a fluid.
- APP: En cours
- Patent: non
- Type of human computer interaction: human for the moment
- OS/Middelware: unix, linux, mac os
- Required library or software: PETSc item Programming language: C
- Documentation: in progess

5.4. S-MPI-2D-3D

Participants: Charles-Henri Bruneau [correspondant], Khodor Khadra.

The software NS-MPI-2D-3D is a numerical platform devoted to the computation of the incompressible flow around bodies in two or three dimensions modelled by Stokes, Navier-Stokes or Oldroyd-B equations. It is based on finite differences or finite volumes approximations on cartesian grid using the volume penalization method to handle the obstacles. The resolution is achieved by means of the multigrid method. Dirichlet, periodic or artificial boundary conditions are implemented to solve various problems in closed or open domains.

- Version: 3
- Keywords: Numerical simulation of incompressible flows,
- Type of human computer interaction: console
- OS/Middelware: unix, linux, Mac OS X item Programming language: Fortran 95 and MPI
- Documentation: included

5.5. Other MC2 codes

- Penalization techniques on cartesian grids to solve incompressible Navier-Stokes equations
  - **Vortex**: sequential, Vortex In-Cell (VIC) scheme : hybrid vortex methods based on the combination of Lagrangian mesh-free schemes and Eulerian grid based schemes on the same flow region.
  - Unstructured body fitted meshes
  - **Richards** : 2D Unstructured finite element code, implicit solver, sequential, to solve the transport-diffusion equations through a porous media including tidal forcing and mechanisms of diagenesis.
  - development inside **FluidBox** software in collaboration with **BACCHUS**. 2D-3D unstructured meshes, Stabilized Finite Elements method (SUPG), RANS turbulence model, parallel: Domain Decomposition and MPI.
- Immersed boundary techniques for:
  - **Compressible flows** : 2D-3D finite volume scheme for compressible Euler equations with solid obstacles on cartesian grids. 3D code parallelized with MPI
  - **Elliptic problems :** 2 2D-3D finite difference scheme for elliptic interface problems, parallelized with PETSc
  - Electropermeabilization: 2D finite difference scheme, parallelized with PETSc to simulate the electropermeablization of biological cells

MCTAO Project-Team

5. Software and Platforms

5.1. Hampath

Participants: Jean-Baptiste Caillau, Olivier Cots [corresponding participant], Joseph Gergaud.

Hampath is a software developped to solve optimal control problems but also to study Hamiltonian flow. It has been developped since 2009 by members of the APO team from Institut de Recherche en Informatique de Toulouse, jointly with colleagues from the Université de Bourgogne. It is now updated with McTAO team members. See more on http://cots.perso.math.cnrs.fr/hampath/.

MICMAC Project-Team (section vide)

MISTIS Project-Team

5. Software and Platforms

5.1. The LOCUS software

Participants: Florence Forbes, Senan James Doyle.

Joint work with: Michel Dojat from Grenoble Institute of Neuroscience and Benoit Scherrer from Harvard Medical School, Boston, MA, USA.

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

The LOCUS software has been developed in the context of a collaboration between Mistis, a computer science team (Magma, LIG) and a Neuroscience methodological team (the Neuroimaging team from Grenoble Institut of Neurosciences, INSERM). This collaboration resulted over the period 2006-2008 into the PhD thesis of B. Scherrer (advised by C. Garbay and M. Dojat) and in a number of publications. In particular, B. Scherrer received a "Young Investigator Award" at the 2008 MICCAI conference.

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

5.2. The P-LOCUS software

Participants: Florence Forbes, Senan James Doyle, Flor Vasseur.

Joint work with: Michel Dojat.

The Locus software was extended to address the delineation of lesions in pathological brains. Its extension P-LOCUS (http://p-locus.com) for lesion detection was realized by S. Doyle with financial support from Gravit with the goal to create a Start-up. P-LOCUS software analyses, in few minutes, a 3D MR brain scan and performs fully automatic brain lesion delineation using a combined dataset of various 3D MRI sequences. Its originality comes from:

- it is fully automatic: no external user interaction and no training data required
- the possibility to combine information from several images (MR sequences)
- a statistical Bayesian framework for robustness to image artefacts and a priori knowledge incorporation
- a voxel-based clustering technique that uses Markov random fields (MRF) incorporating information about neighboring voxels for spatial consistency and robustness to imperfect image features (noise).
- the possibility to select and incorporate relevant a priori knowledge via different atlases, e.g. tissue and vascular territory atlases
- a fully integrated preprocessing steps and lesion ROI identification

P-LOCUS software was presented at various conferences and used for the BRATS Challenge on tumor segmentation organized as a satellite challenge of the Miccai conference in Nagoya, Japan. A paper submitted to IEEE trans. on Medical Imaging reports the challenge results [62].

5.3. The PyHRF software

Participants: Christine Bakhous, Florence Forbes, Thomas Vincent.

Joint work with: Philippe Ciuciu and Solveig Badillo from Parietal Team Inria and CEA NeuroSpin, Lotfi Chaari and Laurent Risser from Toulouse University.

As part of fMRI data analysis, the PyHRF package (http://pyhrf.org) provides a set of tools for addressing the two main issues involved in intra-subject fMRI data analysis: (i) the localization of cerebral regions that elicit evoked activity and (ii) the estimation of the activation dynamics also referenced to as the recovery of the Hemodynamic Response Function (HRF). To tackle these two problems, PyHRF implements the Joint Detection-Estimation framework (JDE) which recovers parcel-level HRFs and embeds an adaptive spatio-temporal regularization scheme of activation maps. With respect to the sole detection issue (i), the classical voxelwise GLM procedure is also available through NIPY, whereas Finite Impulse Response (FIR) and temporally regularized FIR models are implemented to deal with the HRF estimation concern (ii). Several parcellation tools are also integrated such as spatial and functional clusterings. Parcellations may be used for spatial averaging prior to FIR/RFIR analysis or to specify the spatial support of the HRF estimates in the JDE approach. These analysis procedures can be applied either to volumic data sets or to data projected onto the cortical surface. For validation purpose, this package is shipped with artificial and real fMRI data sets. To cope with the high computational needs for inference, PyHRF handles distributing computing by exploiting cluster units as well as multiple cores computers. Finally, a dedicated viewer is available which handles n-dimensional images and provides suitable features for exploring whole brain hemodynamics (display of time series, maps, ROI mask overlay). A paper under revision for Frontiers in Neuroinformatics gives more details on the current PyHRF functionalities.

MODAL Project-Team

5. Software and Platforms

5.1. Rmixmod package for mixed data

Participants: Christophe Biernacki, Serge Iovleff, Parmeet Bhatia.

MIXMOD (MIXture MODelling) is an important software for the mΘdal team since it concerns its main topics: model-based supervised, unsupervised and semi-supervised classification for various data situations. MIXMOD is now a well-distributed software with over 250 downloads/month are recorded for several years. MIXMOD is written in C++ (more than 10 000 lines) and distributed under GNU General Public License. Several other institutions participate in the MIXMOD development since several years: CNRS, Inria Saclay-Île de France, Université de Franche-Comté, Université Lille 1. The software already benefits from several APP depositions and an R package (Rmixmod) has been associated to MIXMOD in 2012.

In 2013, Parmeet Bhatia, under scientific supervision of Christophe Biernacki and Serge Iovleff, has developped possibility in Rmixmod to cluster simultaneously continuous and categorical data with the restrictive conditional independence assumption. It is an important first step towards the long term purpose of mΘdal to cluster heterogeneous (or mixed) data sets. It is a joint work with Florent Langrognet, Rémi Lebret, Gilles Celeux and Gérard Govaert.

5.2. RankClust package for rank data

Participants: Christophe Biernacki, Quentin Grimonprez, Julien Jacques.

Rankcluster package for R proposes a clustering tool for ranking data. Multivariate and partial rankings can be also taken into account. Available on CRAN.

5.3. Clere package for high dimensional regression

Participants: Christophe Biernacki, Loïc Yengo, Julien Jacques.

The Clere package for R proposes variable clustering in high dimensional linear regression. Available on CRAN.

5.4. Clustericat package for correlated categorical variable

Participants: Christophe Biernacki, Matthieu Marbac-Lourdelle, Vincent Vandewalle.

`Clustericat` is a R package for model-based clustering of categorical data. In this package, the model `CCM` [41] where the main conditional dependencies between variables are taken into account is implemented. `Clustericat` performs the model selection and provides the best model according to the BIC criterion and the maximum likelihood estimates. It is available online on Rforge (https://r-forge.r-project.org/R/?group_id=1803).

5.5. CorReg package for correlated variables in regression

Participants: Christophe Biernacki, Clément Théry.

Databases from the steel industry are often large (very long process with many parameters) and have strong correlations between variables. Some variables may be written directly in terms of other via physical models or related by definition. Moreover the process, which is specific to the type of finished product, conditions most of the process parameters and therefore induces strong correlations between variables. The main idea is to consider some form of sub-regression models, some variables defining others. We can then remove temporarily some of the variables to overcome ill-conditioned matrices inherent in linear regression and then reinject the deleted information, based on the structure that links the variables. The final model therefore takes into account all the variables but without suffering from the consequences of correlations between variables or high dimension. This research is placed in a steel industry context (Arcelor-Mittal Dunkerque).

The associated CorReg package is now available on Rforge. It is a joint work with Gaétan Loridant.

5.6. AAM

Participant: Serge Iovleff.

A console based program written in C++ abd dedicated to the estimation of the Auto-Associative Models.

5.7. BlockCluster

Participants: Serge Iovleff, Parmeet Bathia.

Serge Iovleff, Parmeet Bathia
BlockCluster: An R package on top of the coclust C++ library.

5.8. HDPenReg

Participants: Quentin Grimonprez, Serge Iovleff.

R-package written in collaboration based on a C++ code dedicated to the estimation of regression model with l1-penalization.

5.9. STK++ release 0.5

Participant: Serge Iovleff.

New release including new functionalities for templated expression evaluation (similar to the Eigen library offer) and a new subproject offering tools for Clustering.

5.10. Funclustering package for R

Participants: Cristian Preda, Julien Jacques.

Funclustering package for R proposes a clustering tool for functional data. Multivariate curves can be also taken into account. Available on CRAN.

5.11. metaMA

metaMA is a specialised software for microarrays. It is a R package which combines either p-values or modified effect sizes from different studies to find differentially expressed genes. The main competitor of metaMA is geneMeta. Compared to geneMeta, metaMA offers an improvement for small sample size datasets since the corresponding modelling is based on shrinkage approaches.

Guillemette Marot is the main contributor and the maintainer of this package.

This software is routinely used by biologists from INRA, Jouy en Josas (it has been included in a local analysis pipeline) but its diffusion on the CRAN makes it available to a wider community, as attested by the citations of publications related to the methods implemented in the software.

More information is available on the website http://cran.r-project.org/web/packages/metaMA/

5.12. metaRNASeq

Participant: Guillemette Marot.

metaRNASeq is a specialised software for RNA-seq experiments. It is an R package which is an adaptation of the metaMA package presented previously. Both implement the same kind of methods but specificities of the two types of technologies require some adaptations to each one. Guillemette Marot and Andrea Rau are the main contributors of this package and Guillemette Marot is the maintainer of this package.

5.13. MPAGenomics

Participants: Quentin Grimonprez, Guillemette Marot, Alain Celisse.

MPAGenomics is a R package for multi-patients analysis of genomics markers. Its main contributor is Quentin Grimonprez. It enables to study several copy number and SNP data profiles at the same time. It offers wrappers from commonly used packages to offer a pipeline for beginners in R. It also proposes a special way of choosing some crucial parameters to change some default values which were not adapted in the original packages. For multi-patients analysis, it wraps some penalized regression methods implemented in HDPenReg. It is available on the Inria forge and should be released on the R-forge in January.

5.14. SMVar

Participant: Guillemette Marot.

SMVar is a specialised software for microarrays. This R package implements the structural model for variances in order to detect differentially expressed genes from gene expression data. It performs gene expression differential analysis, based on a particular variance modelling. Its main competitor is the Bioconductor R package limma but limma assumes a common variance between the two groups to be compared while SMVar relaxes this assumption.

Guillemette Marot is the main contributor and the maintainer of this package.

More information is available on the website http://cran.r-project.org/web/packages/SMVar/index.html

MOKAPLAN Exploratory Action

5. Software and Platforms

5.1. CFD based MK solvers

5.1.1. Platforms

The core of the ALG2 algorithm [43] for the CFD formulation of the Optimal Mass Transportation problem and many of its generalization is a Poisson solver. Then each problem calls for different but simple modifications the point wise minimization of a given Lagrangian function. We have written such a FreeFem ALG2 platform and are plan to implement a parallel version on Rocquencourt Inria cluster.

5.2. MA based Optimal Mass Transportation solvers

5.2.1. Platforms

Monotone discretisations of the Monge-Ampère operator (the determinant of a Hessian function) is the core of Monge-Ampère Optimal Mass Transportation solvers but is also a useful tool for convexity constraints in infinite dimensional optimization and JKO gradient flows. We are implementing in F90 and comparing several monotone schemes. These modules could be reused in different applications.

NACHOS Project-Team

5. Software and Platforms

5.1. MAXW-DGTD

Participants: Stéphane Lanteri [correspondant], Loula Fezoui, Ludovic Moya, Raphaël Léger, Jonathan Viquerat.

MAXW-DGTD is a software suite for the simulation of time domain electromagnetic wave propagation. It implements a solution method for the Maxwell equations in the time domain. MAXW-DGTD is based on a discontinuous Galerkin method formulated on unstructured triangular (2D case) or tetrahedral (3D case) meshes [16]. Within each element of the mesh, the components of the electromagnetic field are approximated by a arbitrary high order nodal polynomial interpolation method. This discontinuous Galerkin method combines a centered scheme for the evaluation of numerical fluxes at a face shared by two neighboring elements, with an explicit Leap-Frog time scheme. The software and the underlying algorithms are adapted to distributed memory parallel computing platforms thanks to a parallelization strategy that combines a partitioning of the computational domain with message passing programming using the MPI standard. Besides, a peripheral version of the software has been recently developed which is able to exploit the processing capabilities of a hybrid parallel computing system comprising muticore CPU and GPU nodes.

- AMS: AMS 35L50, AMS 35Q60, AMS 35Q61, AMS 65N08, AMS 65N30, AMS 65M60
- Keywords: Computational electromagnetics, Maxwell equations, discontinuous Galerkin, tetrahedral mesh.
- OS/Middelware: Linux
- Required library or software: MPI (Message Passing Interface), CUDA
- Programming language: Fortran 77/95

5.2. MAXW-DGFD

Participants: Stéphane Lanteri [correspondant], Ronan Perrussel.

MAXW-DGFD is a software suite for the simulation of time harmonic electromagnetic wave propagation. It implements a solution method for the Maxwell equations in the frequency domain. MAXW-DGFD is based on a discontinuous Galerkin method formulated on unstructured triangular (2D case) or tetrahedral (3D case) meshes. Within each element of the mesh, the components of the electromagnetic field are approximated by a arbitrary high order nodal polynomial interpolation method. The resolution of the sparse, complex coefficients, linear systems resulting from the discontinuous Galerkin formulation is performed by a hybrid iterative/direct solver whose design is based on domain decomposition principles. The software and the underlying algorithms are adapted to distributed memory parallel computing platforms thanks to a paralleization strategy that combines a partitioning of the computational domain with a message passing programming using the MPI standard. Some recent achievements have been the implementation of non-uniform order DG method in the 2D case and of a new hybridizable discontinuous Galerkin (HDG) formulation also in the 2D and 3D cases.

- AMS: AMS 35L50, AMS 35Q60, AMS 35Q61, AMS 65N08, AMS 65N30, AMS 65M60
- Keywords: Computational electromagnetics, Maxwell equations, discontinuous Galerkin, tetrahedral mesh.
- OS/Middelware: Linux
- Required library or software: MPI (Message Passing Interface)
- Programming language: Fortran 77/95

5.3. SISMO-DGTD

Participants: Nathalie Glinsky [correspondant], Stéphane Lanteri.

SISMO-DGTD is a software for the simulation of time domain seismic wave propagation. It implements a solution method for the velocity-stress equations in the time domain. SISMO-DGTD is based on a discontinuous Galerkin method formulated on unstructured triangular (2D case) or tetrahedral (3D case) meshes [6]. Within each element of the mesh, the components of the electromagnetic field are approximated by a arbitrary high order nodal polynomial interpolation method. This discontinuous Galerkin method combines a centered scheme for the evaluation of numerical fluxes at a face shared by two neighboring elements, with an explicit Leap-Frog time scheme. The software and the underlying algorithms are adapted to distributed memory parallel computing platforms thanks to a paralleization strategy that combines a partitioning of the computational domain with a message passing programming using the MPI standard.

- AMS: AMS 35L50, AMS 35Q74, AMS 35Q86, AMS 65N08, AMS 65N30, AMS 65M60
- Keywords: Computational geoseismics, elastodynamic equations, discontinuous Galerkin, tetrahedral mesh.
- OS/Middelware: Linux
- Required library or software: MPI (Message Passing Interface)
- Programming language: Fortran 77/95

5.4. NUM3SIS

Participants: Nora Aissiouene, Thibaud Kloczko [SED [1] team], Régis Duvigneau [OPALE project-team], Thibaud Kloczko [SED team], Stéphane Lanteri, Julien Wintz [SED team].

NUM3SIS http://num3sis.inria.fr is a modular platform devoted to scientific computing and numerical simulation. It is designed to handle complex multidisciplinary simulations involving several fields such as Computational Fluid Dynamics (CFD), Computational Structural Mechanic (CSM) and Computational ElectroMagnetics (CEM). In this context, the platform provides a comprehensive framework for engineers and researchers that speeds up implementation of new models and algorithms. From a software engineering point of view, num3sis specializes and extends some layers of the meta-platform dtk, especially its core and composition layers. The core layer enables the user to define generic concepts used for numerical simulation such as mesh or finite-volume schemes which are then implemented through a set of plugins. The composition layer provides a visual programming framework that wraps these concepts inside graphical items, nodes. These nodes can then be connected to each other to define data flows (or compositions) corresponding to the solution of scientific problems. NUM3SIS provides a highly flexible, re-usable and efficient approach to develop new computational scenarios and takes advantage of existing tools. The team participates to the development of the NUM3SIS platform through the adaptation and integration of the MAXW-DGTD simulation software. This work is being carried out with the support of two engineers in the framework of an ADT (Action de Développement Technologique) program.

5.5. Medical Image Extractor

Participants: Stéphane Lanteri, Julien Wintz [SED team].

---

[1] Service d'Experimentation et de Développement

Medical Image Extractor http://num3sis.inria.fr/software/apps/extractor provides functionalities needed to extract meshes from labeled MR or PET-CT medical images. It puts the emphasis on consistence, by generating both boundary surfaces, and volume meshes for each label (ideally identifying a tissue) of the input image, using the very same tetreahedrization. As this process requires user interaction, images and meshes are visualized together with tools allowing navigation and both easy and accurate refinement of the generated meshes, that can then be exported to serve as an input for other tools, within a multidisciplinar software toolchain. Using both DTK http://dtk.inria.fr and NUM3SIS SDKs, Medical Image Extractor comes within NUM3SIS'framework. Using cutting edge research algorithms developped by different teams at Inria, spread among different research topics, namely, visualization algorithms from medical image processing, meshing algorithms from algorithmic geometry, it illustrates the possibility to bridge the gap between software that come from different communities, in an innovative and highly non invasive development fashion.

NANO-D Team

4. Software and Platforms

4.1. SAMSON

../../../../projets/nano-d/IMG/SAMSONArchitecture.jpg

Figure 3. SAMSON's architecture.

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

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

The current architecture of SAMSON is visible in Figure 3 . The code is organized into four main parts: a) the Base (in which "Core" contains, in particular, the heart of the adaptive algorithms: signaling mechanisms specifically designed for SAMSON), b) the Software Development Kit (SDK: a subset of the base that will be provided to module developers), c) Modules, and d) the SAMSON application itself.

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

NECS Project-Team

5. Software and Platforms

5.1. GTL – Grenoble Traffic Lab

Participants: C. Canudas de Wit [contact person], I. Bellicot, P. Bellemain, L. Leon Ojeda, D. Pisarski, A. Kibangou, F. Morbidi.

The Grenoble Traffic Lab (GTL) initiative, led by the NECS team, is a real-time traffic data center (platform) that collects traffic road infrastructure information in real-time with minimum latency and fast sampling periods. The main elements of the GTL are: a real-time data-base, a show room, and a calibrated micro-simulator of the Grenoble South Ring. Sensed information comes from a dense wireless sensor network deployed on Grenoble South Ring, providing macroscopic traffic signals such as flows, velocities, densities, and magnetic signatures. This sensor network was set in place in collaboration with Inria spin-off Karrus-ITS, local traffic authorities (DIR-CE, CG38, La Metro), and specialized traffic research centers. In addition to real data, the project also uses simulated data, in order to validate models and to test the ramp-metering; the micro-simulator is a commercial software (developed by TSS AIMSUN ©).

More details at http://necs.inrialpes.fr/pages/grenoble-traffic-lab.php

5.2. NeCSCar

Participants: C. Canudas de Wit [contact person], J. Dumon, V. Ciarla.

NeCSCar is an electrical vehicle (scale 1:3) used as an experimental platform to study new control architectures. The vehicle is designed to be remotely tele-operated from our active steering-wheel platform, ant it will be equipped of a 3D vision system to provide the operator with stereo vision capabilities. Bilateral teleoperation can be performed using wheel contact torque measurements, fed back for force deflexion; wireless connection allows us to test coding algorithms, resource sharing, and robustness against transmission delays.

NeCSCar has been recently used for simulation tests in the framework of the VolHand project, a multi-disciplinary project with the goal to develop a new generation of electronic power assistance steering (EPAS) systems for disabled people.

5.3. Source-seeking robot

Participants: R. Fabbiano [contact person], J. Dumon, Y. Gaudfrin.

The source-seeking algorithms developed in the thesis of Ruggero Fabbiano have been implemented in hardware, with a wheeled robot performing 2-dimensional search. The considered scenario is a source of pollutant in the ocean, where the pollutant can be detected thanks to the fact that it is warmer than water, so that data from an infra-red camera can be used by one or multiple helicopters to move along the ocean surface towards the source. In our experimental equipment, the 2-dimensional movement has been performed with a wheeled vehicle, and the camera was a regular camera, taking pictures of a color-coded image from an actual infra-red image of a pollutant leak. Videos of the experiments are available online: http://necs.inrialpes.fr/pages/platforms.php

NON-A Project-Team

5. Software and Platforms

5.1. SLIM

Multi-robots cooperation can be found as an application in many domains of science and technology: manufacturing, medical robotics, personal assistance, military/security and spatial robots. The market of robots is quickly developing and its capacity is continuously growing. Concerning cooperation of mobile multi-robots, 3 key issues have to be studied: Localization, path planning and robust control, for which Non-A team has worked and proposed new algorithms. Due to the ADT SLIM, we implement our algorithms (localization, path planning and robust control) and integrate them into ROS (Robotic Operating System) as a package, named SLIM.

OPALE Project-Team

5. Software and Platforms

5.1. NUM3SIS

Participants: Régis Duvigneau [correspondant], Nora Aïssiouene, Babett Lekouta.

The Opale project-team has initiated a few years ago the development of NUM3SIS (http://num3sis.inria.fr), which is is a modular platform devoted to scientific computing and numerical simulation. It is not restricted to a particular application field, but is designed to host complex multidisciplinary simulations. Main application fields are currently Computational Fluid Dynamics (by Opale project-team), Computational Electro-Magnetics (by Nachos project-team) and pedestrian traffic simulation (by Opale project-team). Some components of the platform are also used by the Tosca project-team for CO2 market simulation and wind simulation in collaboration with Ciric (Inria-Chile).

NUM3SIS provides innovative software tools to overcome some limitations encountered by classical monolithic simulation codes. In particular, the platform is based on abstract concepts commonly used in scientific computing, such as mesh, fields, finite-elements, linear solvers etc, that can be implemented in plugins. A fast prototyping of algorithms can be achieved using a visual programing interface. A component is dedicated to deployment on parallel architectures. Moreover, the platform relies on a "store" system to foster exchange of plugins, scripts or data.

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

5.2. FAMOSA

Participant: Régis Duvigneau [correspondant].

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

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

The FAMOSA platform is employed by Opale project-team to test its methodological developments. The platform is also used by the Fluid Mechanics Laboratory at Ecole Centrale de Nantes for hydrodynamic design applications and ONERA for multidisciplinary design optimization (MDO). Moreover, it is presently tested by Peugeot Automotive industry for external aerodynamic design purpose.

5.3. Plugins for AXEL

Participant: Régis Duvigneau [correspondant].

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

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

5.4. Integration platform for multidiscipline optimization applications

Participants: Toan Nguyen, Laurentiu Trifan.

A prototype software integration platform is developed and tested for multidiscipline optimization applications. It is based on a workflow management system called YAWL (http://www.yawlfoundation.org). The goal is to design, develop and assess high-performance distributed scientific workflows featuring resilience, i.e., fault-tolerance and exception-handling capabilities. The platform is used to experiment new resilience algorithms, including monitoring and management of application-level errors. Errors include time-outs and out of bounds data values. They can be added and modified by the users. The platform is tested against use-cases provided by the industry partners in the OMD2 project supported by the French Agence Nationale de la Recherche. For example, an optimization of a car air-conditioning pipe was implemented and deployed on the Grid5000 infrastructure. It also takes into account run-time errors related to resource consumption, e.g., memory overflow, to automatically and dynamically relocate the appplications tasks involved on the various clusters. This work was Laurentiu Trifan's PhD thesis, defended in October 2013 [37]. (See Fig. 1 .)

`../../../../projets/opale/IMG/G5K.jpg`

Figure 1. Testcase deployment on the Grid5000 infrastructure.

POEMS Project-Team

5. Software and Platforms

5.1. Software

5.1.1. Introduction

We are led to develop two types of softwares. The first one is prototype softwares : various softwares are developed in the framework of specific research contracts (and sometimes sold to the contractor) or during PhD theses. They may be also contributions to already existing softwares developed by other institutions such as CEA, ONERA or EDF. The second category is an advanced software which are intended to be developed, enriched and maintained over longer periods. Such software is devoted to help us for our own research and/or promote our research. We have chosen to present here our advanced software.

5.1.2. XLiFE++

Participants: Eric Lunéville, Nicolas Kielbasiewicz, Colin Chambeyron, Manh Ha Nguyen.

XLIFE++ is a new Finite Element library in C++ based on philosophy of the previous library MELINA in Fortran but with new capabilities (boundary elements and discontinuous Galerkin methods, more integrated tools – in particular mesh tools – and high performance computing skills, multithread and GPU computation. It is licensed under LGPL and developed in the context of the European project SIMPOSIUM (FP7/ICT, leader CEA/LIST, from september 2011 to august 2014). There are also academic partners : IRMAR, University of Rennes and LAMA, University of Marne-la-Vallée.

In 2012, as a reminder, all development tools were set up and all fundamental and major libraries were done. In 2013, developments have sped up. The Finite Elements, the Spectral Elements and the Boundary Elements computation cores have been implemented and are currently under testing. In addition to the implementation of direct and iterative solvers, an internal eigen solver is operational and coupled to external solver libraries (Arpack++, Umfpack, ...).

As far as inputs/outputs are concerned, XLiFE++ allows to export a solution to the visualization tool PARAVIEW and to read mesh files from GMSH, MELINA and PARAVIEW (vtk). Furthermore, mesh tools have been enriched and a C++ interface to the mesh tool GMSH is under development. XLIFE++ can now solve the Helmholtz equation with Neumann boundary conditions in any mesh. A first version of the library should be published soon.

REALOPT Project-Team

5. Software and Platforms

5.1. BaPCod – a generic Branch-and-Price Code

Participants: Romain Leguay [Software Engineer], Pierre Pesneau, Ruslan Sadykov, François Vanderbeck [correspondant].

BaPCod is a prototype code that solves Mixed Integer Programs (MIP) by application of a Dantzig-Wolfe reformulation technique. The reformulated problem is solved using a branch-and-price (column generation) algorithm. This software platform, made of C++ classes, offers a "*black-box*" implementation that does not require user input and is not application specific. The features are
(i) the automation of the Dantzig-Wolfe reformulation process (the user defines a mixed integer programming problem in a pseudo modeling language, defining variables and constraints, identifying subproblems. He can provide subproblem solvers if available, but he does not need to explicitly define the reformulation, the explicit form of the columns, their reduced cost, or the Lagrangian bounds.
(ii) a default column generation procedure with standard initialization and stabilization [1], [59] [89] [88] [27] and
(iii) a default branching scheme that is generic to all applications [7],
(iv) default primal heuristics specially developed for use in a decomposition framework [64], [79], [90].
The prototype software was/is used as background solver in our application studies and local PhD thesis. It also serves as the framework for our comparative study in a Inria assocaited team project and our transfert projects (the prototype enables us to be very responsive in our industrial contact).

See also the web page https://wiki.bordeaux.inria.fr/realopt/pmwiki.php/Project/BaPCod.

REGULARITY Project-Team

5. Software and Platforms

5.1. FracLab

Participants: Paul Balança, Jacques Lévy Véhel [correspondant].

FracLab was developed for two main purposes:

1. propose a general platform allowing research teams to avoid the need to re-code basic and advanced techniques in the processing of signals based on (local) regularity.
2. provide state of the art algorithms allowing both to disseminate new methods in this area and to compare results on a common basis.

FracLab is a general purpose signal and image processing toolbox based on fractal, multifractal and local regularity methods. FracLab can be approached from two different perspectives:

- (multi-) fractal and local regularity analysis: A large number of procedures allow to compute various quantities associated with 1D or 2D signals, such as dimensions, Hölder and 2-microlocal exponents or multifractal spectra.
- Signal/Image processing: Alternatively, one can use FracLab directly to perform many basic tasks in signal processing, including estimation, detection, denoising, modeling, segmentation, classification, and synthesis.

A graphical interface makes FracLab easy to use and intuitive. In addition, various wavelet-related tools are available in FracLab.

FracLab is a free software. It mainly consists of routines developed in MatLab or C-code interfaced with MatLab. It runs under Linux, MacOS and Windows environments. In addition, a “stand-alone” version (*i.e.* which does not require MatLab to run) is available.

Fraclab has been downloaded several thousands of times in the last years by users all around the world. A few dozens laboratories seem to use it regularly, with more than four hundreds registered users. Our ambition is to make it the standard in fractal softwares for signal and image processing applications. We have signs that this is starting to become the case. To date, its use has been acknowledged in roughly three hundreds research papers in various areas such as astrophysics, chemical engineering, financial modeling, fluid dynamics, internet and road traffic analysis, image and signal processing, geophysics, biomedical applications, computer science, as well as in mathematical studies in analysis and statistics (see http://fraclab.saclay.inria.fr/ for a partial list with papers). In addition, we have opened the development of FracLab so that other teams worldwide may contribute. Additions have been made by groups in Australia, England, France, the USA, and Serbia.

Last year, we produced a major release of FracLab (version 2.1). This year, we corrected a number of bugs.

SCIPORT Team

5. Software and Platforms

5.1. AIRONUM

Participant: Alain Dervieux [correspondant].

AIRONUM is an experimental software that solves the unsteady compressible Navier-Stokes equations with $k-\epsilon$, LES-VMS and hybrid turbulence modelling on parallel platforms with MPI as parallel programming concept. The mesh model is unstructured tetrahedrization, with possible mesh motion. See also http://www-sop.inria.fr/tropics/aironum

- Version: v 1.0
- Programming language: Fortran95 (mostly). About 100,000 lines.

AIRONUM was developed by Inria and university of Montpellier. It is used by Inria, university of Montpellier and university of Pisa (I). AIRONUM is used as an experimental platform for:

- Numerical approximation of compressible flows, such as upwind mixed element volume approximation with superconvergence on regular meshes.
- Numerical solution algorithms for the implicit time advancing of the compressible Navier-Stokes equations, such as parallel scalable deflated additive Schwarz algorithms.
- Turbulence modelling such as the Variational Multiscale Large eddy Simulation and its hybridization with RANS statistical models.

5.2. TAPENADE

Participants: Laurent Hascoet [correspondant], Valérie Pascual, Ala Taftaf.

TAPENADE is an Automatic Differentiation tool that transforms an original program into a new program that computes derivatives of the original program. Automatic Differentiation produces analytical derivatives, that are exact up to machine precision. Adjoint-mode AD can compute gradients at a cost which is independent from the number of input variables. TAPENADE accepts source programs written in Fortran77, Fortran90, or C. It provides differentiation in the following modes: tangent, vector tangent, adjoint, ans vector adjoint. Documentation is provided on the web site of the reserch team http://www-sop.inria.fr/tropics/, in Inria technical report RT-0300, and in [13]. TAPENADE runs under most operating systems and requires installation of Java jdk1.6 or upward.

- Version: v3.8, r4996, November 2013
- ACM: D.3.4 Compilers; G.1.0 Numerical algorithms; G.1.4 Automatic differentiation; I.1.2 Analysis of algorithms
- AMS: 65K10; 68N20
- APP: IDDN.FR.001.040038.002.S.P.2002.000.10600
- Keywords: automatic differentiation, adjoint, gradient, optimisation, inverse problems, static analysis, data-flow analysis, compilation
- Programming language: Java

TAPENADE implements the results of our research about models and static analyses for AD. TAPENADE can be downloaded and installed on most architectures. Alternatively, it can be used as a web server. TAPENADE differentiates computer programs according to the model described in section 3.1 and in [13] Higher-order derivatives can be obtained through repeated application of tangent AD on tangent- and/or adjoint-mode AD.

TAPENADE performs sophisticated data-flow analysis, flow-sensitive and context-sensitive, on the complete source program to produce an efficient differentiated code. Analyses include Type-Checking, Read-Write analysis, and Pointer analysis. AD-specific analysis include:

- **Activity analysis:** Detects variables whose derivative is either null or useless, to reduce the number of derivative instructions.
- **Adjoint Liveness analysis:** Detects the source statements that are dead code for the computation of derivatives.
- **TBR analysis:** In adjoint-mode AD, reduces the set of source variables that need to be recovered.

TAPENADE is not open-source. Academic usage is free. Industrial or commercial usage require a paying license, as detailled on the team's web page. The software has been downloaded several hundred times, and the web tool served several thousands of true connections (not counting robots). The tapenade-users mailing list is over one hundred registered users.

SELECT Project-Team

5. Software and Platforms

5.1. MIXMOD software

Participants: Gilles Celeux [Correspondant], Erwan Le Pennec, Benjamin Auder.

Mixture model, cluster analysis, discriminant analysis

MIXMOD is being developed in collaboration with Christophe Biernacki, Florent Langrognet (Université de Franche-Comté) and Gérard Govaert (Université de Technologie de Compiègne). MIXMOD (MIXture MODelling) software fits mixture models to a given data set with either a clustering or a discriminant analysis purpose. MIXMOD uses a large variety of algorithms to estimate mixture parameters, e.g., EM, Classification EM, and Stochastic EM. They can be combined to create different strategies that lead to a sensible maximum of the likelihood (or completed likelihood) function. Moreover, different information criteria for choosing a parsimonious model, e.g. the number of mixture component, some of them favoring either a cluster analysis or a discriminant analysis view point, are included. Many Gaussian models for continuous variables and multinomial models for discrete variable are available. Written in C++, MIXMOD is interfaced with MATLAB. The software, the statistical documentation and also the user guide are available on the Internet at the following address: http://www.mixmod.org.

Since this 2010, MIXMOD has a proper graphical user interface (Version 1) which has been presented at the MIXMOD day in Lyon in December 2010. A version of MIXMOD in R is now available http://cran.r-project.org/web/packages/Rmixmod/index.html.

Erwan Le Pennec with the help of Serge Cohen has proposed a spatial extension in which the mixture weights can vary spatially.

Benjamin Auder contributes to the informatics improvement of MIXMOD. He implemented an interface to test any mathematical library (Armadillo, Eigen, ...) to replace NEWMAT. He contributed to the continuous integration setup using Jenkins tool and prepared an automated testing framework for unit and non-regression tests.

5.2. BLOCKCLUSTER software

Participants: Vincent Brault, Gilles Celeux, Christine Keribin.

Mixture model, Block cluster analysis,

Blockcluster is a software devoted on model-based block clustering. It is developed by MODAL team (Inria Lille). With Parmeet Bathia (Inria Lille), Vincent Brault has added a Bayesian point of view for the binary, categorial and continuous datas with the variational Bayes algorithm or Gibbs sampler. Criteria ICL and BIC are used for selecting a relevant block clustering.

SequeL Project-Team

5. Software and Platforms

5.1. Computer Games

Participant: Rémi Coulom.

- ***Crazy Stone*** is a top-level Go-playing program that has been developed by Rémi Coulom since 2005. Crazy Stone won several major international Go tournaments in the past. In 2013, a new version was released in Japan. This new version won the 6th edition of the UEC Cup (the most important international computer-Go tournament). It also won the first edition of the Denseisen, by winning a 4-stone handicap game against 9-dan professional player Yoshio Ishida. It is distributed as a commercial product by *Unbalance Corporation* (Japan). 6-month work in 2013. URL: http://remi.coulom.free.fr/CrazyStone/
- ***Kifu Snap*** is an Android image-recognition app. It can automatically recognize a Go board from a picture, and analyze it with Crazy Stone. It was released on Google Play in November, 2013. 6-month work in 2013. URL: http://remi.coulom.free.fr/kifu-snap/

SIERRA Project-Team

5. Software and Platforms

5.1. SPAMS (SPArse Modeling Software)

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

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

5.2. BCFWstruct

Participants: Simon Lacoste-Julien [correspondent], Mark Schmidt.

BCFWstruct is a Matlab implementation of the Block-Coordinate Frank-Wolfe solver for Structural SVMs. See the ICML 2013 paper with the same name.

Participants outside of Sierra: Martin Jaggi (Centre de Mathématiques Appliquées, Ecole Polytechnique); Patrick Pletscher (Machine Learning Laboratory, ETH Zurich)

5.3. SAG

Participant: Mark Schmidt [correspondent].

SAG: Minimizing Finite Sums with the Stochastic Average Gradient.

The SAG code contains C implements (via Matlab mex files) of the stochastic average gradient (SAG) method detailed below, as well as several related methods, for the problem of L2-regularized logistic regression with a finite training set.

The specific methods available in the package are: SGD: The stochastic gradient method with (user-supplied) step-sizes, (optional) projection step, and (optional) (weighted-)averaging. ASGD: A variant of the above code that supports less features, but efficiently implements uniform averaging on sparse data sets. PCD: A basic primal coordinate descent method with step sizes set according the (user-supplied) Lipschitz constants. DCA: A dual coordinate ascent method with a numerical high-accuracy line-search. SAG: The stochastic average gradient method with a (user-supplied) constant step size. SAGlineSearch: The stochastic average gradient method with the line-search described in the paper. SAG-LipschitzLS: The stochastic average gradient method with the line-search and adaptive non-uniform sampling strategy described in the paper.

5.4. fMRI

Participant: Fabian Pedregosa [correspondent].

We showed that HRF estimation improves sensitivity of fMRI encoding and decoding models and propose a new approach for the estimation of Hemodynamic Response Functions from fMRI data. This is an implementation of the methods described in the paper.

SIMPAF Project-Team

5. Software and Platforms

5.1. ns2ddv-M

Participants: Caterina Calgaro [correspondant (Univ. Lille 1)], Emmanuel Creusé [correspondant (Univ. Lille 1)].

Incompressible Navier-Stokes, Variable Density, Rayleigh-Taylor Instability The NS2DDV-M code is based on a hybrid method coupling FV and FE approaches for solving the variable density Navier-Stokes equation in dimension 2. This original approach for variable density flows is described in [41].

Here is the self-assessment of the team effort following the grid provided by Inria (see : http://www.inria.fr/institut/organisation/instances/commission-d-evaluation): A3, SO3-up4, SM2-up3, EM3, SDL4, DA1, CD4, MS4, TPM4.

Software web site : http://math.univ-lille1.fr/~simpaf/SITE-NS2DDV/home.html

5.2. ns2ddv-C++

Participants: Caterina Calgaro [correspondant (Univ. Lille 1)], Emmanuel Creusé [correspondant (Univ. Lille 1)], Thierry Goudon.

Incompressible Navier-Stokes, Variable Density, Kazhikhov-Smagulov model, Rayleigh-Taylor Instability, avalanches phenomena The NS2DVD-C++ code is based on a hybrid method coupling FV and FE approaches for solving the variable density Navier-Stokes equation in dimension 2. The code is developed around the GetFem++ and the Bamg softwares. It allows in particular mesh refinement strategies so that very relevant simulations can be reached (as the falling droplet with very high density ratios, see for example [38]. The current version of the code consider the additional terms in the Kazhikhov-Smagulov model.

Webpage : http://math.univ-lille1.fr/~simpaf/SITE-NS2DDV

Here is the self-assessment of the team effort following the grid provided by Inria (see : http://www.inria.fr/institut/organisation/instances/commission-d-evaluation): A1, SO3-up4, SM1, EM2, SDL1, DA1, CD4, MS4, TPM1.

5.3. RTcodes

Participants: Pauline Lafitte [correspondant (ECP)], Jean-François Coulombel [(CNRS & Univ. Nantes)], Christophe Besse [(Univ. Lille 1)], Thierry Goudon [(Inria)], Giovanni Samaey [(KU Leuven)].

Radiative Transfer, Radiative shocks, AP schemes

We have developed a set of numerical codes, written in Scilab, to compute the solutions of the system coupling the Euler equations to the radiation through energy exchanges, in the non equilibrium regime. This covers several situations in the hierarchy of asymptotic problems. The code treats the one-dimensional framework. In particular the code can be used to investigate radiative shocks profiles. The main advantage of our numerical codes is that they do not require any refinement near the singularities. The numerical tests show a very good agreement with the theoretical predictions. See reference [48].

Here is the self-assessment of the team effort following the grid provided by Inria (see : http://www.inria.fr/institut/organisation/instances/commission-d-evaluation): A2, SO3, SM2, EM1, SDL1.

5.4. FPcodes

Participants: Pauline Lafitte [correspondant (ECP)], Thierry Goudon [(Inria)], Benjamin Boutin [(Univ. Rennes)].

Fluid-Particles flows, Gravity driven flows, AP schemes

We have developed a numerical code, written in Scilab, to compute the solutions of the two-phase flows equations describing particles interacting with a fluid through friction forces. The code treats one-dimensional situation and is well adapted to describe gravity driven flows in either bubbling or flowing regimes. In particular, it can be used to describe the evolution of pollutants in the atmosphere. The numerical strategy, based on a asymptotic-based scheme, is described in details in [43].

Here is the self-assessment of the team effort following the grid provided by Inria (see : http://www.inria.fr/institut/organisation/instances/commission-d-evaluation): A2, SO3, SM2, EM1, SDL1.

5.5. CLAToolBox

Participants: Christophe Besse [correspondant (Univ. Lille 1)], Pauline Klein [Univ. Besançon].

Absorbant boundary conditions, Schrödinger equation

As a byproduct of the review paper [30], a user-friendly interface is offered [1] to trial and compare various numerical methods to solve the 1D Schrödinger equation with absorbant boundary conditions. We also mention [34] for a numerical investigation of blow-up phenomena in the nonlinear Schrödinger equation.

5.6. SPARCS

Participants: Christophe Besse [Univ. Lille 1], Thierry Goudon [correspondant (Inria)], Ingrid Lacroix-Violet [Univ. Lille 1].

Vlasov-Poisson system, Euler-Poisson system. Back-Trajectory method

SPARCS is the code developed by Thales Alenia Space for the simulation of the charge phenomena the spacecrafts are subject to. The current version of the code, according to the PhD thesis of O. Chanrion and M. Chane-Yook performed in collaboration with the team Caiman at Sophia Antipolis, is specialized to geostationary atmospheres. The model consists in the stationary Vlasov-Poisson system, but where instationary effects are taken into account with the boundary condition for the electric field. We participate, in particular through the post doc of N. Vauchelet, to the elaboration of an improved version of the code which includes parallization optimized procedures, the modelling of the natural difference of potential between different dielectric surfaces of the spacecraft, as well as the possible presence of devices emitting charged particles.

5.7. Code-Carmel3D

Participant: Emmanuel Creusé [correspondant (Univ. Lille 1)].

This numerical code, developed in collaboration between EDF R&D and Lille 1 University, is devoted to the electromagnetic fields computation by the use of finite element methods. This code allows in particular to perform nondestructive control by the use of Foucault currents in steam generator pipes, and should be soon coupled with the thermal simulation of Code-Aster. Code-Carmel3D uses the Salomé platform (mesh-generator and post-processing) and Open Turns (uncertainties computation). It will consequently allow to solve multi-physics problems, both for the temporal and harmonic formulations.

---

[1] http://math.univ-lille1.fr/~besse/site/recherche/logiciels/index.html

TAO Project-Team

5. Software and Platforms

5.1. METIS

Participants: Olivier Teytaud [correspondent], Adrien Couëtoux, Jérémie Decock, Jean-Joseph Christophe.

Keywords: Energy, Optimization, Planning.

Many works in Energy Optimization, in particular in the case of high-scale sequential decision making, are based on one software per application, because optimizing the software eventually implies losing generality. Our goal is to develop with Artelys a platform, METIS, which can be used for several applications. In 2012 we interfaced existing codes in Artelys and codes developed in the TAO team; experiments have been performed and test cases have been designed. A main further work is the introduction of generic tools for stochastic dynamic programming into the platform, for comparison and hybridization with other tools from the UCT-SIG.

Our favorite challenge is the hybridization of “classical” tools (based on constraint satisfaction problems, or mixed integer linear programming or mixed integer quadratic programming), which are fast and accurate, with non-linear solvers which can take care of a sophisticated (non-linear) models.

5.2. MoGo

Participants: Olivier Teytaud [correspondent], Jean-Baptiste Hoock.

Keywords:

MoGo and its Franco-Taiwanese counterpart MoGoTW is a Monte-Carlo Tree Search program for the game of Go, which made several milestones of computer-Go in the past (first wins against professional players in 19x19; first win with disadvantageous side in 9x9 Go). Recent results include 7 wins out of 12 against professional players (in Brisbane, 2012) and outpeforming professional players in 7x7. However, the work in the UCT-SIG has now shifted to energy management.

5.3. CMA-ES: Covariance Matrix Adaptation Evolution Strategy

Participant: Nikolaus Hansen [correspondent].

Keywords: Evolutionary Computation, Stochastic Optimization, Real-parameter Optimization.

The Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [72] is considered to be state-of-the-art in continuous domain evolutionary computation [69], and in stochastic optimization at large. It has been shown to be highly competitive on different problem classes even with deterministic continuous algorithms using numerically computed gradients (see the results published on COCO platform). The algorithm is widely used in research and industry as witnessed by hundreds of published applications. We provide source code for the CMA-ES in C, Java, Matlab, Octave, Python, and Scilab including the latest variants of the algorithm.

Link: http://www.lri.fr/~hansen/cmaes_inmatlab.html

5.4. COmparing Continuous Optimizers

Participants: Nikolaus Hansen [correspondent], Anne Auger, Marc Schoenauer, Ouassim Ait Elhara, Asma Atamna.

Keywords: Evolutionary Computation, Stochastic Optimization, Real-parameter Optimization, Benchmarking, Derivative Free Optimization.

COCO (COmparing Continuous Optimizers) is a platform for systematic and sound comparisons of real-parameter global optimizers. COCO provides benchmark function testbeds (noiseless and noisy) and tools for processing and visualizing data generated by one or several optimizers. The code for processing experiments is provided in Matlab, C, Java, and Python. The post-processing code is provided in Python. The code is under continuous developement and has been used for the GECCO 2009, 2010, 2012, and 2013 workshops on "Black Box Optimization Benchmarking" (BBOB) (see Section 6.1). It is now undergoing major changes thanks to the ANR project NumBBO that will add constraint handling and multi-objective benchmarks to the existing platform.

Link: http://coco.gforge.inria.fr/ and http://numbbo.gforge.inria.fr/

5.5. MultiBoost

Participants: Balázs Kégl [correspondent], Djalel Benbouzid.

Keywords: Multi-class, Multi-label Classification.

The MultiBoost package [68] provides a fast C++ implementation of multi-class/multi-label/multi-task boosting algorithms. It is based on ADABOOST.MH but it also implements popular cascade classifiers, ARC-GV, and FILTERBOOST. The package contains common multi-class base learners (stumps, trees, products, Haar filters). Further base learners and strong learners following the boosting paradigm can be easily implemented in a flexible framework.

Link: http://multiboost.org

5.6. Grid Observatory

Participants: Cécile Germain-Renaud [correspondent], Julien Nauroy, Michèle Sebag.

Keywords: Autonomic Computing, Green Computing.

The Grid Observatory (GO) software suite collects and publishes traces of the EGI (European Grid Initiative) grid usage. With the release and extensions of its portal, the Grid Observatory has made a database of grid usage traces available to the wider computer science community since 2008. These data are stored on the grid, and made accessible through a web portal without the need of grid credentials. The GO is fully integrated with the evolution of EGI monitoring. More than 250 users are currently registered. The acquisition has been extended to the University cloud StratusLab hosted by the VirtualData center.

The Green Computing Observatory (GCO) monitors the VirtualData center; it collects data on energy consumption and publishes the data through the Grid Observatory. These data include the detailed monitoring of the processors and motherboards, as well as global site information. The first results on energy saving opportunities have been presented at the Green Days@Luxembourg meeting.

In order to make the GO data readily consistent and complete, as well as understandable for further exploitation, an original approach has been designed, based on a flexible data schema built in collaboration with the users. Its implementation is developed within the FUI project TIMCO. The GO has been supported by an Inria ADT (Action de Développement Technologique) up to September 2013, and by University Paris Sud through the MRM (Moyens de Recherche Mutualisés) program. Stabilization through the VirtualData initiative is currently explored.

Link: http://grid-observatory.org

TOSCA Project-Team

5. Software and Platforms

5.1. SDM

Participant: Mireille Bossy [correspondant].

The computation of the wind at small scale and the estimation of its uncertainties is of particular importance for applications such as wind energy resource estimation. To this aim, starting in 2005, we have developed a new method based on the combination of an existing Numerical Weather Prediction model providing a coarse prediction, and a Lagrangian Stochastic Model for turbulent flows. This Stochastic Downscaling Method (SDM) requires a specific modelling of the turbulence closure, and involves various simulation techniques whose combination is totally original (such as Poisson solvers, optimal transportation mass algorithm, original Euler scheme for confined Langevin stochastic processes, and stochastic particle methods).

In 2013, the SDM code became the kernel of the wind farm modelling of the Fundacion Inria Chile. In France, its development is pursuing through the collaborative Modéol project on the evaluation of wind potential.

This is a joint work with Antoine Rousseau from the project-team MOISE.

- Version: 2.0

5.2. CarbonQuant

Participants: Mireille Bossy [correspondant], Selim Karia.

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 [1] Inria.

See also the web page http://carbonvalue.gforge.inria.fr, from where CarbonQuant can be now downloaded for various architectures.

- Version: 2.0

---

[1] Technology Development Action

CORTEX Team

5. Software and Platforms

5.1. Spiking neural networks simulation

Participants: Dominique Martinez, Yann Boniface.

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. 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.

ABS Project-Team

5. Software and Platforms

5.1. Software

This section briefly comments on all the software distributed by ABS . On the one hand, the software released in 2013 is briefly described as the context is presented in the sections dedicated to new results. On the other hand, the software made available before 2013 is briefly specified in terms of applications targeted.

In any case, the website advertising a given software also makes related publications available.

5.1.1. addict: Stoichiometry Determination from Mass Spectrometry Data

Participants: Deepesh Agarwal, Frédéric Cazals, Noël Malod-Dognin.

Context. Given the individual masses of the proteins present in a complex, together with the mass of that complex, *stoichiometry determination* (SD) consists of computing how many copies of each protein are needed to account for the overall mass of the complex. Our work on the stoichiometry determination (SD) problem for noisy data in structural proteomics is described in [17]. The `addict` software suite not only implements our algorithms `DP++` and `DIOPHANTINE` , but also important algorithms to determine the so-called Frobenius number of a vector of protein masses, and also to estimate the number of solutions of a SD problem, from an unbounded knapsack problem.

Distribution. Binaries for the `addict` software suite are made available from http://team.inria.fr/abs/software/addict/.

5.1.2. vorpatch and compatch: Modeling and Comparing Protein Binding Patches

Participants: Frédéric Cazals, Noël Malod-Dognin.

Context. Modeling protein binding patches, i.e. the sets of atoms responsible of an interaction, is a central problem to foster our understanding of the stability and of the specificity of macro-molecular interactions. We developed a binding patch model which encodes morphological properties, allows an atomic-level comparison of binding patches at the geometric and topological levels, and allows estimating binding affinities—with state-of-the-art results on the protein complexes of the binding affinity benchmark. Given a binary protein complex, `vorpatch` identifies the binding patches, and computes a topological encoding of each patch, defined as an *atom shelling tree* generalizing the core-rim model. The program `compatch` allows comparing two patches via the comparison of their atom shelling trees, by favoring either a geometric or a topological comparison.

Distribution. Binaries for `VORPATCH` and `COMPATCH` are available from http://team.inria.fr/abs/software/vorpatch-compatch.

5.1.3. voratom: Modeling Protein Assemblies with Toleranced Models

Participants: Frédéric Cazals, Tom Dreyfus.

Context. Large protein assemblies such as the Nuclear Pore Complex (NPC), chaperonin cavities, the proteasome or ATP synthases, to name a few, are key to numerous biological functions. Modeling such assemblies is especially challenging due to their plasticity (the proteins involved may change along the cell cycle), their size, and also the flexibility of the sub-units. To cope with these difficulties, a reconstruction strategy known as Reconstruction by Data Integration (RDI), aims at integrating diverse experimental data. But the uncertainties on the input data yield equally uncertain reconstructed models, calling for quantitative assessment strategies.

To leverage these reconstruction results, we introduced TOleranced Model (TOM) framework, which inherently accommodates uncertainties on the shape and position of proteins represented as density maps — maps from cryo electron-microscopy or maps stemming from reconstruction by data integration. In a TOM, a fuzzy molecule is sandwiched between two union of concentric balls, the size of the region between these two unions conveying information on the uncertainties.

The corresponding software package, `VORATOM` , includes programs to (i) perform the segmentation of (probability) density maps, (ii) construct toleranced models, (iii) explore toleranced models (geometrically and topologically), (iv) compute Maximal Common Induced Sub-graphs (MCIS) and Maximal Common Edge Sub-graphs (MCES) to assess the pairwise contacts encoded in a TOM.

Distribution. Binaries for the software package `VORATOM` are made available from http://team.inria.fr/abs/software/voratom/.

5.1.4. intervor: Modeling Macro-molecular Interfaces

Participant: Frédéric Cazals.

In collaboration with S. Loriot (The GEOMETRY FACTORY*)*

Context. Modeling the interfaces of macro-molecular complexes is key to improve our understanding of the stability and specificity of such interactions. We proposed a simple parameter-free model for macro-molecular interfaces, which enables a multi-scale investigation —from the atomic scale to the whole interface scale. Our interface model improves the state-of-the-art to (i) identify interface atoms, (ii) define interface patches, (iii) assess the interface curvature, (iv) investigate correlations between the interface geometry and water dynamics / conservation patterns / polarity of residues.

Distribution. The following website http://team.inria.fr/abs/software/intervor serves two purposes: on the one hand, calculations can be run from the website; on the other hand, binaries are made available. To the best of our knowledge, this software is the only publicly available one for analyzing Voronoi interfaces in macro-molecular complexes.

5.1.5. vorlume: Computing Molecular Surfaces and Volumes with Certificates

Participant: Frédéric Cazals.

In collaboration with S. Loriot (The GEOMETRY FACTORY, *France)*

Context. Molecular surfaces and volumes are paramount to molecular modeling, with applications to electrostatic and energy calculations, interface modeling, scoring and model evaluation, pocket and cavity detection, etc. However, for molecular models represented by collections of balls (Van der Waals and solvent accessible models), such calculations are challenging in particular regarding numerics. Because all available programs are overlooking numerical issues, which in particular prevents them from qualifying the accuracy of the results returned, we developed the first certified algorithm, called `vorlume`. This program is based on so-called certified predicates to guarantee the branching operations of the program, as well as interval arithmetic to return an interval certified to contain the exact value of each statistic of interest—in particular the exact surface area and the exact volume of the molecular model processed.

Distribution. Binaries for `Vorlume` is available from http://team.inria.fr/abs/software/vorlume.

5.1.6. ESBTL: the Easy Structural Biology Template Library

Participant: Frédéric Cazals.

In collaboration with S. Loriot (The GEOMETRY FACTORY, *France) and J. Bernauer (Inria AMIB, France).*

Context. The ESBTL (Easy Structural Biology Template Library) is a lightweight C++ library that allows the handling of PDB data and provides a data structure suitable for geometric constructions and analyses, such as those proposed by `INTERVOR`, `VORPATCH` and `COMPATCH`.

Distribution. The C++ source code is available from http://esbtl.sourceforge.net/http://esbtl.sourceforge.net/.

AMIB Project-Team

4. Software and Platforms

4.1. VARNA

Participants: Yann Ponty [correspondant], Alain Denise.

A lightweight Java Applet dedicated to the quick drawing of an RNA secondary structure. VARNA is open-source and distributed under the terms of the GNU GPL license. Automatically scales up and down to make the most out of a limited space. Can draw multiple structures simultaneously. Accepts a wide range of documented and illustrated options, and offers editing interactions. Exports the final diagrams in various file formats (svg,eps,jpeg,png,xfig) [55]...
VARNA currently ships in its 3.9 version, and consists in ∼50 000 lines of code in ∼250 classes.
Impact: Downloaded ∼10 000 times and is cited by more than ∼170 research manuscripts (source: Google Scholar).
Availability: Distributed under the terms of the GPL v3 licence since 2009 on simple demand to the author(s) at http://varna.lri.fr.

4.2. Cartaj

Participant: Alain Denise [correspondant].

CARTAJ is a software that automatically predicts the topological family of three-way junctions in RNA molecules, from their secondary structure only : :the sequence and the canonical Watson–Crick pairings. The Cartaj software http://cartaj.lri.fr that implements our method can be used online. It is also meant for being part of RNA modelling softwares and platforms. The methodology and the results of CARTAJ are presented in [63]. More than 300 visits since its release in January 2012.

4.3. Rna3Dmotif

Participant: Alain Denise [correspondant].

Rna3Dmotif is a free bundle of three easy-to-install programs aimed to be used in combination to automatically extract recurrent RNA local tertiary motifs. The approach used is based on a graph representation of the RNA tertiary structure using LW nomenclature. It was applied to several widely studied ribosomal RNA structures and the motifs thus found were deposited in a dedicated repository.
Impact: Cited in 17 research manuscripts (source: Google Scholar).
Availability: Distributed under the terms of the licence since 24/03/2009 on simple demand to the author(s) at http://rna3dmotif.lri.fr.

4.4. GenRGenS

Participants: Yann Ponty [correspondant], Alain Denise.

A software dedicated to the random generation of sequences. Supports different lasses of models, including weighted context-free grammars, Markov models, PROSITE patterns... [72] GENRGENS currently ships in its 2.0 version, and consists in ∼25 000 lines of code in ∼120 Java classes.
Impact: Downloaded ∼5 000 times and is cited by more than ∼50 research manuscripts (source: Google Scholar).
Availability: Distributed under the terms of the GPL v3 licence since 2006 on simple demand to the author(s) at https://www.lri.fr/ genrgens/.

4.5. DiMoVo

Participant: Julie Bernauer [correspondant].

DiMoVo, *DIscriminate between Multimers and MOnomers by VOronoi tessellation* : Knowing the oligomeric state of a protein is necessary to understand its function. his tool, accessible as a webserver and still used and maintained, provides a reliable discrimination function to obtain the most favorable state of proteins.
Availability : released in 2008.

4.6. VorScore

Participant: Julie Bernauer [correspondant].

VorScore, *Voronoi Scoring Function Server* : Scoring is a crucial part of a protein-protein procedure and having a quantitave function to evaluate conformations is mandatory. This server provides access to a geometric knowledge-based evaluation function. It is still maintained and widely used. See Bernauer et al., Bioinformatics, 2007 23(5):555-562 for further details.

4.7. GeneValorization

Participants: Bryan Brancotte, Sarah Cohen-Boulakia [correspondant].

High-throughput technologies provide fundamental informations concerning thousands of genes. Most of the current biological research laboratories daily use one or more of these technologies and identify lists of genes. Understanding the results obtained includes accessing to the latest publications concerning individual or multiple genes. Faced to the exponential growth of publications avaliable, this task is becoming particularly difficult to achieve.
Here, we introduce a web-based Java application tool named GeneValorization which aims at making the most of the text-mining effort done downstream to all high throughput technology assays. Regular users come from the Curie Institute, but also the Ebi.
Impact : 925 distinct international users have used GeneValorization and about a hundred use it on a regular basis. The tool is on average used once to twice every day.
Availability : it is available at http://bioguide-project.net/gv with Inter Deposit Digital Number (*depot APP*, June 2013).

4.8. SPFlow

Participant: Sarah Cohen-Boulakia [correspondant].

Scientific workflow systems are numerous and equipped of provenance modules able to collect data produced and consumed during workflow runs to enhance reproducibility. An increasing number of approaches have been developed to help managing provenance information. Some of them are able to process data in a polynomial time but they require workflows to have series-parallel (SP) structures. Rewriting any workflow into an SP workflow is thus particularly important.
Spflow answers this need and takes in a workflow (from the Taverna system) and provide a runnable and provenance equivalent (Taverna) workflow."
Impact: The tool is currently used by Taverna's users from the University of Manchester and more generally by myExperiment users.
Availability: Distributed under the terms of the licence since 04/02/2013 on simple demand to the author(s) at http://www.lri.fr/ chenj/SPFlow/.

4.9. SPChecker

Participant: Sarah Cohen-Boulakia [correspondant].

Scientific workflow systems are numerous and equipped of provenance modules able to collect data produced and consumed during workflow runs to enhance reproducibility. An increasing number of approaches have been developed to help managing provenance information. Some of them are able to process data in a polynomial time but they require workflows to have series-parallel (SP) structures.
SPChecker is able to detect whether or not any Taverna workflow has a series-parallel structure.
Impact: The tool is currently used by Taverna's users from the University of Manchester and more generally by myExperiment users (a collaboration with Manchester has started and should significantly augment the number of potential users).
Availability: Distributed under the terms of the licence since 01/02/2013 on simple demand to the author(s) at http://www.lri.fr/ chenj/SPChecker/.

4.10. BioGuide

Participants: Sarah Cohen-Boulakia [correspondant], Christine Froidevaux.

BioGuide/BioGuideSRS : this software helps the scientists choose suitable sources and tools, find complementary information in sources, and deal with divergent data.
Reference : Sarah Cohen-Boulakia, Olivier Biton, Susan Davidson, Christine Froidevaux, BioGuideSRS: Querying Multiple Sources with a user-centric perspective, Bioinformatics, March, 23(10), 1301-1303, 2007.
Impact: The paper related to the tool has been cited by ∼26 research manuscripts (source: Google Scholar) so far. Since 2007 and up to now, BioGuide has 8,030 distinct users including regular users from the EBI (European Bioinformatics Institute), the Institut Curie and the Children's Hospital of Philadelphia.
Availability: Distributed under the terms of the licence since 01/09/2006 on simple demand to the author(s) at http://bioguide-project.net/.

4.11. HSIM

Participant: Patrick Amar [correspondant].

HSIM (Hyperstructure Simulator) is a simulation tool for studying the dynamics of biochemical processes in a virtual bacteria. The model is given using a language based on probabilistic rewriting rules that mimics the reactions between biochemical species. HSIM is a stochastic automaton that implements an entity-centered model of objects. This kind of modelling approach is an attractive alternative to differential equations for studying the diffusion and interaction of the many different enzymes and metabolites in cells which may be present in either small or large numbers.

The new version of HSIM includes a Stochastic Simulation Algorithm *a la* Gillespie that can be used with the same model in a standalone way or in a mixed way with the entity-centered algorithm. This new version offers also the possibility to export the model in SciLab for a ODE integration. Last, HSIM can export the differential equations system, equivalent to the model, to LaTeX for pretty-printing.

This software is freely available at http://www.lri.fr/~pa/Hsim; A compiled version is available for the Windows, Linux and MacOSX operating systems.

ANGE Team

5. Software and Platforms

5.1. FRESHKISS

Although the Saint-Venant system is the cornerstone of flow modelling in geosciences, this does not mean that the transfer of the efficient dedicated simulation tools is achieved in the geoscience community.

ANGE collaborates with scientists, laboratories and companies that are interested in scientific advances which makes the valuation and the transfer of results easier.

The development of robust and efficient numerical tools has been a strong point of the activities within the BANG project-team. ANGE aims at pursuing this effort as most publications of the team members contain both modelling and simulation/validation aspects. For the simulation of the free surface Navier-Stokes equations, numerical tools have been developed namely FRESHKISS2D[1] and FRESHKISS3D. These tools are used by several scientists typically in the BIOCORE Inria project-team, at EDF and in public research laboratories.

FRESHKISS3D is a numerical code solving the 3D hydrostatic and incompressible Navier-Stokes equations with variable density. This code was initially dedicated to research activities within the team but we now aim at turning it into a numerical tool being used by non-mathematicians. Indeed, there is a demand in research laboratories and companies to use this tool. A young engineer (R. Hamouda) has been hired (ADT In@lgae funded by Inria) and its assignment is to improve/enrich the code and to make it user-friendly. Notice that FRESHKISS3D is used for teaching (master students in geosciences) at university Denis Diderot Paris 7 and IPGP.

---

[1] FRESHKISS: FREe Surface Hydrodynamics using KInetic SchemeS

ARAMIS Team

5. Software and Platforms

5.1. SACHA

Participants: Marie Chupin [Correspondant], Ludovic Fillon.

SACHA (“Segmentation Automatisée Compétitive de l’Hippocampe et de l’Amygdale”) is a software for the fully automatic segmentation of the hippocampus and the amygdala from MRI 3D T1 brain scans. It has been validated in various populations including healthy controls and patients with Alzheimer’s disease, epilepsy and depression. It has been successfully applied to over 3,000 subjects, both controls, from adolescents to elderly subjects, and patients with different types of pathologies. The current stable version is fully automatic and focused on cross-sectional segmentation. The software can be used both as a command-line program or through a graphical user interface (GUI). The core of the program is coded in C++. It has a dependency to the AIMS library (http://www.brainvisa.info) and preprocessing steps rely on processes in Matlab from SPM (http://www.fil.ion.ucl.ac.uk/spm/). The GUI is coded in Python and is based on BrainVISA (http://www.brainvisa.info). The software has been registered at the APP (French agency for software protection).

5.2. WHASA

Participants: Marie Chupin [Correspondant], Ludovic Fillon, Thomas Samaille.

WHASA (“White matter Hyperintensity Automatic Segmentation Algorithm”) is a software for the fully automatic segmentation of age-related white matter hyperintensities from MRI FLAIR and 3D T1 brain scans. It has been validated on a population showing a wide range of lesion load, and is being further evaluated on elderly subjects with few clinical abnormalisties and with different acquisition characteristics. The current stable version is fully automatic and focused on cross-sectional segmentation. The software can be used both as a Matlab command-line or through a graphical user interface (GUI). The core of the program is coded in Matlab. It has a dependency to the SPM environment (http://www.fil.ion.ucl.ac.uk/spm/). The GUI is coded in Python and is based on BrainVISA (http://www.brainvisa.info). The software has been registered at the APP (French agency for software protection).

5.3. Deformetrica

Participants: Stanley Durrleman [Correspondant], Alexandre Routier, Pietro Gori.

Deformetrica is a software which estimates diffeomorphic deformations between sets of geometric objects in 2D and 3D. Those deformations are estimated either for the registration of two of such objects sets or for the construction of an atlas from several of such sets (a template model set and deformations mapping the template model to each set). Geometric objects could be grey-level images, surface meshes, polygonal lines or unstructured point sets. The method relies on the metric on currents for the comparison of point sets and the sum of squared differences for the comparison of images.

The software is written in C++ and relies on the ITK and VTK libraries. It is a command-line software.

The release of the software to the scientific community is planned for 2014.

5.4. qualiCATI

Participants: Marie Chupin [Correspondant], Hugo Dary, Nicolas Vibet, Urielle Thoprakarn, Aude Costard, Amadou Tall, Cyril Poupon, Vincent Perlbarg, Mélanie Pélégrini-Issac.

qualiCATI is a software designed for comprehensive quality control of multimodal MRI data acquisition in large multicentre clinical studies. The software is built as a platform receiving several modules, developped by several CATI engineers. The first module is dedicated to acquisition requirement checking and conversion to nifti format. The second module aims at making 3DT1 acquisition quality check more systematic, and relies both on visual inspection and quantitative indices. The third module allows a simultaneous evaluation of the clinical part of the CATI acquisition protocol. The fourth module embeds automatic indices to evaluate resting state fMRI acquisition. The last module, up to now, is dedicated to first preprocessings and quality indices for dMRI. qualiCATI requires training for the visual parts, and is closely linked with a team of clinical research assistants. It has been used to analyse over 3000 subjects from over 10 multi centre research projects initiated before or after the CATI started. Other modules will be added in the future to embed new aspects of the MRI protocol proposed by the CATI. The Aramis team is in charge of the second and third modules and jointly in charge of the first module. The software is centered on a graphical user interface (GUI). The whole program is coded in Python within the pyPTK environment developped by Cyril Poupon (Neurospin). It has dependencies to SPM (http://www.fil.ion.ucl.ac.uk/spm/) and brainVISA environments as well as specific tools for DICOM management.

ASCLEPIOS Project-Team

4. Software and Platforms

4.1. SOFA

Participants: Hervé Delingette [correspondant], Brina Goyette, Federico Spadoni, Stéphanie Marchesseau, Hugo Talbot.

SOFA is an Open Source framework primarily targeted at real-time simulation, with an emphasis on medical simulation. It is mostly intended for the research community to help develop new algorithms, but can also be used as an efficient prototyping tool. Based on an advanced software architecture, it allows : the creation of complex and evolving simulations by combining new algorithms with algorithms already included in SOFA; the modification of most parameters of the simulation (deformable behavior, surface representation, solver, constraints, collision algorithm, etc.) by simply editing an XML file; the building of complex models from simpler ones using a scene-graph description; the efficient simulation of the dynamics of interacting objects using abstract equation solvers; the reuse and easy comparison of a variety of available methods. It was developed mainly by the Inria team projects Shaman, Evasion and Asclepios.
See also the web page http://www.sofa-framework.org/.

- ACM: J.2 Physics, J.3 LIFE AND MEDICAL SCIENCES
- Software benefit:- Simulation of the human body
- License: GPL
- License: LGPL
- Type of human computer interaction: console, opengl, qt
- OS/Middelware: linux, windows, mac
- Required library or software: Qt - GPL - GLEW - BSD/MIT - Tinyxml - zlib
- Programming language: C/C++
- Documentation: - each function of the core API and each class in the SOFA modules - doxygen
- ACM: J.3
- Programming language: C/C++

4.2. MedInria

Participants: Maxime Sermesant [Correspondant], Florian Vichot, Moulay Fadil, Loïc Cadour.

MedInria is a medical imaging software platform developed by the Asclepios research project in collaboration with the Athena, Parietal and Visages Inria research projects. It aims at providing clinicians with state-of-the-art algorithms dedicated to medical image processing and visualization. Efforts have been made to simplify the user interface, while keeping high-level algorithms.

The core of medInria is open source with a BSD license; additional plug-ins can have any license.

The latest release of medInria, 2.1.2, was made in September 2013. See also the web page http://med.inria.fr.

- Version: 2.1.2
- License: BSD
- Keywords: Medical Image Processing
- Dependencies: Qt, DTK, VTK, ITK, TTK, MIPS
- Programming language: C++
- Supported OSes: Windows (XP/Vista/7/8), Linux (Fedora/Ubutu), Mac OS X (10.6-10.9)

ATHENA Project-Team

5. Software and Platforms

5.1. OpenMEEG

Participants: Théodore Papadopoulo, Maureen Clerc, Alexandre Gramfort [Telecom ParisTech].

OpenMEEG provides state-of-the art tools for low-frequency bio-electromagnetism, notably solving forward problems related to EEG and MEG [57], [58]. It implements the symmetric BEM which provides excellent accuracy and versatility. OpenMEEG is a free open software written in C++. It can be accessed either through a command line interface or through Python/Matlab interfaces. The first release has been directly downloaded about 600 times since October 2008. Our last release (in September 2011) has been downloaded more than 2000 times to this date. OpenMEEG has been integrated in the neuro-debian distribution (http://neuro.debian.net/) and matlab suites (such as BrainStorm, FieldTrip or SPM) which may represent several more indirect downloads. Work is under progress to integrate it in a commercial package (BESA).

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

- Version: 2.2
- License: French opensource license CeCILL-B
- Multiplatform: Windows - Linux - MacOSX
- Programming language: C++
- 17 000 lines of code.
- 1800 downloads in 2012-2013.
- Web: http://openmeeg.gforge.inria.fr

5.2. Diffusion MRI

Participants: Aurobrata Ghosh, Théodore Papadopoulo, Rachid Deriche.

We have been closely involved in pushing the frontiers of the diffusion MRI (dMRI) in the recent years, especially in the mathematical modelling and processing of the dMRI signal and have developed state-of-the-art software implementations in the form of a C++ library that can be effectively used to infer the complex microstructure of the cerebral white matter. These algorithms and software fall into four categories : (i) local tissue modelling, which includes both popular 2nd order models and advanced higher than 2nd order models such as DTI, higher order Cartesian tensors (HOTs), ODF, FOD, EAP, maxima extraction, regularization and segmentation; (ii) generation of scalar indices (or biomarkers), which include DTI biomarkers, Diffusion Kurtosis Imaging (DKI) and invariants of 4th order tensors; (iii) global structure estimation, which includes deterministic and probabilistic tractography; and (iv) data visualisation for scalar indices, local models and global structures.

So far, ODF estimation from the ATHENA-dMRI C++ library has been successfully included in medInria 1.9, and in the process to be re-adapted for medInria 2.1. Otherwise, the ATHENA-dMRI C++ library has been mostly used internally for research purposes. However, this is now changing with a fresh restructuring of the entire library so that it can be successfully ported and used externally – primarily to be included in parts with the cutting-edge software developed by OLEA MEDICAL.

- License: French opensource license CeCILL-B - To change when it is to be sourced to OLEA MEDICAL.
- Platform: Linux and (medInria platforms)
- Programming language: C++

5.3. medInria

Participants: Jaime Garcia Guevara, Théodore Papadopoulo.

The ATHENA team is heavily involved in the development of medInria 2.0 along with the ASCLEPIOS, PARIETAL and VISAGES research teams. medInria is a free software platform dedicated to medical data visualization and processing. medInria 2.0, it is a complete re-write of the first version of medInria in order to be modular and allow a distributed development. It aims at providing an integrative platform for medical image processing and to be a framework for disseminating various research tools not only to other researchers but also to clinicians. New algorithms or data formats can be added as plugins.

It aims at providing to clinicians and researchers state-of-the-art algorithms developed at Inria and elsewhere (for the future), through an intuitive user interface. medInria offers from standard to cutting-edge processing functionalities for medical images such as 2D/3D/4D image visualization, image registration, diffusion MR processing and tractography.

ATHENA's contributions so far consist in various improvements on the infrastructure, the core application as well as several plugins which are already available with version 2.1 (ODF vizualization) or in future ones: advanced dMRI processing, M/EEG signal visualisation (by integrating code from the software AnyWave developed by Bruno Colombet and J.-M. Badier INSERM U1106 and Aix-Marseille University).

In 2013, the source code of the core of medInria was made public. Regular releases and bug fixes are provided on a large number of Linux, Windows and Mac versions, thanks to the Continuous Integration platform proposed at Inria.

After 4 years of important development, medInria is now rather mature and can be used as a basis for collaborations and projects. We now receive regular feedback through the forum and the mailing list, from both academic and clinical users.

- Version: 2.1
- Keywords: Medical Image Processing and Visualization
- License: BSD 4
- Multiplatform: Windows - Linux - MacOSX
- Programming language: C++
- 250 000 lines of code.
- 5000 downloads on 2012-2013.
- Web: http://med.inria.fr.

5.4. FindSources3D

Participants: Maureen Clerc, Juliette Leblond [APICS project-team], Jean-Paul Marmorat [APICS project-team], Théodore Papadopoulo.

FindSources3D is a Matlab software program dedicated to solving inverse source localization problems in electroencephalography (EEG), and in the future, magnetoencephalography (MEG). FindSources3D implements a new formalism for source localization, based on rational approximations in the complex plane. It is able to estimate, with high precision, and with no a priori on the number of sources, pointwise dipolar current sources within the brain. The head model used is a spherical model with concentric layers of homogenous conductivity.

Contributors: APICS and ATHENA Project Teams, Inria Sophia-Antipolis Méditerranée, Centre de Mathématiques Appliquées (CMA), Ecole des Mines de Paris.

- Version: 1.0
- Keywords: Medical Image Processing and Visualization
- License: CeCILL
- Multiplatform: Windows - Linux - MacOSX
- Programming language: Matlab
- Web: http://www-sop.inria.fr/apics/FindSources3D/fr/index.html

5.5. ImplicitFEM

Participants: Théodore Papadopoulo, Sylvain Vallaghé.

ImplicitFEM is a software to simulate the forward EEG/MEG problem. It uses a volumic finite element approach (FEM) that allows the modeling of anisotropic conductivities (which OpenMEEG cannot). Its main originality is to avoid the need of meshes that can be very complicated to build for the head. Instead, it uses directly representations of tissue interfaces as levelsets (that can be provided directly by some segmentation program based on levelsets or can be generated from other representations). It also uses non-differentiable elements so as to properly model continuity of both potential and normal current across the tissues interfaces (which correspond to conductivity discontinuities). This tool is currently used only internally by students and researchers.

- Version: 0.5
- Programming language: C++

5.6. External Stimulator for OpenViBE

Participants: Maureen Clerc, Loïc Mahé, Dieter Devlaminck.

In the domain of Brain Computer Interfaces, extracting relevant features requires a precise timing of all events occurring in the system. In particular, when dealing with evoked responses as in the P300 speller, the timing of the visual stimulations must be well controlled. To alleviate some timing issues with the P300 speller initially provided with OpenViBE, we have implemented an external visual stimulator that allows to flash the visual targets, in a time-robust manner.

- Version: 1.0
- Keywords: Brain Computer Interfaces
- Multiplatform: Windows - Linux - MacOSX
- Programming language: C++

BAMBOO Project-Team

5. Software and Platforms

5.1. AcypiCyc

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

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

5.2. AlViE

Participants: Pierluigi Crescenzi [Contact, pierluigi.crescenzi@unifi.it, ext. member EPI], Giorgio Gambosi, Roberto Grossi, Carlo Nocentini, Tommaso Papini, Walter Verdese.

ALVIE is a post-mortem algorithm visualization Java environment, which is based on the interesting event paradigm. The current distribution of ALVIE includes more than forty visualizations. Almost all visualizations include the representation of the corresponding algorithm C-like pseudo-code. The ALVIE distribution allows a programmer to develop new algorithms with their corresponding visualization: the included Java class library, indeed, makes the creation of a visualization quite an easy task (once the interesting events have been identified).
http://piluc.dsi.unifi.it/alvie/

5.3. Cassis

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

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

5.4. Coala

Participants: Christian Baudet [EPI, Contact, christian.baudet@univ-lyon1.fr], Pielrluigi Crescenzi, Bea Donati [EPI, Contact, bea.donati@inria.fr], Christian Gautier [EPI], Catherine Matias, Blerina Sinaimeri [EPI, Contact, blerina.sinaimeri@inria.fr], Marie-France Sagot [EPI, Contact, marie-france.sagot@inria.fr].

COALA stands for “CO-evolution Assessment by a Likelihood-free Approach”. It is thus a likelihood-free method for the co-phylogeny reconstruction problem which is based on an Approximative Bayesian Computation (ABC).
http://coala.gforge.inria.fr/

5.5. C3Part & Isofun

Participants: Frédéric Boyer, Yves-Pol Deniélou, Anne Morgat [EPI, ext. member], Marie-France Sagot [EPI], Alain Viari [EPI, Contact, alain.viari@inria.fr].

The C3Part / Isofun package implements a generic approach to the local alignment of two or more graphs representing biological data, such as genomes, metabolic pathways or protein-protein interactions, in order to infer a functional coupling between them. It is based on the notion of “common connected components” between graphs.http://www.inrialpes.fr/helix/people/viari/lxgraph/index.html

5.6. CycADS

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

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

5.7. Eucalypt

Participants: Christian Baudet [EPI, Contact, christian.baudet@univ-lyon1.fr], Pielrluigi Crescenzi, Bea Donati [Contact, bea.donati@inria.fr], Blerina Sinaimeri, Marie-France Sagot [EPI].

Algorithm for enumerating all optimal (possibly time-unfeasible) mappings of a parasite tree unto a host tree.
http://eucalypt.gforge.inria.fr/

5.8. Gobbolino & Touché

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

Designed to solve the metabolic stories problem, which consists in finding all maximal directed acyclic subgraphs of a directed graph G whose sources and targets belong to a subset of the nodes of G, called the black nodes. Biologically, stories correspond to alternative metabolic pathways that may explain some stress that affected the metabolites corresponding to the black nodes by changing their concentration (measured by metabolomics experiments).
http://gforge.inria.fr/projects/gobbolino

5.9. KisSNP

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

Algorithm for identifying SNPs without a reference genome by comparing raw reads.
http://alcovna.genouest.org/kissnp/

5.10. kisSplice & KisSplice2igv

Participants: Lilia Brinza [EPI], Rayan Chikhi, Alice Julien-Lafferière [EPI], Janice Kielbassa, Vincent Lacroix [Contact, EPI], Camille Marchet [EPI], Claire Lemaitre, Pierre Peterlongo, Gustavo Sacomoto [EPI], Marie-France Sagot [EPI], Raluca Uricaru.

Enables to analyse RNA-seq data with or without a reference genome. It is an exact local transcriptome assembler, which can identify SNPs, indels and alternative splicing events. It can deal with an arbitrary number of biological conditions, and will quantify each variant in each condition. KISSPLICE2IGV is a pipeline that combines the outputs of KISSPLICE to a reference transcriptome (obtained with a full-length transcriptome assembler or a reference database). It provides a visualisation of the events found by KISSPLICE in a longer context using a genome browser (IGV).
http://kissplice.prabi.fr/

5.11. LASAGNE

Participants: Pierluigi Crescenzi [Contact, pierluigi.crescenzi@unifi.it, ext. member EPI], Roberto Grossi, Michel Habib, Claudio Imbrenda, Leonardo Lanzi, Andrea Marino.

LASAGNE is a Java application which allows the user to compute distance measures on graphs by making a clever use either of the breadth-first search or of the Dijkstra algorithm. In particular, the current version of LASAGNE can compute the exact value of the diameter of a graph: the graph can be directed or undirected and it can be weighted or unweighted. Moreover, LASAGNE can compute an approximation of the distance distribution of an undirected unweighted graph. These two features are integrated within a graphical user interface along with other features, such as computing the maximum (strongly) connected component of a graph.
http://amici.dsi.unifi.it/lasagne/?page_id=324

5.12. MetExplore

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

Web server to link metabolomic experiments and genome-scale metabolic networks.
http://metexplore.toulouse.inra.fr/metexplore/

5.13. Migal

Participants: Julien Allali [Contact, julien.allali@labri.fr], Marie-France Sagot [EPI, Contact, marie-france.sagot@inria.fr].

RNA, tree comparison
Algorithm for comparing RNA structures.
http://www-igm.univ-mlv.fr/~allali/logiciels/index.en.php

5.14. Mirinho

Participants: Cyril Fournier [EPI], Susan Higashi [EPI, Contact, susan.higashi@inria.fr], Christian Gautier [EPI], Christine Gaspin, Marie-France Sagot [EPI].

Predicts, at a genome-wide scale, microRNA candidates.
http://mirinho.gforge.inria.fr/

5.15. MotusWEB

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

Algorithm for searching and inferring coloured motifs in metabolic networks (web-based version - offers different functionalities from the downloadable version).
http://pbil.univ-lyon1.fr/software/motus_web/

5.16. Motus

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

Algorithm for searching and inferring coloured motifs in undirected graphs (downloadable version - offers different functionalities from the web-based version).
http://pbil.univ-lyon1.fr/software/motus/

5.17. PhEVER

Participants: Christian Gautier [EPI], Vincent Lotteau, Leonor Palmeira [Contact, mlpalmeira@ulg.ac.be], Chantal Rabourdin-Combe, Simon Penel.

Database of homologous gene families built from the complete genomes of all available viruses, prokaryotes and eukaryotes and aimed at the detection of virus/virus and virus/host lateral gene transfers.
http://pbil.univ-lyon1.fr/databases/phever/

5.18. PepLine

Participants: Jérôme Garin, Alain Viari [EPI, Contact, alain.viari@inria.fr].

Pipeline for the high-throughput analysis of proteomic data.

5.19. Pitufo and family

Participants: Vicente Acuña [EPI], Ludovic Cottret [Contact, Ludovic.Cottret@toulouse.inra.fr], Alberto Marchetti-Spaccamela [EPI, ext. member], Paulo Vieira Milreu [EPI, Contact, pvmilreu@gmail.com], Marie-France Sagot [EPI], Leen Stougie [EPI, ext. member], Fabio Viduani-Martinez.

Algorithms to enumerate all minimal sets of precursors of target compounds in a metabolic network.
http://sites.google.com/site/pitufosoftware/

5.20. Repseek

Participants: Guillaume Achaz [Contact, achaz@abi.snv.jussieu.fr], Eric Coissac, Alain Viari [EPI].

Finding approximate repeats in large DNA sequences.
http://wwwabi.snv.jussieu.fr/public/RepSeek/

5.21. Smile

Participants: Laurent Marsan, Marie-France Sagot [EPI, Contact, marie-france.sagot@inria.fr].

Motif inference algorithm taking as input a set of biological sequences.

5.22. Tuiuiu

Participants: Alair Pereira Do Lago, Pierre Peterlongo [Contact, pierre.peterlongo@inria.fr], Nadia Pisanti, Gustavo Sacomoto [EPI], Marie-France Sagot [EPI].

Multiple repeat search filter with edit distance.
http://mobyle.genouest.org/cgi-bin/Mobyle/portal.py?form=tuiuiu

5.23. UniPathway

Participants: Eric Coissac, Anne Morgat [EPI, Contact, anne.morgat@inria.fr], Alain Viari [EPI].

Database of manually curated pathways developed with the Swiss-Prot group.
http://www.unipathway.org

BANG Project-Team

5. Software and Platforms

5.1. Software and Platforms

5.1.1. Continuation of M3N

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

5.1.2. CellSys

Participants: Géraldine Cellière [PhD student], Dirk Drasdo [correspondent], Stefan Höhme, Adrian Friebel [PhD student, University of Leipzig], Tim Johann [Software Engineer, University of Leipzig], Johannes Neitsch [PhD student], Paul Van Liedekerke [Research Engineer].

Based on an earlier submitted software (Hoehme and Drasdo, Bioinformatics, 2010) a modular computer simulation software for image analysis of tissue samples at histological scales, as well as for individual cell (agent)-based modeling of tumour and tissue growth, and tissue regeneration has been developed. Cell movement is solved either by systems of coupled equations of motion for each individual cell or by Kinetic Monte Carlo methods. The software uses a git framework to facilitate coordinated contributions of multiple developers. The image analysis part allows analysis of structures down to sub-cellular scale such as liver micro-capillaries and bile cannaliculi structures. So far, blood flow as well as growth and regeneration processes, fluxes of chemicals by diffusion and flow etc can be modelled, finite element solvers, ITK and VTK have been integrated.

The software CellSys is calibrated to allow use by external and internal researchers. The idea is to perspectively go open-source and offer consultancy for potential users.

Moreover in 2013 the image processing and analysis chain was refined to capture high resolution laser scanning micrographs. The algorithms were integrated into CELLSYS (see: software) and our experimental partner labs within the projects VLN and NOTOX were provided with the software to allow image analysis directly in their lab and with their people. Along the same line an experimental partner lab at the German Cancer center was provide with a small image analysis tool permitting them to efficiently analyze their bright field images on growing and invasive cancer cell populations in vitro (LUNGSYS).

BEAGLE Project-Team

5. Software and Platforms

5.1. aevol (artificial evolution)

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

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

5.2. FluoBacTracker

Participants: Hugues Berry, David P Parsons, Magali Vangkeosay.

- Contact: Hugues Berry (hugues.berry@inria.fr)
- FluoBacTracker is a software for automated quantification of bacterial cells in microscopy movies, developed in collaboration with INSERM U1001 and Paris 5 MAP (Applied Mathematics) Labs. The development (started october 2012) is supported by is a 2-year grant (ADT) funded by Inria's Technological Development Department (Sept 2012- July 2014, project name: "MultiPop"). We hope this software will be useful to all the experimental biology labs that tries to derive single-cell data from bacteria growth microscopy movies. Co-developers include Magali Vangkeosay (Beagle), David P Parsons (SED, Inria Grenoble) and Xiaohu Song (INSERM U1001).

5.3. Ancestral Genome Reconstructions

Participant: Eric Tannier.

- Contact: Eric Tannier (eric.tannier@inria.fr).
- We participated in the development of a series of softwares for genome organization analysis:
  - ANGES, for ANcestral GEnomeS maps, is a toolkit for ordering ancestral genomic markers in chromosomes. An application note has been published in *Bioinformatics* in 2012 to advertise its first release. It is hosted at SFU in Vancouver, URL: http://paleogenomics.irmacs.sfu.ca/ANGES/, under a GNU license, 2012.
  - DeCo and DeCoLT, for Detection of Co-evolution (with Lateral gene Transfer), reconstruct neighborhood relationships between genes of ancient genomes, in the presence of gene duplications, transfer and losses. Both are hosted at the PRABI, the bioinformatics platform in Lyon, under a Cecill license, 2012 and 2013. URL: http://pbil.univ-lyon1.fr/software/DeCo/ and http://pbil.univ-lyon1.fr/software/DeCoLT/.
  - DCJ2HP provides bayesian samples of rearrangements scenarios between two genomes. It is hosted at the Renyi Institute in Budapest. URL: http://www.renyi.hu/~miklosi/DCJ2HP/

5.4. DMT4SP mining tool

Participant: Christophe Rigotti.

- Contact: Christophe Rigotti (christophe.rigotti@insa-lyon.fr).
- DMT4SP (Data-Mining Tool For Sequential Patterns) – DMT4SP is command-line tool to extract episodes and episode rules over a single sequence or several sequences of events. It allows to specify constraints on the episodes or on the rules. Three kinds of patterns can be extracted: (1) serial episodes, (2) serial episode rules having a single event type in the consequent, and (3) quantitative episodes (aka grouping of "homogeneous" occurrences of serial episodes with respect to the time gap between events). DMT4SP is a prototype that is freely distributed (http://liris.cnrs.fr/~crigotti/dmt4sp.html).

BIGS Project-Team

5. Software and Platforms

5.1. Light diffusion into tissues

We are currently considering the possibility to implement our Matlab algorithms concerning light diffusion into tissues into the Matlab toolbox *Contsid*, developed by the System Identification team of the CRAN (http://www.iris.cran.uhp-nancy.fr/contsid/).

5.2. Online data analysis

An R package performing most of the methods of factorial analysis in an online way has been developed by R. Bar and J-M. Monnez. Starting from a simulated data flow, the main goal of the program is to perform online factorial analyses (Principal Component Analyses, Canonical Correlation Analysis, Canonical Discriminant Analysis, Correspondence Analysis). Data are supposed to be independent and identically distributed observations of a random vector (whose distribution is a priori unknown). Defining stochastic approximation processes, the procedure is adaptative in the sense that the results of the analyses are updated recursively each time that a new data is taken into account.

From a theoretical point of view, the i.i.d case has been recently extended to the case of an expectation and/or covariance matrix of the random vector varying with time. We plan to include these improvements into our software.

5.3. Socio-economic index

A R package called SesIndexCreatoR has been written by B. Lalloué and J-M. Monnez in order to implement our socio-economic index for health inequalities. The version 1.0 of this package is currently freely available on the website of the Equit'Area project: http://www.equitarea.org/documents/packages_1.0-0/. It contains the functions needed to run the procedure (either integrally or partially) and obtain the corresponding SES index. The user may also create categories of this index with different methods (hierarchical clustering with or without k-nearest neighbors, quantiles, or intervals) and generate automatic reports of the results. Visualization and plotting functions are provided in the package.

BIOCORE Project-Team

5. Software and Platforms

5.1. Supervision software

We are developing a software for the supervision of bioreactors: this platform, named ODIN, has been built for the smart management of bioreactors (data acquisition, fault diagnosis, automatic control algorithm,...). This software was developed in C++ and uses a Scilab engine to run the advanced algorithms developed within BIOCORE. It has been implemented and validated with four different applications.

BONSAI Project-Team

5. Software and Platforms

5.1. YASS – Local homology search

Actively maintained.
Software self-assessment following the mechanisms provided by Inria Evaluation Committee for software evaluation: **A-4**, **SO-3**, **SM-2**, **EM-3**, **SDL-4**, DA-4, CD-4, MS-4, TPM-4

Software web site : http://bioinfo.lifl.fr/yass/

Licence: GPL

YASS is a software devoted to the classical problem of genomic pairwise alignment, and use most of our knowledge to design and implement efficient seeding techniques these last years. It is frequently used, it always receives more than 300 web queries per month (excluding local queries), and is also frequently downloaded and cited.

5.2. RNA tools – RNA structure prediction and comparison

http://bioinfo.lifl.fr/rna/

Actively maintained/Actively developed

Inria Evaluation Committee Criteria for Software Self-Assessment: A-4, SO-3, SM-2, EM-3, SDL-4, DA-4, CD-4, MS-4, TPM-4

The RNA tools provide a suite of programs to help analysing RNA secondary structures, together with visualisation tools for RNA 2D structures and RNA multiple alignments. Our first tool was *carnac* for RNA structure prediction by comparative analysis. *carnac* was issued in 2004 [1], independently benchmarked [2], and re-designed in 2009. It is still cited and used. Over the years, we have add new programs: *regliss* for locally optimal secondary structures, *gardenia* for structure comparison, *CG-seq* for gene prediction by comparative analysis, ...

5.3. TFM-Explorer – Identification and analysis of transcription factor binding sites

Actively maintained.
Software self-assessment: **A-4**, **SO-3**, **SM-2**, **EM-3**, **SDL-4**, DA-4, CD-4, MS-4, TPM-4

Software web site : http://bioinfo.lifl.fr/TFM/

Licence: GPL

The TFM suite is a set of tools for analysis of transcription factor binding sites modeled by Position Weight Matrices. In this suite, the TFM-EXPLORER tool is designed to analyze regulatory regions of eukaryotic genomes using comparative genomics and local over-representation.

5.4. RNAspace – A platform for noncoding RNA annotation

Actively developped.
Software self-assessment: **A-5**, **SO-3**, **SM-3-up4**, **EM-2-up3**, **SDL-4**, DA-4, CD-4, MS-4, TPM-4

Software web site : http://www.rnaspace.org/

---

[1] CARNAC: folding families of related RNAs. H. Touzet *et al.*, Nucleic Acids Research, 2004

[2] A comprehensive comparison of comparative RNA structure prediction approaches. P. Gardner *et al.*, BMC Bioinformatics, 2004

Licence: GPL

RNAspace is a national collaborative initiative conducted with Genopole Midi-Pyrénées and originally supported by IBISA [3]. The goal is to develop an open source platform for structural and functional noncoding RNA annotation in genomes (see Section 6.2): http://www.rnaspace.org. The project will be pursued within France Génomique (see Section 7.2.1).

5.5. CGseq – A toolbox for comparative analysis

Actively maintained.
Software self-assessment: **A-4**, **SO-3**, **SM-2**, **EM-3**, **SDL-4**, DA-4, CD-4, MS-4, TPM-4

Software web site : http://bioinfo.lifl.fr/CGseq/

Licence: GPL

CG-seq is a toolbox for identifying functional regions in a genomic sequence by comparative analysis using multispecies comparison.

5.6. SortMeRNA – Metatranscriptome classification

Actively developed.
Software self-assessment: **A-4**, **SO-3**, **SM-2**, **EM-3**, **SDL-4**, DA-4, CD-4, MS-4, TPM-4

Software web site: http://bioinfo.lifl.fr/RNA/sortmerna

Licence: GPL

+ *SortMeRNA*: Metatranscriptome classification

http://bioinfo.lifl.fr/RNA/sortmerna/

Actively developed

Inria Evaluation Committee Criteria for Software Self-Assessment: A-4, SO-3, SM-3, EM-3, SDL-4, DA-4, CD-4, MS-4, TPM-4

SortMeRNA is a tool designed to rapidly filter ribosomal RNA fragments from metatranscriptomic data produced by next-generation sequencers. The distribution includes curated ribosomal RNA databases. It is available for download from our website, or through the open web-based platform Galaxy. *SortMeRNA* was released in October 2012, and is used in production by Genoscope (French National Center for Sequencing) to process metatranscriptomic data. Moreover, it has already been integrated in two published computational pipelines [4,5] and have identified users in multiple research laboratories worldwide [6].

5.7. Vidjil – Quantifying lymphocytes rearrangements in high-throughput sequencing data

Actively developed
Software self-assessment: **A-3-up4**, **SO-3**, **SM-2-up3**, **EM-3**, **SDL-4**, DA-4, CD-4, MS-4, TPM-4

Software web site : http://bioinfo.lifl.fr/vidjil

---

[3] IBISA is a French consortium for evaluating and funding national technological platforms in life sciences.

[4] A comprehensive metatranscriptome analysis pipeline and its validation using human small intestine microbiota datasets. M. Leimena et al., *BMC genomics*, 2013

[5] Metagenome survey of a multispecies and algae-associated biofilm reveals key elements of bacterial-algae interactions in photobioreactors. I. Krohn-Molt et al. Applied and environmental microbiology, 2013

[6] Umeå University (Sweden), Leibniz Institute DSMZ (Germany), NGS department of Campus Science Support Facilities GmbH (Austria), Oxford Centre for Integrative Systems Biology (Great Britain), Laboratoire d'Ecologie Alpine (Grenoble), PRABI (Lyon), Wageningen University (Netherlands), ...

Vidjil implements a two-stage strategy for fast clustering and quantification of clones coming from immunological rearrangements in genomic sequences. It is currently used in “minimal residual disease” following, but could have other uses in immunology research. *Vidjil* is currently under test at the Lille hospital, and is planned to be tested in another hematological lab. In 2013, the development of Vidjil was supported by the regional project ABILES: an engineer (Marc Duez) developed for 5 months a graphical interface for using *Vidjil*. We plan to release a first production version to the hospital during 2014.

5.8. Biomanycores.org – A community for bioinformatics on manycore processors

Actively developped.
Software self-assessment: **A-3**, **SO-2**, **SM-3**, **EM-3down2**, **SDL-4up5**, OC-4 (DA-4, CD-4, MS-4, TPM-4)

Software web site : http://biomanycores.org/

Manycore architectures are an emerging field of research full of promises for parallel bioinformatics. However the usage of GPUs is not so widespread in the end-user bioinformatics community. The goal of the `biomanycores.org` project is to gather open-source CUDA and OpenCL parallel codes and to provide easy installation, benchmarking, and interoperability. The last point includes interfaces to popular frameworks such as Biopython, BioPerl and BioJava.

The development of Biomanycores was supported by a national ADT [7] from October 2010 to October 2012.

5.9. Norine – A resource for nonribosomal peptides

Actively maintained.
Software self-assessment: **A-5**, **SO-3**, **SM-3-up4**, **EM-2-up3**, **SDL-4**, DA-4, CD-4, MS-4, TPM-4

Software web site : http://bioinfo.lifl.fr/norine/ Norine is a public computational resource that contains a database of NRPs with a web interface and dedicated tools, such as a 2D graph viewer and editor for peptides or comparison of NRPs. Norine was created and is maintained by members of BONSAI team, in tight collaboration with members of the ProBioGEM lab, a microbial laboratory of Lille1 University. Since its creation in 2006, Norine has gained an international recognition as the unique database dedicated to nonribosomal peptides because of its high quality and manually curated annotations, and has been selected by wwPDB as a reference database. It is queried from all around the world by biologists or biochemists. It receives more than 3000 queries per month. Norine main users come for 13% from the United States of America, for 12% from the United Kingdom, for 5% from China or for 4% from Germany where renowned biology laboratories work on nonribosomal peptides (NRPs) or on their synthetases.

5.10. Crac – RNA-seq read analysis

Actively maintained.
Software self-assessment: **A-4**, **SO-3**, **SM-3**, **EM-3**, **SDL-4**, DA-4, CD-4, MS-4, TPM-3

Software web site: http://crac.gforge.inria.fr/

Objective: CRAC aims at identifying biological variations in RNAs by comparing short reads to a reference genome. It detects point mutations, short indels, splice events, and fusion genes or transcripts.

This library is the result of a collaboration with N. Philippe and T. Commes (IGH laboratory, Montpellier) and É. Rivals (LIRMM laboratory, Montpellier).

5.11. GkArrays – Indexing high throughput sequencer reads

Actively maintained.
Software self-assessment: **A-3**, **SO-3**, **SM-3**, **EM-2**, **SDL-4**, DA-4, CD-4, MS-4, TPM-3

---

[7] ADT (Action for Technological Development) is an Inria internal call

Software web site : http://crac.gforge.inria.fr/gkarrays/

Objective : Gk-Arrays is a C++ library specifically dedicated to indexing reads produced by high-throughput sequencers. This index allows to answer queries centred on reads. It also takes benefits from the input specificity to lower space consumption.

This library is the result of a collaboration with N. Philippe and T. Commes (IGH laboratory, Montpellier), M. Léonard and T. Lecroq (LITIS laboratory, Rouen) and É. Rivals (LIRMM laboratory, Montpellier).

CARMEN Team

5. Software and Platforms

5.1. CEPS: a Cardiac ElectroPhysiology Simulator

The Carmen team develops a software code to perform high performance numerical simulations in cardiac electrophysiology using unstructured three-dimensional grids. The software, called CEPS (*Cardiac Electrophysiology Simulation*), is developped as a common tool for researchers in the Carmen team and for our partners and colleagues in scientific computing and biomedical engineering. The goal of CEPS is to easily allow the development of new numerical methods and new physical models. Thanks to an ADT, actual developments started at the end of 2012 and still continue.

As compared to other existing softwares, CEPS aims at providing a more general framework of integration for new methods or models and a better efficiency in parallel. CEPS is designed to run on massively parallel architectures, and to make use of state-of-the-art and well known computing libraries to achieve realistic and complex heart simulations. CEPS also includes software engineering and and validation tools [30]. We use the platform GForge (gforge.inria.fr/projects/ceps) based on Subversion. This allows to keep a history of developments for developers and users.

Some of our collaborators actively participate to the testing and discussion for the development of CEPS, namely:

- C. Pierre, LMA Université de Pau et des Pays de l'Adour;
- R. Turpault, LMA Université de Nantes;
- L. Gerardo-Giorda, BCAM Bilbao.

5.2. PROPAG

The workhorse for our applied simulation studies of the whole human heart is PROPAG, a code that has its origins at the Université de Montréal in Canada, and has been further developed by the Institute of Computational Science in Lugano, Switzerland. PROPAG is highly configurable, runs with complex model geometries, and runs efficiently on high-performance computing systems with many thousands of cores. It is particularly useful for whole-heart studies, which typically rely on very large model sizes (in the order of 10^8 elements), several different membrane models and cell types in a single simulation run, and several regionally varying parameters.

PROPAG is presently used in our group to study the relation between the substrate, complexity, and electrocardiographic features of atrial fibrillation and of cardiomyopathy-related ventricular arrhythmia, providing the efficiency and flexibility that is required to handle the complex anatomical structures that are involved.

5.3. Model construction – A new project

Many of our projects rely on realistic or even patient-tailored meshes to represent the anatomy of the human heart and torso. The construction of such meshes provides challenges on many levels, from the delineation of the anatomical structures in medical images to the construction of high-quality meshes. The construction of such meshes provides challenges on many levels, from the delineation of the anatomical structures in medical images to the construction of high-quality meshes. We presently use a variety of in-house and public software packages to perform this work and are able to produce meshes of sufficient quality, but we strive for an important streamlining of this work. We have initiated a discussion with several groups inside and outside Inria who have similar needs or can offer solutions. We specifically investigate the possibility to build a common software which combines and complements our present solutions. The new code should make various methods easily accessible and automate the work as much as possible. Because accuracy and mesh quality are important requirements, the new code should also provide convenient options for human intervention where algorithms fall short. For example, manual segmentation and mesh editing should be as easy and efficient as they are in medical-imaging tools and 3D-editing software, respectively, but well integrated into the workflow.

CASTOR Team

5. Software and Platforms

5.1. FluidBox

Participants: Boniface Nkonga [contact], Hervé Guillard.

FluidBox is a software dedicated to the simulation of inert or reactive flows. It is also able to simulate multiphase, multi-material and MDH flows. There exist 2D and 3D dimensional versions. The 2D version is used to test new ideas that are later implemented in 3D. Two classes of schemes are available : a classical finite volume scheme and the more recent residual distribution schemes. Several low Mach number preconditioning are also implemented. The code has been parallelized with and without domain overlapping. The linear solver PaStiX is integrated in FluidBox. A partitioning tool exists in the package and uses Scotch.

5.2. PlaTo

Participant: Hervé Guillard [contact].

PlaTo (Platform for Tokamak simulation) is a specialized set of softwares dedicated to the geometry of Tokamaks whose main objective is to provide the researchers of the CASTOR team a common development tools. The platform integrates Fortran90 modules using the MPI communication library for parallel computations and some python and C codes. The PlaTo platform has been developed thanks to a 2010 ADT of Inria and the ANR ESPOIR. The construction of this platform integrates the following developments :

- The set-up of a (small) database corresponding to axisymmetric solutions of the equilibrium plasma equations for realistic geometrical and magnetic configurations (JET, ITER and the Tore-Supra upgrade WEST).The construction of meshes is always an important and time consuming task. PlaTo provides meshes and solutions corresponding to equilibrium solutions that will be used as initial data for more complex computations.
- A set of interfaces (PlaTo ToolBox) allowing easy transfer between different solution and mesh formats.
- Numerical templates allowing the use of 3D discretization schemes using finite element/volume methods. At present, several applications (reduced MHD, Euler equations, two fluid Euler model) are available in PlaTo .

5.3. PaMPA

Participants: Cécile Dobrzynski [Bacchus], Hervé Guillard, Laurent Hascoët [Ecuador], Cédric Lachat, François Pellegrini [Bacchus].

PaMPA (“Parallel Mesh Partitioning and Adaptation”) is a middleware library dedicated to the management of distributed meshes. Its purpose is to relieve solver writers from the tedious and error prone task of writing again and again service routines for mesh handling, data communication and exchange, remeshing, and data redistribution. An API of the future platform has been devised, and the coding of the mesh handling and redistribution routines is in progress. PaMPA will be used as a base module for the PLATO solvers, to balance dynamically, refine and coarsen its distributed mesh.

5.4. Cedres++

Participants: Jacques Blum, Cédric Boulbe, Blaise Faugeras, Sylvain Bremond [CEA], Eric Nardon [CEA].

In Tokamaks, at the slow resistive diffusion time scale, the magnetic configuration in the plasma can be described by the MHD equilibirum equations inside the plasma and the Maxwell equations outside. Moreover, the magnetic field is often supposed not to depend on the azimutal angle.

Under this assumption of axisymmetric configuration, the equilibrium in the whole space reduces to solving a 2D problem in which the magnetic field in the plasma is described by the well known Grad Shafranov equation. The unknown of this problem is the poloidal magnetic flux. The P1 finite element code CEDRES++ solves this free boundary equilibrium problem in direct and inverse mode. The direct problem consists in the computation of the magnetic configuration and of the plasma boundary, given a plasma current density profile and the total current in each poloidal field coils (PF coils). The aim of the inverse problem is to find currents in the PF coils in order to best fit a given plasma shape. An evolutive version of the code has also been recently developed. This version takes into account the circuit equations in the PF coils. These equations give a time dependent relation between the voltages, the total current in the coils and the time derivative of the magnetic flux. Induced currents in passive structures like the vaccuum vessel are also considered in this dynamic equilibrium problem. This new version of the code is an important tool for plasma scenario development and Tokamak design studies.

5.5. Equinox

Participants: Jacques Blum, Cédric Boulbe, Blaise Faugeras.

EQUINOX is a code dedicated to the numerical reconstruction of the equilibrium of the plasma in a Tokamak. The problem solved consists in the identification of the plasma current density, a non-linear source in the 2D Grad-Shafranov equation which governs the axisymmetric equilibrium of a plasma in a Tokamak. The experimental measurements that enable this identification are the magnetics on the vacuum vessel, but also polarimetric and interferometric measures on several chords, as well as motional Stark effect measurements. The reconstruction can be obtained in real-time and the numerical method implemented involves a finite element method, a fixed-point algorithm and a least-square optimization procedure.

CLIME Project-Team

5. Software and Platforms

5.1. Polyphemus

Participants: Sylvain Doré, Vivien Mallet, Florian Couvidat [CEREA], Yiguo Wang [CEREA], Nora Duhanyan [CEREA], Yelva Roustan [CEREA].

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

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

Figure 1 depicts a typical result produced by Polyphemus.

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

In 2013, Polyphemus has received numerous improvements on aerosol modeling, including better dynamics for organic aerosol formation and interactions between organic and inorganic aerosols. The data assimilation part of Polyphemus can now perform 3D data assimilation, taking advantage of Lidar data. Further integration of the data assimilation library Verdandi was also carried out.

5.2. Data assimilation library: Verdandi

Participants: Vivien Mallet, Dominique Chapelle [M3DISIM], Philippe Moireau [M3DISIM], Anne Tilloy, Paul Baudin, Tristan Perotin.

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

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

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

In 2013, version 1.5 was released with better consistency between the methods. Verdandi received improvements in its test cases. Increased flexibility was introduced in error descriptions, especially for uncertainty quantification.

`../../../../projets/clime/IMG/uncertainty.png`

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

A C++ interface to the Nucleus for European Modelling of the Ocean (see the web site NEMO http://www.nemo-ocean.eu/) has been developed so that it can be plugged to Verdandi. The interface currently enables the application of Monte Carlo simulations and the ensemble Kalman filter.

5.3. Urban air quality analysis

Participants: Anne Tilloy, Vivien Mallet, Raphaël Périllat.

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

In 2013, the software introduced new models for error covariances. It may now take into account tunnels. New options were added to filter out certain observations. The software was extended to handle new file formats.

COFFEE Project-Team

5. Software and Platforms

5.1. NS2DDV

The code NS2DDV is developed jointly with the team SIMPAF, of the Inria Research Centre Lille Nord Europe. It is devoted to the simulation of non-homogeneous viscous flows, in two-dimensional geometries. The code is based on an original hybrid Finite Volume/Finite Element scheme; it works on unstructured meshes and can include mesh refinements strategies. Further details can be found in the research papers J. Comput. Phys., 227, 4671–4696, 2008 and J. Comput. Phys., 229 (17), 6027–6046, 2010. The code exists in two versions: a `Matlab` public version, a `C++` prototype version allowing more ambitious simulations. Both versions are still subject to developments. The current versions is restricted to incompressible flows but on-going progress are concerned with the simulation of avalanches. The source code of the public version is downloadable and several benchmarks tests can be reproduced directly.

5.2. Compass

for Computing Parallel Architecture to Speed up Simulation is a parallel code for the discretization of polyphasic flows by Finite Volumes methods. The code is mainly devoted to applications in porous media. It works on quite general polyhedral meshes. A first step in the code development has been made during the 2012 edition of CEMRACS and then pursued by C. Guichard, R. Masson and R. Eymard in 2013. A first version of the code has been deposited at the Agency for the Protection of Programs (APP). This current version of `ComPASS` has been tested on a gas storage two phase flow benchmark with GDFSuez using the Vertex Approximate Gradient spatial discretization. The results have shown a very good parallel scalability on the CICADA Cluster at UNS with a few millions of cells and up to 1024 cores. The objective is to develop a generic simulator for multiphase Darcy flows. This simulator will implement advanced finite volume methods on general 3D meshes and on heterogeneous anisotropic media, taking into account discrete fracture networks represented as interfaces of codimension one and coupled with the surrounding matrix. It will be able to treat a large range of multiphase Darcy flow models accounting for thermodynamical equilibrium and the coupling with an energy conservation equation. The simulator will run on massively parallel architectures with a few thousands of cores. It will be applied to several type of industrial applications starting with the simulation of high energy geothermal systems as a carbon-free source of power production.

5.3. SimBiof

We are developing numerical methods, currently by using Finite Differences approaches, for the simulation of biofilms growth. The underlying system of PDEs takes the form of multiphase flows equations with conservation constraints and vanishing phases. The numerical experiments have permitted to bring out the influence of physical parameters on the multidimensional growth dynamics.

5.4. AP_PartFlow

We are developing experimental codes, mainly based on Finite Differences, for the simulation of particulate flows. A particular attention is paid to guaranty the asymptotic properties of the scheme, with respect to relaxation parameters.

DEMAR Project-Team

5. Software and Platforms

5.1. Software and Platforms

5.1.1. RdP to VHDL tool

Participants: Gregory Angles, David Andreu, Thierry Gil, Robin Passama.

Our SENIS (Stimulation Electrique Neurale dIStribuee) based FES architecture relies on distributed stimulation units (DSU) which are interconnected by means of a 2-wire based network. A DSU is a complex digital system since its embeds among others a dedicated processor (micro-machine with a specific reduced instruction set), a monitoring module and a 3-layer protocol stack. To face the complexity of the unit's digital part and to ease its prototyping on programmable digital devices (e.g. FPGA), we developed an approach for high level hardware component programming (HILECOP). To support the modularity and the reusability of sub-parts of complex hardware systems, the HILECOP methodology is based on components. An HILECOP component has: a Petri Net (PN) based behavior, a set of functions whose execution is controlled by the PN, and a set of variables and signals. Its interface contains places and transitions from which its PN model can be inter-connected as well as signals it exports or imports. The interconnection of those components, from a behavioral point out view, consists in the interconnection of places and/or transitions according to well-defined mechanisms: interconnection by means of oriented arcs or by means of the "merging" operator (existing for both places and transitions).

GALS (Globally Asynchronous Locally Synchronous) systems can be specified, connecting different clocks to HILECOP components, and interconnecting them by means of asynchronous signals.

Undergoing work includes the modification of the formalism in order to allow behavior agregation as well as exception handling, both for analysis and implementation sides.

The Eclipse-based version of HILECOP is regulary updated. The last version of HILECOP (registered at the french Agence de Protection des Programmes (APP)) is accessible to the academic community (http://www.lirmm.fr/~gil/Temp/).

5.1.2. SENISManager

Participants: Robin Passama, David Andreu.

We developed a specific software environment called SENISManager allowing to remotely manage and control a network of DSUs, i.e. the distributed FES architecture. SENISManager performs self-detection of the architecture being deployed. This environment allows the manipulation of micro-programs from their edition to their remote control. It also allows the programming of control sequences executed by an external controller in charge of automatically piloting a stimulator.

SENISManager has been transferred to the industrial partner and a new version is under development according to an Eclipse-based design. This new version should be available by the end of 2014.

DRACULA Project-Team

5. Software and Platforms

5.1. CelDyn

Participants: Laurent Pujo-Menjouet, Alen Tosenberger, Vitaly Volpert [correspondant].

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

DYLISS Project-Team

5. Software and Platforms

5.1. Platform

Our tools are based on formal systems. They aim at guiding the user to progressively reduce the space of models (gene or protein families, set of main actors involved in a system response, dynamical models) which are compatible with both knowledge and experimental observations. Most of our tools are available both as stand-alone software and through portals such as Mobyle or Galaxy interfaces. Tools are developed in collaboration with the GenOuest resource and data center hosted in the IRISA laboratory, including their computer facilities [more info]. .

5.2. Integrative Biology: (constraint-based) toolbox for network filtering

Participants: Anne Siegel [contact], Andres Aravena, Jeanne Cambefort [contact], Guillaume Collet, Damien Eveillard, Sylvain Prigent, Sven Thiele [contact].

The goal is to offer a toolbox for the reconstruction of networks from genome, literature and large-scale observation data (expression data, metabolomics...) in order to elucidate the main regulators of an observed phenotype. Most of the optimization issues are addressed with Answer Set Programming.

MeMap and MeMerge. We develop a workflow for the **Au**tomatic **Re**construction of **Me**tabolic networks (AuReMe). In this workflow, we use heterogeneous sources of data with identifiers from different namespaces. MeMap (**Me**tabolic network **Map**ping) consists in mapping identifiers from different namespaces to a unified namespace. Then, MeMerge (**Me**tabolic network **Merge**) merges two metabolic networks previously mapped on the same namespace. [web server].

meneco [*input*: draft metabolic network & metabolic profiles. *output*: metabolic network]. It is a qualitative approach to elaborate the biosynthetic capacities of metabolic networks. In fact, large-scale metabolic networks as well as measured datasets suffer from substantial incompleteness. Moreover, traditional formal approaches to biosynthesis require kinetic information, which is rarely available. Our approach builds upon formal systems for analyzing large-scale metabolic networks. Mapping its principles into Answer Set Programming allows us to address various biologically relevant problems [44] [27] [python package][web server].

shogen [*input*: genome & metabolic network. *output* : functional regulatory modules]. This software is able to identify genome portions which contain a large density of genes coding for enzymes that regulate successive reactions of metabolic pathways [26] [python package].

lombarde [*input*: genome, modules & several gene-expression datasets. *output*: oriented regulation network]. This tool is useful to enhance key causalities within a regulatory transcriptional network when it is challenged by several environmental perturbations [13] [web server].

bioquali [*input*: signed regulation network & one gene-expression dataset. *output*: consistency-checking and gene-expression prediction]. It is a plugin of the Cytoscape environment. BioQuali analyses regulatory networks and expression datasets by checking a global consistency between the regulatory model and the expression data. It diagnoses a regulatory network searching for the regulations that are not consistent with the expression data, and it outputs a set of genes which predicted expression is decided in order to explain the expression inputed data. It also provides the visualization of this analysis with a friendly environment to encourage users of different disciplines to analyze their regulatory networks [6] [web server][cytoscape plugin] .

ingranalyze [*input*: signed regulation network & one gene-expression dataset. *output*: network repair gene-expression prediction] This tool is an extension to the bioquali tool. It proposes a range of different operations for altering experimental data and/or a biological network in order to re-establish their mutual consistency, an indispensable prerequisite for automated prediction. For accomplishing repair and prediction, we take advantage of the distinguished modeling and reasoning capacities of Answer Set Programming [5] [Python package][web server].

5.3. Dynamics: invariant-based prediction

Participants: Oumarou Abdou-Arbi, Geoffroy Andrieux, Jérémie Bourdon [contact], Jeanne Cambefort [contact], Damien Eveillard, Michel Le Borgne, Anne Siegel, Sven Thiele, Santiago Videla [contact].

We develop tools predicting some characteristics of a biological system behavior from incomplete sets of parameters or observations.

cadbiom. Based on Guarded transition semantic, this software provides a formal framework to help the modeling of biological systems such as cell signaling network. It allows investigating synchronization events in biological networks. [software][web server].

caspo: Cell ASP Optimizer This soft provides an easy to use software for learning Boolean logic models describing the immediate-early response of protein signaling networks. Given a network describing causal interactions, and a phospho-proteomics dataset, caspo is able to searches for optimal Boolean logic models explaining the dataset. Optimality includes both the size of the boolean network and the distance of predictions to real-data observations. It is useful to boolean networks inference, cancer research, drug discovery, and experimental design. It is used in the CellNOpt environment [1]. [python package][web server].

nutritionAnalyzer. This tool is dedicated to the computation of allocation for an extremal flux distribution. It allows quantifying the precursor composition of each system output (AIO) and to discuss the biological relevance of a set of flux in a given metabolic network by computing the extremal values of AIO coefficients. This approach enables to discriminate diets without making any assumption on the internal behaviour of the system [15][webserver][software and doc].

POGG. The POGG software allows scoring the importance and sensibility of regulatory interactions with a biological system with respect to the observation of a time-series quantitative phenotype. This is done by solving nonlinear problems to infer and explore the family of weighted Markov chains having a relevant asymptotic behavior at the population scale. Its possible application fields are systems biology, sensitive interactions, maximal entropy models, natural language processing. It results from our collaboration with the LINA-Nantes [2][matlab package].

5.4. Sequence annotation

Participants: François Coste [contact], Aymeric Antoine-Lorquin, Catherine Belleannée [contact], Gaëlle Garet, Olivier Quenez, Jacques Nicolas.

We develop tools for discovery and search of complex pattern signatures within biological sequences, with a focus on protein sequences.

Logol Logol is a swiss-army-knife for Pattern matching on DNA/RNA/Protein sequences, using a high-level grammar to permit a large expressivity. Allowed patterns can consist in a combination of motifs, structures (stem-loops, repeats), indels etc. It allows pseudo-knot identification, context sensitive grammatical formalism and full genome analysis. Possible fields of application are the detection of mutated binding sites or stem-loop identification (e.g. in CRISPR [2] [10]) [software]

---

[1] http://www.cellnopt.org/

[2] http://crispi.genouest.org/

Protomata learner This tool is a grammatical inference framework suitable for learning the specific signature of a functional protein family from unaligned sequences by partial and local multiple alignment and automata modeling. It performs a syntactic characterization of proteins by identification of conservation blocks on sequence subsets and modelling of their succession. Possible fields of application are new members discovery or study (for instance, for site-directed mutagenesis) of, possibly non-homologous, functional families and subfamilies such as enzymatic, signaling or transporting proteins [38][4] [web server]

5.5. Integration of our tools in larger software environments

Most of our software were designed as "bricks" that can combined through workflow application such as Mobyle. It worths considering them into larger dedicated environments to benefit from the expertise of other research groups.

Web servers In collaboration with the GenOuest ressource center, most our tools are made available through several web portals.

- The **mobyle@GenOuest portal** is the generic web server of our ressource center. It hosts the ingranalysis, meneco, caspo, lombarde and shogun tools [website].
- The **Mobyle@Biotempo server** is a mobyle portal for system biology with formal approaches. It hosts the memap, memerge, meneco, ingranalysis, cadbiom and pogg tools [website].

Dr Motif This resource aims at the integration of different software commonly used in pattern discovery and matching. This resource also integrates Dyliss pattern search and discovery software [website].

ASP4biology and BioASP It is a meta-package to create a powerful environment of biological data integration and analysis in system biology, based on knowledge representation and combinatorial optimization technologies (ASP). It provides a collection of python applications which encapsulates ASP tools and several encodings making them easy to use by non-expert users out-of-the-box. [Python package][website].

ASP encodings repository This suite comprises projects related to applications of Answer Set Programming using Potassco systems (the Potsdam Answer Set Solving Collection, bundles tools for Answer Set Programming developed at the University of Potsdam). These are usually a set of encodings possibly including auxiliary software and scripts [respository].

FLUMINANCE Project-Team

5. Software and Platforms

5.1. DenseMotion software - Estimation of 2D dense motion fields

Participants: Thomas Corpetti, Patrick Héas, Etienne Mémin.

This code allows the computation from two consecutive images of a dense motion field. The estimator is expressed as a global energy function minimization. The code enables the choice of different data models and different regularization functionals depending on the targeted application. Generic motion estimators for video sequences or fluid flows dedicated estimators can be set up. This software allows in addition the users to specify additional correlation based matching measurements. It enables also the inclusion of a temporal smoothing prior relying on a velocity vorticity formulation of the Navier-Stoke equation for Fluid motion analysis applications. The different variants of this code correspond to research studies that have been published in IEEE transaction on Pattern Analysis and machine Intelligence, Experiments in Fluids, IEEE transaction on Image Processing, IEEE transaction on Geo-Science end Remote Sensing. The binary of this code can be freely downloaded on the FLUID web site http://fluid.irisa.fr.

5.2. 2DLayeredMotion software - Estimation of 2D independent mesoscale layered atmospheric motion fields

Participants: Patrick Héas, Etienne Mémin.

This software enables to estimate a stack of 2D horizontal wind fields corresponding to a mesoscale dynamics of atmospheric pressure layers. This estimator is formulated as the minimization of a global energy function. It relies on a vertical decomposition of the atmosphere into pressure layers. This estimator uses pressure data and classification clouds maps and top of clouds pressure maps (or infra-red images). All these images are routinely supplied by the EUMETSAT consortium which handles the Meteosat and MSG satellite data distribution. The energy function relies on a data model built from the integration of the mass conservation on each layer. The estimator also includes a simplified and filtered shallow water dynamical model as temporal smoother and second-order div-curl spatial regularizer. The estimator may also incorporate correlation-based vector fields as additional observations. These correlation vectors are also routinely provided by the Eumetsat consortium. This code corresponds to research studies published in IEEE transaction on Geo-Science and Remote Sensing. It can be freely downloaded on the FLUID web site http://fluid.irisa.fr.

5.3. 3DLayeredMotion software - Estimation of 3D interconnected layered atmospheric motion fields

Participants: Patrick Héas, Etienne Mémin.

This software extends the previous 2D version. It allows (for the first time to our knowledge) the recovery of 3D wind fields from satellite image sequences. As with the previous techniques, the atmosphere is decomposed into a stack of pressure layers. The estimation relies also on pressure data and classification clouds maps and top of clouds pressure maps. In order to recover the 3D missing velocity information, physical knowledge on 3D mass exchanges between layers has been introduced in the data model. The corresponding data model appears to be a generalization of the previous data model constructed from a vertical integration of the continuity equation. This research study has been published in IEEE trans. on Geo-Science and Remote Sensing. The binary of this code can be freely downloaded on the FLUID web site http://fluid.irisa.fr.

5.4. Low-Order-Motion - Estimation of low order representation of fluid motion

Participants: Anne Cuzol, Etienne Mémin.

This code enables the estimation of a low order representation of a fluid motion field from two consecutive images.The fluid motion representation is obtained using a discretization of the vorticity and divergence maps through regularized Dirac measure. The irrotational and solenoidal components of the motion fields are expressed as linear combinations of basis functions obtained through the Biot-Savart law. The coefficient values and the basis function parameters are formalized as the minimizer of a functional relying on an intensity variation model obtained from an integrated version of the mass conservation principle of fluid mechanics. Different versions of this estimation are available. The code which includes a Matlab user interface can be downloaded on the FLUID web site http://fluid.irisa.fr. This program corresponds to a research study that has been published in the International Journal on computer Vision.

GALEN Project-Team

5. Software and Platforms

5.1. Deformable Registration Software

Participant: Nikos Paragios [Correspondant].

deformable image and volume registration, is a deformable registration platform in C++ for the medical imaging community (publicly available at http://www.mrf-registration.net) developed mainly at Ecole Centrale, Technical University of Munich and University of Crete. This is the first publicly available platform which contains most of the existing metrics to perform registration under the same concept. The platform is used for clinical research from approximately 3,000 users worldwide.

5.2. Dense image and surface descriptors

Participant: Iasonas Kokkinos [Correspondant].

Scale-Invariant Descriptor, Scale-Invariant Heat Kernel Signatures DISD (publicly available at http://vision.mas.ecp.fr/Personnel/iasonas/descriptors.html) implements the SID, SI-HKS and ISC descriptors. SID (Scale-Invariant Descriptor) is a densely computable, scale- and rotation- invariant descriptor. We use a log-polar grid around every point to turn rotation/scalings into translation, and then use the Fourier Transform Modulus (FTM) to achieve invariance. SI-HKS (Scale-Invariant Heat Kernel Signatures) extract scale-invariant shape signatures by exploiting the fact that surface scaling amounts to multiplication and scaling of a properly sampled HKS descriptor. We apply the FTM trick on HKS to achieve invariance to scale changes. ISC (Intrinsic Shape Context) constructs a net-like grid around every surface point by shooting outwards and tracking geodesics. This allows us to build a meta-descriptor on top of HKS/SI-HKS that takes neighborhood into account, while being invariant to surface isometries.

5.3. Dissimilarity Coefficient learning

Participant: Pawan Kumar [Correspondant].

weakly supervised learning, dissimilarity coefficient, structured prediction DISC (publicly available at http://cvn.ecp.fr/personnel/pawan/code/DISCAPI.zip) software provides a convenient API for dissimilarity coefficient (DISC) based learning. DISC allows the use of weakly supervised datasets (with missing information) by jointly learning a structured prediction classifier and a conditional probability distribution of the missing information. The parameters of the classifier and the distribution are learned by minimizing a user-specified dissimilarity coefficient between them.

5.4. Efficient bounding-based object detection

Participant: Iasonas Kokkinos [Correspondant].

branch-and-bound, parts detection, segmentation, DPMS implements branch-and-bound object detection, cutting down the complexity of detection from linear in the number of pixels to logarithmic (publicly available at http://vision.mas.ecp.fr/Personnel/iasonas/dpms.html). The results delivered are identical to those of the standard deformable part model detector, but are available in 5 to 20 times less time. This website has been visited 1500 times in 10 months.

5.5. Fast Primal Dual Strategies for Optimization of Markov Random Fields

Participant: Nikos Komodakis [Correspondant].

discrete optimization, Markov random field, duality, graph cuts, FASTPD is an optimization platform in C++ for the computer vision and medical imaging community (publicly available at http://www.csd.uoc.gr/~komod/FastPD/) developed mainly at Ecole Centrale and University of Crete. This is the most efficient publicly available platform in terms of a compromise of computational efficiency and ability to converge to a good minimum for the optimization of generic MRFs. The platform is used from approximately 1,500 users worldwide.

5.6. imaGe-based Procedural Modeling Using Shape Grammars

Participant: Iasonas Kokkinos [Correspondant].

procedural modeling, image-based building reconstruction, shape grammars GRAPES is a generic image parsing library based on re-inforcement learning (publicly available at http://vision.mas.ecp.fr/Personnel/teboul/grapesPage/index.php). It can handle grammars (binary-split, four-color, Hausmannian) and image-based rewards (Gaussian mixtures, Randomized Forests) of varying complexity while being modular and computationally efficient both in terms of grammar and image rewards. The platform is used from approximately 500 users worldwide.

5.7. Learning-based symmetry detection

Participant: Stavros Tsogkas [Correspondant].

Scale-Invariant Descriptor, Scale-Invariant Heat Kernel Signatures LBSD (publicly available at http://cvn.ecp.fr/personnel/tsogkas/code.html implements the learning-based approach to symmetry detection. It includes the code for running a detector, alongside with the ground-truth symmetry annotations that we have introduced for the Berkeley Segmentation Dataset (BSD) benchmark.

5.8. Texture Analysis Using Modulation Features and Generative Models

Participant: Iasonas Kokkinos [Correspondant].

Texture, modulation, generative models, segmentation, TEXMEG is a front-end for texture analysis and edge detection platform in Matlab that relies on Gabor filtering and image demodulation (publicly available at http://cvsp.cs.ntua.gr/software/texture/). Includes frequency- and time- based definition of Gabor- and other Quadrature-pair filterbanks, demodulation with the Regularized Energy Separation Algorithm and Texture/Edge/Smooth classification based on MDL criterion. The platform is used from approximately 250 users worldwide.

5.9. Sparse Prediction

Participant: Andreas Argyriou [Correspondant].

Sparse prediction, K-support norm, SPARSE_K is a sparse prediction code (publicly available at http://cvn.ecp.fr/personnel/andreas/code/sparse_k/sparse_k.tar) using regularization with the k-support norm, which we have introduced [39]. The algorithm uses an accelerated first-order method similar to Nesterov's method.

GENSCALE Project-Team

5. Software and Platforms

5.1. Next Generation Sequencing

Participants: Alexan Andrieux, Dominique Lavenier, Claire Lemaitre, Nicolas Maillet, Pierre Peterlongo, Guillaume Rizk, Erwan Drezen, Charles Deltel.

- **Genome assembly** [contact: P. Peterlongo]
  - **Minia : ultra low memory footprint assembly** Minia is a short-read assembler based on a de Bruijn graph, capable of assembling a human genome on a desktop computer in a day. The output of Minia is a set of contigs. Minia produces results of similar contiguity and accuracy to other de Bruijn assemblers (e.g. Velvet). http://minia.genouest.org/
  - **Mapsembler: targeted assembly software.** Mapsembler is a targeted assembly software. From sets of NGS raw reads and a set of input sequences (starters), it determines if each starter could be constructed from the reads. Then for each "read-coherent" starter, Mapsembler outputs its sequence neighborhood as a linear sequence or as a graph, depending on the user choice. http://colibread.inria.fr/mapsembler2/
  - **Bloocoo: memory-efficient read correction** Bloocoo is a software to identify sequencing errors in short-read datasets and correct them. It is based on an efficient data structure that enables to keep a very low memory footprint. http://gatb.inria.fr
- **Variant detection** [contact: C. Lemaitre]
  - **discoSnp and kisSplice : variant identification without the use of a reference genome**. discoSnp is a tool to find single nucleotide polymorphisms (SNP) by comparing two sets of raw NGS reads. http://colibread.inria.fr/discosnp/ KisSplice finds alternative splicings but also short insertions, deletions and duplications, SNPs and sequencing errors in one or two RNA-seq sets, without assembly nor mapping on a reference genome. http://colibread.inria.fr/software/kissplice/
  - **Kissreads: quantification of variants** Kissreads considers sets of NGS raw reads and a set of input sequences (starters). Mapping reads to each starter, it provides quantitative (coverage depth) and qualitative (mapped read quality) information about each starter.
  - **MindTheGap : detection of large insertions** MindTheGap is a tool to detect large insertion events in re-sequencing data with respect to a reference genome. http://gatb.inria.fr
- **Read mapping** [contact: D. Lavenier]
  - **GASSST: short reads mapper** The GASSST software (Global Alignment Short Sequence Search Tool) is a general purpose mapper. GASSST finds global alignments of short DNA sequences against large DNA banks. One main characteristic of GASSST is its ability to perform fast gapped alignments and to process long reads compared to other current similar tools. http://www.irisa.fr/symbiose/projects/gassst/

5.2. High throughput sequence analysis

Participants: Erwan Drezen, Dominique Lavenier, Claire Lemaitre, Nicolas Maillet, Pierre Peterlongo.

- **PLAST : efficient bank-to-bank alignments** PLAST (Parallel Local Alignment Search Tool) is a parallel alignment search tool for comparing large protein banks. PLAST runs 3 to 5 times faster than the NCBI-BLAST software. An improved version is commercialized by the Korilog Company, including the DNA bank-to-bank option. [contact: D. Lavenier] http://www.irisa.fr/symbiose/projects/plast/
- **Compareads : efficient comparison of large metagenomics NGS datasets** This software extracts similar DNA sequences (reads) between two metagenomic datasets. It requires a small and fixed amount of memory and can thus be used on huge datasets. [contact: P. Peterlongo] http://alcovna.genouest.org/compareads/

5.3. 3D Protein structures

Participants: Rumen Andonov, Guillaume Chapuis, Mathilde Le Boudic-Jamin, Antonio Mucherino.

- **CSA and DALIX** CSA (Comparative Structural Alignment) is a webserver for computing and comparing protein structure alignments. CSA is able to compute score-optimal alignments with respect to various inter-residue distance-based scoring schemes. [contact: R. Andonov] http://csa.project.cwi.nl/
- **A_purva** A_purva is a Contact Map Overlap maximization (CMO) solver. Given two protein structures represented by two contact maps, A_purva computes the amino-acid alignment which maximize the number of common contacts. [contact: R. Andonov] http://mobyle.genouest.org/cgi-bin/Mobyle/portal.py?forms::A_Purva
- **MD-Jeep** MD-jeep is a software tool for solving distance geometry problems. It is able to solve a subclass of instances of the problem for which a discrete reformulation can be supplied. We refer to this subclass of instances as the Discretizable Molecular Distance Geometry Problem (DMDGP). We employ a Branch & Prune (BP) algorithm for the solution of DMDGPs. [contact: A. Mucherino] http://www.antoniomucherino.it/en/mdjeep.php

5.4. HPC and Parallelism

Participants: Guillaume Chapuis, Dominique Lavenier, François Moreews, Charles Deltel.

- **QTLmap** QTLMap is a tool dedicated to the detection of Quantitative Trait Loci (QTL) from experimental designs in outbred population. QTLMap was recently ported to GPU and offers reduced run times. [contact: D. Lavenier] http://www.inra.fr/qtlmap/
- **SLICEE** (Service Layer for Intensive Computation Execution Environment) is part of the BioWIC project. This software proposes (1) to abstract the calls to the cluster scheduler by handling command submission; (2) to take care of exploiting the data parallelism with data specific methods; (3) to manage data using a cache references mecanism and route data between tasks. [contact: F. Moreews] http://vapor.gforge.inria.fr/

IBIS Project-Team

4. Software and Platforms

4.1. Genetic Network Analyzer (GNA)

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

Keywords. Gene regulatory networks, qualitative simulation, model checking

GENETIC NETWORK ANALYZER (GNA) is the implementation of methods for the qualitative modeling and simulation of gene regulatory networks developed in the IBIS project. The input of GNA consists of a model of the regulatory network in the form of a system of piecewise-linear differential equations (PLDEs), supplemented by inequality constraints on the parameters and initial conditions. From this information, GNA generates a state transition graph summarizing the qualitative dynamics of the system. In order to analyze large graphs, GNA allows the user to specify properties of the qualitative dynamics of a network in temporal logic, using high-level query templates, and to verify these properties on the state transition graph by means of standard model-checking tools, either locally installed or accessible through a remote web server. GNA is currently distributed by the company Genostar, but remains freely available for academic research purposes. The current version is GNA 8.5. In comparison with the previously distributed versions, GNA 8.5 has the following additional functionalities: (1) it supports the editing and visualization of regulatory networks, in an SBGN-compatible format, (2) it semi-automatically generates a prototype model from the network structure, thus accelerating the modeling process, and (3) it allows models to be exported in the SBML Qual standard [6]. For more information, see http://www-helix.inrialpes.fr/gna.

4.2. WellReader

Participants: Johannes Geiselmann, Hidde de Jong [Correspondent], Michel Page, Delphine Ropers.

Keywords. Gene expression, reporter gene data

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

M3DISIM Team

5. Software and Platforms

5.1. FELISCE

Participants: Dominique Chapelle, Sébastien Gilles [correspondant], Philippe Moireau.

FELISCE – standing for “Finite Elements for LIfe SCiences and Engineering” – is a new finite element code which the MACS and REO teams have decided to jointly develop in order to build up on their respective experiences concerning finite element simulations. One specific objective of this code is to provide in a unified software environment all the state-of-the-art tools needed to perform simulations of the complex cardiovascular models considered in the two teams – namely involving fluid and solid mechanics, electrophysiology, and the various associated coupling phenomena. FELISCE is written in C++, and may be later released as an opensource library. See https://gforge.inria.fr/projects/felisce/.

5.2. HeartLab

Participants: Matthieu Caruel, Dominique Chapelle, Alexandre Imperiale, Philippe Moireau [correspondant].

The heartLab software is a library written in (64 bits compatible) Matlab and C (mex functions) designed to perform both simulation and estimation (based on various types of measurements, e.g. images) of the heart mechanical behavior. Started in 2006, it is already quite large (about 60,000 lines), and is used within various collaborations.

The code relies on OpenFEM – to which the team has previously contributed, see http://www.openfem.net – for the finite element computations, and the implementation was performed with a particular concern for modularity, since modeling and estimation use the same finite element operators. This modularity also allows to couple the code with other FEM solvers, such as LifeV and Mistral developed in the Reo team-project. In particular, we are now able to include perfusion and electrical coupling with LifeV using PVM, and fluid-structure interaction using Mistral.

We also included geometric data and tools in the code to define cardiac anatomical models compatible with the simulation requirements in terms of mesh quality, fiber direction data defined within each element, and the referencing necessary for handling boundary conditions and estimation, in particular. These geometries are analytical or come from computerized tomography (CT) or magnetic resonance (MR) image data of humans or animals.

We recently incorporated numerous non-linear data assimilation observation operators based on medical imaging post-processing to be able to now perform estimation with a large variety of medical imaging modalities.

5.3. Verdandi

Participants: Dominique Chapelle, Marc Fragu, Vivien Mallet [Clime team], Philippe Moireau [correspondant].

Verdandi is an opensource (LGPL) software library aiming at providing assimilation data methods and related tools. Mainly targeted at large systems arising from the discretization of PDEs, it is intentionally devised as generic, which allows for applications in a wide range of problems (biology and medicine, environment, image processing...). See also the web page http://verdandi.gforge.inria.fr/, with a complete documentation in English. The first stable version (1.0) was released in June 2012 and contains most of the major data assimilation algorithms of both variational and sequential types. The current version (1.5) contains additional estimation algorithm and parallel capabilities. Note that some specific developments are performed with particular regard to cardiac modeling applications, as Verdandi is partly funded by – and distributed within – the VPH-Share project and is now referenced in the peer-reviewed article [16].

- ACM: Mathematical software
- AMS: System theory; control
- Software benefit: Verdandi is the only *generic* data assimilation library
- License: LGPL (2.1 or any later version)
- Type of human computer interaction: Command line and configuration files
- OS/Middelware: Linux, MacOS ou Windows
- Required library or software: Seldon (LGPL, http://seldon.sourceforge.net/)
- Programming language: C++, ISO/IEC 14882: I998(E) Python, version 2.6
- Documentation: Doxygen and utilisation manual in English

MAGIQUE-3D Project-Team

5. Software and Platforms

5.1. Hou10ni

Participant: Julien Diaz [correspondant].

5.1.1. *Hou10ni-Time-Domain*

This software, written in FORTRAN 90, simulates the propagation of acoustic waves in heterogeneous 2D and 3D media. It is based on an Interior Penalty Discontinuous Galerkin Method (IPDGM). The 2D version of the code has been implemented in the Reverse Time Migration (RTM) software of TOTAL in the framework of the Ph.D thesis of Caroline Baldassari. The 2D code allows for the use of meshes composed of cells of various order (p-adaptivity in space). For the time discretization, we used the local time stepping strategy described at section 3.2 , item **High-Order Schemes in Space and Time** which permits not only the use of different time-step, but also to adapt the order of the time-discretization to the order of each cells (hp-adaptivity in time).

The main competitors of Hou10ni are codes based on Finite Differences, Spectral Element Method or other Discontinuous Galerkin Methods (such as the ADER schemes). During her Ph.D thesis, Caroline Baldassari compared the solution obtained by Hou10ni to the solution obtained by a Finite Difference Method and by a Spectral Element Method (SPECFEM). To evaluate the accuracy of the solutions, we have compared them to analytical solutions provided by the codes Gar6more (see below). The results of these comparisons are: a) that Hou10ni outperforms the Finite Difference Methods both in terms of accuracy and of computational burden and b) that its performances are similar to Spectral Element Methods. Since Hou10ni allows for the use of meshes based on tetrahedrons, which are more appropriate to mesh complex topographies, and for the p-adaptivity, we decided to implement it in the RTM code of TOTAL. Of course, we also used these comparisons to validate the code.

5.1.2. *Hou10ni-Frequency-Domain*

Recently, we have extended the 2D version of Hou10ni for computing the solution of the harmonic wave equation (Helmholtz), in the framework of the PhD thesis of Élodie Estécahandy. This new version is able to deal with both acoustic and elastodynamic media, but also to model elastoacoustic problems. The surfaces between the different media can be approximated by curved elements. We can use up to P^{15} elements when dealing with curved elements and element of arbitrary order (with of course a limitation depending on the machine precision) when dealing with non-curved elements. The construction of the global matrix is perform using OpenMP and the extension to hybrid MPI/OpenMP parallelism is on development. This code has been also implemented in a solver which determine the shape of an elastic obstacle from the knowledge of its scattered field.

The 3D version of Hou10ni-Frequency-Domain is under development. The code is now able to solve acoustic problems up to P^3 elements. It has been parallelized using MPI and is able to deal with partitioned meshes. Preliminary tests have been performed up to 16.000.000 unknowns. We are now considering the following features : hybrid MPI/OpenMP parallelism; extension to arbitrary polynomial degrees; extension to elastodynamic.

5.2. Gar6more2D and Gar6more3D

Participant: Julien Diaz [correspondant].

This codes compute the analytical solution of problems of waves propagation in two layered 3D media such as-acoustic/acoustic- acoustic/elastodynamic- acoustic/porous- porous/porous, based on the Cagniard-de Hoop method.

See also the web page http://web.univ-pau.fr/~jdiaz1/software.html.

The main objective of these codes is to provide reference solutions in order to validate numerical codes. They have been already used by J. Tromp and C. Morency to validate their code of poroelastic wave propagation [96]. They are freely distributed under a CECILL license and can be downloaded on the website http://web.univ-pau.fr/~jdiaz1/software.html. As far as we know, the main competitor of this code is EX2DELDEL (available on http://www.spice-rtn.org), but this code only deals with 2D acoustic or elastic media. Our codes seem to be the only ones able to deal with bilayered poroelastic media and to handle the three dimensional cases.

- ACM: J.2
- AMS: 34B27 35L05 35L15 74F10 74J05
- Programming language: Fortran 90

5.3. Montjoie

Participant: Marc Duruflé [correspondant].

`Montjoie` is a code developed by Marc Duruflé with contributions of students, including Juliette Chabassier during her PhD. It provides a C++ framework for solving partial differential equations on unstructured meshes with finite element-like methods (continuous finite element, discontinuous Galerkin formulation, edge elements and facet elements). The handling of mixed elements (tetrahedra, prisms, pyramids and hexahedra) has been implemented for these different types of finite elements methods in the context of Morgane Bergot's PhD. Several applications are currently available : wave equation, elastodynamics, aeroacoustics, Maxwell's equations. In 2013, an implementation of non-linear 1-D Maxwell's equations (non-linear Kerr effect) has been added as well as various 1-D non-linear Schrödinger-like equations. Thin layer models are also available for Maxwell's equations and elastodynamics.

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

MAGNOME Project-Team

5. Software and Platforms

5.1. Magus: Genome exploration and analysis

Participants: David James Sherman [correspondant], Pascal Durrens, Natalia Golenetskaya, Florian Lajus, Xavier Calcas.

The MAGUS genome annotation system integrates genome sequences and sequences features, *in silico* analyses, and views of external data resources into a familiar user interface requiring only a Web navigator. MAGUS implements annotation workflows and enforces curation standards to guarantee consistency and integrity. As a novel feature the system provides a workflow for simultaneous annotation of related genomes through the use of protein families identified by *in silico* analyses; this results in an n-fold increase in curation speed, compared to curation of individual genes. This allows us to maintain standards of high-quality manual annotation while efficiently using the time of volunteer curators. For more information see the MAGUS Gforge web site. [1] MAGUS $1.x$ is mature software used since 2006 by our collaboration partners. MAGUS 2.0 is developed in an Inria Technology Development Action (ADT) with an open-source license and is being deposited with the APP.

5.2. Pantograph: Inference of metabolic networks

Participants: David James Sherman [correspondant], Pascal Durrens, Nicolás Loira, Anna Zhukova.

Pantograph is a software tool for inferring whole-genome metabolic models for eukaryote cell factories. It is based on metabolic scaffolds, abstract descriptions of reactions and pathways on which inferred reactions are hung are are eventually connected by an interative mapping and specialization process. Scaffold fragments can be repeatedly used to build specialized subnetworks of the complete model. A novel feature of Pantograph is that it uses expert knowledge implicitly encoded in the scaffold's gene associations, and explicitly transfers this knowledge to the new model. Pantograph is available under an open-source license. For more information see the Pantograph Gforge web site. [2].

5.3. MetaModGen: Generalizing Metabolic Models

Participants: Anna Zhukova [correspondant], David James Sherman.

The metabolic model generalization and navigation software allows a human expert to explore a metabolic model in a layered manner. The software creates an on-line semantically zoomable representation of a model submitted by the user in SBML [3] format. The most general view represents the compartments of the model; the next view shows the visualization of generalized versions of reactions and metabolites in each compartment (see section 6.3); and the most detailed view visualizes the initial model with the generalization-based layout (where similar metabolites and reactions are placed next to each other). Zoomable representation is implemented using the Leaflet[4] JavaScript library for mobile-friendly interactive maps. Users can click on reactions and compounds to see the information about their annotations. An example of a zoomable representation of the peroxisome compartment of *Y. lipolytica* is available at http://metamogen.gforge.inria.fr/map.html.

5.4. BioRica: Multi-scale Stochastic Modeling

Participants: David James Sherman [correspondant], Rodrigo Assar Cuevas, Joaquin Fernandez.

---

[1] http://magus.gforge.inria.fr

[2] http://pathtastic.gforge.inria.fr

[3] http://sbml.org

[4] http://leafletjs.com

BioRica is a high-level modeling framework integrating discrete and continuous multi-scale dynamics within the same semantics field. A model in BioRica node is hierarchically composed of nodes, which may be existing models. Individual nodes can be of two types:

- Discrete nodes are composed of states and transitions described by guarded events. Behavior can be stochastic (defined by the likelihood that an event fires when activated) and timed (defined by the delay between an event's activation and the moment that its transition occurs).
- Continuous nodes are described by ODE systems, potentially a hybrid system whose internal state flows continuously while having discrete jumps.

The system has been implemented as a distributable software package. The BioRica compiler reads a specification for hierarchical model and compiles it into an executable simulator. The modeling language is a stochastic extension to the AltaRica [5] Dataflow language, inspired by work of Antoine Rauzy. Input parsers for SBML 2 version 4 are curently being validated. The compiled code uses the Python runtime environment and can be run stand-alone on most systems. For more information see the BioRica Gforge web site. [6] BioRica was developed as an Inria Technology Development Action (ADT) with an open-source license and is deposited with the APP.

5.5. Génolevures On Line: Comparative Genomics of Yeasts

Participants: Pascal Durrens [correspondant], Natalia Golenetskaya, Tiphaine Martin, David James Sherman.

The Génolevures online database provides tools and data for exploring the annotated genome sequences of more than 20 genomes, determined and manually annotated by the Génolevures Consortium to facilitate comparative genomic studies of hemiascomycetous yeasts. Data are presented with a focus on relations between genes and genomes: conservation of genes and gene families, speciation, chromosomal reorganization and synteny. The Génolevures site includes a private collaboration area for specific studies by members of its international community. The contents of the knowledge base are expanded and maintained by the CNRS through GDR 2354 Génolevures, and full data may be downloaded from the site. Génolevures online uses our open-source MAGUS system for genome navigation, with project-specific extensions developed by David Sherman, Pascal Durrens, and Tiphaine Martin; these extensions are not made available due to incertainty about intellectual property rights. For more information see the Génolevures web site. [7]

5.6. Inria Bioscience Resources

Participants: Olivier Collin [correspondant], Frédéric Cazals, Mireille Régnier, Marie-France Sagot, Hélène Touzet, Hidde De jong, David James Sherman, Marie-Dominique Devignes, Dominique Lavenier.

Inria Bioscience Resources is a portal designed to improve the visibility of bioinformatics tools and resources developed by Inria teams. This portal will help the community of biologists and bioinformatians understand the variety of bioinformatics projects in Inria, test the different applications, and contact project-teams. Eight project-teams participate in the development of this portal. Inria Bioscience Resources is developed in an Inria Technology Development Action (ADT).

---

[5] http://altarica.labri.fr

[6] http://biorica.gforge.inria.fr

[7] http://www.genolevures.org/

MASAIE Project-Team (section vide)

MNEMOSYNE Team

5. Software and Platforms

5.1. Positioning

Our previous works in the domain of well-defined distributed asynchronous adaptive computations [48], [45], [50] have already made us define a library (DANA [44]), closely related to both the notion of artificial neural networks and cellular automata. From a conceptual point of view, the computational paradigm supporting the library is grounded on the notion of a unit that is essentially a (vector of) potential that can vary along time under the influence of other units and learning. Those units can be organized into layers, maps and networks.

We also gather in the middleware EnaS (that stands for *Event Neural Assembly Simulation*; cf. http://gforge.inria.fr/projects/enas) our numerical and theoretical developments, allowing to simulate and analyze so called "event neural assemblies". In 2013, we have also integrated in this C/C++ middleware early-vision perception routines developed within the scope of the KEOpS project.

We will also have to interact with the High Performance Computing (HPC) community, since having large scale simulations at that mesoscopic level is an important challenge in our systemic view of computational neuroscience. Our approach implies to emulate the dynamics of thousands, or even millions, of integrated computational units, each of them playing the role of a whole elementary neural circuit (e.g. the microcolumn for the cortex). Mesoscopic models are considered in such an integrative approach, in order to exhibit global dynamical effect that would be hardly reachable by compartment models involving membrane equations or even spiking neuron networks.

The vast majority of high performance computing softwares for computational neuroscience addresses sub-neural or neural models [34], but coarser grained population models are also demanding for large scale simulations, with fully distributed computations, without global memory or time reference, as it is specified in (*cf.* § 3.2).

5.2. Dana

Participant: Nicolas Rougier.

DANA [44] is a python framework (http://dana.loria.fr) whose computational paradigm is grounded on the notion of a unit that is essentially a set of time dependent values varying under the influence of other units via adaptive weighted connections. The evolutions of a unit's value are defined by a set of differential equations expressed in standard mathematical notation which greatly ease their definition. The units are organized into groups that form a model. Each unit can be connected to any other unit (including itself) using a weighted connection. The DANA framework offers a set of core objects needed to design and run such models. The modeler only has to define the equations of a unit as well as the equations governing the training of the connections. The simulation is completely transparent to the modeler and is handled by DANA. This allows DANA to be used for a wide range of numerical and distributed models as long as they fit the proposed framework (e.g. cellular automata, reaction-diffusion system, decentralized neural networks, recurrent neural networks, kernel-based image processing, etc.).

5.3. ENAS: Event Neural Assembly Simulation

Participants: Frédéric Alexandre, 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. On the mnemosyne side, this middleware includes since 2013 functional simulations of the non-standard perceptive behavior of the retina (detection of visual events) based on intensity and local spatial intensity cues, while in 2014 we are going to extend these new developments to motion cues. We have also made a strong effort in terms of interoperability on our side, since main functions are now usable from other languages (especially the python wrapper, actually in use), while we still maintain the interoperability with . This development is a complement of what has been developed on the Neruomathcomp side where colleagues have invested in a GUI for their routines.

5.4. Virtual Enaction

Participants: Frédéric Alexandre, André Garenne, Nicolas Rougier, Thierry Viéville.

The computational models studied in this project have applications that extend far beyond what is possible to experiment yet in human or non-human primate subjects. Real robotics experimentations are also impaired by rather heavy technological constraints; for instance, it is not easy to dismantle a given embedded system in the course of emerging ideas. The only versatile environment in which such complex behaviors can be studied both globally and at the level of details of the available modeling is a virtual environment, as in video games, Such a system can be implemented as “brainy-bot” (a programmed player based on our knowledge of the brain architecture) which goal is to survive in a complete manipulable environment.

In order to attain this rather ambitious objective we are going to both (i) deploy an existing open-source video game middleware in order to be able to shape the survival situation to be studied and (ii) revisit the existing models in order to be able to integrate them as an effective brainy-bot. It will consist of a platform associated to a scenario that would be the closest possible to a survival situation (foraging, predator-prey relationship, partner approach to reproduction) and in which it would be easy to integrate an artificial agent with sensory inputs (visual, touch and smell), emotional and somatosensory cues (hunger, thirst, fear, ..) and motor outputs (movement, gesture, ..) connected to a "brain" whose architecture will correspond to the major anatomical regions involved in the issues of learning and action selection (cortex areas detailed here, basal ganglia, hippocampus, and areas dedicated to sensorimotor processes). The internal game clock will be slowed down enough to be able to run non trivial brainy-bot implementations. This platform has already being used by two students of the team and is now a new deliverable of the KEOpS project.

MODEMIC Project-Team

5. Software and Platforms

5.1. SMC Demos (Sequential Monte Carlo demos)

Participant: Fabien Campillo.

SMC Demos proposes a set of demonstration Matlab procedures for nonlinear filtering approximation via particle filtering (sequential Monte Carlo): bearing-only tracking with obstacles, tracking in digital terrain model, track-before-detect in a sequence of digital picture, mobile phone tracking based on the signal strength to nearby antenna. This software is deposited with the "Agence pour la Protection des Programmes" (APP, 7/7/2009) [2].

5.2. IBM Cellulose

Participant: Fabien Campillo.

In the context of the DISCO/ANR and MnMs/RNSC projects (see Sections 7.2.1 and 7.2.2), in collaboration with Ariane Bize (Irstea), the team has developed an individual-based model for the degradation of one cellulose bead (dozens of micrometers in diameter) by cellulolytic bacteria [3].

5.3. VITELBIO (VIrtual TELluric BIOreactors)

Participants: Jérôme Harmand, Alain Rapaport.

VITELBIO is a simulation tool for studying networks of interconnected chemostat models with the objective of mimicking microbial activities in heterogeneous media, such as the soil. This software, that has been developed with the help of ITK Company, is accessible on a server from any web navigator [4] and makes use of Flex for the user interface and Octave for the numerical integration. It is is no longer maintained but serves as a teaching support.

---

[2] http://www-sop.inria.fr/members/Fabien.Campillo/software/smc-demos/

[3] http://www-sop.inria.fr/members/Fabien.Campillo/software/ibm-cellulose/

[4] https://sites.google.com/site/vitelbio/logiciel

MOISE Project-Team

5. Software and Platforms

5.1. Adaptive Grid Refinement

Participants: Laurent Debreu, Marc Honnorat.

AGRIF (Adaptive Grid Refinement In Fortran, [85], [84]) is a Fortran 90 package for the integration of full adaptive mesh refinement (AMR) features within a multidimensional finite difference model written in Fortran. Its main objective is to simplify the integration of AMR potentialities within an existing model with minimal changes. Capabilities of this package include the management of an arbitrary number of grids, horizontal and/or vertical refinements, dynamic regridding, parallelization of the grids interactions on distributed memory computers. AGRIF requires the model to be discretized on a structured grid, like it is typically done in ocean or atmosphere modelling. As an example, AGRIF is currently used in the following ocean models: MARS (a coastal model developed at IFREMER-France), ROMS (a regional model developed jointly at Rutgers and UCLA universities), NEMO ocean modelling system (a general circulation model used by the French and European scientific community) and HYCOM (a regional model developed jointly by University of Miami and the French Navy).
In 2013, a new contract has been signed with IFREMER to add online degradation capabilities. The software will be used operationally to attain a resolution of 500meters along the French coasts. (http://www.previmer.org) AGRIF is licensed under a GNU (GPL) license and can be downloaded at its web site (http://ljk.imag.fr/MOISE/AGRIF/index.html).

5.2. NEMOVAR

Participants: Arthur Vidard, Pierre-Antoine Bouttier, Bénédicte Lemieux-Dudon.

NEMOVAR is a state-of-the-art multi-incremental variational data assimilation system dedicated to the European ocean modelling platform NEMO for research and operational applications. It is co-developed by MOISE, CERFACS (FR), ECMWF (EU) and MetOffice (UK) under the CeCILL license, written in fortran and python. It is now in use in both ECMWF and MetOffice for their operational oceanic forecasting systems. It has also been used for specific studies in collaboration with Mercator-Ocean, LPO, LOCEAN and LEGI in France and University of Namur in Belgium. It is also a likely candidate for becoming the future Black-Sea forecasting system of the Marine Hydrographical Institute of Ukraine with whom we collaborate actively. Previously part of NEMOVAR, NEMO-TAM (Tangent and adjoint models for NEMO) that have been developed by the MOISE team will be now distributed directly by the NEMO consortium. The first official tagged release including NEMO-TAM has been published early 2013.

5.3. DatIce

Participant: Bénédicte Lemieux-Dudon.

Antarctic and Greenland ice cores provide a mean to study the phase relationships of climate changes in both hemispheres. They also enable to study the timing between climate and greenhouse gases or orbital forcings. One key step for such studies is to improve the absolute and relative precisions of ice core age scales (for ice and trapped gas), and beyond that, to try to reach the best consistency between chronologies of paleo-records of any kind.

The DatIce tool is designed to increase the consistency between pre-existing core chronologies (also called background). It formulates a variational inverse problem which aims at correcting three key quantities that uniquely define the core age scales: the accumulation rate, the total thinning function, and the close-off depth. For that purpose, it integrates paleo-data constraints of many types among which age markers (with for instance documented volcanoes eruptions), and stratigraphic links (with for instance abrupt changes in methane concentration). A cost function is built that enables to calculate new chronologies by making a trade-off between all the constraints (background chronologies and paleo-data).

DatIce enables to circumvent the limits encountered with other dating approaches, in particular because it controls the model errors, which are still large despite efforts to better describe the firn densification, the ice flow and the forcing fields (ice sheet elevation, temperature and accumulation rate histories). Controlling the model error makes it possible to assimilate large set of observations, to constrain both the gas and ice age scales, and to apply the process on several cores at the same time by including stratigraphic links between cores. This approach greatly improves the consistency of ice cores age scales.

The method presented in [93], [94] has already been applied simultaneously to EPICA EDML and EDC, Vostok and NGRIP drillings. The code has also been applied in two publications [78] and [106] which aimed at the construction of a unified chronology for Antarctic ice cores. LGGE, LSCE and MOISE are partners to extend the code to marine and terrestrial cores. On going development efforts are made to ensure the robustness of the dating solution (diagnostics on the assimilation system, calibration of the background error covariance matrices).

5.4. SDM toolbox

Participant: Antoine Rousseau.

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

5.5. CompModSA package

Participants: Clémentine Prieur, Alexandre Janon, Céline Helbert.

Alexandre Janon is a contributor of the packages CompModSA - Sensitivity Analysis for Complex Computer Models (see http://cran.open-source-solution.org/web/packages/CompModSA/index.html), and sensitivity (see http://cran.r-project.org/web/packages/sensitivity/index.html). These packages are useful for conducting sensitivity analysis of complex computer codes.

Celine Helbert is now the maintainer of the packages DiceDesign (see http://cran.r-project.org/web/packages/DiceDesign/index.html) and DiceEval (see http://cran.r-project.org/web/packages/DiceEval/index.html). These packages are useful for conducting design and analysis of computer experiments.

MORPHEME Project-Team (section vide)

NEUROMATHCOMP Project-Team

4. Software and Platforms

4.1. Virtual Retina: A Large-Scale Simulator of Biological Retina

Participants: Bruno Cessac, Maria-Jose Escobar [Universidad Técnica Federico Santa María, Valparaiso, Chile], Christobal Nettle [Universidad Técnica Federico Santa María, Valparaiso, Chile], Pierre Kornprobst, Adrien Wohrer [Group for Neural Theory - ENS, Paris, France].

Virtual Retina is a simulation software developped by Adrien Wohrer during his PhD [73], [72] that allows large-scale simulations of biologically-plausible retinas.

Virtual Retina has a variety of biological features implemented such as (i) spatio-temporal linear filter implementing the basic center/surround organization of retinal filtering, (ii) non-linear contrast gain control mechanism providing instantaneous adaptation to the local level of contrast; (iii) spike generation by one or several layers of ganglion cells paving the visual field.

Virtual Retina is under Inria CeCill C open-source licence, so that one can download it, install it and run it on one's own image sequences. Virtual Retina also offers a web service (v 2.1), so that users may test directly the main software on their own data, without any installation. This webservice was developed in collaboration with Nicolas Debeissat (engineer, 2002).

We are now interested in the analysis of the collective behavior of ganglion cells responses. To take this collective behavior into account, Virtual Retina needs to be extended since in its current version, ganglion cells are independent. The goal is to produce better retinal models from experimental recordings obtained with our collaborators at the Institut de la Vision (Olivier Marre and Serge Picaud), Evelyne Sernagor (New Castle University) and Luca Berdondini (IIT) using e.g. multi-electrode arrays. This will allow us to better understand the correlations between retina spikes trains and to improve the Virtual Retina model [72] in such a way that it could reproduce the retinal response at the population level. Another application is to the electric stimulation of a retina with implanted multi-electrode arrays in collaboration with the Institut de la Vision and the INT (Frédéric Chavane). Other evolutions of Virtual Retina are also investigated by external partners like the role/implementation of starbust amacrine cells involved in direction selectivity (collaboration with Universidad Técnica Federico Santa María, Valparaiso, Chile, and Centro de Neurociencia de Valaparaiso) (see also e.g., [64]).

- IDDN number: IDDN.FR.001.210034.000.S.P.2007.000.31235
- Version: v 2.2.2 (September 2011)
- Link: http://www-sop.inria.fr/neuromathcomp/public/software/virtualretina

4.2. Event Neural Assembly Simulation

Participants: Bruno Cessac, Sélim Kraria [Inria DREAM], Gaia Lombardi, Hassan Nasser, Wahiba Tahouali.

Enas is a library providing numerical tools for the simulation of neural networks and the analysis of spike trains either coming from neural simulators or from biological experiments. The goal is to provide statistical methods allowing to estimate a spatio-temporal statistical model of spike train statistics (including thus pairwise spatio-temporal correlations, but also higher order correlations) from experimental rasters. More precisely, the algorithms are based on our theoretical results on spike trains statistical analysis via Gibbs distributions. We estimate a parametric Gibbs potential optimally characterizing the statistics of empirical spike trains (by minimisation of the Kullback-Leibler divergence between the empirical measure and the Gibbs measure). From this, classical statistical indicators such as firing rate, correlations, higher order moments and statistical entropy are obtained. Also, the form of the Gibbs potential provides essential informations on the underlying neural network and its structure. This method does not only allows us to estimate the spikes statistics but also to compare different models, thus answering such questions about the neural code as: are correlations (or time synchrony or a given set of spike patterns,...) significant with respect to rate coding?

Compared to existing software (Pandora; Sigtool; Spyke Viewer; Orbital Spikes) Enas offers new computational methods taking into account time constraints in neural networks (such as memory effects), based on theoretical methods rooted in statistical physics and applied mathematics. The algorithms used are based on linear programming, nonlinear parameter estimations, statistical methods. The C/C++ code has been organized as “bean java” to ease its use by programmers non specialized in advanced object programming. As a consequence the code is distributed in the form of an include source for the lightest and the most universal integration into users codes.

Event neural assembly simulation is developed under CeCILL C licence
APP logiciel Enas: IDDN.FR.OO1.360008.000.S.P.2009.000.10600.

It has benefited from the support of an ADT Inria from 2011 to 2013.

The software is freely downloadable at http://enas.gforge.inria.fr/v3/download.html.

Website: `http://enas.gforge.inria.fr/`

NEUROSYS Team

5. Software and Platforms

5.1. Software and Platform

5.1.1. Visualization

- The NeuralFieldSimulator [1] computes numerically activity in two-dimensional neural fields by solving integral-differential equations involving transmission delays and visualizes the spatio-temporal activity. The tool includes a GUI that allows the user to choose field parameters. It is written in Python, open-source and is aimed to be promoted to become a major graphical visualization tool in the domain of neural field theory.
- *A*naesthesiaSimulator [2] simulates the activity of networks of spiking neurons subject to specific receptor dynamics. The tool is a platform to test effects of anaesthetics on neural activity and is still in its first stage of development. The neural activity is planned to be visualized in a 2D and 3D-plot evolving in time. It is written in Python, open-source and involves heavily the simulation package BRIAN [3].

5.1.2. Platforms

OpenViBE [4] 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, high-performance, portability, its multiple-users facilities and its connection with high-end/Virtual Reality displays. The designer tool 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 [5] under the terms of the LGPL-V2 license. The development of OpenVibe is done in association with other Inria research teams (Hybrid, Athena, Potioc) for the national Inria project: ADT OpenViBE-NT. Neurosys is in charge of machine learning techniques and the interoperability with other tools such as Matlab, BCI2000, or TOBI.

5.1.3. Others

The package DEvariants [6] includes Matlab routines which implements new variants of the Differential Evolution (an evolutionary algorithm) strategies. The novelty lies in the selection process where we proposed to use a multinomial law to recombine the individuals/vectors. Compared to the standard strategies, our variants allow a faster convergence and a better avoidance of local minima. The different variants are provided with a test sample of functions, the DeJong benchmark. The audience is any scientific user familiar with evolutionary optimization.

---

[1] https://gforge.inria.fr/projects/nfsimulator/

[2] https://gforge.inria.fr/projects/anasim/

[3] http://briansimulator.org/

[4] http://openvibe.inria.fr/

[5] https://gforge.inria.fr/projects/openvibe/

[6] https://sites.google.com/site/laurebuhry/publications/optimization-algorithms

NUMED Project-Team

4. Software and Platforms

4.1. SimPHyT

SimPHyt has been developed by Morgan Martinet (junior engineer). SimPHyt is an implementation in Python of the low grad glioma model developped by Benjamin Ribba. The aim is to predict the evolution of the glioma size of patients. It is used by Dr François Ducray in Pierre Wertheimer Hospital in Lyon.

4.2. SETIS

We are currently developing the SETIS software which is a GUI allowing to treat DICOM medical images to extract pathological data. These data can then be exported and used in a SAEM software (including Monolix (Inria & Lixoft)) for the parameters' estimation of models in the context of population approaches. As an example SETIS can be used to segment and compute the tumor size of a patients from MRI scans taken at different times. The software is sufficiently general to be used in various situations by clinicians (already done by our colleagues in Lyon Hospital). It will be freely distributed and is based on open source technology, so that it can easily be adapted to specific needs by other users.

4.3. Zebre

Participant: Thierry Dumont [correspondant].

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

This code was first designed to serve as demonstrator of the theoretical results of Descombes and Massot on the solution of stiff reaction-diffusion systems by alternate directions methods, and as a first step towards complex chemistry simulations. Later it was used and improved to solve the ionic model of strokes, and incorporated stabilized explicit Runge Kutta methods for diffusion steps. Coded in C++, it solves stiff systems with various schemes in dimension 1, 2 and 3, in complex geometries. The code is multithreaded.

4.4. OptimChemo

Participants: Violaine Louvet [correspondant], Emmanuel Grenier.

OptimChemo is a userfriendly software designed to study numerically the effect of multiple chemotherapies on simple models of tumour growth and to optimize chemotherapy schedules.

4.5. Simstab

Stability prediction of vaccine, intellectual property of Sanofi, covered by a US patent demand (Sanofi, Benjamin Ribba, Emmanuel Grenier).

4.6. Bingham flows

A 1D and 2D code with a new method for the computation of viscoplatic flows with free-surface. It essentially couples Optimization methods and Well-Balanced Finite-Volumes schemes for viscous shallow-water equations (induced by the viscoplastic nature of the fluid). Currently applied to avalanches of dense snow, it is a private code currently actively developed (in C++). One of the key feature is that its well-balanced property allows to obtained the stationary states which are linked to the stopping of the snow avalanche for this highly non-linear type of fluid.

PARIETAL Project-Team

5. Software and Platforms

5.1. Scikit learn

Participants: Bertrand Thirion, Gaël Varoquaux, Olivier Grisel [correspondant], Jaques Grobler, Alexandre Gramfort, Fabian Pedregosa, Virgile Fritsch.

Scikit-learn is an open-source machine learning toolkit written in Python/C that provides generic tools to learn information for the classification of various kinds of data, such as images or texts. It is tightly associated to the scientific Python software suite (numpy/scipy) for which it aims at providing a complementary toolkit for machine learning (classification, clustering, dimension reduction, regression). There is an important focus on code quality (API consistency, code readability, tests, documentation and examples), and on efficiency, as the scikit-learn compares favorably to state-of-the-art modules developed in R in terms of computation time or memory requirements. Scikit-learn is currently developed by more than 60 contributors, but the core developer team has been with the Parietal Inria team at Saclay-Île-de-France since January 2010. The scikit-learn has recently become the reference machine learning library in Python.

- Version: 0.14
- Programming language: Python, C/Cython

5.2. Nilearn

Participants: Bertrand Thirion, Gaël Varoquaux [correspondant], Philippe Gervais, Jaques Grobler, Alexandre Gramfort, Fabian Pedregosa, Alexandre Abraham, Michael Eickenberg.

NiLearn is the neuroimaging library that adapts the concepts and tools of the scikit learn to neuroimaging problems. As a pure Python library, it depends on scikit learn and nibabel, the main Python library for neuroimaging I/O. It is an open-source project, available under BSD license. The two key components of NiLearn are *i)* the analysis of functional connectivity (spatial decompositions and covariance learning) and *ii)* the most common tools for multivariate pattern analysis. A great deal of efforts has been put on the efficiency of the procedures both in terms of memory cost and computation time. NiLearn is maintained both through the help of Inria: (a developer funded by Saclay CRI in 2012-2013, a 2013-2014 ADT, and through the NiConnect project (P. Gervais).

- Version: 0.1
- Programming language: Python

5.3. Mayavi

Participant: Gaël Varoquaux [Correspondant].

Mayavi is the most used scientific 3D visualization Python software (http://mayavi.sourceforge.net/). It has been developed by Prabhu Ramachandran (IIT Bombay) and Gaël Varoquaux (PARIETAL, Inria Saclay). Mayavi can be used as a visualization tool, through interactive command line or as a library. It is distributed under Linux through Ubuntu, Debian, Fedora and Mandriva, as well as in PythonXY and EPD Python scientific distributions. Mayavi is used by several software platforms, such as PDE solvers (fipy, sfepy), molecule visualization tools (http://pyrx.scripps.edu) and brain connectivity analysis tools (connectomeViewer).

See also the web page http://mayavi.sourceforge.net/ and the following paper http://hal.inria.fr/inria-00528985/en.

- Version: 3.4.0

5.4. Nipy

Participants: Bertrand Thirion [correspondant], Virgile Fritsch, Elvis Dohmatob, Gaël Varoquaux.

Nipy is an open-source Python library for neuroimaging data analysis, developed mainly at Berkeley, Stanford, MIT and Neurospin. It is open to any contributors and aims at developing code and tools sharing. Some parts of the library are completely developed by Parietal and LNAO (CEA, DSV, Neurospin). It is devoted to algorithmic solutions for various issues in neuroimaging data analysis. All the nipy project is freely available, under BSD license. It is available in NeuroDebian.

See also the web page http://nipy.org.

- Version: 0.3

5.5. MedInria

Participants: Pierre Fillard [correspondant], Sergio Medina, Viviana Siless.

MedInria is a free collection of softwares developed within the ASCLEPIOS, ATHENA and VISAGES research projects. It aims at providing to clinicians state-of-the-art algorithms dedicated to medical image processing and visualization. Efforts have been made to simplify the user interface, while keeping high-level algorithms. MedInria is available for Microsoft windows XP/Vista, Linux Fedora Core, MacOSX, and is fully multi-threaded.
See also the web page http://med.inria.fr/.

- Version: 2.0

5.6. PyHRF

Participants: Philippe Ciuciu [correspondant], Solveig Badillo, Aina Frau Pascual.

PyHRF is a set of tools for within-subject fMRI data analysis, focused on the characterization of the hemodynamics. Within the chain of fMRI data processing, these tools provide alternatives to the classical within-subject GLM estimation step. The inputs are preprocessed within-subject data and the outputs are statistical maps and/or fitted HRFs. The package is mainly written in Python and provides the implementation of the two following methods:

- The joint-detection estimation (JDE) approach, that divides the brain into functionally homogeneous regions and provides one HRF estimate per region as well as response levels specific to each voxel and each experimental condition. This method embeds a temporal regularization on the estimated HRFs and an adaptive spatial regularization on the response levels.
- The Regularized Finite Impulse Response (RFIR) approach, that provides HRF estimates for each voxel and experimental conditions. This method embeds a temporal regularization on the HRF shapes, but proceeds independently across voxels (no spatial model).

The development of PyHRF is now funded by an Inria ADT, in collaboration with MISTIS.

- Version: 0.1
- Keywords: Hemodynamic response function; estimation; detection; fMRI
- License: BSD 4
- Multiplatform: Windows - Linux - MacOSX
- Programming language: Python

POMDAPI Project-Team

4. Software and Platforms

4.1. LifeV

Participant: Michel Kern.

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

Version 3.1.1

Programming language: C++

http://www.lifev.org/

4.2. M1cg1

Participant: Jean Charles Gilbert.

M1cg1 solves convex quadratic optimization problems and builds preconditioning matrices.

Version: 1.2

Programming language: Fortran 77

14 downloads in 2013

https://who.rocq.inria.fr/Jean-Charles.Gilbert/modulopt/optimization-routines/m1cg1/m1cg1.html

4.3. M1qn3

Participant: Jean Charles Gilbert.

M1qn3 solves very large scale differentiable optimization problems.

Version: 3.3

Programming language: Fortran 77

36 downloads in 2013

https://who.rocq.inria.fr/Jean-Charles.Gilbert/modulopt/optimization-routines/m1qn3/m1qn3.html

In collaboration with Claude Lemaréchal (project-team Bipop)

4.4. Oqla, Qpalm

Participants: Jean Charles Gilbert, Émilie Joannopoulos.

Oqla and Qpalm aim at solving large scale convex quadratic functions on a polyhedron by an augmented Lagrangian method.

Versions (in development): 0.1 (Oqla), 0.2 (Qpalm)

Programming languages: `C++` (Oqla), `Matlab` (Qpalm)

4.5. Ref-image

Participants: Hend Ben Ameur, François Clément, Pierre Weis.

Ref-image is an image segmentation program using optimal control techniques. Slogan is “no gestalt inside”. Ref-image implements the refinement indicator algorithm, specialized to the case of the inversion of the identity map. It is a first step towards the implementation of a generic inversion platform using the refinement indicator algorithm.

Version: 1.0+pl0

Programming language: OCaml

http://refinement.inria.fr/ref-image/

4.6. SQPlab

Participant: Jean Charles Gilbert.

SQPlab solves constrained differentiable optimization problems.

Version: 0.4.5

Programming language: Matlab

200 downloads in 2013

https://who.rocq.inria.fr/Jean-Charles.Gilbert/modulopt/optimization-routines/sqplab/sqplab.html

4.7. Sklml

Participants: François Clément, Pierre Weis.

Sklml is a functional parallel skeleton compiler and programming system for OCaml programs. Slogan is “easy coarse grain parallelization”.

Version: 1.1+pl0

Programming language: OCaml

http://sklml.inria.fr/

4.8. FreeFem++

Participants: Martin Vohralík, Martin Čermák, Zuqi Tang.

The scientific calculation code FreeFem++ is an excellent example of a complex software numerical simulation tool. It in particular encompasses all specification of the problem, the choice and implementation of the numerical method, the choice and implementation of the linearization method (nonlinear solver), and the choice and implementation of the method of solution of the associated linear systems (linear solver). In the post-doc stays of M. Čermák and Z. Tang, we integrate there the most recent advances of the theory of a posteriori error estimation and of adaptive algorithms. In particular, adaptive stopping criteria for the linear and nonlinear solvers are being implemented.

Version 3.26-2

Programming language: C++

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

Popix Team

5. Software and Platforms

5.1. Monolix

Participants: Marc Lavielle, Hector Mesa, Célia Barthélémy.

MONOLIX is an easy, fast and powerful tool for parameter estimation in nonlinear mixed-effect models, model diagnosis and assessment, and advanced graphical representation. It is a platform of reference for model-based drug development. Pharmacometricians and biostatisticians can rely on MONOLIX for population analysis and to model PK/PD and other complex biochemical and physiological processes.

MONOLIX was developed by Inria until June 2011. The start-up Lixoft now develops and supports MONOLIX. POPIX collaborates closely with Lixoft to convert research results into new user features available in MONOLIX.

5.2. MLXtran

Participant: Marc Lavielle.

MONOLIX is associated with MLXtran, a powerful and immediately readable declarative language for describing complex pharmacometric and statistical models. MLXtran can be used and interfaced with various environments, e.g., R, Matlab, etc.

POPIX collaborates closely with Lixoft on the definition of the specifications and the syntax of MLXtran. Implementation is then ensured by Lixoft.

5.3. Clinical trial simulator

Participants: Marc Lavielle, Elodie Maillot, Laura Brocco, Fazia Bellal, Célia Barthélémy.

A clinical trial simulator (CTS) enables effective implementation of the learn-and-confirm paradigm in drug development. Through simulations the anticipated success rate of a future trial can be estimated. For various reasons industry has not embraced currently available software for trial simulation. A new tool is essential for Model Based Drug Development (MBDD).

POPIX is responsible for developing a new CTS within the DDMoRe project (see below). Version 3 of the CTS is available since June 2013. The capabilities of this new version comprise:

- Flexible study designs used in Phase 2 of clinical drug development: parallel group studies, crossover studies, complex treatments defined as a combination of different treatments
- Simulation of patients sampled from a joint distribution or using an external data file
- Simulation of exposure to the investigated drug and several types of drug effects related to drug exposure (continuous, categorical, count, time-to-event)
- Graphics and statistical tests
- Automatic reporting

5.4. MLXplore

Participants: Marc Lavielle, Laura Brocco.

MLXplore is a graphical and interactive software for the exploration and visualization of complex pharmacometric models. MLXplore also includes the ability to study the statistical variability of the models, and to model and study complex administration designs.

MLXplore does not require MONOLIX, although they make for a powerful combination, enabling to use the same, human-readable model description, to finely explore the properties of the model on the one hand, and on the other hand use the same model for advanced parameter estimation in the context of population analysis and mixed effect statistics.

MLXplore is an ideal tool to learn about pharmacometric models and population analysis, and is used extensively in the online wiki WikiPopix created by POPIX, found at: https://wiki.inria.fr/popix.

MLXplore is developed by Lixoft. POPIX collaborates closely with Lixoft on on the definition of the specifications of MLXplore.

REO Project-Team

5. Software and Platforms

5.1. LiFE-V library

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

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

5.2. Mistral library

Participant: Jean-Frédéric Gerbeau [correspondant].

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

5.3. FELiScE

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

FELiScE – standing for "Finite Elements for Life Sciences and Engineering" – is a new finite element code which the MACS and REO project-teams have decided to jointly develop in order to build up on their respective experiences concerning finite element simulations. One specific objective of this code is to provide in a unified software environment all the state-of-the-art tools needed to perform simulations of the complex cardiovascular models considered in the two teams – namely involving fluid and solid mechanics, electrophysiology, and the various associated coupling phenomena. FELISCE is written in C++, and may be later released as an opensource library. https://gforge.inria.fr/projects/felisce/

5.4. SHELDDON

Participant: Marina Vidrascu [correspondant].

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

---

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

SAGE Project-Team

5. Software and Platforms

5.1. Hydrogeology

5.1.1. H2OLab

Participants: Thomas Dufaud, Jocelyne Erhel [correspondant], Grégoire Lecourt, Aurélien Le Gentil, Géraldine Pichot.

The software platform H2OLab is devoted to stochastic simulations of groundwater flow and contaminant transport in highly heterogeneous porous and fractured geological media. It contains a database which is interfaced through the web portal H2OWeb. It contains also software modules which can be used through the interface H2OGuilde. The platform H2OLab is an essential tool for the dissemination of scientific results. Currently, software and database are shared by the partners of the h2mno4 project (see 8.2.1). Software integrated in the platform and registered at APP are GW-UTIL, GW-NUM, PARADIS, MP-FRAC.

See also the web page http://h2olab.inria.fr.

5.1.2. GW-UTIL

Participants: Jocelyne Erhel, Grégoire Lecourt, Aurélien Le Gentil, Géraldine Pichot [correspondant].

- Version: version 1.0, May 2008
- APP: registered
- Programming language: C++
- See also: http://h2olab.inria.fr.
- Abstract: The software GW-UTIL allows to discretize PDE for flow and transport in aquifers and to deal with stochastic models. It contains a set of utilitary modules for geometry, input, output, random numbers, visualization, parallel computing, numerical algorithms, etc. A package is devoted to launch applications.
- Current work: refactoring.

5.1.3. GW-NUM

Participants: Thomas Dufaud, Jocelyne Erhel, Grégoire Lecourt, Aurélien Le Gentil, Géraldine Pichot [correspondant].

- Version: version 1.0, May 2008
- APP: registered
- Programming language: C++
- See also: http://h2olab.inria.fr.
- Abstract: The software GW-NUM is a set of generic modules to discretize PDE of flow and transport in 2D computational domains in order to deal with stochastic models. Methods for flow simulations are either Finite Volume on structured meshes or Mixed Finite Element with unstructured meshes. Method for transport simulations is a particle tracker for advection and a random walker for diffusion. Uncertainty Quantification method is Monte-Carlo. For flow computations, the involved linear system is solved by external software devoted to sparse matrices.
- Current work: refactoring.

5.1.4. MP-FRAC

Participants: Thomas Dufaud, Jocelyne Erhel, Aurélien Le Gentil, Géraldine Pichot [correspondant].

- Version: version 1.0, May 2008
- APP: registered
- Programming language: C++
- See also: http://h2olab.inria.fr.
- Abstract: The software MP-FRAC aims at modelling and simulating numerically flow in a fractured aquifer. The physical domain is a network of fractures, either deterministic or stochastic, with a permeability field either deterministic or stochastic. The software computes the velocity field in the aquifer, by assuming that the medium is saturated and that flow is steady-state. Physical equations are stochastic PDEs, handled by a Monte-Carlo method. This non intrusive approach generates a set of random samples, which are used for simulations. Then, the software analyzes statistically the flow in the stochastic case. The objective is to characterize hydraulic properties in Discrete Fracture Networks. The software MP-FRAC handles a simulation corresponding to one sample, whereas Monte-Carlo method is implemented in a generic way by the software GW-NUM. The software is specific of the physical model (Discrete Fracture Network) and of the application (steady-state flow). Generic numerical methods to discretize PDE are implemented in the software GW-NUM.
- Current work: refactoring and design of libraries.

5.1.5. PARADIS

Participants: Jocelyne Erhel, Grégoire Lecourt, Aurélien Le Gentil, Géraldine Pichot [correspondant].

- Version: version 1.0, May 2008
- APP: registered
- Programming language: C++
- See also: http://h2olab.inria.fr/.
- Abstract: The software PARADIS aims at modelling and simulating numerically flow in a porous aquifer and transport by convection-diffusion of an inert solute. The porous medium is heterogeneous, with a stochastic or deterministic permeability field. A first step computes the velocity filed in the aquifer, by assuming that the medium is saturated and that flow is steady-state. A second step computes the distribution of solute concentration, by assuming a transport by convection and by molecular diffusion. Physical equations are stochastic PDEs, handled by a Monte-Carlo method and discretized by numerical methods. This non intrusive approach generates a set of random samples, which are used for simulations. Then, the software analyzes statistically the flow in the stochastic case. The objectives are to determine asymptotic laws of transport, to characterize pre-asymptotic behavior and to define global laws.

  The software PARADIS handles a simulation corresponding to one sample, whereas Monte-Carlo method is implemented in a generic way by the software GW-NUM. The software is specific of the physical model (heterogeneous porous medium) and of the application (steady-state flow then transport with macro-dispersion). Generic numerical methods to discretize PDE are implemented in the software GW-NUM.
- Current work: refactoring and design of libraries.

5.1.6. GRT3D

Participants: Édouard Canot, Jocelyne Erhel [correspondant], Souhila Sabit.

- Version: version 1.0, April 2011
- APP: registered
- Programming language: C

- Abstract: Reactive transport modeling has become an essential tool for understanding complex environmental problems. It is an important issue for MoMaS partners (see section 8.2.7), in particular Andra (see section 7.1). We have developed a method coupling transport and chemistry, based on a method of lines such that spatial discretization leads to a semi-discrete system of algebraic differential equations (DAE system). The main advantage is to use a complex DAE solver, which controls simultaneously the timestep and the convergence of Newton algorithm. The approach SIA uses a fixed-point method to solve the nonlinear system at each timestep, whereas the approach SNIA uses an explicit scheme.

  The software suite GRT3D has four executable modules:

  - SIA1D: Sequential Iterative Approach for 1D domains;
  - GDAE1D: Global DAE approach for 1D domains;
  - SNIA3D: Sequential Non Iterative Approach for 1D, 2D or 3D domains.
  - GDAE3D: Global DAE approach for 1D, 2D or 3D domains. This module has three variants: the original one with logarithms, an optimized one still with logarithms, an optimized one which does not use logarithms.
- Current work: extension of the chemistry module and parallelization.

5.1.7. SBM

Participant: Géraldine Pichot [correspondant].

- Version: version 1.0, November 2013
- Programming language: C
- Abstract: SBM (Skew Brownian Motion) is a code developed with A. Lejay (Inria, Nancy). This code allows exact or approximated simulations of the Skew Brownian Motion. This code is used for the simulation, with a Monte-Carlo approach, of a 1D diffusion process with a discontinuous diffusion coefficient. Several benchmark tests are also implemented.
- Current work: paper about benchmarking results.

5.2. High Performance Scientific Computing

5.2.1. PALMTREE

Participants: Lionel Lenôtre [correspondant], Géraldine Pichot.

- Version: version 1.0, November 2013
- Programming language: C++
- Abstract: We present an easy-to-use package for the parallelization of Lagrangian methods for partial differential equations. In addition to the reduction of computation time, the code aims at satisfying three properties:

  - simplicity: the user just has to add the algorithm governing the behaviour of the particles.
  - portability: the possibility to use the package with any compiler and OS.
  - action-replay: the ability of the package to replay a selected batch of particles.

  The last property allows the user to replay and capture the whole sample path for selected particles of a batch. This feature is very useful for debugging and catching some relevant information.
- Current work: paper about performance results.

5.2.2. GPREMS

Participants: Édouard Canot, Jocelyne Erhel [correspondant].

- Version: version 1.0, May 2008
- APP: registered
- Programming language: C++
- See also: http://www.irisa.fr/sage/.
- Abstract: GPREMS implements a robust hybrid solver for large sparse linear systems that combines a Krylov subspace method as accelerator with a Schwarz-based preconditioner. This preconditioner uses an explicit formulation associated to one iteration of the multiplicative Schwarz method. The Newton-basis GMRES, which aims at expressing a good data parallelism between subdomains is used as accelerator.

5.2.3. DGMRES

Participant: Jocelyne Erhel [correspondant].

- Version: version 1.0, June 2011
- APP: distributed with the free software PETSC
- Programming language: C
- See also: http://www.irisa.fr/sage/.
- Abstract: DGMRES implements a preconditioner based on adaptive deflation, which can be used with any preconditioner for the GMRES algorithm.

5.2.4. AGMRES

Participant: Jocelyne Erhel [correspondant].

- Version: version 1.0, November 2011
- APP: distributed with the free software PETSC
- Programming language: C
- See also: http://www.irisa.fr/sage/.
- Abstract: AGMRES implements an augmented subspace approach, based on adaptive deflation, which can be used with any preconditioner for the GMRES algorithm. It also implements a Newton basis for enhancing parallelism.

5.2.5. PPAT

Participants: Édouard Canot [corresponding author], Bernard Philippe.

PPAT (Parallel PATh following software) is a parallel code, developed by D. Mezher, W. Najem (University of Saint-Joseph, Beirut, Lebanon) and B. Philippe. This tool can follow the contours of a functional from $\mathbb{C}$ to $\mathbb{R}^+$. The present version is adapted for determining the level curves of the function $f(z) = \sigma_{\min}(A - zI)$ which gives the pseudospectrum of matrix A.

The algorithm is reliable: it does not assume that the curve has a derivative everywhere. The process is proved to terminate even when taking into account roundoff errors. The structure of the code spawns many independent tasks which provide a good efficiency in the parallel runs.

The software can be downloaded under the GPL licence from: http://sourceforge.net/projects/ppat.

5.2.6. MUESLI

Participant: Édouard Canot [corresponding author].

Doing linear algebra with sparse and dense matrices is somehow difficult in scientific computing. Specific libraries do exist to deal with this area (*e.g.* BLAS and LAPACK for dense matrices, SPARSKIT for sparse ones) but their use is often awful and tedious, mainly because of the large number of arguments which must be used. Moreover, classical libraries do not provide dynamic allocation. Lastly, the two types of storage (sparse and dense) are so different that the user must know in advance the storage used in order to declare correctly the corresponding numerical arrays.

MUESLI is designed to help in dealing with such structures and it provides the convenience of coding in Fortran with a matrix-oriented syntax; its aim is therefore to speed-up development process and to enhance portability. It is a Fortran 95 library split in two modules: (i) FML (Fortran Muesli Library) contains all necessary material to numerically work with a dynamic array (dynamic in size, type and structure), called `mfArray`; (ii) FGL (Fortran Graphics Library) contains graphical routines (some are interactive) which use the `mfArray` objects.

MUESLI includes some parts of the following numerical libraries: Arpack, Slatec, SuiteSparse, Triangle, BLAS and LAPACK.

Linux is the platform which has been used for developing and testing MUESLI. Whereas the FML part (numerical computations) should work on any platform (*e.g.* Win32, Mac OS X, Unix), the FGL part is intended to be used only with X11 (*i.e.* under all UNIXes).

Last version of MUESLI is 2.6.6 (2012-08-29). More information can be found at: http://people.irisa.fr/Edouard.Canot/muesli

5.2.7. CANARD

Participant: Édouard Canot [corresponding author].

When dealing with non-linear free-surface flows, mixed Eulerian-Lagrangian methods have numerous advantages, because we can follow marker particles distributed on the free-surface and then compute with accuracy the surface position without the need of interpolation over a grid. Besides, if the liquid velocity is large enough, Navier-Stokes equations can be reduced to a Laplace equation, which is numerically solved by a Boundary Element Method (BEM); this latter method is very fast and efficient because computing occur only on the fluid boundary. This method has been applied to the spreading of a liquid drop impacting on a solid wall and to the droplet formation at a nozzle; applications take place, among others, in ink-jet printing processes.

The code used (CANARD) has been developped with Jean-Luc Achard (LEGI, Grenoble) for fifteen years and is used today mainly through collaborations with Carmen Georgescu at UPB (University Polytechnica of Bucarest, Romania), and with Alain Glière (CEA-LETI, Grenoble).

SERPICO Project-Team

5. Software and Platforms

5.1. Software for live cell imaging

Participants: Charles Kervrann [(contact)], Patrick Bouthemy, Tristan Lecorgne, Thierry Pécot.

Motion2d: parametric motion model estimation

The MOTION2D software written in C++ (APP deposit number: FR.001.520021.001.S.A.1998.000.21000 / release 1.3.11, January 2005) and JAVA (plug-in IMAGEJ (http://rsbweb.nih.gov/ij/) is a multi-platform object-oriented library to estimate 2D parametric motion models in an image sequence. It can handle several types of motion models, namely, constant (translation), affine, and quadratic models. Moreover, it includes the possibility of accounting for a global variation of illumination and more recently for temporal image intensity decay (e.g. due to photo-bleaching decay in fluorescence microscopy). The use of such motion models has been proved adequate and efficient for solving problems such as optic flow computation, motion segmentation, detection of independent moving objects, object tracking, or camera motion estimation, and in numerous application domains (video surveillance, visual servoing for robots, video coding, video indexing), including biological imaging (image stack registration, motion compensation in videomicroscopy). Motion2D is an extended and optimized implementation of the robust, multi-resolution and incremental estimation method (exploiting only the spatio-temporal derivatives of the image intensity function) [47]. Real-time processing is achievable for motion models involving up to six parameters. Motion2D can be applied to the entire image or to any pre-defined window or region in the image.

Free academic software distribution: Motion2D Free Edition is the version of Motion2D available for development of Free and Open Source software only. More information on Motion2D can be found at http://www.irisa.fr/vista/Motion2D and the software can be downloaded at the same Web address (about 1650 downloads registered).
On-line demo: Mobyle@SERPICO http://mobyle-serpico.rennes.inria.fr/cgi-bin/portal.py#forms::Motion2D.

Partner: Fabien Spindler (Inria Lagadic team).

ND-Safir and Fast2D-SAFIR: Image denoising software

The ND-SAFIR software (APP deposit number: IDDN.FR.001.190033.002.S.A.2007.000.21000 / new release 3.0 in 2013) written in C++, JAVA and MATLAB, removes additive Gaussian and non-Gaussian noise in still 2D or 3D images or in 2D or 3D image sequences (without any motion computation) (see Figure 3) [4]. The method is unsupervised and is based on a pointwise selection of small image patches of fixed size (a data-driven adapted way) in spatial or space-time neighbourhood of each pixel (or voxel). The main idea is to modify each pixel (or voxel) using the weighted sum of intensities within an adaptive 2D or 3D (or 2D or 3D + time) neighbourhood and to use image patches to take into account complex spatial interactions. The neighbourhood size is selected at each spatial or space-time position according to a bias-variance criterion. The algorithm requires no tuning of control parameters (already calibrated with statistical arguments) and no library of image patches. The method has been applied to real noisy images (old photographs, JPEG-coded images, videos, ...) and is exploited in different biomedical application domains (time-lapse fluorescence microscopy, video-microscopy, MRI imagery, X-ray imagery, ultrasound imagery, ...).

The FAST-2D-SAFIR software (APP deposit number: IDDN.FR.001.190033.001.S.A.2007.000.21000) written in C++ removes mixed Gaussian-Poisson noise in large 2D images, typically $10^3 \times 10^3$ pixels, in a few seconds. The method is unsupervised and is a simplified version of the method related to the SAFIR-nD software. The software dedicated to microarrays image denoising, was licensed to the INNOPSYS company which develops scanners for disease diagnosis and multiple applications (gene expression, genotyping, aCGH, ChIP-chip, microRNA, ...).

On-line demo: Mobyle@SERPICO http://mobyle-serpico.rennes.inria.fr/cgi-bin/portal.py#forms::NDSafir

Free download binaries: Binaries of the software ND-SAFIR are freely and electronically distributed. Developed in standard C/C++ under Linux using the CImg library, it has been tested over several platforms such as Linux/Unix, Windows XP and Mac OS.

Academic licence agreements: Institut Curie, CNRS, ENS Ulm, Oxford University, Weizmann Institute, UCSF San-Francisco, Harvard University, Berkeley University, Stanford University, Princeton University, Georgia-Tech, Kyoto UNiversity, IMCB Singapore ...

Partners: J. Boulanger, J. Salamero (UMR 144, CNRS-Institut Curie), P. Elbau (RICAM Linz, Austria), J.B. Sibarita (UMR 5091, University of Bordeaux 2).

../../../../projets/serpico/IMG/DN-Rab6AGFP-WF.png

Figure 3. ND-SAFIR *software: denoising of a 3D image sequence in wide-field microscopy (GFP-Rab6A (Hela cell), UMR 144, CNRS-Institut Curie).*

HullkGround: Background subtraction by convex hull estimation

The HullkGround software (APP deposit number: IDDN.FR.001.400005.000.S.P.2009.000.21000) written in Java (plug-in ImageJ, see Fig. 4) decomposes a fluorescence microscopy image sequence into two dynamic components: i/ an image sequence showing mobile objects; ii/ an image sequence showing the slightly moving background. Each temporal signal of the sequence is processed individually and analyzed with computational geometry tools. The convex hull is estimated automatically for each pixel and subtracted to the original signal. The method is unsupervised, requires no parameter tuning and is a simplified version of the α shapes-based scale-space method [32].

On-line demo: Mobyle@SERPICO http://mobyle-serpico.rennes.inria.fr/cgi-bin/portal.py#forms::Hullkground
Partners: A. Chessel and J. Salamero (UMR 144, CNRS-Institut Curie)

`../../../../projets/serpico/IMG/hullkground_imagej.png`

Figure 4. HullkGround *software: plug-in* ImageJ.

5.2. Software for cryo-electron tomography

Participant: Charles Kervrann [(contact)].

TubuleJ: Straightening of microtubule cryo-EM projection views

The TubuleJ software (APP deposit number: IDDN.FR.001.240023.000.S.P.2011.000.21000) written in Java (plug-in ImageJ) is devoted to the analysis of microtubules and helical structures in 2D cryo-electron microscope images. The software straightens curved microtubule images by estimating automatically points locations on the microtubule axis. The estimation of microtubule principal axis relies on microtubule cylindrical shape analyzed in the Fourier domain. A user-friendly interface enables to filter straight fiber images by selecting manually the layer lines of interest in the Fourier domain. This software can be used to generate a set of 2D projection views from a single microtubule projection view and a few parameters of this microtubule structure. These projection views are then back projected, by using the IMOD plug-in (http://rsbweb.nih.gov/ij/), to reconstruct 3D microtubules.

On-line demo: see http://equipes.igdr.univ-rennes1.fr/en/tips/Software/TubuleJ/

Partners: S. Blestel and D. Chrétien (UMR 6290, CNRS, University of Rennes 1)

Cryo-Seg: Segmentation of tomograms in cryo-electron microscopy

The CRYO-SEG software written in C++ and JAVA (plug-in MAGEJ) has been developed to detect microtubule structures and helical structures in 2D cryo-electron microscope images. Cryo-electron tomography allows 3D observation of biological specimens in their hydrated state. Segmentation is formulated as Maximum A Posteriori estimation problem and exploits image patches to take into account spatial contexts (Markov Random Fields). Because of the contrast anisotropy in the specimen thickness direction, the whole tomogram is segmented section by section, with an automatic update of reference patches. This algorithm has been evaluated on synthetic data and on cryo-electron tomograms of in vitro microtubules. On real data, this segmentation method extracts the most contrasted regions of microtubules, and 3D visualization is improved.

Partners: S. Blestel and D. Chrétien (UMR 6290, CNRS-University of Rennes 1)

5.3. Image Processing software distribution

Participants: Tristan Lecorgne, Tinaherinantenaina Rakotoarivelo, Thierry Pécot [(contact)], Charles Kervrann.

The objective is to disseminate the distribution of SERPICO image processing software for biologist users:

- *Free binaries:* software packages have been compiled for the main operating systems (Linux, MacOS, Windows) using CMake (see http://www.cmake.org/). They are freely available on the team website under a proprietary license (e.g. ND-SAFIR and HULLKGROUND are distributed this way at http://serpico.rennes.inria.fr/doku.php?id=software:index).
- *Mobyle@SERPICO web portal*: An on-line version of the image processing algorithms has been developped using the Mobyle framework (Institut Pasteur, see http://mobyle.pasteur.fr/). The main role of this web portal (see Fig. 5) is to demonstrate the performance of the programs developed by the team: CRFMOVINGSPOTDETECTION (under review), HOTSPOTDETECTION [50], HULLKGROUND [32], KLTRACKER [48], MOTION2D[19], MS-DETECT [35], ND-SAFIR[4] and OPTICALFLOW. The web interface makes our image processing methods available for biologist users at Mobyle@SERPICO (http://mobyle-serpico.rennes.inria.fr/cgi-bin/portal.py#welcome) without any installation or configuration on their own. The size of submitted images is limited to 200 MegaBytes per user and all the results are kept 15 days. The web portal and calculations run on a server with 2 CPU x 8 cores, 64 GigaBytes of RAM.
- IMAGEJ *plug-ins*: IMAGEJ (see http://rsb.info.nih.gov/ij/) is a widely used image visualization and analysis software for biologist users. We have developed IMAGEJ plug-in JAVA versions of the following software: ND-SAFIR[4], HULLKGROUND [32] (see Fig. 4), MOTION2D[19], HOTSPOTDETECTION [50].
- *Institut Curie CID iManage database*: Institut Curie is currently acquiring a commercial database system (CID iManage / Strand Avadis company) to store mass of data. The database can be searched via meta-data and includes menu selections that enable to run remote processing. We have integrated ND-SAFIR in the interface environment to allow the database users to denoise images easily.

Partners: C. Deltel (Inria Rennes SED) and Perrine Paul-Gilloteaux (UMR 144, PICT IBiSA, CNRS-Institut Curie)

../../../../projets/serpico/IMG/MobyleatSERPICO.png

Figure 5. Mobyle@SERPICO web portal.

SHACRA Project-Team

5. Software and Platforms

5.1. SOFA

SOFA http://www.sofa-framework.org is an open-source software framework targeted at interactive computational (medical) simulation. The idea of SOFA was initiated by members of the SHACRA team, and strongly supported by Inria through a development program that we lead. SOFA facilitates collaborations between specialists from various domains, by decomposing complex simulators into components designed independently. Each component encapsulates one of the key aspects of a simulation, such as the degrees of freedom, the forces and constraints, the differential equations, the linear solvers, the collision detection algorithms or the interaction devices. The simulated objects can be represented using several models, each of them optimized for a different task such as the computation of internal forces, collision detection, haptics or visual display. These models are synchronized during the simulation using a mapping mechanism. CPU and GPU implementations can be transparently combined to exploit the computational power of modern hardware architectures. Thanks to this flexible yet efficient architecture, SOFA can be used as a test-bed to compare models and algorithms, or as a basis for the development of complex, high-performance simulators. As proof of its success, SOFA has been downloaded nearly 150,000 times, and is used today by many research groups around the world, as well as a number of companies. The mailing list used to exchange with the community includes several hundreds of researchers, from about 50 different institutions. SOFA is at the heart of a number of research projects, including cardiac electro-physiology modeling, interventional radiology planning and guidance, planning for cryosurgery and deep brain stimulation, robotics, percutaneous procedures, laparoscopic surgery, non-rigid registration, etc. SOFA is the only software developed by our team, but practically speaking it is a collection of plugins (each one aimed at a specific application) organized around a common core that provides a large number of functionalities. As mentioned previously, SOFA is currently used by a number of companies (Siemens Corporate Research, Digital Trainers, Epona Medical, Moog, SenseGraphics, etc.) and also provides the key technology on which our newly created start-up (InSimo) is relying. We strongly believe that today SOFA has become a reference for academic research, and is increasingly gaining recognition for product prototyping and development. The best illustration of this worldwide positioning is the role of SOFA in the challenge set by the HelpMeSee foundation to win the contract for the development of a very ambitious and high-risk project on cataract surgery simulation.

We also gave a 4 hours workshop on SOFA at MMVR/NextMed conference in february 2013 in San Diego. This workshop was done in collaboration with the swedish company SenseGraphics. The topic was to demonstrate the setup of a dental surgery simulation in Sofa, and use SenseGraphics visual tools for the rendering. The attendees feedback was beyond our expectations, with an unexpected interest in new SOFA features like the SofaPython plugin. Still about SOFA, like last year we gave in october a 3 days training session in Montpellier for about twenty SOFA beginners (mostly engineers). These are new engineers of the three teams involved in SOFA development, and employees of companies using SOFA in their business. Last, a "SOFA Day" in november in prelude of the Vriphys conference gave us a unique opportunity to meet SOFA users from various research institutes or companies, and exchange about the future improvements and development of the engine. We use these occasions to share and discuss with SOFA users, to refine the roadmap and stay tuned with our audience.

SISYPHE Project-Team

5. Software and Platforms

5.1. SITB: The Matlab System Identification ToolBox

Participant: Qinghua Zhang.

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

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

5.2. ISTL: Inverse Scattering for Transmission Lines

Participants: Michel Sorine, Qinghua Zhang.

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

5.3. CGAO: Contrôle Glycémique Assisté par Ordinateur

Participants: Alexandre Guerrini, Michel Sorine.

The software CGAO developed with LK2 and P. Kalfon (Hospital Louis Pasteur, Chartres) provides efficient monitoring and control tools that will help physicians and nursing staff to avoid hyperglycaemia and hypoglycaemia episodes in Intensive Care Units. The controller determines the insulin infusion rate, glucose bolus and scheduling of blood glucose measurement on the basis of the standard available glycaemia measurements. A first version, CGAO_v1, has been used in a large clinical study CGAO-REA (see Section 6.3.1). An improved version, CGAO_v2 registered at APP under the number IDDN.FR.001.360019.002.S.P.2009.000.31230 is now used by the company Fresenius Kabi (see Section 7.1).

5.4. DYNPEAK: a Scilab toolbox and a Web service for the analysis of LH (Luteinizing Hormone) secretion rhythms

Participants: Frédérique Clément, Claire Médigue, Serge Steer, Mouhamadoul Bachir Syll, Alexandre Vidal, Qinghua Zhang.

DYNPEAK is a software dedicated to the analysis of the pulsatile rhythm of secretion of the pituitary hormone LH, that aims at providing the final users (experimentalists and clinicians) with a simple-to-use version of the algorithm developed in [25]. It has been implemented as a Scilab atom toolbox (http://atoms.scilab.org/toolboxes/Dynpeak) and registered in APP under the reference DynPeak V1.0, IDDN.FR.001.360015.000.S.P.2013.000.10000. The web service version of DynPeak (https://dynpeak.inria.fr), still in test, has also been updated and a new release is planned for a next future.

5.5. The Cardiovascular toolbox for Scilab

Participants: Claire Médigue, Michel Sorine, Serge Steer.

This Cardiovascular toolbox is an “atom” of Scilab developed by Serge Steer to distribute the cardiovascular signal processing tools designed and intensively used internally in the team for several years by Claire Médigue, Alessandro Monti and Michel Sorine. It includes baroreflex analysis using a multi channel non stationary signal analysis method ; the cardiovascular signal spectral analysis using time-frequency decomposition and signal demodulations methods, e.g. for respiratory sinus arrhythmia analysis. It replaces LARY_CR, the former software package dedicated to the study of cardiovascular and respiratory rhythms [108].

5.6. K-Assessor: assessment of controllers

Participants: Habib Jreige, Michel Sorine.

This development is made in collaboration with the small business enterprise SciWorks Technologies (Jim Pioche). We have defined a method to assess SISO (Single Input / Single Output) controllers based on square or cubic tables of metadata easily manipulated on a computer and easily interpretable by control experts and field experts (emergency doctors in our case) who can use them to jointly tune a risk estimator. The agreement between experts is obtained using a ROC-analysis approach. The software K-Assessor implements this methodology. It is registered at APP under the number IDDN.FR.001.390011.000.S.P.2013.000.10000.

STEEP Team

5. Software and Platforms

5.1. TEOS: Tranus Exploration and Optimization Software

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

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

5.2. REDEM: REDuction Of GHG EMission software

Participant: Emmanuel Prados.

REDEM software (REDuction of EMissions) is a tool designed for the benchmarking of national GHG emission reduction trajectories. We have developed REDEM in collaboration with EDDEN Laboratory (Patrick Criqui and Constantin Ilasca). The actual version of the software is implemented in Visual Basic under Microsoft Excel in order to facilitate handling and diffusion to climate/energy economists. We envisage to distribute this software as an open source software.

5.3. Wassily

Participants: Julien Alapetite, Jean-Yves Courtonne, Lara Antonela Colombo, Pablo Virgolini.

In collaboration with the association “Groupe de Réflexion sur les Empreintes Ecologiques Locales” (eco-data.fr), STEEP contributes to the development of Wassily (in tribute to Wassily Leontief who first designed the relevant concepts), to perform input-output analyses applied to environmental issues (see section 4.2). The purpose of this software is to automatize most of the work of standard input-output analysis and to visualize the results in a user-friendly way in order to efficiently address the related key environmental questions.

The software is structured in three different modules:

- the database module stores all the input-output data coming from Eurostat, OCDE, Insee or other sources.
- the computation module performs the input-output calculations
- the visualization module displays the results in a synthetic manner.

The database module is based on the SQlite format and makes use of SQL to manipulate the various tables involved in the process. The goal of this module is to provide a normalized data interface for the computation module, from various types of input-output data which are often stored as Excel sheet on web sites.
The computation module is based on QT and C++ and deals mostly with matrix manipulation.
The visualization module is based on a JavaScript library called D3 and allows the user to visualize the results in a number of different ways, such as bar charts, pie charts, sankey diagrams to name a few. The integration between the C++ and JavaScript pieces of code is performed with QTScript.

VIRTUAL PLANTS Project-Team

4. Software and Platforms

4.1. V-Plants

Participants: Frédéric Boudon, Christophe Godin [coordinator], Yann Guédon, Christophe Pradal [software architect], Jean-Baptiste Durand, Pascal Ferraro, Julien Coste, Guillaume Baty.

Computer algorithms and tools developed by the Virtual Plants team are integrated in a common software suite *V-Plants*, dedicated to the modeling and analysis of plant development at different scales (e.g. cellular tissue, whole plant, stand). The VPlants packages are integrated in OpenAlea as Python components. Several components are distributed and usable through the visual programming environment (see figure 1):

- Multi-scale geometric modeling and visualization. VPlants.PlantGL is a geometric library which provides a set of graphical tools and algorithms for 3D plant modeling at different scales [7]. It is used by many other components to represent the geometry of biological shapes from 3D meristems, plant architectures to plant populations. VPlants.PlantGL is built around a scene-graph data structure and provides efficient algorithms and original geometrical shapes (parametric surfaces, dedicated envelops), that are useful for plant modeling.
- Statistical sequence and tree analysis. Different statistical packages (i.e. VPlants.StatTool, VPlants.SequenceAnalysis, VPlants.TreeMatching and VPlants.TreeAnalysis) are now available in OpenAlea. They provide different models and algorithms for plant architecture analysis and simulation.
- Meristem functioning and development. A first set of components has been created in the last 4-years period to model meristem development in OpenAlea. These tools are currently being integrated thoroughly in the platform so that modelers and biologists can use them, and reuse components easily (for meristem 3D reconstruction, cell tracking, statistical analysis of tissues, creating and manipulating atlases, creating or loading models of growth that can further be run on digitized structures, etc).
- Standard data structure for plants. A new implementation of the MTG formalism for representing and manipulating multiscale plant architecture has been developed. It provides a central data-structure to represent plants in a generic way in OpenAlea. This implementation is available through the packages OpenAlea.MTG. These components make it possible to share plant representations between users and fosters the interoperability of new models.
- Simulation system. The study of plant development requires increasingly powerful modeling tools to help understand and simulate the growth and functioning of plants. In the last decade, the formalism of L-systems has emerged as a major paradigm for modeling plant development. Previous implementations of this formalism were made based on static languages, i.e. languages that require explicit definition of variable types before using them. These languages are often efficient but involve quite a lot of syntactic overhead, thus restricting the flexibility of use for modelers. We developed L-Py [2] an adaptation of L-systems to the Python language (basis of OpenAlea). Thanks to its dynamic typing property, syntax is simple, code execution is made easy and introspection property of the language makes it possible to parameterize and manipulate simply complex models. Independent L-systems can be composed to build-up more complex modular models. MTG structures (that are a common way to represent plants at several scales) can be translated back and forth into L-system data-structure and thus make it easy to reuse in L-systems tools for the analysis of plant architecture based on MTGs. Extensions to integrate multiscale dynamic models are currently being developed in collaboration with P. Prusinkiewicz and his team from the University of Calgary.

../../../../projets/virtual_plants/IMG/alea.png

Figure 1. V-Plants components of the OpenAlea platform: simulating plant processes at different scales. Top Left: Reconstruction of a virtual meristem, analysis and simulation of the auxin fluxes inside the meristem. Top Right: Reconstruction of a virtual apple tree from digitized data. Bottom: Simulation of an ecosystem (A beech "Fagus Sylvatica L." trees forest) with a multi level approaches. On the left, explicit representation of the crown volumes that serves as input to generate the detailed representation, on the right.

4.2. OpenAlea

Participants: Frédéric Boudon, Christophe Godin, Yann Guédon, Christophe Pradal [coordinator], Christian Fournier, Julien Coste.

This research theme is supported by the Inria ADT Grant OpenAlea 2.0 and by an Agropolis RTRA Grant named OpenAlea.

OpenAlea[8] is an open source and collaborative software project primarily dedicated to the plant research community. It is designed as a component framework to dynamically glue together models from different plant research labs, and to enhance re-usability of existing models in the plant research community.

The architecture of OpenAlea is based on a component architecture. It provides a set of standard components (OpenAlea.Stdlib), a package manager to dynamically add and retrieve new components, and a port graph data-structure to compose models by interconnecting components into a data-flow.

Visualea provides a visual programming environment, used by scientists to build new model interactively by connecting available components together through an easy-to-use graphical user interface.

In 2013, the following progresses were accomplished:

1. Development and extension of OpenAlea and Visualea:
   - The standard library of components has been extended with useful scientific packages such as a flexible data plotting package (Openalea.Pylab), 2D and 3D image manipulation (Openalea.Image) and linear algebra operations (Openalea.Numpy).
   - Several models of computation have been implemented on the data-flow data-structure to enable discrete event simulation and control flow inside OpenAlea.
2. Animation and diffusion
   - A scientific board has been defined to manage the development and diffusion of OpenAlea. It is composed by 12 scientists.
   - StandAlone binary installers have been released on Windows and Mac to ease the installation of a large number of packages without relying on a web server. A Ubuntu repository has been set up on Launchpad.
   - The OpenAlea project is hosted at the Inria gforge (link http://openalea.gforge.inria.fr). The web site is visited by more than 370 unique visitor each month; 650000 web pages have been visited and the different available components of OpenAlea have been downloaded more than 540,000 times during the last two years. OpenAlea is the first project at Inria Gforge in term of number of downloads and of page views.

4.3. Alinea

Participants: Christian Fournier, Christophe Pradal, Frédéric Boudon.

Other participants : Bruno Andrieu, Michael Chelle, Gaëtan Louarn, Benoit de Solan, Mariem Abichou, Liqi Han, Elmer Ccopa-Rivera, Frédéric Baret, Rafaële Casa, Guillaume Garin, Corinne Robert, Sébastien Saint-Jean, Didier Combes, Camille Chambon, Romain Barillot, Jean-Christophe Soulie, Delphine Luquet.

The Alinea software suite is distributed as a meta-package of the OpenAlea Platform. It is produced by a consortium of modelers from INRA , and consists of various ecophysiological and biophysical models of simulation (radiative transfer, interaction between plant and pest, circulation of hydric fluxes, and dispersion). The project is suported by 5 INRA teams and the Inria Virtual Plants project. This project has been homologated as strategic ressource for INRA, and is integrated in the CATI IUMA (Centre Automatisé de Traitement de l'Information, : Informatisation et Utilisation des Modèles dédiés aux Agro-Ecosystemes). The five following components are distributed with the OpenAlea platform:

- Alinea.Adel is a module to simulate the 3D architectural development of gramineous crops. In 2013, a new parameterisation procedure was developed for wheat, that allows to use the model for the simulation of agronomic experiments with a minimal set of measurements [28].

- Alinea.Alep is a generic model developed in 2013 to simulate pathosystems from the scale of leaf to the scale of the canopy
- Alinea.Caribu is a modeling suite for lighting 3D virtual scenes, especially designed for the illumination of virtual plant canopies such as virtual crop fields. It uses a special algorithm, the nested radiosity, that allows for a precise estimation of light absorption at the level of small canopy elements.
- Alinea.pyRATP allows to simulate the light interception, photosynthesis and transpiration of a plant canopy.
- Alinea.TopVine is a component to reconstruct grapevine canopy structure. Other components are in developmental state and will be released after publication by their autors.

VISAGES Project-Team

5. Software and Platforms

5.1. CLARCS: C++ Library for Automated Registration and Comparison of Surfaces

Participants: Juan Francisco Garamendi Bragado, Sylvain Prima.

In collaboration with Benoit Combès (Géosciences Rennes, UMR 6118) and Alexandre Abadie (Inria Saclay Île-de-France), within the 3D-MORPHINE ARC project (http://3dmorphine.inria.fr), we conceived and implemented a C++ library (named CLARCS) for the automated analysis and comparison of surfaces. One of the primary goal of this library is to allow the assessment and quantification of morphological differences of free-form surfaces from medical or paleoanthropological data.

- APP: IDDN.FR.001.130002.000.S.P.2011.000.21000
- Programming language: CC++

CLARCS was presented at the MeshMed MICCAI workshop (http://www2.imm.dtu.dk/projects/MeshMed/2011/index.html) [57] and is to be distributed through a dedicated website (http://clarcs.inria.fr).

We also developed a surface viewer (named 'Surface').

- APP: IDDN.FR.001.110019.000.S.P.2011.000.21000
- Programming language: C++, Python

5.2. Shanoir

Participants: Justine Guillaumont, Michael Kain, Christian Barillot.

Shanoir (Sharing NeurOImaging Resources) is an open source neuroinformatics platform designed to share, archive, search and visualize neuroimaging data. It provides a user-friendly secure web access and offers an intuitive workflow to facilitate the collecting and retrieving of neuroimaging data from multiple sources and a wizzard to make the completion of metadata easy. Shanoir comes along many features such as anonymization of data, support for multi-centres clinical studies on subjects or group of subjects. For a better distribution/replication of stored data on a Shanoir server an export and import function on base of XML has been developed for the usage of server administrators (Figure 2).

Shanoir APP registration number is: IDDN.FR.001.520021.000.S.P.2008.000.31230

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

- Keywords: neuroimaging, ontology, sharing neuroimage
- Software benefit: full featured neuroimaging management system with additionnal web services
- APP: IDDN.FR.001.520021.000.S.P.2008.000.31230
- License: Licence QPL
- Type of human computer interaction: Online web application, web service (SOAP messages based)
- OS/Middelware: Windows, Mac et Linux.
- Required library or software: Java 1.6, JBoss server, JBoss Seam, JSF, JPA Hibernate, EJB, Richfaces, Faceless, Ajax4JSF, Dcmtk, Dcm4chee.
- Programming language: Java
- Documentation: see the website

`../../../../projets/visages/IMG/shanoir1.png../../projets/visages/IMG/shanoir2.p`

Figure 2. The SHANOIR software is a web application to share, archive, search and visualize neuroimaging data.

5.3. ShanoirUploader

Participants: Justine Guillaumont, Michael Kain.

The ShanoirUploader is a desktop application on base of JavaWebStart (JWS). The app can be downloaded and installed using an internet browser. The app interacts with a PACS to query and retrieve the data stored on any PACS. After this the ShanoirUploader sends the data to a Shanoir server instance to import these data into a Shanoir server instance. This app bypasses the situation, that in most of the clinical network infrastructures a server to server connection is complicated to set up between the PACS and a Shanoir server instance.

An APP registration is in progress.

See also the web page http://www.shanoir.org as the ShanoirUploader documentation is integrated on this page.

- Keywords: neuroimaging, ontology, sharing neuroimage
- Software benefit: offers a great solution to query a PACS server, download the data and send the data to a Shanoir server
- License: no defined licence for the moment
- Type of human computer interaction: desktop application on base of JavaWebStart (JWS), web service (SOAP messages based)
- OS/Middelware: Linux, Windows and Mac
- Required library or software : Java SDK, installed on client machine
- Programming language: Java
- Documentation : see the website

5.4. AutoMRI

Participants: Camille Maumet, Isabelle Corouge, Pierre Maurel, Fang Cao, Elise Bannier.

AutoMRI Based on MATLAB and the SPM8 toolbox, autoMRI provides complete pipelines to pre-process and analyze various types of images (anatomical, funcional, perfusion, metabolic, relaxometry, vascular). This software is highly configurable in order to fit to a wide range of needs. Pre-processing includes segmentation of anatomical data, as well as co-registration, spatial normalisation and atlas building of all data types. The analysis pipelines perform either within-group analysis or between-group or one subject-versus-group comparison and produce statistical maps of regions with significant differences. These pipelines can be applied to structural data to exhibit patterns of atrophy or lesions, to ASL or PET data to detect perfusion or metabolic abnormalities, to relaxometry data to detect deviations from a template, to functional data - either BOLD or ASL - to outline brain activations related to block or event-related paradigms. In addition to the standard General Linear Model approach, the ASL pipelines implement an a contrario approach and, for patient-specific perfusion study, an heteroscedastic variance model. Besides, the vascular pipeline processes 4D MRA data and enables accurate assessment of hemodynamic patterns (Figure 3).

- Keywords: fMRI, MRI, ASL, fASL, SPM, automation
- Software benefit: Automatic MRI data analysis based on SPM. Once the parameters are set, the analysis is performed without human interaction.
- APP: Part in IDDN.FR.001.130017.000.S.A.2012.000.31230
- License: Part under CeCILL
- Type of human computer interaction: Matlab function (script, no GUI)
- OS/Middleware: Windows, OS X, Linux
- Required library or software: Matlab, SPM, SPM toolboxes : Marsbar, LI-toolbox, NS
- Programming language: Matlab
- Documentation: available at https://gforge.inria.fr/projects/autofmri/ and https://gforge.inria.fr/projects/asl/

`../../../../projets/visages/IMG/automri1.png`

`../../projets/visages/IMG/automri2.p`

Figure 3. Illustrations of results obtained with autoMRI: Conjunction map showing areas of hypoperfusion and hypometabolism in semantic dementia (right), Detection of relaxometry defect in an MS patient (left).

5.5. Medinria

Participants: René-Paul Debroize, Guillaume Pasquier, Laurence Catanese, Olivier Commowick.

Medinria is a national Inria project shared between 4 Inria teams (Asclepios, Athena, Parietal and Visages). It aims at creating an easily extensible platform for the distribution of research algorithms developed at Inria for medical image processing. This project has been funded by the D2T (ADT MedInria-NT) in 2010 and renewed for two years in 2012. The Visages team leads this Inria national project and participates in the development of the common core architecture and features of the software as well as in the development of specific plugins for the team's algorithm. Medinria 2.1.2 has been released in September 2013 for the main distribution platforms. medInria core API source code has also been released under a BSD license.

See also Figure 4 and the web page http://med.inria.fr

- Keywords: medical imaging, diffusion imaging, registration, filtering, user-friendly interface
- Software benefit: user-friendly interface to cutting-edge research tools for research clinicians. Straightforward to add functionalities through plugins.
- License: core: BSD, plugins: choice of each team.
- Type of human computer interaction: Qt-based GUI
- OS/Middelware: Windows, Mac et Linux.
- Required library or software : Qt, DTK, ITK, VTK.
- Programming language: C++

`../../../../projets/visages/IMG/fibers.png/../../projets/visages/IMG/registrati`

Figure 4. The medInria software platform : Side by side registration using fast algorithms Optimus (right), Tractography overlapped with 3D image (left)

5.6. Anima

Participants: René-Paul Debroize, Guillaume Pasquier, Aymeric Stamm, Fang Cao, Olivier Commowick.

Anima is a set of libraries and tools developed by the team as a common repository of research algorithms. As of now, it contains tools for image registration, statistical analysis (group comparison, patient to group comparison), diffusion imaging (model estimation, tractography, etc.), quantitative MRI processing (quantitative relaxation times estimation, MR simulation), image denoising and filtering, and segmentation tools. All of these tools are based on stable libraries (ITK, VTK), making it simpler to maintain.

- Keywords: medical imaging, diffusion imaging, registration, filtering, relaxometry
- Software benefit: New methodological image processing, common place for team code
- Type of human computer interaction: C++ API
- OS/Middleware: Windows, Mac and Linux.
- Required library or software : ITK, VTK.
- Programming language: C++

ACES Project-Team

4. Software and Platforms

4.1. Software and Platforms

4.1.1. THE GAME: THeory of Evidence in a lanGuage Adapted for Many Embedded systems

Context-aware applications have to sense the environment in order to adapt themselves and provide with contextual services. This is the case of Smart Homes equipped with sensors and augmented appliances. However, sensors can be numerous, heterogeneous and unreliable. Thus the data fusion is complex and requires a solid theory to handle those problems. The aim of the data fusion, in our case, is to compute small pieces of context we call context attributes. Those context attributes are diverse and could be for example the presence in a room, the number of people in a room or even that someone may be sleeping in a room. For this purpose, we developed an implementation of the belief functions theory (BFT). THE GAME (THeory of Evidence in a lanGuage Adapted for Many Embedded systems) is made of a set of C-Libraries. It provides the basics of belief functions theory, computations are optimized for an embedded environment (binary representation of sets, conditional compilation and diverse algorithmic optimizations).

THE GAME has been developed within the ACES-EDF collaboration (see 6.1.1), and is published under apache licence (https://github.com/bpietropaoli/THEGAME/). It is maintained and experimented by Aurélien Richez within a sensor network platform developed by ACES since June 2013.

ADAM Project-Team

5. Software and Platforms

5.1. APISENSE

Participants: María Gómez Lacruz, Nicolas Haderer, Christophe Ribeiro, Romain Rouvoy [correspondant].

APISENSE® is a distributed platform dedicated to crowdsensing activities [30], [31], [24], [77], [67], [66]. Crowdsensing intends to leverage mobile devices to seamlessly collect valuable dataset for different categories of stakeholders. APISENSE® intends to be used in a wide variety of scientific and industrial domains, including network quality monitoring, social behavior analysis, epidemy predictions, emergency crisis support, open maps initiatives, wild applications debugging. APISENSE® is composed of a HIVE and an HONEYCOMB delivered as a *Platform-as-a-Service* (PaaS) to the stakeholders who can pilot and customize their own crowdsensing environment [77], and *Bee.mob* supporting participants with a mobile application to control the sensors to be shared with the rest of the world [30], [31]. The platform is used by the Metroscope project, an Internet scientific observatory initiative supported by Inria.

Web site: http://www.apisense.fr. Registered with the APP (*Agence pour la Protection des Programmes*) under reference IDDN.FR.001.080006.000.S.P.2013.000.10000 is pending. License: Proprietary.

5.2. FraSCAti

Participants: Gwenaël Cattez, Philippe Merle [correspondant], Fawaz Paraïso, Romain Rouvoy, Lionel Seinturier.

FRASCATI is a service-oriented component-based middleware platform implementing OASIS *Service Component Architecture* (SCA) specifications.The main originality of OW2 FRASCATI is to bring FRACTAL-based reflectivity to SCA, *i.e.*, any FRASCATI software component is equipped with both the SOA capabilities brought by SCA and the reflective capabilities (*i.e.*, introspection and reconfiguration) brought by FRACTAL. Various micro-benchmarks have shown that FRASCATI reflectivity is achieved without hindering its performance relative to the de facto reference SCA implementation, *i.e.*, Apache Tuscany. Non-functional concerns (logging, transaction, security, etc.), so called intents in SCA terms, are also programmed as FRASCATI components and are (un)woven on business components dynamically at runtime, this is based on aspect-oriented concepts defined in FAC [78]. OW2 FRASCATI supports various implementation technologies (SCA Composite, Java, WS-BPEL, Spring Framework, OSGi, Fractal ADL, native C library, Apache Velocity templates, and seven scripting languages as BeanShell, FScript, Groovy, JavaScript, JRuby, Jython, XQuery) for programming services or integrating legacy code, various binding protocols (SOAP, REST, JSON-RPC, UPnP, HTTP servlets, Java RMI, JMS, JGroups) and interface definition languages (WSDL, Java, WADL) for interoperating with existing services. OW2 FRASCATI provides management tools like standalone, Web-based, and JMX-based graphical consoles and a dedicated scripting language for reconfiguring SCA applications. The whole OW2 FRASCATI platform is itself built as a set of reflective SCA components.

Inria Evaluation Committee Criteria for Software Self-Assessment: A-4-up, SO-4, SM-4-up, EM-3-up, SDL-4-up, DA-4, CD-4, MS-4, TPM-4. FRASCATI is a project of the OW2 consortium for open-source middleware. Web site: http://frascati.ow2.org. 292 Kloc (mainly Java). Registered with the APP (Agence pour la Protection des Programmes) under reference FR.001.050017.000.S.P.2010.000.10000. License: LGPL. Embedded into several industrial software systems: EasySOA, Petals Link EasyViper, EasyBPEL, EasyESB, OW2 PEtALS, OW2 Scarbo. Various demonstrators built during funded projects: ANR SCOrWare, FP7 SOA4All, ANR ITEmIS, ANR SALTY, ANR SocEDA, FUI Macchiato, FUI EasySOA, ADT Galaxy and ADT Adapt. Main publications: [82], [81], [70], [71], [62], [61].

5.3. PowerAPI

Participants: Aurélien Bourdon, Maxime Colmant, Lo√øc Huertas, Adel Noureddine, Romain Rouvoy [correspondant].

PowerAPI is a Scala-based library for monitoring energy at the process-level. It is based on a modular and asynchronous event-driven architecture using the Akka library. PowerAPI differs from existing energy process-level monitoring tool in its pure software, fully customizable and modular aspect which let users precisely define what they want to monitor, without plugging any external device. PowerAPI offers an API which can be used to express requests about energy spent by a process, following its hardware resource utilization (in terms of CPU, memory, disk, network, etc.). Its applications cover energy-driven benchmarking [74], [50], [49], [23], energy hotspots and bugs detection [75], [76] and real-time distributed system monitoring.

Web site: http://www.powerapi.org. Registered with the APP (Agence pour la Protection des Programmes) under reference IDDN.FR.001.400015.000.S.P.2012.000.10000. License: AGPL.

ALGORILLE Project-Team

5. Software and Platforms

5.1. Introduction

Software is a central part of our output. In the following we present the main tools to which we contribute. We use the Inria software self-assessment catalog for a classification.

5.2. Implementing parallel models

Several software platforms have served us to implement and promote our ideas in the domain of coarse grained computation and application structuring.

5.2.1. ORWL and P99

Participants: Jens Gustedt, Rodrigo Campos-Catelin, Stéphane Vialle, Mariem Saied.

ORWL is a reference implementation of the Ordered Read-Write Lock tools as described in [4]. The macro definitions and tools for programming in C99 that have been implemented for ORWL have been separated out into a toolbox called P99. ORWL is intended to become opensource, once it will be in a publishable state. P99 is available under a QPL at http://p99.gforge.inria.fr/.
Software classification: A-3-up, SO-4, SM-3, EM-3, SDL (P99: 4, ORWL: 2-up), DA-4, CD-4, MS-3, TPM-4

5.2.2. parXXL

Participants: Jens Gustedt, Stéphane Vialle.

ParXXL is a library for large scale computation and communication that executes fine grained algorithms on coarse grained architectures (clusters, grids, mainframes). It has been one of the software bases of the InterCell project and has been proven to be a stable support, there. It is available under a GPLv2 at http://parxxl.gforge.inria.fr/. ParXXL is not under active development anymore, but still maintained in the case of bugs or portability problems.
Software classification: A-3, SO-4, SM-3, EM-2, SDL-4, DA-4, CD-4, MS-2, TPM-2

5.3. Distem

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

Wrekavoc and Distem are distributed system emulators. They enable researchers to evaluate unmodified distributed applications on heterogeneous distributed platforms created from an homogeneous cluster: CPU performance and network characteristics are altered by the emulator.
Wrekavoc was developed until 2010, and we then focused our efforts on **Distem**, that shares the same goals with a different design. Distem is available from http://distem.gforge.inria.fr/ under GPLv3.
Software classification: A-3-up, SO-4, SM-3-up, EM-3, SDL-4, DA-4, CD-4, MS-4, TPM-4.

5.4. SimGrid

SimGrid is a toolkit for the simulation of distributed applications in heterogeneous distributed environments. The specific goal of the project is to facilitate research in the area of parallel and distributed large scale systems, such as Grids, P2P systems and clouds. Its use cases encompass heuristic evaluation, application prototyping or even real application development and tuning.

5.4.1. Core distribution

Participants: Martin Quinson, Marion Guthmuller, Paul Bédaride, Gabriel Corona, Lucas Nussbaum.

SimGrid has an active user community of more than one hundred members, and is available under GPLv3 from http://simgrid.gforge.inria.fr/. One third of the source code is devoted to about 12000 unit tests and 500 full integration tests. These tests are run for each commit for 4 package configurations and on 4 operating systems thanks to the Inria continuous integration platform.
Software classification: A-4-up, SO-4, SM-4, EM-4, SDL-5, DA-4, CD-4, MS-3, TPM-4.

5.4.2. SimGridMC

Participants: Martin Quinson, Marion Guthmuller, Gabriel Corona.

SimGridMC is a module of SimGrid that can be used to formally assess any distributed system that can be simulated within SimGrid. It explores all possible message interleavings searching for states violating the provided properties. We recently added the ability to assess liveness properties over arbitrary C codes, thanks to a system-level introspection tool that provides a finely detailed view of the running application to the model checker. This can for example be leveraged to verify arbitrary MPI code written in C.
Software classification: A-3-up, SO-4, SM-3-up, EM-3-up, SDL-5, DA-4, CD-4, MS-4, TPM-4.

5.4.3. SCHIaaS

Participants: Julien Gossa, Stéphane Genaud, Luke Bertot, Rajni Aron, Étienne Michon.

The *Simulation of Clouds, Hypervisor and IaaS* (SCHIaaS) is an extension of SimGrid that can be used to comprehensively simulate clouds, from the hypervisor/system level, to the IaaS/administrator level. The hypervisor level includes models about virtualization overhead and VMs operations like boot, start, suspend, migrate, and network capping. The IaaS level includes models about instances management like image storage and deployment and VM scheduling. This extension allows to fully simulate any cloud infrastructure, whatever the hypervisor or the IaaS manager. This can be used by both cloud administrators to dimension and tune clouds, and cloud users to simulate cloud applications and assess provisioning strategies in term of performances and cost.
Software classification: A-4-up, SO-4, SM-2-up, EM-2-up, SDL-2, DA-4, CD-4, MS-4, TPM-4.

5.5. Kadeploy

Participants: Luc Sarzyniec, Emmanuel Jeanvoine, Lucas Nussbaum.

Kadeploy is a scalable, efficient and reliable deployment (provisioning) system for clusters and grids. It provides a set of tools for cloning, configuring (post installation) and managing cluster nodes. It can deploy a 300-nodes cluster in a few minutes, without intervention from the system administrator. It plays a key role on the Grid'5000 testbed, where it allows users to reconfigure the software environment on the nodes, and is also used on a dozen of production clusters both inside and outside INRIA. It is available from http://kadeploy3.gforge.inria.fr/ under the Cecill license.
Software classification: A-4-up, SO-3, SM-4, EM-4, SDL-4-up, DA-4, CD-4, MS-4, TPM-4.

5.6. XPFlow

Participants: Tomasz Buchert, Lucas Nussbaum.

XPFlow is an implementation of a new, workflow-inspired approach to control of experiments involving large-scale computer installations. Such systems pose many difficult problems to researchers due to their complexity, their numerous constituents and scalability problems. The main idea of the approach consists in describing the experiment as a workflow and execute it using achievements of Business Workflow Management (BPM), workflow management techniques and scientific workflows. The website of XPFlow is http://xpflow.gforge.inria.fr/. The software itself is not released to the general public yet, a fact that will be addressed during the year 2014.
Software classification: A-2-up, SO-3-up, SM-2-up, EM-3-up, SDL-2-up, DA-4, CD-4, MS-4, TPM-4.

5.7. Grid’5000 testbed

Participants: Luc Sarzyniec, Emmanuel Jeanvoine, Émile Morel, Lucas Nussbaum.

Grid’5000 (http://www.grid5000.fr) is a scientific instrument designed to support experiment-driven research in all areas of computer science related to parallel, large-scale or distributed computing and networking. It gathers 10 sites, 25 clusters, 1200 nodes, for a total of 8000 cores. It provides its users with a fully reconfigurable environment (bare metal OS deployment with Kadeploy, network isolation with KaVLAN) and a strong focus on enabling high-quality, reproducible experiments.

The AlGorille team contributes to the design of Grid’5000, to the administration of the local Grid’5000 site in Nancy, and to the design and development of Kadeploy (in close cooperation with the Grid’5000 technical team). The AlGorille engineers also administer *Inria Nancy – Grand Est*’s local production cluster, named *Talc*, leveraging the experience and tools from Grid’5000.

Software classification: A-5, SO-4, SM-4, EM-4, SDL-N/A, DA-4, CD-4, MS-4, TPM-4.

ALPINES Team

5. Software and Platforms

5.1. Software and Platforms

5.1.1. Platforms

5.1.1.1. FreeFem++, http://www.freefem.org/ff++/

`FreeFem++` is a PDE solver based on a flexible language that allows a large number of problems to be expressed (elasticity, fluids, etc) with different finite element approximations on different meshes. There are more than 2000 users, and on the mailing list there are 430 members. Among those, we are aware of at least 10 industrial companies, 8 french companies and 2 non-french companies. It is used for teaching at Ecole Polytechnique, Ecole Centrale, Ecole des Ponts, Ecole des Mines, University Paris 11, University Paris Dauphine, La Rochelle, Nancy, Metz, Lyon, etc. Outside France, it is used for example at universities in Japan (Tokyo, Kyoto, Hiroshima, there is a userguide FreeFem++ in japan), Spain (Sevilla, BCAM, userguide available in spanish), UK (Oxford), Slovenia, Switzerland (EPFL, ETH), China. For every new version, there are 350 regression tests, and we provide a rapid correction of reported bugs. The licence of FreeFem++ is LGPL.

5.1.1.2. Library for preconditioned iterative methods

In the project-team we develop a library that integrates the direction preserving and low rank approximation preconditioners for both approached factorizations and domain decomposition like methods. It will be available through `FreeFem++` and also as a stand alone library, and we expect to have one version of this library available in 2014.

ARLES Project-Team

5. Software and Platforms

5.1. Introduction

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

5.2. iCONNECT – Emergent Middleware Enablers

Participant: Valérie Issarny [correspondent].

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

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

The Synthesis Enabler assumes semantically-annotated system descriptions *à la* OWL-S, which are made available by the Discovery Enabler, together with a domain ontology, and produces the mediators that enable functionally compatible networked systems to interoperate. The semantically-annotated interfaces of the NSs that need to communicate are processed to compute the semantic mapping between their respective operations using a constraint solver. The resulting mapping serves generating a mediator that coordinates the behaviors of the NSs and guarantees their successful interaction. Only when the mediator includes all the details about the communication of NSs, can interoperability be achieved, which calls for the adequate concretization of synthesized mediators.

The *concretization of mediators* bridges the gap between the application level, which provides the abstraction necessary to reason about interoperability and synthesize mediators, and the middleware-level, which provides the techniques necessary to implement these mediators. Concretization entails the instantiation of the data structures expected by each NS and their delivery according to the interaction pattern defined by the middleware, based on which the NS is implemented. Therefore, we have been developing a mediation engine that, besides executing the data translations specified by the mediator, generates composed parsers and composers, which can process complex messages, by relying on existing libraries associated with standard protocols and state-of-the-art middleware solutions.

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

5.3. Service-oriented Middleware for Pervasive Computing

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

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

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

5.4. XSB – eXtensible Service Bus for the Future Internet

Participant: Nikolaos Georgantas [correspondent].

The eXtensible Service Bus (XSB) is a development and runtime environment dedicated to complex distributed applications of the Future Internet. Such applications will be based, to a large extent, on the open integration of extremely heterogeneous systems, such as lightweight embedded systems (e.g., sensors, actuators and networks of them), mobile systems (e.g., smartphone applications), and resource-rich IT systems (e.g., systems hosted on enterprise servers and Cloud infrastructures). Such heterogeneous systems are supported by enabling middleware platforms, particularly for their interaction. With regard to middleware-supported interaction, the client-service (CS), publish-subscribe (PS), and tuple space (TS) paradigms are among the most widely employed ones, with numerous related middleware platforms, such as: Web Services, Java RMI for CS; JMS, SIENA for PS; and JavaSpaces, Lime for TS. XSB then provides support for the seamless integration of heterogeneous interaction paradigms (CS, PS and TS).

In a nutshell, our systematic interoperability approach implemented by the proposed XSB is carried out in two stages. First, a middleware platform is abstracted under a corresponding interaction paradigm among the three base ones, i.e., CS, PS and TS. To this aim, we have elicited a connector model for each paradigm, which comprehensively covers its essential semantics. Then, these three models are abstracted further into a single generic application (GA) connector model, which encompasses their common interaction semantics. Based on GA, we build abstract connector converters that enable interconnecting the base interaction paradigms.

Following the above, XSB is an abstract service bus that prescribes only the high-level semantics of the common bus protocol, which is the GA semantics. Furthermore, we provide an implementation of the XSB, building upon existing SOA and ESB realizations. XSB features richer interaction semantics than common ESBs to deal effectively with the increased Future Internet heterogeneity. Moreover, from its very conception, XSB incorporates special consideration for the cross-integration of heterogeneous interaction paradigms. Services relying on different interaction paradigms can be plugged into XSB by employing binding components (BCs) that adapt between their native middleware and the common bus protocol. This adaptation is based on the abstractions, and in particular on the conversion between the native middleware, the corresponding CS/PS/TS abstraction, and the GA abstraction.

Furthermore, we provide a companion implementation, named Light Service Bus (LSB), targeting the Internet of Things(IoT) domain. LSB forms a concrete access solution for IoT systems as it is able to cope with the diversity of the involved interaction protocols and take care of the IoTS specifics, such as resource constraints, dynamic environments, data orientation, etc. It is implemented to be lightweight in nature and uses REST as the common protocol/bus in place of an ESB solution. In LSB, we confirm the wide use of the aforementioned interaction paradigms (CS/PS/TS) but also underline the existence of an additional paradigm focused on continuous interaction known as Streaming (STR).

Both the XSB and LSB solutions are available for download under open source licenses at http://xsb.inria.fr and http://websvn.ow2.org/listing.php?repname=choreos&path=%2Ftrunk%2Fextensible-service-access%2Flsb%2FLsbBindingComponents%2F respectively.

5.5. MobIoT – Service-oriented Middleware for the Mobile IoT

Participant: Valérie Issarny [correspondent].

MobIoT is a service-oriented middleware aimed at the mobile Internet of Things (IoT), which in particular deals with the ultra-large scale, heterogeneity and dynamics of the target networking environment. MobIoT offers novel probabilistic service discovery and composition approaches, and wraps legacy access protocols to be seamlessly executed by the middleware. The middleware exposes two levels of service abstractions: Thing as a service (on the service provider side); and Things measurements/actions as a service (on the service consumer side).

Key features of MobIoT lie in: (i) the exploitation of ontologies to overcome the heterogeneity of the Things network, (ii) the introduction of probabilistic approaches for both registering and retrieving networked things so as to filter out the ones that are redundant with already known alternatives, and finally, (iii) the exploitation of Thing services composition for responding to user queries asking information about the physical world so as to ease interaction with such a complex and dynamic networking environment.

MobIoT is implemented using Java and the Android platform, and consists of two complementary components: The MobIoT Mobile middleware and the MobIoT Web Service. The MobIoT Mobile middleware is deployed on mobile devices (e.g., smartphones, tablets, sensor devices). It wraps: (i) the Query component that enables the querying of the physical world, (ii) the Registration component that deals with the probabilistic registration of local sensors and actuators, (iii) the domain ontology that allows reasoning about the features of Things, and (iv) the Sensor Access component that enables the sensor data retrieval and exposure. The MobIoT Web Service wraps: (i) the Registry component that keeps tracks of the registered services, (ii) the probabilistic Lookup component that allow retrieving relevant services in a scalable way, and (iii) the Composition & Estimation component to answer queries over the physical world using available Thing services, and finally domain and devices ontologies.

The MobIoT middleware is available for download under an open source license at http://choreos.eu/bin/Documentation/IoTS_Middleware.

5.6. Srijan: Data-driven Macroprogramming for Sensor Networks

Participant: Animesh Pathak [correspondent].

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

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

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

The current implementation of *Srijan* targets both the Sun SPOT sensor nodes and larger nodes with J2SE. Most recently, *Srijan* also includes rudimentary support for incorporating Web services in the application being designed.

The software is released under open source license, and available as an Eclipse plug-in at http://code.google.com/p/srijan-toolkit/.

5.7. Yarta: Middleware for supporting Mobile Social Applications

Participant: Animesh Pathak [correspondent].

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

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

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

The current implementation of the Yarta middleware targets both desktop/laptop nodes running Java 2 SE, as well as Android smart phones.

The software is released under open source license at https://gforge.inria.fr/projects/yarta/.

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

Participant: Valérie Issarny [correspondent].

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

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

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

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

The iBICOOP middleware has been licensed by AMBIENTIC (http://www.ambientic.com/), a start-up that specifically develops innovative mobile distributed services on top of the iBICOOP middleware that allows for seamless interaction and content sharing in today's multi-* networks.

ASAP Project-Team

5. Software and Platforms

5.1. MediEgo: A recommendation solution for webmasters

Participants: Antoine Boutet, Jacques Falcou, Arnaud Jégou, Anne-Marie Kermarrec, Jean-François Verdonck.

Contact:	Anne-Marie Kermarrec
Licence:	Proprietary
Presentation:	Recommendation solution for webmasters
Status:	Beta version, IDDN.FR.001.490030.000.S.P.2013.000.30000 on 09/12/2013

MEDIEGO is a solution for content recommendation based on the users navigation history. The solution 1) collects the usages of the Web users and store them in a profile; 2) uses this profile to associate to each user her most similar users; 3) leverages this implicit network of close users in order to infer their preferences and recommend advertisements and recommendations. MEDIEGO achieves scalability using a sampling method, which provides very good results at a drastically reduced cost.

5.2. MediEgo Dashboard: A personalized news dashboard

Participants: Yuri Barssi, Antoine Boutet, Anne-Marie Kermarrec, Jean-François Verdonck.

Contact:	Antoine Boutet
Licence:	Proprietary
Status:	Beta version

This work has led to the development of MEDIEGO Dashboard, a personalized news recommendation system. In MEDIEGO Dashboard, users benefit from a personalized stream of news matching their interests. Additionally, users can use explicit subscriptions as well as post content and navigate through tags. MEDIEGO Dashboard is available through a web interface and a mobile-based Android application. To provide personalization, MEDIEGO Dashboard exploits the users' opinions regarding their received news to identify users with similar interests. MEDIEGO Dashboard is centralized and it allows us to test and evaluate different recommendation schemes. In collaboration with EIT/ICT Lab, an experiment has been conducted with a set of users at Trento (Italie). This experiment allowed us to collect traces and to perform a user survey to assess and improve our solution. This solution will soon be interconnected to AllYours-P2P.

5.3. AllYours-P2P: A distributed news recommender (former WhatsUp)

Participants: Heverson Borba Ribeiro, Antoine Boutet, Davide Frey, Arnaud Jégou, Anne-Marie Kermarrec, Jean-François Verdonck.

Contact:	Antoine Boutet
Licence:	AGPL 3.0
Presentation:	A distributed news recommender
Status:	Beta version, IDDN.FR.001.500002.000.S.P.2013.000.30000 on 09/12/2013

In the context of the AllYours EIT/ICT Labs project, we refined the implementation of WhatsUp into the AllYours-P2P application. The application provides a distributed recommendation system aimed to distribute instant news in a large scale dynamic system. It consists of two parts, running on each peer: an embedded application server, based on Jetty, and a web interface accessible from any web browser. The application server exchanges information with other peers in the system, while the web interface displays news items and collects the opinions of the user.

5.4. HyRec: A hybrid recommender system

Participants: Antoine Boutet, Davide Frey, Anne-Marie Kermarrec.

Contact:	Antoine Boutet
Licence:	Proprietary
Status:	Beta version, IDDN.FR.001.500007.000.S.P.2013.000.30000 on 09/12/2013

This work leads to the development of HyRec, a hybrid recommender system. The motivation of this work is to explore solutions that could in some sense democratize personalization by making it accessible to any content provider company without generating huge investments. HyRec implements a user-based collaborative filtering scheme and offloads CPU-intensive recommendation tasks to front-end client browsers, while retaining storage and orchestration tasks within back-end servers. HyRec seeks to provide the scalability of P2P approaches without forcing content providers to give up the control of the system.

5.5. GossipLib: A library for gossip-based applications

Participants: Heverson Borba Ribeiro, Davide Frey, Anne-Marie Kermarrec.

Contact:	Heverson Borba Ribeiro, Davide Frey
Licence:	AGPL 3.0
Presentation:	Library for gossip protocols
Status:	Alpha version, IDDN.FR.001.500001.000.S.P.2013.000.10000 on 09/12/2013

GossipLib is a library consisting of a set of JAVA classes aimed to facilitate the development of gossip-based application in a large-scale setting. It provides developers with a set of support classes that constitute a solid starting point for building any gossip-based application. GossipLib is designed to facilitate code reuse and testing of distributed application, and provides also the implementation of a number of standard gossip protocols that may be used out of the box or extended to build more complex protocols and applications. These include for example the peer-sampling protocols for overlay management. GossipLib also provides facility for the configuration and deployment of applications as final-product but also as research prototype in environments like PlanetLab, clusters, network emulators, and even as event-based simulation. The code developed with GossipLib can be run both as a real application and in simulation.

5.6. YALPS: A library for P2P applications

Participants: Heverson Borba Ribeiro, Davide Frey, Anne-Marie Kermarrec.

Contact:	Heverson Borba Ribeiro, Davide Frey
Licence:	Open Source
Presentation:	Library for p2p applications
Status:	Beta version, IDDN.FR.001.500003.000.S.P.2013.000.10000 on 09/12/2013

YALPS is an open-source Java library designed to facilitate the development, deployment, and testing of distributed applications. Applications written using YALPS can be run both in simulation and in real-world mode without changing a line of code or even recompiling the sources. A simple change in a configuration file will load the application in the proper environment. A number of features make YALPS useful both for the design and evaluation of research prototypes and for the development of applications to be released to the public. Specifically, YALPS makes it possible to run the same application as a simulation or in a real deployment. Applications communicate by means of application-defined messages which are then routed either through UDP/TCP or through YALPS's simulation infrastructure. In both cases, YALPS's communication layer offers features for testing and evaluating distributed protocols and applications. Communication channels can be tuned to incorporate message losses or to constrain their outgoing bandwidth. Finally, YALPS includes facilities to support operation in the presence of NATs and firewalls using relaying and NAT-traversal techniques. This work was done in collaboration with Maxime Monod (EPFL).

5.7. HEAP: Heterogeneity-aware gossip protocol

Participants: Davide Frey, Arnaud Jégou, Anne-Marie Kermarrec.

Contact:	Davide Frey
Licence:	Open Source
Presentation:	Java Application
Status:	Release & ongoing development

This work has been done in collaboration with Vivien Quéma (CNRS Grenoble), Maxime Monod and Rachid Guerraoui (EPFL), and has lead to the development of a video streaming platform based on HEAP, *HEterogeneity-Aware gossip Protocol*. The platform is particularly suited for environment characterized by heterogeneous bandwidth capabilities such as those comprising ADSL edge nodes. HEAP is, in fact, able to dynamically leverage the most capable nodes and increase their contribution to the protocol, while decreasing by the same proportion that of less capable nodes. During the last few months, we have integrated HEAP with the ability to dynamically measure the available bandwidth of nodes, thereby making it independent of the input of the user.

ASCOLA Project-Team

5. Software and Platforms

5.1. btrCloud (and Entropy)

Participants: Jean-Marc Menaud [correspondent], Guillaume Le Louët, Thierry Bernard, Frédéric Dumont.

Orchestration, virtualization, energy, autonomic system, placement, cloud computing, cluster, data center, scheduler, grid

btrCloud is a virtual machine manager for clusters and provides a complete solution for the management and optimization of virtualized data center. btrCloud (acronym of better cloud) is composed of three parts.

The analysis function enables operatives and people in charge to monitor and analyze how a data-center works, be it on a daily basis or on the long run and predict future trends. This feature includes a performance, an analysis and a trends board.

btrCloud, by the integration of btrScript, provides (semi-)automated VM lifecycle management, including provisioning, resource pool management, VM tracking, cost accounting, and scheduled deprovisioning. Key features include a thin client interface, template-based provisioning, approval workflows, and policy-based VM placement.

Finally, several kinds of optimizations are currently available, such as energy and load balancing. The former can help save up to around 20% of the data-center energy consumption. The latter provides optimized quality of service properties for applications that are hosted in the virtualized datacenters.

btrCloud is available at http://www.btrcloud.org.

5.2. EScala and JEScala

Participants: Jacques Noyé [correspondent], Jurgen Van Ham.

AOP, inheritance, event-based programming, events, declarative events, asynchronous events, join operator, Scala

EScala is an extension of the programming language Scala with support for events as object members. EScala combines ideas of event-driven, aspect-oriented and functional reactive programming.

Events are natural abstractions for describing interactive behavior as part of an object interface. In conventional object-oriented languages, events are implemented indirectly, typically using the Observer pattern. C# eliminates the corresponding glue code and directly supports events as object members. However, events are still *explicitly* triggered at specific locations within the program.

EScala goes much further. First, it also supports *implicit* events. Akin to join points in aspect-oriented languages, these events are implicitly produced at specific execution points, such as the beginning or the end of the execution of a method. Second, *declarative events* make it possible to compose events using logical operators as well as to filter them and alter their content.

EScala events are fully integrated with object-oriented features. An event is defined in the context of its owner object. Event definitions are inherited in subclasses and event uses are late-bound. Unlike typical aspect-oriented languages, EScala preserves object-oriented encapsulation and modular reasoning.

JEScala extends EScala with support for concurrent programming (see Sec. 6.2). Events can be declared as *asynchronous* so that their handling takes place concurrently. A new composition operator, the *join* operator, inspired by the join calculus, can also be used to synchronize the concurrent activities created by asynchronous events and communicate between them.

This is joint work with the Software Technology Group at TU Darmstadt.

Prototype implementations of these languages are available through http://www.stg.tu-darmstadt.de/research.

5.3. CSLA

Participants: Thomas Ledoux [correspondent], Yousri Kouki.

Service-level agreement, Cloud computing, elasticity

Verifying non-functional properties like performance, dependability, energy consumption and economical costs of Clouds is challenging today due to ad-hoc management in terms of Quality-of-Service (QoS). We believe that a differentiating element between Cloud computing environments will be the QoS and the service-level agreement (SLA) provided by the Cloud.

CSLA, the Cloud Service Level Agreement language, allows the definition of SLA properties for arbitrary Cloud services (XaaS). CSLA addresses QoS uncertainty in unpredictable and dynamic environment and provides a cost model of Cloud computing. Besides the standard formal definition of contracts – comprising validity, parties, services definition and guarantees/violations – CSLA is enriched with features, such as QoS degradation and an advanced penalty model, thus introducing fine-grained language support for Cloud elasticity management [13].

CSLA is available at http://www.emn.fr/z-info/csla.

5.4. SAdapt

Participants: Ronan-Alexandre Cherrueau [correspondent], Mario Südholt.

Service-oriented systems, distributed programming, event-based programming, workflow patterns

The SAdapt tool provides an implementation of workflow adaptation patterns and allows the transformation of service-oriented systems implemented using Apache’s CXF service infrastructure in terms of high-level declarative service transformations. The transformations are defined using an expressive language that supports matching of the execution of service-based systems in terms of flexible patterns over service compositions.

The SAdapt tool has partially been developed and is employed in the A4Cloud EU project (see Sec. 8.2) as a basis for our work on the enforcement of accountability properties in complex cloud-based systems.

The SAdapt tool and its application, notably to the security hardening of service systems that use OAuth 2 for the authorization of resource accesses is available at http://a4cloud.gforge.inria.fr/doku.php?id=start:advservcomp.

ATLANMOD Project-Team

5. Software and Platforms

5.1. The ATL Model Transformation Language

URL: http://www.eclipse.org/atl/

With an eye on the normative work of the OMG (MOF, OCL, QVT, etc.), a new conceptual framework has been developed based on a second generation model transformation language called ATL. Although ATL influenced the OMG standard, the approach is more general as discussed in [8]. In 2004 IBM gave an Eclipse innovation award to the ATL project. In 2007 Eclipse recognized ATL as one central solution for model transformation and promoted it to the M2M project (see *Eclipse.org/m2m*). There are more than 200 industrial and academic sites using ATL today, and several Ph.D. thesis in the world are based on this work.

In 2011 we started a new evolution phase for ATL. Our mid-term plan is making of ATL the leading solution for building autonomous reactive transformation systems, i.e. transformation networks that can autonomously manage a set of dataflows among the application models.

Following this line, we first implemented a new refinement mode for ATL, to support in-place transformations. This extension allows the dynamic manipulation of models while keeping them connected to runtime applications. Next, we presented a lazy execution algorithm for ATL. With it, the elements of the target model are generated only when and if they are accessed. This extension allows to build reactive transformation systems that react to requests of model elements, by triggering the necessary computation. Our lazy version of ATL enables also transformations that generate infinite target models, extending the application space of the model-transformation paradigm.

The latest (still ongoing) work in this direction is the development of a full reactive ATL engine, able to activate the minimal computation for responding to updates or request on the involved models. This engine is studied to scale up with large ATL networks. In this line we also introduced an algorithm for simplifying ATL transformation chains.

Performing just the required work on model transformation improves scalability, an open issue the previous described works contribute to solve. An efficient execution, as in the the lazy and reactive scenarios, may help with scalability problems by focusing the tasks in the required part of a very large transformation. However, this is not always the case and we might have to perform operations in the whole model. In this scenario, a solution for the scalability problem would be to take advantage of multi-core architectures that are very popular today, to improve computation times in the transformation of very large models. In this sense, a first step explores the strong parallelization properties rule-based languages like ATL have. A new prototype implementation of a parallel ATL engine have been developed showing how transformations can be developed without taking into account concurrency concerns, and that a transformation engine can automatically parallelize operations improving execution times.

5.2. MoDisco (Model Discovery)

URL: http://www.eclipse.org/MoDisco/

MoDisco is an open source Eclipse project that provides a generic and extensible framework dedicated to the elaboration of Model Driven Reverse Engineering (MDRE) solutions. Gathering contributions from both academics and industrials, the goal of the project is to federate common efforts in the model-based transformation of legacy software systems implemented using different technologies (e.g. Java, COBOL, C). The first principle is to discover models out of legacy artifacts, representing appropriately all the relevant information, to be then used as part of reverse engineering processes for software understanding, evolution or modernization. Targeted scenarios include software (technical or architectural) migration of large legacy systems, but also retro-documentation, refactoring, quality assurance, etc. Within this context, MoDisco has collaborations with the OMG Architecture Driven Modernization (ADM) Task Force, for which the project provides several reference implementations of its standards: Knowledge Discovery Metamodel (KDM), Software Measurement Metamodel (SMM) and Abstract Syntax Tree Metamodel (ASTM).

The MoDisco framework is composed of a set of Eclipse plugins, and relies on the de-facto standard Eclipse Modeling Framework (EMF) for model handling. Thanks to its modular architecture, it allows completely covering the three steps of a standard MDRE approach: 1) Discovery (i.e. extracting a complete model of the source code), 2) Understanding (i.e. browsing and providing views on this model for a given purpose) and 3) Transformation (evolving the model towards a new technology, architecture, etc). More specifically, as part of its *Infrastructure* layer, MoDisco offers the set of generic (i.e.; legacy technology-independent) reusable components really useful to build the core of MDRE solutions: Discovery Manager and Workflow for MDRE task orchestration, Model Browser for advanced navigation in complex models, model extension and customization capabilities for understanding (e.g. views definition), etc. As part of its *Technologies* layer, it provides an advanced support for the Java, JEE and XML technologies, including complete metamodels, corresponding model discoverers, transformations, code generators, customizations, query libraries, etc.

MoDisco (or some of its components) is being used by different partners including other academics, industrials (e.g. Sodifrance on several of their real modernization projects for their customers) or Eclipse projects (e.g. Eclipse-MDT Papyrus as developed by CEA). Moreover, the Eclipse-EMFT EMF Facet project has been initiated as a MoDisco spin-off, in order to externalize some features which are not actually specific to reverse engineering problems and thus may be reused in many different contexts (cf. corresponding EMF Facet section).

The initiative continues to be developed within the context of the European FP7-ICT project named ARTIST [2], and also to a lower extent within the context of the French FUI 13 project named TEAP.

5.3. Community-driven language development

URL: http://code.google.com/a/eclipselabs.org/p/collaboro/

Software development processes are collaborative in nature. Neglecting the key role of end-users leads to software that does not satisfy their needs. This collaboration becomes specially important when creating Domain-Specific Languages (DSLs), which are (modeling) languages specifically designed to carry out the tasks of a particular domain. While end-users are actually the experts of the domain for which a DSL is developed, their participation in the DSL specification process is still rather limited nowadays.

Thus, Collaboro is an approach to make language development processes more participative, meaning that both developers and users of the language can collaborate together to design it and make it evolve. The tool has been developed as an Eclipse plugin, with currently the following features implemented:

- Version view to navigate through the Proposals of a version of a language. For each Proposal, the solutions and comments are shown.
- Collaboration view to show the data related to a Collaboration selected in the version view. This view also shows the changes to apply if the selected element is a Solution.
- The user can login to the Collaboro system and create proposals, solutions and comments by right-clicking in the version view. The user can also vote for/against the collaborations.

---

[2] http://www.artist-project.eu/

- Decision engine based on a total agreement (i.e., all the community users must vote for the collaboration). The decision engine can be launch by using the menu bar.
- Notation engine and Notation view to render SVG snapshots of the DSL concrete syntax.
- Support for example-driven development of DSMLs, thus incoporating a graphical editor which allows end-users to draw examples of the DSML they are developing.

5.4. JSON Discoverer

URL: http://atlanmod.github.io/json-discoverer/

Given a set of JSON documents, the tool (distributed as an open source Eclipse plugin contributed to MoDisco) returns a model describing their implicit schema. We follow an iterative process where new JSON documents (from the same or different services within the API) contribute to enrich the generated model. The model helps to both understand single services and to infer possible relationships between them, thus suggesting possible compositions and providing an overall view of the application domain. The tool has also been released as a web site, thus allowing any web developer to use our approach without the need of installing Eclipse.

5.5. EMF-REST

URL: http://emf-rest.com/

EMF is *the modeling framework* of the Eclipse community. While EMF is able to automatically generate Java APIs from Ecore models, it is still missing support to deal with Web APIs such as RESTful ones that could boost the use of modeling techniques in the Web. However, the creation of RESTful APIs requires from developers not only an investment in implementation but also a good understanding of the REST Principles to apply them correctly. We therefore created EMF-REST, a tool that empowers EMF to get Truly RESTFul APIs from Ecore models, thus allowing web developers to generate JSON-based Web APIs for their applications. It generates both a JavaScript API to work with models as Javascript Objects in the client-side (without any EMF dependency) and REST services in the server-side based on the Java JAX-RS specification.

5.6. EMF Views (Model Views)

URL: http://emfviews.jdvillacalle.com/

The Eclipse Modeling Framework (EMF) is widely used in the Eclipse community: defining domain models and generating corresponding source code, modeling software architectures, specifying DSL concepts or simply representing software/user data in different contexts. This implies that any software project involves a large number of heterogeneous but interrelated EMF models.

To make matters worse, not all participants in the project should have the same kind of access/views on the models. Some users only need to see some parts of one model, others have to get the full model extended with data from another model, or simply access to a combination of information coming from different interconnected models. Up to now, creating such perspectives transparently in EMF was almost impossible.

Based on the unquestionable success/usefulness of database views to solve similar problems in databases, EMF Views aims to bring the same concept to the modeling world. Thanks to the three main constructs (inspired from SQL) offered by the tool, designers can create new model views: SELECTing a subset of elements from a model, PROJECTing only some of the properties of those elements and/or JOINing them with elements from other models. A model view is a special type of model whose instances are directly computed at runtime based on the model view definition and concerned actual model(s).

EMF Views is currently being developed in the context of the TEAP industrial project http://www.teap-project.org/, by showing different possible applications of model views including:

- Software architect/developer views relating UML design models and Java code models (cf. Eclipse MoDisco);
- Enterprise architect views linking (BPMN) business process models, (ReqIF) requirements models and (TOGAF) architecture models;
- View querying using dedicated technologies (e.g. Eclipse IncQuery);
- View transformation using dedicated technologies (e.g. Eclipse ATL).

5.7. EMFtoCSP

URL: http://code.google.com/a/eclipselabs.org/p/emftocsp/

EMFtoCSP is a tool for the verification of precisely defined conceptual models and metamodels. For these models, the definition of the general model structure (using UML or EMF) is supplemented by OCL constraints. The Eclipse Modeling Development Tools (MDT [3]) provides mature tool support for such OCL-annotated models with respect to model definition, transformation, and validation.

However, an additional important task that is not supported by Eclipse MDT is the assurance of model quality. A systematical assessment of the correctness of such models is a key issue to ensure the quality of the final application. EMFtoCSP fills this gap by provided support for automated model verification in Eclipse.

Essentially, the EMFtoCSP is a sophisticated bounded model finder that yields instances of the model that conform not only to the structural definition of the model (e.g. the multiplicity constraints), but also to the OCL constraints. Based on this core, several correctness properties can be verified:

1. Satisfiability – is the model able to express our domain? For this check, the minimal number of instances and links can be specified to ensure non-trivial instances.
2. Unsatisfiability – is the model unable to express undesirable states? To verify this, we add further constraints to the model that state undesired conditions. Then we can check if is it impossible to instantiate the amended model.
3. Constraint subsumption – is one constraint already implied by others (and could therefore be removed)?
4. Constraint redundancy – do different constraints express the same fact (and could therefore be removed)?

To solve these search problems, EMFtoCSP translates the EMF/OCL (resp. UML/OCL) model into a constraint satisfaction problem and employs the Eclipse CLP solver [4]to solve it. This way, constraint propagation is exploited to tackle the (generally NP-hard) search.

The tool is a continuation of the UMLtoCSP approach [47] developed previously by Jordi Cabot, Robert Clarisó and Daniel Riera. It provides a generic plugin framework for Eclipse to solve OCL-annotated models using constraint logic programming. Apart from already supported Ecore and UML metamodels, further metamodels can be added easily in the future. Similarly, other constraint solving back-ends can be integrated. It is provided under the Eclipse Public License.

5.8. EMF Facet

URL: http://www.eclipse.org/modeling/emft/facet/

---

[3] http://www.eclipse.org/modeling/mdt/?project=ocl

[4] http://eclipseclp.org/

EMF Facet is an open source Eclipse project, under the Eclipse Public License (EPL), that provides a generic and extensible framework dedicated to the dynamic and non-intrusive extension of models. It can be used to extend already existing metamodels with additional concepts and properties, the corresponding models being then transparently augmented, reduced or modified accordingly at runtime. Such a metamodel extension is called a facet, and can be specified on top of any metamodel in EMF Ecore. The underlying mechanism is based on the runtime execution of queries on the models corresponding to the *faceted* metamodels. Facets are notably particularly relevant for obtaining different views on existing models without having to actually alter them with any extra data.

The EMF Facet framework is composed of several Eclipse plugins, and relies on the de-facto standard Eclipse Modeling Framework (EMF) for model handling. The facet definitions are stored as facet models, allowing them to be exchanged and reused in various contexts. The queries can be implemented using any suitable query language (e.g. ATL, OCL, Java, XPath), as far as the corresponding adapters exist and are correctly registered within the framework. The proposed tooling includes dedicated editors for creating, editing and saving both facet and query definitions, the implemented support for Java, OCL and ATL queries, a Table Editor for visualizing query results. An advanced support for the model display customization (e.g. icons, colors, fonts) is also provided as part of the framework.

EMF Facet is currently intensively used in MoDisco for extracting and displaying different specific views from large models of legacy systems. Its extension and customization capabilities are actually integrated into several MoDisco components, such as notably the MoDisco Model Browser. However, different other integration possibilities will be also explored in the future.

The initiative continues to be developed within the context of the European FP7-ICT project ARTIST.

5.9. Neo4EMF

URL: http://www.neo4emf.com

Neo4EMF is an open source software distributed under the terms of the Eclipse Public License, that provides a persistence backend for big, complex and highly interconnected EMF models.

Neo4EMF is a model repository and persistence framework allowing on-demand loading, storage, and unloading of large-scale EMF models. Neo4EMF uses a sophisticated unloading approach apart from simple Soft/Weak references. Moreover, Neo4EMF provides a No-SQL database persistence framework based on Neo4j [5], which is a transactional property-graph database that proved a remarkable running speed for connected data operations compared to relational databases.

In terms of performance, Neo4EMF eases data access and storage not only in a manner to reduce time and memory usage but also to allow big models to fit into small memory. This is established through an on-demand loading mechanism that offers :

- Lightweight first time loading of model elements : we separated EMF objects and their data fields, thus, data objects are only instantiated if an access request to one of their fields is established
- Dynamic partitioning of model elements : a partition represents a group of model elements to be unloaded all together. Hence, after each EMF operation call, first time loading objects are organized in their suitable partition
- Unloading of model partitions : when memory reaches a given threshold, we use a selection strategy to choose one or more partitions to be removed from the memory

A session about Neo4EMF took place at eclipseCon Europe 2013 [6], held in Ludwigsburgh Germany.

However, works are still going over Neo4EMF (within the context of the project ITM Factory -FUI14), to provide more utilities such as concurrent access, model distribution, and other Ecore utilities.

---

[5] http://www.neo4j.org

[6] https://www.eclipsecon.org/europe2013/neo4emf-big-models-made-possible

AVALON Team

5. Software and Platforms

5.1. BitDew

Participants: Gilles Fedak [correspondant], Haiwu He.

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

BITDEW is distributed open source under the GPLv3 or Cecill licence at the user's choice. 10 releases were produced over the last two years, and it has been downloaded approximately 6,000 times on the Inria forge. Known users are Université Paris-XI, Université Paris-XIII, University of Florida (USA), Cardiff University (UK) and University of Sfax (Tunisia). In terms of support, the development of BitDew is partly funded by the Inria ADT BitDew and by the ANR MapReduce projects. Thanks to this support, we have developed and released the first prototype of the MapReduce programming model for Desktop Grids on top of BitDew. In 2012, 8 versions of the software have been released, including the version 1.2.0 considered as a stable release of BitDew with many advanced features. Our most current work focuses on providing reliable storage on top of hybrid distributed computing infrastructures.

5.2. DIET

Participants: Daniel Balouek, Eddy Caron [correspondant], Frédéric Desprez, Maurice-Djibril Faye, Arnaud Lefray, Guillaume Verger, Jonathan Rouzaud-Cornabas, Lamiel Toch, Huaxi Zhang.

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

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

DIET provide a support for Cloud architecture. and it takes benefits from virtualized resources. As cloud resources are dynamic, we have on-going research in the field of automatic and elastic deployment for middleware systems. DIET will be able to extend and reduce the amount on aggregated resources and adjust itself when resources fail.

In the context of the Seed4C project, we have studied how secured our platform, authenticated and secured interactions between the different parts of our middleware and between our middleware and its users. By the way, we have added the SSL support into the DIET communication layer. We have worked to show how to securely use public cloud storage without taking the risk of losing confidentiality of data stored on them.

We have started a work to design a plug-in schedulers into DIET to deal with energy management. Using this scheduler we have obtain a significatif gain close to 25% with a minor weakening of performance (6%). Moreover we have experimented some dynamic resources management through DIET based on the energy criteria.

5.3. Pilgrim

Participants: Eddy Caron, Matthieu Imbert [correspondant].

Pilgrim (http://pilgrim.gforge.inria.fr) is an open metrology and prediction performance framework whose goal is to provide easy and powerful tools for instrumenting computer platforms and predicting their behavior. Those tools are aimed at being used not only by humans but also by programs, in particular by resource managers and schedulers. Pilgrim is designed to be a loosely coupled integration of various custom-developed or off-the-shelf tools.

5.4. Sam4c

Participants: Eddy Caron, Arnaud Lefray [correspondant], Jonathan Rouzaud-Cornabas.

Sam4C (https://gforge.inria.fr/projects/sam4c/) -Security-Aware Models for Clouds- is a graphical and textual editor to model Cloud applications (as virtual machines, processes, files and communications) and describe its security policy. Sam4C is suitable to represent any static application without deadline or execution time such as n-tiers or parallel applications. This editor is generated in Java from an EMF -Eclipse Modeling Framework- metamodel to simplify any modifications or extensions. The application model and the associated security policy are compiled in a single XML file which serves as input for an external Cloud security-aware scheduler. Alongside with this editor, Cloud architecture models and provisioning algorithms are provided for simulation (in the current version) or real deployments (in future versions). During this step of development this software is private and available only for Seed4C project members. The design of Sam4c is a joint effort with ENSIB (Bourges).

5.5. SimGrid

Participants: Georgios Markomanolis, Jonathan Rouzaud-Cornabas, Frédéric Suter [correspondant].

SIMGRID is a toolkit for the simulation of distributed applications in heterogeneous distributed environments. The specific goal of the project is to facilitate research in the area of parallel and distributed large scale systems, such as Grids, P2P systems and clouds. Its use cases encompass heuristic evaluation, application prototyping or even real application development and tuning. SIMGRID has an active user community of more than one hundred members, and is available under GPLv3 from http://simgrid.gforge.inria.fr/.

5.6. HLCMi, L^2C, & Gluon++

Participants: Zhengxiong Hou, Vincent Lanore, Christian Perez [correspondant].

HLCMI (http://hlcm.gforge.inria.fr) is an implementation of the HLCM component model. HLCM is a generic extensible component model with respect to component implementations and interaction concerns. Moreover, HLCM is abstract; it is its specialization—such as HLCM/L^2C—that defines the primitive elements of the model, such as the primitive components and the primitive interactions.

HLCMI is making use of Model-driven Engineering (MDE) methodology to generate a concrete assembly from an high level description. It is based on the Eclipse Modeling Framework (EMF). HLCMI contains 700 Emfatic lines to describe its models and 7000 JAVA lines for utility and model transformation purposes. HLCMI is a general framework that supports several HLCM specializations: HLCM/CCM, HLCM/JAVA, HLCM/L^2C and HLCM/Charm++ (known as Gluon++).

L^2C (http://hlcm.gforge.inria.fr) is a *Low Level Component* model implementation targeting at use-cases where overhead matters such as High-Performance Computing. L^2C does not offer network transparency neither language transparency. Instead, L^2C lets the user choose between various kinds of interactions between components, some with ultra low overhead and others that support network transport. L^2C is extensible as additional interaction kinds can be added quite easily. L^2C currently supports C++, FORTRAN 2013, MPI and CORBA interactions.

Gluon++(http://hlcm.gforge.inria.fr) is a thin component model layer added on top of Charm++ (http://charm.cs.uiuc.edu/). It defines chare components as a Charm++ chare with minimal metadata, C++ components as a C++ class with minimal metadata, (asynchronous) entry method calls between components, and plain C++ method calls between components.

L^2C and Gluon++ are implemented in the LLCMc++ framework (http://hlcm.gforge.inria.fr). It is distributed under a LGPL licence and represents 6400 lines of C++.

5.7. Execo

Participants: Matthieu Imbert [correspondant], Laurent Pouilloux.

Execo (http://execo.gforge.inria.fr) offers a Python API for local or remote, standalone or parallel, processes execution. It is especially well suited for scripting workflows of parallel/distributed operations on local or remote hosts: automating a scientific workflow, conducting computer science experiments, performing automated tests, etc. The core python package is `Execo`. The `execo_g5k` package provides a set of tools and extensions for GRID'5000. The `execo_engine` package and `execo-run` script provide an extendable experiment engine.

Execo currently has more than 10 users in and outside the AVALON team, who rely on it to automate experimental workflows, mainly on GRID'5000 ([26]).

It is distributed under GPLv3 and it is made of 5200 lines of Python.

5.8. Grid'5000

Participants: Frédéric Desprez, Simon Delamare, Laurent Lefèvre, Christian Perez.

The GRID'5000 experimental platform (http://www.grid5000.fr) is a scientific instrument to support computer science research related to distributed systems, including parallel processing, high performance computing, cloud computing, operating systems, peer-to-peer systems and networks. It is distributed on 10 sites in France and Luxembourg, including Lyon. GRID'5000 is a unique platform as it offers to researchers many and varied hardware resources and a complete software stack to conduct complex experiments, ensure reproducibility and ease understanding of results.

Not only GRID'5000 is heavily used for Avalon research, but several team members are also involved in GRID'5000 direction:

- Frédéric Desprez is leading the "Groupement d'Intérêt Scientifique Groupement Grille 5K" which drives GRID'5000.
- Laurent Lefèvre is responsible of the GRID'5000 Lyon platform and member of the GRID'5000 direction committee.
- Christian Perez is leading the Hemera initiative (https://www.grid5000.fr/Hemera) and he is a member of the GRID'5000 direction committee.
- Simon Delamare is the operational manager of the technical team.

CEPAGE Project-Team

5. Software and Platforms

5.1. SimGrid

Participants: Paul Renaud-Goud, Lionel Eyraud-Dubois [correspondant].

SimGrid (http://simgrid.gforge.inria.fr/) is a toolkit that provides core functionalities for the simulation of distributed applications in heterogeneous distributed environments. The specific goal of the project is to facilitate research in the area of parallel and distributed large scale systems, such as Grids, P2P systems and clouds. Its use cases encompass heuristic evaluation, application prototyping or even real application development and tuning. It is based on experimentally validated models, and features very high scalability, which allows to perform very large scale simulations. It is used by over a hundred academic users all over the world, and has been used in about one hundred scientific articles.

CEPAGE has contributed to this software by participating in the management of the project and in many design decisions. As a part of the SONGS project, we also participated in the development and validation of new models and interfaces for the HPC and Cloud Computing platforms.

Software assessment : A-4, SO-4, SM-4, EM-3, SDL-5
Contribution : DA-2, CD-2, MS-2, TPM-3.

5.2. Hubble

Participants: Ludovic Courtes, Nicolas Bonichon [correspondant].

Hubble is implemented in Scheme, using GNU Guile version 2. Details of the simulation, such as keeping track of processor occupation and network usage, are taken care of by SimGrid, a toolkit for the simulation of distributed applications in heterogeneous distributed environments.

The input to Hubble is an XML description of the DAG of build tasks. For each task, a build duration and the size in bytes of the build output are specified. For our evaluation purposes, we collected this data on a production system, the http://hydra.nixos.org/ build farm hosted at the Technical University of Delft. The DAG itself is the snapshot of the Nix Package Collection (Nixpkgs) corresponding to this data. Hubble has its own in-memory representation of the DAG in the form of a purely functional data structure.

The Nixpkgs DAG contains fixed-output nodes, i.e., nodes whose output is known in advance and does not require any computation. These nodes are typically downloads of source code from external web sites. The raw data collected on http://hydra.nixos.org/ specifies a non-zero duration for these nodes, which represents the time it took to perform the download. This duration info is irrelevant in our context, since they don’t require any computation, and Hubble views these nodes as instantaneous.

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

Software assessment: A-3, SO-3, SM-2, EM-1, SDL-2.
Contribution: DA-4, CD-4, MS-4, TPM-4.

5.3. Gengraph

Participant: Cyril Gavoille [correspondant].

This is a command-line tool for generating graphs. There are several output formats, includes the dot format from GraphViz. It generates also .pdf files for visualization. Several graph algorithms have been implemented (diameter, connectivity, treewidth, etc.) which can be tested on the graphs. The software has been originally designed for teaching purpose so that students can test their project algorithms on many non trivial families like random geometric graphs, graphs of given density, given treewidth. It is also used for research purpose, in particular the exhaustive search results in the Emilie Diot's thesis are based on `gengraph`. The program can filter a list of graphs based to many criteria, as for instance it can extract all graphs of a given list that are 2-connected, of diameter at least four, and that exclude some minor (or some induced subgraph).

Currently, more than 100 parametrized graph families are implemented, supporting simple operators like complementation, random edge/vertex removal, and others. The source has more than 10,000 lines including a command-line documentation of 2,000 lines. The single source file is available at http://dept-info.labri.fr/~gavoille/gengraph.c

Software assessment: A-3, SO-3, SM-2, EM-2, SDL-2.
Contribution: DA-4, CD-4, MS-4, TPM-4.

5.4. Bedibe

Participants: Lionel Eyraud-Dubois [correspondant], Przemyslaw Uznanski.

Bedibe (Benchmarking Distributed Bandwidth Estimation) is a software to compare different models for bandwidth estimation on the Internet, and their associated instantiation algorithms. The goal is to ease the development of new models and algorithms, and the comparison with existing solutions. Additionally, we have developed a measuring framework which can be deployed on PlanetLab to obtain available bandwidth datasets.

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

Software assessment : A-1-up2, SO-3, SM-1-up2, EM-2, SDL-1-up2.

5.5. MineWithRounds

Participants: Sofian Maabout [correspondant], Nicolas Hanusse.

The software implements a parallel algorithm aiming at computing *Borders* that's sets of maximal/minimal subsets of objects satisfying some anti-monotone condition. It is implemented in `C++` together with the `openMP` library to exploit multi-core machines. In its current status, it outperforms state of the art implementations addressing the Maximal Frequent Itemsets problem.

Software assessment: A-2, SO-4, SM-2, EM-2, SDL-2.
Contribution: DA-4, CD-4, MS-4, TPM-4.

5.6. FSM: Full Skycube Materialization

Participant: Sofian Maabout [correspondant].

The software implements a parallel algorithm aiming at fully materializing skycubes: the set of all possible skyline queries. It is implemented in C++ together with the openMP library to exploit multi-core machines. In its current status, it outperforms state of the art implementations such as QSkycube and PSkyCube (VLDB Journal '13)

Software assessment: A-2, SO-4, SM-2, EM-2, SDL-2.
Contribution: DA-4, CD-4, MS-4, TPM-4.

ADT: Pierre Matri has been hired as an engineer for two year starting from September 15th, 2013. He is in charge to implement our algorithm on top of a large scale infrastructure especially by using Map-Reduce paradigm and its Hadoop implementation. His main focus during 2013 concerned the functional dependencies extraction from large distributed data bases by considering exact and approximate solutions.

5. Software and Platforms

5.1. Intrusion Detection

Members of the team have developed several intrusion detectors and security tools.

Blare implements our approach of illegal information flow detection at the OS level for a single node **and** a set of nodes. Two implementations have been realized: a first one for standard Linux distributions and a second one dedicated to Android operating systems (smartphones, tablets, etc). These implementations imply modification of the standard OS kernel; it monitors information flows between typical OS containers as files, sockets or IPC. System active entities are processes viewed as black-boxes as we only observe their inputs and outputs. Thanks to the work conducted by Christophe Hauser during its PhD [34], it is now possible to extend this information flow monitoring between a set of cooperating nodes. This is made possible by using dedicated tags carried out by IPv4 packets header (CIPSO tags).

However, detection at the OS level is in some cases too coarse-grained to avoid the generation of false positives and to detect attacks targeting the application logic. Even if it remains convenient to define the security policy at the OS-level, sound illegal information flow detection implies an additional detection at the language level. This has led us to implement a detector for Java applications, **JBlare**, to complement the detection at the OS level. JBlare extends the OS-level one by refining the observation of information flows at the language level.

Both **Blare** and **JBlare** development have been supported by an Inria ADT grant since January 2013. Thanks to this grant, Guillaume Brogi has been hired as an engineer to improve the development process of these tools and their quality. He also participates in the dissemination of these tools to the scientific community and potential industrial partners. Blare tools source code and documentation are now available on a dedicated Web site [1].

GNG is an intrusion detection system that correlates different sources (such as different logs) in order to identify attacks against the system. The attack scenarios are defined using the Attack Description Langage (**ADeLe**) proposed by our team, and are internally translated to attack recognition automatons. GNG intends to define time efficient algorithms based on these automatons to recognize complex attack scenarios.

SIDAN (Software Instrumentation for Detecting Attacks on Non-control-data) is a tool that aims to instrument automatically C-language software with assertions whose role is to detect attacks against the software. This tool is implemented as a plugin of the FRAMA-C framework that provides an implementation of static analysis techniques.

Netzob is an open-source tool for reverse engineering, traffic generation and fuzzing of communication protocols. It helps security experts to infer both the message format and the state machine of a protocol using passive and active inference approaches. The model can afterward be used to simulate realistic traffic. This tool is developed by AMOSSYS company and Cidre members. Netzob source code and documentation are available on a dedicated Web site [2]

BSPL policy manager is a tool that aims to charge a security policy in a Android device. Policies are fine-grained information flow policies written in BSPL (Blare Security Policies Languages). Such policies precisely describe how a piece of data owned by an application is allowed to disseminate in the operating system. The BSPL policy manager permits to load a policy, checks if the policy is consistent or not. The policy manager permits to compose policies coming with different applications to obtain the policy of the whole device. A policy defined by the manager is enforced by Blare.

---

[1] https://www.blare-ids.org/

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

5.2. Privacy

GEPETO (GEoPrivacy-Enhancing TOolkit) is an open source software for managing location data (currently in development in cooperation with LAAS). GEPETO can be used to visualize, sanitize, perform inference attacks and measure the utility of a particular geolocated dataset. For each of these actions, a set of different techniques and algorithms can be applied. The global objective of GEPETO is to enable a user to design, tune, experiment and evaluate various sanitization algorithms and inference attacks as well as visualizing the following results and evaluating the resulting trade-off between privacy and utility. An engineer (Izabela Moïse) has contributed to the development of a distributed version of GEPETO based on the MapReduce paradigm and the Hadoop framework that is able to analyze datasets composed of millions of mobility traces in a few minutes [30].

GNOME (Geoprivacy eNhancing tOol for MobilE) is an application for Android smartphone whose main objectives are to (1) to help the user to understand which type of personal information can be learnt from his mobility traces through inference attacks as well as (2) to allow him to decide if he want to sanitize his location data before it is released to a third party (for instance this data could be perturbed according to the desired level of privacy of the user). In addition of the inference attacks such as the extraction of the points of interests and the construction of the mobility model, different mechanisms for generating fake yet realistic mobility traces have been implemented. In particular, one of this method leverages on the mobility model learnt while the second one perturbs the location based on a location variant of differential privacy, a well-established privacy model. These fake mobility traces, that are hard to distinguish from real ones, can be fed to applications running on the smartphone instead of his real location upon request of the user. This application is actually available as a beta release and experiments are actually being conducted with real users in order to test the functionalities of the application. This application has been developed by David Lanoë, an engineer hired as part of the "security and privacy for location-based services" EIT ICT labs activity.

COATI Project-Team

5. Software and Platforms

5.1. Grph

Participants: David Coudert, Luc Hogie [correspondant], Aurélien Lancin, Issam Tahiri, Michel Syska.

Around 20,000 lines of code, developed in Java, and licensed under LGPL. See http://grph.inria.fr.

The objective of GRPH is to provide researchers and engineers a suitable graph library for graph algorithms experimentation and network simulation. GRPH is primarily a software library, but it also comes with a set of executable files for user interaction and graph format conversion; as such, it can be used autonomously. Performance and accessibility are the primary targets of the GRPH library. It allows manipulating large graphs (millions of nodes). Its model considers mixed graphs composed of directed and undirected simple- and hyper-edges. GRPH comes with a collection of graph algorithms which is regularly augmented.

GRPH includes bridges to other graph libraries such as JUNG, JGraphT, CORESE (a software developed by the WIMMICS team Inria-I3S), LAD (Christine Solnon, LIRIS), Nauty (Brendan D. McKay), SageMath, as well as specific algorithms developed by Matthieu Latapy and Jean-Loup Guillaume (LIP6), etc.

In 2013, we have added several graph algorithms to GRPH (e.g., subgraph isomorphism, subgraph search as sets or regular expressions, transitive closure, etc.). In particular, a significant effort has been put on the support for paths with multiple data-structures for more efficient in-memory representation of paths, and the implementation of algorithms for the enumeration of paths, the characterization of paths, the computation of the k-shortest paths, etc. Furthermore, we have improved the support of weigths in graphs and developed software bridges to SageMath and OGDF. We have also added several models (link-failures, node mobility) for graph dynamics using the discrete-event simulator included in GRPH, as well as models for the development of decentralized algorithms (useful for instance for the simulation of routing schemes). Finally, we have redesigned the website which now includes a forum gathering the community of users.

5.2. SageMath

Participant: David Coudert.

Sagemath is a free open-source mathematics software aiming at becoming an alternative to Maple and Matlab. Initially created by William Stein (Professor of mathematics at Washington University), Sagemath is currently developed by more than 180 contributors around the world (mostly researchers) and its source code, developed in Python, Cython, and C++, has reached 350 MB.

It is of interest for COATI members because it combines a large collection of graph algorithms with various libraries in algebra, calculus, combinatorics, linear programming, statistics, etc. We use SageMath for quickly testing algorithms, analyzing graphs, and disseminating algorithms. We also use it for teaching purposes in the Master 2 IFI, stream UBINET.

In 2013, David Coudert has contributed to the development of the SageMath releases 5.0 to 6.0 with 10 patches (from bug fix to advanced graph algorithms) and participated to the reviewing process of more than 20 patches that are now part of the standard distribution.

5.3. DRMSim

Participants: David Coudert, Luc Hogie [correspondant], Aurélien Lancin, Nicolas Nisse, Issam Tahiri.

Around 45,000 lines, developed in Java, collaboration between COATI, LaBRI, and Alcatel-Lucent Bell labs.

DRMSim relies on a discrete-event simulation engine aiming at enabling the large-scale simulations of routing models. DRMSim is developped in the framework of the FP7 EULER project. It proposes a general routing model which accommodates any network configuration. Aside to this, it includes specific models for Generalized Linear Preference (GLP), and k-chordal network topologies, as well as implementations of routing protocols, including a previously defined routing protocol and lightweight versions of BGP (Border Gateway Protocol).

The metric model takes measures along a discrete-event simulation which can be performed in many ways.

Commonly, a simulation campaign consists in iterating over the set of combinations of parameter values, calling the simulation function for every combination. These combinations are most often complex, impeding their description by a set of mathematical functions. Thus DRMSim provides a simulation methodology that describes (programmatically) the way a simulation campaign should be conducted.

DRMSim stores on disk every step of the execution of a simulation campaign. In a simulation campaign, simulation runs are independent (no simulation depends on the result computed by another simulation). Consequently they can be executed in parallel. Because one simulation is most likely to use large amount of memory and to be multi-threaded, parallelizing the simulation campaign on one single computer is a poor parallelization scheme. Instead, we currently work at enabling the remote parallel execution of several simulation runs, with the same distribution framework that is used in the GRPH library.

DRMSim relies on the Mascsim abstract discrete-event simulation framework, the GRPH library and the Java4Unix integration framework.

In 2013, the work on DRMSim consisted (1) in the implementation of a full support for dynamic networks, including topological modifications and evolving transfer loads in the simulated network. The implementation of the BGP protocol was updated so as to support these dynamic properties. (2) This implementation of BGP was also augmented with a framework enabling its dynamic profiling. (3) Finally DRMSim does no longer relies on the Dipergrafs library. Instead it now uses GRPH, which brings better performance, stability and a broader set of graph algorithms.

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

5.4. Utilities

5.4.1. P2PVSim

Participant: Remigiusz Modrzejewski [correspondant].

Around 12,000 lines, developed in Python.

P2PVSim is a discrete-event simulator created for analyzing theoretical properties of peer-to-peer live video streaming algorithms. Implemented in Python it was designed with clarity and extensibility in mind from the beginning. It is capable of simulating overlays of a few thousands of peers. Multiple control protocols have been implemented. At the same time, a lot of work was put into the performance and scalability aspects of the software. Currently it is meant for simulating overlays of a few thousand peers running multiple control protocols that have been implemented. And in 2012, a distributed version of P2PVSim was developed running on an arbitrary number of computers. It has been so far used with success on a dozen computers with multiple cores all located in the same LAN.

5.4.2. Papareto

Participant: Luc Hogie [correspondant].

About 500 lines, developed in Java.

PAPARETO is a Java framework for the development of evolutionary solutions to computational problems. The primary motivation for developing an evolutionary framework was to give the GRPH library the ability to generate particular graph instances. Papareto differs from other evolutionary frameworks (ECJ, WatchMaker, JGAP, etc) in the following ways:

- it is *multi-objective*;
- it is *not a genetic algorithms (GAs)* framework because it manipulates objects *as is*. It does not consider their chromosomes representation. Performance consequently is no longer impacted by the computational cost of encoding/decoding;
- it *parallelizes* the creation and the evaluation of a new generation, adaptively to the evolving load of the computer;
- it is *self-adaptive* in the sense that it dynamically evaluates the performance of the crossover and mutation operators, then gives greater priority to most efficient ones;
- it is *easy to use*, by exposing the cleanest and more natural API possible and the minimal set of functionality enabling researchers and engineers to perform evolutionary computing;

See also the webpage http://www.i3s.unice.fr/~hogie/papareto/.

5.4.3. Toools

Participants: Luc Hogie [correspondant], Aurélien Lancin.

Around 3,000 lines, developed in Java.

TOOOLS is a general purpose Java toolbox which, much like Google Guava and Apache Commons, aims at providing classes useful in daily programming tasks. It focuses on the following topics:

- runtime (threads, control of parallel executions of SIMD code, execution of external processes, management of I/O operations, piping);
- input/output files (a complete, easier to use and more complete new model for files on disk is provided) and streams;
- reflection, including dynamic loading of classes, classpath management, Java beans, and access to the source code at runtime;
- application configuration files (parsing, querying, saving);
- plain text, XML;
- collections, including Java collection utilities and efficient sets of primitive integers;
- mathematical and statistical operations.

See also the webpage http://www-sop.inria.fr/members/Luc.Hogie/toools/.

5.4.4. Other software

We ensure the maintenance of various tools developped in the past:

Java4unix	a software glue for the integration of Java applications intro the UNIX environment; http://www-sop.inria.fr/members/Luc.Hogie/java4unix/;
Jalinopt	a Java toolkit for linear optimization; http://www-sop.inria.fr/members/Luc.Hogie/jalinopt/;
JavaFarm	a minimal middleware infrastructure for practical distributed computing; see http://www-sop.inria.fr/members/Luc.Hogie/javafarm/;
Macsim	a discrete event simulation engine use in the DRMSIM routing model simulator; http://www-sop.inria.fr/mascotte/software/mascsim/;
Jaseto	a Java toolkit for the XML (de)serialization of Java objects; http://www-sop.inria.fr/members/Luc.Hogie/jaseto/;

DANTE Team

5. Software and Platforms

5.1. Sensor Network Tools: drivers, OS and more

Participants: Éric Fleury [correspondant], Sandrine Avakian.

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

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

5.2. Queueing Systems

Participant: Thomas Begin [correspondant].

Queueing models, steady-state solution, online tool, web interface Online tool: http://queueing-systems.ens-lyon.fr

This tool aims at providing an ergonomic web-based interface to promote the use of our proposed solutions to numerically solve classical queueing systems. This tool was launched in 2011 and presented at the conference [30]. It attracts each month hundreds of visitors scattered accross the world. Its initial implementation only includes the solution for a queue with multiple servers, general arrivals, exponential services and a possibly finite buffer (*i.e.*, $Ph/M/c/N$-like queue). The steady-state solution to this queue is based on a simple and stable recurrence [31] and was performed in collaboration with Pr. Brandwajn (UCSC). Since then, we added new models such that a mono server queue with Poisson arrivals, general services and a possibly finite buffer (*i.e.*, $M/Ph/1/N$-like queue). In 2013, we extended our tool so as to include the solution for a queue with multiple servers, general service times and Poisson arrivals (*i.e.*, $M/Ph/c/N$-like queue). The solution is based on a new approximation that we developed this year in collaboration with Pr. Brandwajn (UCSC) [32]. Associated URL is: http://queueing-systems.ens-lyon.fr

DIANA Team

5. Software and Platforms

5.1. FIT platform

We have started, since 2011, the procedure of building a new experimental platform at Sophia-Antipolis, in the context of the FIT Equipment of Excellence project. This platform has two main goals : the first one is to enable highly controllable experiments due to its anechoic environment. These experiments can be either hybrid-experiments (as NEPI will be deployed, see section 5.4) or federated experiments through several testbeds. The second goal is to make resource consuming experiments (like CCNx) possible due to some powerful servers that will be installed and connected to the PlanetLab testbed. During 2013, a first call for bids has been made during March/April and has been unfortunately declared unsuccessful due to an overestimation of the building's price. As some premises became vacant at the same time, a second call for bids has been launched during September/October. This latter was a success because three interesting offers have been received and negotiated during the end of the year. The notification is planned for the 14th of January 2014.

5.2. ns-3

Participants: Thierry Turletti [correspondant], Daniel Camara, Walid Dabbous.

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

- Version: ns-3.19
- Keywords: networking event-driven simulation
- License: GPL (GPLv2)
- Type of human computer interaction: programmation C++/python, No GUI
- OS/Middleware: Linux, cygwin, osX
- Required library or software: standard C++ library: GPLv2
- Programming language: C++, python
- Documentation: doxygen

5.3. DCE

Participants: Emilio Mancini [correspondant], Daniel Camara, Walid Dabbous, Thierry Turletti.

Direct Code Execution (DCE) enables developers and researchers to develop their protocols and applications in a fully controllable and deterministic environment, where tests can be repeated with reproducible results. It allows unmodified protocol implementations and application code to be tested over large and possibly complex network topologies through the ns-3 discrete-event network simulator. The single-process model used in the DCE virtualization core brings key features, such as the possibility to easily debug a distributed system over multiple simulated nodes without the need of a distributed and complex debugger. Examples of tested applications over DCE include Quagga, iperf, torrent, thttpd, CCNx and various Linux kernel versions (from 2.6.36 to 3.12 versions).

DCE is free software, licensed under the GNU GPLv2 license, and is publicly available for research, development, and use.

See also the web page https://www.nsnam.org/overview/projects/direct-code-execution/

- Version: DCE-1.2
- Keywords: emulation, virtualization, networking event-driven simulation
- License: GPL (GPLv2)
- Type of human computer interaction: programmation C/C++, No GUI
- OS/Middleware: Linux
- Required library or software: standard C++ library: GPLv2
- Programming language: C++, python
- Documentation: doxygen

5.4. NEPI

Participants: Thierry Turletti [correspondant], Alina Quereilhac, Julien Tribino, Lucia Guevgeozian Odizzio.

NEPI stands for Network Experimentation Programming Interface. NEPI is a generic framework to manage network experiments, which allows to describe experiments in a simple way and automates experiment execution and result collection in a variety of experimentation environments, including simulators and live testbeds. NEPI was designed to be extensible for potentially any experimentation environment, which can be done by extending its well defined description and execution API's.

During the year 2013 we fully re-implemented NEPI in order to support new important requirements that arose from our participation in the FED4fire and OpenLab European projects. Among those requirements we can enumerate the ability to dynamically provision experiment resources during the experiment run time (e.i. Add new resources to the experiment at any moment), the interactive experiment execution (i.e. NEPI can be used as an interactive experiment management tool), and the description of experiment work-flows (e.i. Allow to include execution/deployment dependencies across resources).

This re-implementation of NEPI gave place to an improved, more extensible and user friendly framework which was officially released as NEPI 3.0 in December 2013. This new version of NEPI was shipped with support for new testbeds, including any testbeds supporting SSH key authentication and OMF (cOntrol and Management Framework) testbeds. Support for OMF technology is a very important features since it is a key part of the federation control management framework proposed in the FED4Fire project. The version of OMF currently supported by NEPI is 5.4. OMF version 6.0, which is the new mainstream release, will be supported during 2014. In order to comply with the requirements of FED4Fire for a federation framework, work is undergo in NEPI to fully support SFA (Slice Federation Architecture), in its latest version, for resource discovery and provisioning across federated testbeds.

Additional improvements to the NEPI framework during 2013 include out-of-the box support for Future Internet technologies such as CCN and OpenFlow for certain testbeds.

Finally, during 2013 a new NEPI web site was released, including an improved look&feel (http://nepi.inria.fr), user manual and code reference pages, detailed experiment examples and a issue tracking page.

- Version: 3.0
- ACM: C.2.2, C.2.4
- Keywords: networking experimentation
- License: GPL (3)
- Type of human computer interaction: python library, QT GUI
- OS/Middelware: Linux
- Required library or software: python – http://www.python.org – http://rpyc.sourceforge.net
- Programming language: python

5.5. Bake

Participants: Daniel Camara [correspondant], Walid Dabbous, Thierry Turletti.

Bake is an integration tool, which can be used by software developers to automate the reproducible build of a number of projects which depend on each other and which might be developed, and hosted by unrelated parties. Bake was developed to automate the reproducible build of ns-3 taking into account the particular needs of it. However, Bake is not specific for ns-3, it can be used by any open source project composed of a number of interdependent sub projects and that needs to simplify and automatize the assembly of these pieces of software in a coherent and useful way.

Bake is free software, licensed under the GNU GPLv2 license, and is publicly available for research, development, and use.

See also the web page http://planete.inria.fr/software/bake/index.html

- Version: Bake-0.1
- Keywords: Integration tool, distributed project, build and installation version control
- License: GPL (GPLv2)
- Type of human computer interaction: command line tool, No GUI
- OS/Middleware: Linux, Mac Os
- Required library or software: Python, GNU C++, Mercurial, CVS, GIT, Bazaar, Tar, Unzip, Unrar, 7z, XZ, Make, cMake, patch, autoreconf
- Programming language: python
- Documentation: doxygen

5.6. Com4Innov Network Testing Platform

Participants: Emilio Mancini [correspondant], Walid Dabbous.

Developed for the Com4Innov project, the platform integrates a set of tools in a virtual appliance, in order to conduct network experiments, store and share their results, and collect network diagnostic information. The platform's core is developed as a Web Application in the Apache Tomcat application server. It has a web user interface, and a public API accessible to external tools. Its architecture is designed to be integrated with SQL or NoSQL databases, and with HTTP or REST client. The core platform is completed by measurement tools and mobile phone clients.

The implementation is available at the url: http://gforge.inria.fr/projects/com4innov. It is currently in advanced development stage.

- Version: 0.5
- Keywords: network, diagnostic
- License: not yet public
- Type of human computer interaction: Web Interface
- OS/Middelware: Linux
- Required library or software: Apache Tomcat Java, Linux Distribution and standard development environment, GPL.
- Programming languages: Java, C++, C#
- Documentation: latex, javadoc

5.7. Mobile Devicess Network Analyzer Tool

Participants: Emilio Mancini [correspondant], Arnaud Legout.

This tool has been developed as a client for the Com4Innov Network Testing Platform, but it evolved in an autonomous one. It samples the signal's strength of wireless networks, correlating it with the GPS position, and then produces coverage maps. Once the sampling is done, it allows the user to upload such data, completed with diagnostic information to the platform.

The implementation is available at the url: http://gforge.inria.fr/projects/com4innov.

- Version: 1.0
- Keywords: network, diagnostic
- License: not yet public
- Type of human computer interaction: Android Application
- OS/Middelware: Android
- Required library or software: Android, GPL.
- Programming languages: Java
- Documentation: latex, javadoc

5.8. SfaWrap

Participants: Thierry Parmentelat [correspondant], Mohamed Larabi.

The SfaWrap is a reference implementation of the Slice-based Federation Architecture (SFA), the emerging standard for networking experimental testbed federation. We are codeveloping the SfaWrap with Princeton University, and during 2013, we have focused on:

- Implementing the Aggregate Manager (AM) API v3.
- Implementing a compatibility layer between AM API v2 and AM API v3.
- Supporting testbed providers in exposing their testbeds through SFA in order to enlarge the federation of testbeds.
- Maintaning the codebase of SFAWrap.
- Releasing SFAWrap software packages for the latest versions of different Linux Distributions (Namely: Fedora, Debian and Ubuntu).
- Version: sfa-3.1-1, myplc-5.3.1
- Keywords: networking testbed federation
- License: Various Open Source Licenses
- Type of human computer interaction: Web-UI, XMLRPC-based API, Qt-based graphical client
- OS/Middelware: Linux
- Required library or software: python2.7 or superior
- Programming languages: python
- Documentation: http://svn.planet-lab.org/#SFAUser-leveldocumentation
- Codebase: http://git.onelab.eu/?p=sfa.git;a=summary

5.9. Experimentation Software

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

ACQUA ACQUA is an Application for Collaborative Estimation of QUality of Internet Access. It was supported by the French ANR CMON project on collaborative monitoring. ACQUA is based on the principle of active measurements to a predefined set of landmarks and on the estimation of the Internet access quality by correlating these measurements together. This correlation will point to the importance of observed problems and to estimates to the quality the end user should expect from its access when running his applications over the Internet (one can see the measurements to the landmarks as samples of the global set of possible paths). In its first version (the version available online), ACQUA was concentring on delay measurements at the access and on the detection and estimation of the impact of delay anomalies (local problems, far away problems, etc). The current work is concentring on using the ACQUA principle in the estimation and prediction of the quality of experience of main applications. More details and code can be found at http://planete.inria.fr/acqua/.

ElectroSmart We are developing the application ElectroSmart as part of the Inria ADT ElectroSmart. The ElectroSmart application will enable crowd sourcing of electromagnetic exposures based on the electromagnetic radiations measured by smartphones.

DIONYSOS Project-Team

5. Software and Platforms

5.1. T3devKit testing toolkit and IPv6 test suites

Participants: César Viho, Anthony Baire.

We have built a toolkit for easing executing tests written in the standardized TTCN-3 test specification language. This toolkit is made of a `C++` library together with a highly customizable CoDec generator that allows fast development of external components (that are required to execute a test suite) such as CoDec (for message Coding/Decoding), System and Platform Adapters. It also provides a framework for representing and manipulating TTCN-3 events so as to ease the production of test reports. The toolkit addresses issues that are not yet covered by ETSI standards while being fully compatible with the existing standard interfaces: TRI (Test Runtime Interfaces) and TCI (Test Control Interfaces), it has been tested with four TTCN-3 environments (IBM, Elvior, Danet and Go4IT) and on three different platforms (Linux, Windows and Cygwin). It is publicly released under the CeCILL-C License.

All these tools with associated test suites (for RIPng, DHCPv6 and examples for DNS) are freely available at http://www.irisa.fr/tipi.

5.2. Interoperability Assessment

Participants: César Viho, Anthony Baire.

Our experience in interoperability assessment (since 1996) and in using the TTCN-3 standard allowed us to develop a tool (called `ttproto`) that helps in: (i) experimenting new concepts for long term evolution of the TTCN-3 standard and (ii) facilitating new approaches and methods for interoperability assessment. For instance, new passive approaches that we developed have been implemented and validated using `ttproto`. This tool `ttproto` has been used to develop test suites for 6LoWPAN-ND (IPv6 for Low Power Networks) and CoAP (Constrained Application Protocol). The CoAP test suites have been successfully used for two Plugtest interoperability events organized by ETSI, IPSO Alliance and the FP7 PROBE-IT project. The tool `ttproto` and the test suites indicated above are freely available at http://www.irisa.fr/tipi.

5.3. Performance and dependability evaluation

Participants: Gerardo Rubino, Bruno Sericola, Bruno Tuffin.

We develop software tools for the evaluation of two classes of models: Markov models and reliability networks. The main objective is to quantify dependability aspects of the behaviors of the modeled systems, but other aspects of the systems can be handled (performance, performability, vulnerability). The tools are specialized libraries implementing numerical, Monte Carlo and Quasi-Monte Carlo algorithms.

One of these libraries has been developed for the Celar (DGA), and its goal is the evaluation of dependability and vulnerability metrics of wide area communication networks (WANs). The algorithms in this library can also evaluate the sensitivities of the implemented dependability measures with respect to the parameters characterizing the behavior of the components of the networks (nodes, lines).

We are also developing tools with the objective of building Markovian models and to compute bounds of asymptotic metrics such as the asymptotic availability of standard metrics of models in equilibrium, loss probabilities, blocking probabilities, mean backlogs, etc. A set of functions designed for dependability analysis is being built under the name `DependLib`.

DYOGENE Project-Team

5. Software and Platforms

5.1. SINR-based k-coverage probability in cellular networks

H. P. Keeler, "SINR-based k-coverage probability in cellular networks" MATLAB Central File Exchange, 2013.

Available at: http://www.mathworks.fr/matlabcentral/fileexchange/40087-sinr-based-k-coverage-probability-in-cellular-networks

The scripts calculate the SINR k-coverage probability in a single-tier cellular network using a method based on a homogeneous Poisson process model. Details can be found in [20] which presents the model that these scripts are based on.

FOCUS Project-Team

5. Software and Platforms

5.1. Jolie

Members of Focus have developed Jolie [8] (Java Orchestration Language Interpreter Engine, see http://www.jolie-lang.org/). Jolie is a service-oriented programming language. Jolie can be used to program services that interact over the Internet using different communication protocols. Differently from other Web Services programming languages such as WS-BPEL, Jolie is based on a user-friendly C/Java-like syntax (more readable than the verbose XML syntax of WS-BPEL) and, moreover, the language is equipped with a formal operational semantics. This language is used for the *proof of concepts* developed around Focus activities. For instance, contract theories can be exploited for checking the conformance of a Jolie program with respect to a given contract. A spin-off, called “Italiana Software”, has been launched around Jolie, its general aim is to transfer the expertise in formal methods for Web Services matured in the last few years onto Service Oriented Business Applications. The spin-off is a software producer and consulting company that offers service-oriented solutions (for instance, a "‘single sign-on” application) based on the Jolie language.

In 2013 the development of the Jolie language has mainly focused on tools for the programming environment and on the integration of the language with cloud infrastructure. More in detail, we have produced the following software.

- PaaSSOA. This is a prototype for the deployment of Jolie services in a cloud infrastructure. We have also worked on the integration of Jolie with the Drools engine. Drools is used in PaaSSOA for storing and managing monitor events from services.
- JEye. This is web GUI prototype for designing Jolie graphical workflows which can be deployed into PaaSSOA.
- WSOA. We have developed a SaaS (Software as a Service) layer for the publication of Jolie APIs using different formats (http, JSON, SOAP, XML, and so on). We have also enhanced some libraries for the integration between Jolie and GWT technology.

All the server side code of PaaSSOA, JEye and WSOA has been developed by using Jolie.

5.2. Others

Below we list some software that has been developed, or is under development, in Focus.

- *Deadlock analysis* We have prototyped a framework for statically detecting deadlocks in a concurrent object-oriented language with asynchronous method calls and cooperative scheduling of method activations (the language is inspired by the ABS language developed in the EU project HATS).

  Since this language features recursion and dynamic resource creation, deadlock detection is extremely complex and state-of-the-art solutions either give imprecise answers or do not scale.

  In order to augment precision and scalability we propose a modular framework that allows several techniques to be combined. The basic component of the framework is a front-end inference algorithm that extracts abstract behavioral descriptions of methods, called contracts, which retain resource dependency information. Then this algorithm is integrated with a number of possible different back-ends that analyze contracts and derive deadlock information. We have prototyped two such back-ends:

  1. an evaluator that computes a fixpoint semantics, and
  2. an evaluator using abstract model checking.

  The evaluator (1) is available at http://www.cs.unibo.it/~laneve/deadlock/index.html

The evaluator (2) is available at http://www.cs.unibo.it/~giachino/siteDat/index.php

- *CaReDeb* (http://proton.inrialpes.fr/~mezzina/deb/).

  Reversible debugging provides developers with a way to execute their applications both forward and backward, seeking the cause of an unexpected or undesired event. We have developed CaReDeb, the first prototype of a causal-consistent reversible debugger. Causal consistent here means that independent actions are undone independently, while dependent actions are undone in reverse order. This allows the programmer to concentrate on the threads responsible of the bug, independently of the actual interleaving. CaReDeb provides primitives that given a misbehavior, e.g., a variable has not the expected value, allow one to go back to the action responsible for it, e.g., the one that assigned the wrong value to the variable. Notably, the programmer has no need to know which thread the action belongs to, since this is found automatically by the debugger. The procedure can be iterated till the bug is found. CaReDeb targets a fragment of the language Oz, which is at the basis of Mozart. The considered fragment provides functional variables, procedures, threads, and asynchronous communication via ports.

- *AIOCJ* (http://www.cs.unibo.it/projects/jolie/aiocj.html).

  AIOCJ is a framework for programming adaptive distributed systems based on message passing. AIOCJ comes as a plugin for Eclipse, AIOCJ-ecl, allowing to edit descriptions of distributed systems as adaptive interaction-oriented choreographies (AIOC). From interaction-oriented choreographies the description of single participants can be automatically derived. Adaptation is specified by rules allowing to replace predetermined parts of the AIOC with a new behaviour. A suitable protocol ensures that all the participants are updated in a coordinated way. As a result, the distributed system follows the specification given by the AIOC under all changing sets of adaptation rules and environment conditions. In particular, the system is always deadlock-free.

- *METIS* (https://github.com/aeolus-project/metis)

  As partners of the Aeolus project we have developed a tool for the automatic synthesis of deployment plans. A deployment plan is a sequence of actions that, when performed, allows the deployment of a given configuration of components. METIS (Modern Engineered Tool for Installing Software systems) is a tool that enables one to automatically generate a deployment plan, starting from a description of the configuration following the Aeolus model. The software is open source. It is written entirely in OCaml and is about 3.5K lines of source code. The tool is based on theoretical results that guarantee its soundness and completeness, while maintaining polynomial computational complexity. Experimental results are encouraging as METIS looks quite effective in practice by handling problem instances with hundreds of components in less than a minute. This is a key ingredient in the solution to the automation problem addressed by the Aeolus project. The paper [48] is dedicated to the description of the tool, while [41] addresses the formal aspects of the technique.

The sofware below have not undergone substantial modifications during 2013

- *Croll-pi Interpreter* (http://proton.inrialpes.fr/~mlienhar/croll-pi/implem/).

  Croll-pi is a concurrent reversible language featuring a rollback operator to undo a past action (together with all the actions depending on it), and a compensation mechanism to avoid cycling by redoing the same action again and again. We have developed an interpreter for croll-pi using Maude.

  We used the interpreter to test the expressive power of croll-pi on various problems, including the 8-queen problem, error handling in an automotive scenario from the EU project Sensoria, and constructs for distributed error handling such as stabilizers.

- *IntML* is a functional programming language guaranteeing sublinear space bounds for all programs [50]. See the Activity Reports of previous years (in particular 2010) for more details.

- *Lideal* (http://lideal.cs.unibo.it/) is an experimental tool implementing type inference for dependently linear type systems. The tool reduces the problem of evaluating the complexity of PCF (i.e.

functional programs with primitive integers and recursive definitions) to checking a set of first-order inequalities for validity. The latter can then be handled through SMT solvers or put in a form suitable for managing them with tools such as CoQ.

FUN Project-Team

4. Software and Platforms

4.1. Distributed ONS

Participants: Nathalie Mitton, Roberto Quilez [correspondant].

This module implements a DHT-based Distributed EPC Global ONS issued from the ANR WINGS project and published in [61]. APP number: IDDN.FR.001.180033.000.S.P.2012.000.10000.

- Version: version 1

4.2. GOLIATH 1.0

Participants: Fadila Khadar [correspondant], Nathalie Mitton.

GOLIATH (Generic Optimized LIghtweight communication stack for Ambient TecHnologies) is a full protocol stack for wireless sensor networks.

See also the web page https://gforge.inria.fr/projects/goliath/.

4.3. Linear variable energy module for WSNET.

Participants: Tony Ducrocq [correspondant], Nathalie Mitton.

This module is to be integrated in the WSNET event-based simulator for wireless networks. It implements a Linear transmission variable energy module for WSNET.

- Version: 1.0

4.4. New ALE module for ASPIRERFID middleware.

Participants: Rim Driss [correspondant], Nathalie Mitton, Ibrahim Amadou, Julien Vandaele.

AspireRFID middleware is a modular OW2 open source RFID middleware. It is compliant with EPC Global standards. This new module integrates the modifications of the new standard release, including new RP and LLRP definitions and fixing bugs.

- Version: 1.0

GANG Project-Team (section vide)

GRAND-LARGE Project-Team

4. Software and Platforms

4.1. APMC-CA

Participants: Sylvain Peyronnet [correspondant], Joel Falcou, Pierre Esterie, Khaled Hamidouche, Alexandre Borghi.

The APMC model checker implements the state-of-the-art approximate probabilistic model checking methods. Last year we develop a version of the tool dedicated to the CELL architecture. Clearly, it was very pedagogic, but the conclusion is that the CELL is not adapted to sampling based verification methods.

This year we develop, thanks to the BSP++ framework, a version compatible with SPM/multicores machines, clusters and hybrid architectures. This version outperforms all previous ones, thus showing the interest of both these new architectures and of the BSP++ framework.

4.2. YML

Participants: Serge Petiton [correspondant], Nahid Emad, Maxime Hugues.

Scientific end-users face difficulties to program P2P large scale applications using low level languages and middleware. We provide a high level language and a set of tools designed to develop and execute large coarse grain applications on peer-to-peer systems. Thus, we introduced, developed and experimented the YML for parallel programming on P2P architectures. This work was done in collaboration with the PRiSM laboratory (team of Nahid Emad).

The main contribution of YML is its high level language for scientific end-users to develop parallel programs for P2P platforms. This language integrates two different aspects. The first aspect is a component description language. The second aspect allows to link components together. A coordination language called YvetteML can express graphs of components which represent applications for peer-to-peer systems.

Moreover, we designed a framework to take advantage of the YML language. It is based on two component catalogues and an YML engine. The first one concerns end-user's components and the second one is related to middleware criteria. This separation enhances portability of applications and permits real time optimizations. Currently we provide support for the XtremWeb Peer-to-Peer middleware and the OmniRPC grid system. The support for Condor is currently under development and a beta-release will be delivered soon (in this release, we plan to propagate semantic data from the end-users to the middleware). The next development of YML concerns the implementation of a multi-backend scheduler. Therefore, YML will be able to schedule at runtime computing tasks to any global computing platform using any of the targeted middleware.

We experimented YML with basic linear algebra methods on a XtremWeb P2P platform deployed between France and Japan. Recently, we have implemented complex iterative restarted Krylov methods, such as Lanczos-Bisection, GMRES and MERAM methods, using YML with the OmniRPC back-end. The experiments are performed either on the Grid5000 testbed of on a Network of Workstations deployed between Lille, Versailles and Tsukuba in Japan. Demos was proposed on these testbeds from conferences in USA. We recently finished evaluations of the overhead generated using YML, without smart schedulers and with extrapolations due to the lack of smart scheduling strategies inside targeted middleware.

In the context of the FP3C project funded by ANR-JST, we have recently extended YML to support a directive distributed parallel language, XcalableMP http://www.xcalablemp.org/. This extension is based on the support of the XcalableMP language inside YML components. This allows to develop parallel programs with two programming paradigm and thus two parallelism levels. This work is a part of the project that targets post-Petascale supercomputer that would be composed of heterogeneous and massively parallel hardware.

The software is available at http://yml.prism.uvsq.fr/

4.3. The Scientific Programming InterNet (SPIN)

Participant: Serge Petiton [correspondant].

SPIN (Scientific Programming on the InterNet), is a scalable, integrated and interactive set of tools for scientific computations on distributed and heterogeneous environments. These tools create a collaborative environment allowing the access to remote resources.

The goal of SPIN is to provide the following advantages: Platform independence, Flexible parameterization, Incremental capacity growth, Portability and interoperability, and Web integration. The need to develop a tool such as SPIN was recognized by the GRID community of the researchers in scientific domains, such as linear algebra. Since the P2P arrives as a new programming paradigm, the end-users need to have such tools. It becomes a real need for the scientific community to make possible the development of scientific applications assembling basic components hiding the architecture and the middleware. Another use of SPIN consists in allowing to build an application from predefined components ("building blocks") existing in the system or developed by the developer. The SPIN users community can collaborate in order to make more and more predefined components available to be shared via the Internet in order to develop new more specialized components or new applications combining existing and new components thanks to the SPIN user interface.

SPIN was launched at ASCI CNRS lab in 1998 and is now developed in collaboration with the University of Versailles, PRiSM lab. SPIN is currently under adaptation to incorporate YML, cf. above. Nevertheless, we study another solution based on the Linear Algebra KErnel (LAKE), developed by the Nahid Emad team at the University of Versailles, which would be an alternative to SPIN as a component oriented integration with YML.

4.4. V-DS

Participant: Franck Cappello [correspondant].

This project started officially in September 2004, under the name V-Grid. V-DS stands for Virtualization environment for large-scale Distributed Systems. It is a virtualization software for large scale distributed system emulation. This software allows folding a distributed systems 100 or 1000 times larger than the experimental testbed. V-DS virtualizes distributed systems nodes on PC clusters, providing every virtual node its proper and confined operating system and execution environment. Thus compared to large scale distributed system simulators or emulators (like MicroGrid), V-DS virtualizes and schedules a full software environment for every distributed system node. V-DS research concerns emulation realism and performance.

A first work concerns the definition and implementation of metrics and methodologies to compare the merits of distributed system virtualization tools. Since there is no previous work in this domain, it is important to define what and how to measure in order to qualify a virtualization system relatively to realism and performance. We defined a set of metrics and methodologies in order to evaluate and compared virtualization tools for sequential system. For example a key parameter for the realism is the event timing: in the emulated environment, events should occur with a time consistent with a real environment. An example of key parameter for the performance is the linearity. The performance degradation for every virtual machine should evolve linearly with the increase of the number of virtual machines. We conducted a large set of experiments, comparing several virtualization tools including Vserver, VMware, User Mode Linux, Xen, etc. The result demonstrates that none of them provides both enough isolation and performance. As a consequence, we are currently studying approaches to cope with these limits.

We have made a virtual platform on the GDX cluster with the Vserver virtualization tool. On this platform, we have launched more than 20K virtual machines (VM) with a folding of 100 (100 VM on each physical machine). However, some recent experiments have shown that a too high folding factor may cause a too long execution time because of some problems like swapping. Currently, we are conducting experiments on another platform based on the virtualization tool named Xen which has been strongly improved since 2 years. We expect to get better result with Xen than with Vserver. Recently, we have been using the V-DS version based on Xen to evaluate at large scales three P2P middleware [74].

This software is available at http://v-ds.lri.fr/

4.5. PVC: Private Virtual Cluster

Participant: Franck Cappello [correspondant].

Current complexity of Grid technologies, the lack of security of Peer-to-Peer systems and the rigidity of VPN technologies make sharing resources belonging to different institutions still technically difficult.

We propose a new approach called "Instant Grid" (IG), which combines various Grid, P2P and VPN approaches, allowing simple deployment of applications over different administration domains. Three main requirements should be fulfilled to make Instant Grids realistic: simple networking configuration (Firewall and NAT), no degradation of resource security, no need to re-implement existing distributed applications.

Private Virtual Cluster, is a low-level middle-ware that meets Instant Grid requirements. PVC turns dynamically a set of resources belonging to different administration domains into a virtual cluster where existing cluster runtime environments and applications can be run. The major objective of PVC is to establish direct connections between distributed peers. To connect firewall protected nodes in the current implementation, we have integrated three techniques: UPnP, TCP/UDP Hole Punching and a novel technique Traversing-TCP.

One of the major application of PVC is the third generation desktop Grid middleware. Unlike BOINC and XtremWeb (which belong to the second generation of desktop Grid middleware), PVC allows the users to build their Desktop Grid environment and run their favorite batch scheduler, distributed file system, resource monitoring and parallel programming library and runtime software. PVC ensures the connectivity layer and provide a virtual IP network where the user can install and run existing cluster software.

By offering only the connectivity layer, PVC allows to deploy P2P systems with specific applications, like file sharing, distributed computing, distributed storage and archive, video broadcasting, etc.

4.6. OpenWP

Participant: Franck Cappello [correspondant].

Distributed applications can be programmed on the Grid using workflow languages, object oriented approaches (Proactive, IBIS, etc), RPC programming environments (Grid-RPC, DIET), component based environments (generally based on Corba) and parallel programming libraries like MPI.

For high performance computing applications, most of the existing codes are programmed in C, Fortran and Java. These codes have 100,000 to millions of lines. Programmers are not inclined to rewrite then in a "non standard" programming language, like UPC, CoArray Fortran or Global Array. Thus environments like MPI and OpenMPI remain popular even if they require hybrid approaches for programming hierarchical computing infrastructures like cluster of multi-processors equipped with multi-core processors.

Programming applications on the Grid add a novel level in the hierarchy by clustering the cluster of multi-processors. The programmer will face strong difficulties in adapting or programming a new application for these runtime infrastructures featuring a deep hierarchy. Directive based parallel and distributed computing is appealing to reduce the programming difficulty by allowing incremental parallelization and distribution. The programmer add directives on a sequential or parallel code and may check for every inserted directive its correction and performance improvement.

We believe that directive based parallel and distributed computing may play a significant role in the next years for programming High performance parallel computers and Grids. We have started the development of OpenWP. OpenWP is a directive based programming environment and runtime allowing expressing workflows to be executed on Grids. OpenWP is compliant with OpenMP and can be used in conjunction with OpenMP or hybrid parallel programs using MPI + OpenMP.

The OpenWP environment consists in a source to source compiler and a runtime. The OpenWP parser, interprets the user directives and extracts functional blocks from the code. These blocks are inserted in a library distributed on all computing nodes. In the original program, the functional blocks are replaced by RPC calls and calls to synchronization. During the execution, the main program launches non blocking RPC calls to functions on remote nodes and synchronize the execution of remote functions based on the synchronization directives inserted by the programmer in the main code. Compared to OpenMP, OpenWP does not consider a shared memory programming approach. Instead, the source to source compiler insert data movements calls in the main code. Since the data set can be large in Grid application, the OpenWP runtime organize the storage of data sets in a distributed way. Moreover, the parameters and results of RPC calls are passed by reference, using a DHT. Thus, during the execution, parameter and result references are stored in the DHT along with the current position of the datasets. When a remote function is called, the DHT is consulted to obtain the position of the parameter data sets in the system. When a remote function terminates its execution, it stores the result data sets and store a reference to the data set in the DHT.

We are evaluating OpenWP from an industrial application (Amibe), used by the European aerospace company EADS. Amibe is the mesher module of jCAE [1]. Amibe generates a mesh from a CAD geometry in three steps. It first creates edges between every patch of the CAD (mesh in one dimension), then generates a surface mesh for every unfolded patch (mesh in two dimensions) and finally adds the third dimension to the mesh by projecting the 2D mesh into the original CAD surfaces. The first and third operation cannot be distributed. However the second step can easily be distributed following a master/worker approach, transferring the mesh1d results to every computing node and launching the distributed execution of the patches.

4.7. OpenScop

Participant: Cédric Bastoul.

OpenScop is an open specification which defines a file format and a set of data structures to represent a *static control part* (SCoP for short), i.e., a program part that can be represented in the *polyhedral model*, an algebraic representation of programs used for automatic parallelization and optimization (used, e.g., in GNU GCC, LLVM, IBM XL or Reservoir Labs R-Stream compilers). The goal of OpenScop is to provide a common interface to various polyhedral compilation tools in order to simplify their interaction.

OpenScop provides a single format for tools that may have different purposes (e.g., as different as code generation and data dependence analysis). We could observe that most available polyhedral compilation tools during the last decade were manipulating the same kind of data (polyhedra, affine functions...) and were actually sharing a part of their input (e.g., iteration domains and context concepts are nearly everywhere). We could also observe that those tools may rely on different internal representations, mostly based on one of the major polyhedral libraries (e.g., Polylib, PPL or isl), and this representation may change over time (e.g., when switching to a more convenient polyhedral library). OpenScop aims at providing a stable, unified format that offers a durable guarantee that a tool can use an output or provide an input to another tool without breaking a compilation chain because of some internal changes in one element of this chain. The other promise of OpenScop is the ability to assemble or replace the basic blocks of a polyhedral compilation framework at no, or at least low engineering cost. The OpenScop Library (licensed under the 3-clause BSD license) has been developped as an example, yet powerful, implementation of the OpenScop specification.

4.8. Clay

Participant: Cédric Bastoul.

Clay is a free software and library devoted to semi-automatic optimization using the polyhedral model. It can input a high-level program or its polyhedral representation and transform it according to a transformation script. Classic loop transformations primitives are provided. Clay is able to check for the legality of the complete sequence of transformation and to suggest corrections to the user if the original semantics is not preserved (experimental at this document redaction time). Main authors include Joël Poudroux and Cédric Bastoul.

---

[1] project page: http://jcae.sourceforge.net

4.9. Fast linear system solvers in public domain libraries

Participant: Marc Baboulin [correspondant].

Hybrid multicore+GPU architectures are becoming commonly used systems in high performance computing simulations. In this research, we develop linear algebra solvers where we split the computation over multicore and graphics processors, and use particular techniques to reduce the amount of pivoting and communication between the hybrid components. This results in efficient algorithms that take advantage of each computational unit [16]. Our research in randomized algorithms yields to several contributions to propose public domain libraries PLASMA and MAGMA in the area of fast linear system solvers for general and symmetric indefinite systems. These solvers minimize communication by removing the overhead due to pivoting in LU and $LDLT$ factorization. Different approaches to reduce communication are compared in [2].

See also the web page http://icl.cs.utk.edu/magma/.

4.10. cTuning: Repository and Tools for Collective Characterization and Optimization of Computing Systems

Participant: Grigori Fursin [correspondant].

Designing, porting and optimizing applications for rapidly evolving computing systems is often complex, ad-hoc, repetitive, costly and error prone process due to an enormous number of available design and optimization choices combined with the complex interactions between all components. We attempt to solve this fundamental problem based on collective participation of users combined with empirical tuning and machine learning.

We developed cTuning framework that allows to continuously collect various knowledge about application characterization and optimization in the public repository at cTuning.org. With continuously increasing and systematized knowledge about behavior of computer systems, users should be able to obtain scientifically motivated advices about anomalies in the behavior of their applications and possible solutions to effectively balance performance and power consumption or other important characteristics.

Currently, we use cTuning repository to analyze and learn profitable optimizations for various programs, datasets and architectures using machine learning enabled compiler (MILEPOST GCC). Using collected knowledge, we can quickly suggest better optimizations for a previously unseen programs based on their semantic or dynamic features [8].

We believe that such approach will be vital for developing efficient Exascale computing systems. We are currently developing the new extensible cTuning2 framework for automatic performance and power tuning of HPC applications.

For more information, see the web page http://cTuning.org.

HIEPACS Project-Team

5. Software and Platforms

5.1. Introduction

We describe in this section the software that we are developing. The first list will be the main milestones of our project. The other software developments will be conducted in collaboration with academic partners or in collaboration with some industrial partners in the context of their private R&D or production activities. For all these software developments, we will use first the various (very) large parallel platforms available through GENCI in France (CCRT, CINES and IDRIS Computational Centers), and next the high-end parallel platforms that will be available via European and US initiatives or projects such that PRACE.

5.2. MaPHyS

Participant: Emmanuel Agullo [corresponding member].

`MaPHyS` (Massivelly Parallel Hybrid Solver) is a software package whose prototype was initially developed in the framework of a PhD thesis and further consolidated first thanks to the ANR-CIS Solstice funding and the Inria Parscali ADT. This parallel linear solver couples direct and iterative approaches. The underlying idea is to apply to general unstructured linear systems domain decomposition ideas developed for the solution of linear systems arising from PDEs. The interface problem, associated with the so called Schur complement system, is solved using a block preconditioner with overlap between the blocks that is referred to as Algebraic Additive Schwarz.

The `MaPHyS` package is very much a first outcome of the research activity described in Section 3.3 . Finally, `MaPHyS` is a preconditioner that can be used to speed-up the convergence of any Krylov subspace method. We forsee to either embed in `MaPHyS` some Krylov solvers or to release them as standalone packages, in particular for the block variants that will be some outcome of the studies discussed in Section 3.3 .

`MaPHyS` can be found at http://maphys.gforge.inria.fr.

5.3. PaStiX

Participant: Pierre Ramet [corresponding member].

Complete and incomplete supernodal sparse parallel factorizations.

`PaStiX` (Parallel Sparse matriX package) is a scientific library that provides a high performance parallel solver for very large sparse linear systems based on block direct and block ILU(k) iterative methods. Numerical algorithms are implemented in single or double precision (real or complex): LLt (Cholesky), LDLt (Crout) and LU with static pivoting (for non symmetric matrices having a symmetric pattern).

The `PaStiX` library uses the graph partitioning and sparse matrix block ordering package `Scotch`. `PaStiX` is based on an efficient static scheduling and memory manager, in order to solve 3D problems with more than 50 million of unknowns. The mapping and scheduling algorithm handles a combination of 1D and 2D block distributions. This algorithm computes an efficient static scheduling of the block computations for our supernodal parallel solver which uses a local aggregation of contribution blocks. This can be done by taking into account very precisely the computational costs of the BLAS 3 primitives, the communication costs and the cost of local aggregations. We also improved this static computation and communication scheduling algorithm to anticipate the sending of partially aggregated blocks, in order to free memory dynamically. By doing this, we are able to reduce the aggregated memory overhead, while keeping good performance.

Another important point is that our study is suitable for any heterogeneous parallel/distributed architecture when its performance is predictable, such as clusters of multicore nodes. In particular, we now offer a high performance version with a low memory overhead for multicore node architectures, which fully exploits the advantage of shared memory by using an hybrid MPI-thread implementation.

Direct methods are numerically robust methods, but the very large three dimensional problems may lead to systems that would require a huge amount of memory despite any memory optimization. A studied approach consists in defining an adaptive blockwise incomplete factorization that is much more accurate (and numerically more robust) than the scalar incomplete factorizations commonly used to precondition iterative solvers. Such incomplete factorization can take advantage of the latest breakthroughs in sparse direct methods and particularly should be very competitive in CPU time (effective power used from processors and good scalability) while avoiding the memory limitation encountered by direct methods.

`PaStiX` is publicly available at http://pastix.gforge.inria.fr under the Inria CeCILL licence.

5.4. HIPS

Participant: Pierre Ramet [corresponding member].

Multilevel method, domain decomposition, Schur complement, parallel iterative solver.

`HIPS` (Hierarchical Iterative Parallel Solver) is a scientific library that provides an efficient parallel iterative solver for very large sparse linear systems.

The key point of the methods implemented in `HIPS` is to define an ordering and a partition of the unknowns that relies on a form of nested dissection ordering in which cross points in the separators play a special role (Hierarchical Interface Decomposition ordering). The subgraphs obtained by nested dissection correspond to the unknowns that are eliminated using a direct method and the Schur complement system on the remaining of the unknowns (that correspond to the interface between the sub-graphs viewed as sub-domains) is solved using an iterative method (GMRES or Conjugate Gradient at the time being). This special ordering and partitioning allows for the use of dense block algorithms both in the direct and iterative part of the solver and provides a high degree of parallelism to these algorithms. The code provides a hybrid method which blends direct and iterative solvers. `HIPS` exploits the partitioning and multistage ILU techniques to enable a highly parallel scheme where several subdomains can be assigned to the same process. It also provides a scalar preconditioner based on the multistage ILUT factorization.

`HIPS` can be used as a standalone program that reads a sparse linear system from a file ; it also provides an interface to be called from any C, C++ or Fortran code. It handles symmetric, unsymmetric, real or complex matrices. Thus, `HIPS` is a software library that provides several methods to build an efficient preconditioner in almost all situations.

`HIPS` is publicly available at http://hips.gforge.inria.fr under the Inria CeCILL licence.

5.5. LBC2

Participant: Aurélien Esnard [corresponding member].

`LBC2` (Load-Balancing for Code Coupling) is a library providing different methods for partitioning or repartitioning graphs and hypergraphs. These methods are designed for effective communications during dynamic load balancing with a variable number of processors or coupled code interactions. Repartitioning is achieved with an enriched graph model partitioned using third-party partitioning tools, as `Scotch`, `PaToH`, `METIS`, `Zoltan` or `Mondrian`.

The framework is publicy available at Inria Gforge as a part of the MPICPL framework: http://mpicpl.gforge.inria.fr.

5.6. MPICPL

Participant: Aurélien Esnard [corresponding member].

MPICPL (MPI CouPLing) is a software library dedicated to the coupling of parallel legacy codes, that are based on the well-known MPI standard. It proposes a lightweight and comprehensive programing interface that simplifies the coupling of several MPI codes (2, 3 or more). MPICPL facilitates the deployment of these codes thanks to the *mpicplrun* tool and it interconnects them automatically through standard MPI inter-communicators. Moreover, it generates the universe communicator, that merges the world communicators of all coupled-codes. The coupling infrastructure is described by a simple XML file, that is just loaded by the *mpicplrun* tool.

MPICPL was developed by HIEPACS for the purpose of the ANR NOSSI. It uses advanced features of MPI2 standard. The framework is publicy available at Inria Gforge: http://mpicpl.gforge.inria.fr.

5.7. ScalFMM

Participant: Olivier Coulaud [corresponding member].

`ScalFMM` (Parallel Fast Multipole Library for Large Scale Simulations) is a software library to simulate N-body interactions using the Fast Multipole Method.

`ScalFMM` intends to offer all the functionalities needed to perform large parallel simulations while enabling an easy customization of the simulation components: kernels, particles and cells. It works in parallel in a shared/distributed memory model using OpenMP and MPI. The software architecture has been designed with two major objectives: being easy to maintain and easy to understand. There is two main parts: 1) the management of the octree and the parallelization of the method ; 2) the kernels. This new architecture allow us to easily add new FMM algorithm or kernels and new paradigm of parallelization. The code is extremely documented and the naming convention fully respected. Driven by its user-oriented philosophy, `ScalFMM` is using CMAKE as a compiler/installer tool. Even if `ScalFMM` is written in C++ it will support a C and fortran API soon.

The library offers two methods to compute interactions between bodies when the potential decays like $1/r$. The first method is the classical FMM based on spherical harmonic expansions and the second is the Black-Box method which is an independent kernel formulation (introduced by E. Darve at Stanford). With this method, we can now easily add new non oscillatory kernels in our library. For the classical method, two approaches are used to decrease the complexity of the operators. We consider either matrix formulation that allows us to use BLAS routines or rotation matrix to speed up the M2L operator.

The `ScalFMM` package is available at http://scalfmm.gforge.inria.fr

5.8. ViTE

Participant: Mathieu Faverge [corresponding member].

Visualization, Execution trace

`ViTE` is a trace explorer. It is a tool made to visualize execution traces of large parallel programs. It supports Pajé, a trace format created by Inria Grenoble, and OTF and OTF2 formats, developed by the University of Dresden and allows the programmer a simpler way to anlyse, debug and/or profile large parallel applications. It is an open source software licenced under CeCILL-A.

The `ViTE` software is available at http://vite.gforge.inria.fr and has been developed in collaboration with the Inria Bordeaux - Sud-Ouest SED team, Telecom SudParis and Inria Grenoble.

In the same context we also contribute to the EZtrace and GTG libraries in collaboration with F. Trahay from Telecom SudParis. EZTrace (http://eztrace.gforge.inria.fr) is a tool that aims at generating automatically execution trace from HPC programs. It generates execution trace files thanks to the GTG library (http://gtg.gforge.inria.fr) that can be later interpreted by visualization tools such as `ViTE`.

5.9. Other software

For the materials physics applications, a lot of development will be done in the context of ANR projects (NOSSI and OPTIDIS, see Section 4.1) in collaboration with LaBRI, CPMOH, IPREM, EPFL and with CEA Saclay and Bruyère-le-Châtel.

- **FAST**
  Participant: Olivier Coulaud [corresponding member].

  FAST is a linear response time dependent density functional program for computing the electronic absorption spectrum of molecular systems. It uses an O(N3) linear response method based on finite numerical atomic orbitals and deflation of linear dependence in atomic orbital product space. This version is designed to work with data produced by the SIESTA DFT code. The code produces as principal output a numerical absorption spectrum (complex part of the polarisability, loosely called the polarisability below) and a list of transition energies and oscillator strengths deduced from fitting Lorentzians to the numerical spectrum. Considering the absence of hybrid functionals in SIESTA and that concerning calculation of spectra, generalized gradient Hamiltonians are not usually considered to be notably better than the local density approximation, the present release of FAST works only with LDA, which despite its limitations, has provided useful results on the systems to which the present authors have applied it. The FAST library is available at http://people.bordeaux.inria.fr/coulaud/Softwares/FAST/index.html.

- **OptiDis**
  Participant: Olivier Coulaud [corresponding member].

  OptiDis is a new code for large scale dislocation dynamics simulations. its aim is to simulate real life dislocation densities (up until 5.10^{22} dislocations/m^{-2}) in order to understand plastic deformation and study strain hardening. The main application is to observe and understand plastic deformation on irradiated zirconium. Zirconium alloys is the first containment barrier against the dissemination of radioactive elements. More precisely, with neutron irradiated zirconium alloys we are talking of channeling mechanism, which mean to stick with the reality,more than tens of thousands of induced loops so 10^8 degrees of freedom in the simulation.

  The code is based on Numodis code developed at CEA Saclay and the ScalFMM library developed in our Inria project. The code is written in C++ language and using the last features of C++11. One of the main aspects is the hybrid parallelism MPI/OpenMP that gives the software the ability to scale on large cluster while the computation load rise. In order to achieve that, we use different level of parallelism. First of all, the simulation box is spread over MPI process, we then use a thinner level for threads, dividing the domain using an Octree representation. All theses parts are driven by the ScalFMM library. On the last level our data are stored in an adaptive structure absorbing dynamic of this kind of simulation and handling well task parallelism.

The two following packages are mainly designed and developed in the context of a US initiative led by ICL and to which we closely collaborate through the associate team MORSE.

- **PLASMA**
  Participant: Mathieu Faverge [corresponding member].

  The PLASMA (Parallel Linear Algebra for Scalable Multi-core Architectures) project aims to address the critical and highly disruptive situation that is facing the Linear Algebra and High Performance Computing community due to the introduction of multi-core architectures.

  The PLASMA ultimate goal is to create software frameworks that enable programmers to simplify the process of developing applications that can achieve both high performance and portability across a range of new architectures.

  The development of programming models that enforce asynchronous, out of order scheduling of operations is the concept used as the basis for the definition of a scalable yet highly efficient software framework for Computational Linear Algebra applications.

The `PLASMA` library is available at http://icl.cs.utk.edu/plasma.

- **PaRSEC/DPLASMA**
  Participant: Mathieu Faverge [corresponding member].

  `PaRSEC` Parallel Runtime Scheduling and Execution Controller, is a generic framework for architecture aware scheduling and management of micro-tasks on distributed many-core heterogeneous architectures. Applications we consider can be expressed as a Direct Acyclic Graph of tasks with labeled edges designating data dependencies. DAGs are represented in a compact problem-size independent format that can be queried on-demand to discover data dependencies in a totally distributed fashion. `PaRSEC` assigns computation threads to the cores, overlaps communications and computations and uses a dynamic, fully-distributed scheduler based on architectural features such as NUMA nodes and algorithmic features such as data reuse.

  The framework includes libraries, a runtime system, and development tools to help application developers tackle the difficult task of porting their applications to highly heterogeneous and diverse environment.

  `DPLASMA` (Distributed Parallel Linear Algebra Software for Multicore Architectures) is the leading implementation of a dense linear algebra package for distributed heterogeneous systems. It is designed to deliver sustained performance for distributed systems where each node featuring multiple sockets of multicore processors, and if available, accelerators like GPUs or Intel Xeon Phi. `DPLASMA` achieves this objective through the state of the art `PaRSEC` runtime, porting the `PLASMA` algorithms to the distributed memory realm.

  The `PaRSEC` runtime and the `DPLASMA` library are available at http://icl.cs.utk.edu/parsec.

HIPERCOM2 Team

5. Software and Platforms

5.1. Software and Platforms

5.1.1. Platforms

5.1.1.1. SensLab and FIT

Participants: Cédric Adjih, Alaeddine Weslati, Vincent Ladeveze.

This is a joint work with Emmanuel Baccelli from Inria Saclay.

Period: 2011 - 2021

Partners: Inria (Lille, Sophia-Antipolis, Grenoble), INSA, UPMC, Institut Télécom Paris, Institut Télécom Evry, LSIIT Strasbourg.

- Deployment: during the year 2013, most of the practical deployment has been planned, designed and realized. A location has been found for the new testbed of the EQUIPEX FIT: the basement of building 1 at Rocquencourt. The preparation of the deployment space, power, GPS and network infrastructures for the deployment of IoT-Lab testbed has been finished: including designing and installing support structures, amenaging space (ventilation, ...), acquiring and installing network equipment, servers, GPS, ...
  The Senslab testbed has been moved to the building 1 and has been integrated into the new platform. Deployment is in a finalization phase and should go in beta testing starting from Q1 2014. The testbed will offer 344 open nodes, including 120 WSN430 nodes, 200 Cortex A8 based nodes, 24 Cortex M3.
- Support of external projects: Support for RIOT-OS and OpenWSN projects has been developed for IoT-Lab hardware and is being tested.
  - RIOT-OS, a joint work between Inria and FU-Berlin to create an Operating System for the Internet of Things.
  - OpenWSN, an open-source protocol stack for Internet of Things developed by UC Berkley.
  - IoT-Lab hardware based on STM stm32f1 series ARM Cortex-M3 MCU and Atmel AT86RF231 radio transceiver.

5.1.2. Software

5.1.2.1. NS3

Participants: Cédric Adjih, Hana Baccouch.

Ey-Wifi, Elimination-Yield for WiFi networks, is a module developed for the ns-3 simulation tool to integrate the features of the EY-NPMA channel access scheme. EY-NPMA (Elimination-Yield Non-Pre-emptive Priority Multiple Access) is a contention based protocol using active signaling (black burst): a node requests access to the medium by transmitting a burst signal. More precisely, the channel access cycle comprises three phases: priority phase, elimination phase and yield phase.

This software was developped thanks to the ADT MOBSIM.

The Ey-Wifi module has been publically released and is available, along with a detailed tutorial explaining how to use it, at: http://hipercom.inria.fr/Ey-Wifi

5.1.2.2. OPERA

Participants: Cédric Adjih, Ichrak Amdouni, Pascale Minet, Saoucene Ridene, Ridha Soua.

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

More details and documentation about this software are available in the website made by the Hipercom2 team: http://opera.gforge.inria.fr/index.html

5.1.2.3. SAHARA

Participants: Erwan Livolant, Pascale Minet, Ridha Soua, Cédric Adjih.

The software modules developed by the Hipercom2 team in the SAHARA project have been registered by the APP in July 2013:

- Mundi-Safeti V1.0, Reference: IDDN.FR.001.270022.000.S.P.2013.000.10000
- SAHARA-Network V1.0, Reference: IDDN.FR.001.270021.000.S.P.2013.000.10000

KERDATA Project-Team

5. Software and Platforms

5.1. BlobSeer

Participants: Zhe Li, Rohit Saxena, Alexandru Costan, Gabriel Antoniu, Luc Bougé.

Contact: Gabriel Antoniu.

Presentation: BlobSeer is the core software platform for most current projects of the KerData team. It is a data storage service specifically designed to deal with the requirements of large-scale data-intensive distributed applications that abstract data as huge sequences of bytes, called BLOBs (Binary Large OBjects). It provides a versatile versioning interface for manipulating BLOBs that enables reading, writing and appending to them.

BlobSeer offers both scalability and performance with respect to a series of issues typically associated with the data-intensive context: *scalable aggregation of storage space* from the participating nodes with minimal overhead, ability to store *huge data objects*, *efficient fine-grain access* to data subsets, *high throughput in spite of heavy access concurrency*, as well as *fault-tolerance*.

Users: Work is currently in progress in several formalized projects (see previous section) to integrate and leverage BlobSeer as a data storage back-end in the reference cloud environments: a) Microsoft Azure; b) the Nimbus cloud toolkit developed at Argonne National Lab (USA); and c) the OpenNebula IaaS cloud toolkit developed at UCM (Madrid).

URL: http://blobseer.gforge.inria.fr/

License: GNU Lesser General Public License (LGPL) version 3.

Status: This software is available on Inria's forge. Version 1.0 (released late 2010) registered with APP: IDDN.FR.001.310009.000.S.P.000.10700.

A *Technology Research Action* (ADT, *Action de recherche technologique*) started in November 2012 for two years, aiming at robustifying the BlobSeer software and making it a safely distributable product. This project is funded by Inria *Technological Development Office* (D2T, *Direction du Développement Technologique*). Loïc Cloatre, has been hired as a senior engineer for the second year of this project, as a successor of Zhe Li, starting in February 2014.

5.2. BlobSeer-WAN

Participants: Rohit Saxena, Alexandru Costan, Gabriel Antoniu.

Contact: Gabriel Antoniu.

Presentation: BlobSeer-WAN was initially designed as an extension of BlobSeer, targeting geographically distributed environments. With BlobSeer-WAN, the metadata is replicated asynchronously for low latency. There is a version manager on each site and vector clocks are used to allow collision detection and resolution under highly concurrent access. Several experiments have been conducted with this setup on the Grid'5000 testbed which have shown scalable metadata performance under geographically distributed environments. Currently, BlobSeer-WAN is integrated within BlobSeer, as a new release of the latter.

Users: BlobSeer-WAN has been preliminarily evaluated at University of Tsukuba (Japan) in the context of the FP3C project. BlobSeer-WAN is used as a storage backend for HGMDS, a multi master metadata server designed for a global distributed file system.

URL: http://blobseer.gforge.inria.fr./doku.php?id=ci:blobseer-wan

License: GNU Lesser General Public License (LGPL) version 3.

Status: This software is available on Inria's forge as part of BlobSeer. Registration with APP is in progress.

5.3. Damaris

Participants: Matthieu Dorier, Lokman Rahmani, Gabriel Antoniu.

Contact: Gabriel Antoniu.

Presentation: Damaris is a middleware for multicore SMP nodes enabling them to efficiently handle data transfers for storage and visualization. The key idea is to dedicate one or a few cores of each SMP node to the application I/O. It is developed within the framework of a collaboration between KerData and the *Joint Laboratory for Petascale Computing* (JLPC). The current version enables efficient asynchronous I/O, hiding all I/O related overheads such as data compression and post-processing, as well as direct (*in situ*) interactive visualization of the generated data.

Users: Damaris has been preliminarily evaluated at NCSA (Urbana-Champaign) with the CM1 tornado simulation code. CM1 is one of the target applications of the Blue Waters supercomputer in production at NCSA/UIUC (USA), in the framework of the Inria-UIUC-ANL Joint Lab (JLPC). Damaris now has external users, including (to our knowledge) visualization specialists from NCSA and researchers from the France/Brazil Associated research team on Parallel Computing (joint team between Inria/LIG Grenoble and the UFRGS in Brazil). Damaris has been successfully integrated into three large-scale simulations (CM1, OLAM, Nek5000). Works are in progress to evaluate it in the context of several other simulations including HACC (cosmology code) and GTC (fusion).

URL: http://damaris.gforge.inria.fr/

License: GNU Lesser General Public License (LGPL) version 3.

Status: This software is available on Inria's forge and registered with APP. Registration of the latest version with APP is in progress.

5.4. TomusBlobs

Participants: Radu Tudoran, Alexandru Costan, Gabriel Antoniu.

Contact: Gabriel Antoniu.

Presentation: TomusBlobs is a software library for concurrency-optimized data storage for data-intensive applications running on Azure clouds, including MapReduce applications. It is being developed by the KerData Inria Project-Team in the framework of the A-Brain MSR-Inria project. It uses the BlobSeer library.

Users: TomusBlobs has been preliminarily evaluated within the A-Brain project where it was used to execute a real-life application aiming to search for significant associations between brain images and genetics data. The TomusBlobs data-storage layer developed in the framework of the A-Brain MSR-Inria project was demonstrated to scale up to 1000 cores on 3 Azure data centers; it exhibits improvements in execution time up to 50 % compared to standard solutions based on Azure BLOB storage. Based on this storage infrastructure, the A-Brain project consortium has provided the first statistical evidence of the heritability of functional signals in a failed stop task in basal ganglia, using a ridge regression approach, while relying on the Azure cloud to address the computational burden.

License: GNU Lesser General Public License (LGPL) version 3.

Status: This software is available on Inria's forge. Registration with APP is in progress.

5.5. Darshan-Ruby

Participant: Matthieu Dorier.

Contact: Matthieu Dorier.

Presentation: Darshan-Ruby is a Ruby extension to the Darshan scalable HPC I/O characterization tool (developed by the Mathematics and Computer Science division at Argonne National Lab). It simplifies the access to the contents of Darshan-generated log files, in an object-oriented manner through the Ruby scripting language.

Users: Darshan-Ruby is available as a Ruby gem package on the official Rubygems website (http://rubygems.org/), and is referenced on the Darshan website (http://www.mcs.anl.gov/research/projects/darshan/).

URL: http://darshan-ruby.gforge.inria.fr/

License: GNU Lesser General Public License (LGPL) version 3.

Status: This software is available on Inria's forge.

5.6. Derived software

Derived from BlobSeer, an additional platform is currently being developed within KerData: Pyramid, a software service for array-oriented active storage developed within the framework of Viet-Trung Tran's PhD thesis.

LOGNET Team

5. Software and Platforms

5.1. myMed

Our flagship software is called myMed. myMed is a highly innovative project in which three main orthogonal components are brought together:

- a software development kit, SDKmyMed, with which we can build social networks in “rush time”;
- a novel distributed hosting cloud, CLOUDmyMed, with which the social applications (developed by us and by third parties) can be hosted and run;
- a pull of 5-10 social network applications, aka “sociapps” developed in our team to test the SDKmyMed.

The sociapp can be enjoyed in almost all platforms, from web browsers, to mobile web, until IOS and Android devices.

5.2. myMed backbone

Participants: Luigi Liquori [contact], The Mymed Engineer Team.

`../../../../projets/lognet/IMG/splash3.pn`

`../../../../projets/lognet/IMG/myMed-backbone.jpg`

Figure 4. The myMed backbone and the myMed LaunchPad

We have implemented a “backbone” for the myMed social network using a nosql database called Cassandra http://cassandra.apache.org, the latter used also by social networks like Facebook and Twitter. The backbone relies on 50 PC quad code HP400, equipped with 2Tb of hard drive each.

5.3. myMed frontend

Participants: Luigi Liquori [contact], The Mymed Engineer Team.

We have implemented a front-end with which all the social application can be used and downloaded via a “store” mechanism similar to the ones of Apple and Google stores. Social applications can be chosen, voted for via a reputation system, and uninstalled (including all personal data) if the user wants. We have also implemented a “template” allowing to build “proofs-of-concept” of social networks in a very short time.

5.4. Synapse simulator in Oversim

Participants: Vincenzo Ciancaglini [contact], Luigi Liquori.

Synapse-Oversim is an implementation of the Synapse overlay interconnection protocol in the Oversim overlay simulator. The software presents two main contributions: first of all, a fork of the original Oversim simulator has been implemented in order to support running multiple protocol modules in a single instance of Oversim, a necessary feature in order to simulate a set of heterogeneous interconnected networks. Secondly, the whole Synapse protocol has been implemented on top of Oversim, in order to allow for the efficient inter-routing of messages between heterogeneous overlays. The Synapse code has been developed in C++, by running in Oversim, its correctness and its performances can be evaluated, while then the code can be easily ported to a real-world application.

5.5. Synapse model Erlang validator

Participant: Vincenzo Ciancaglini [contact].

During the work on the Synapse protocol, we devised a mathematical model which would allow us to estimate performance indexes of an interconnected system without having to deploy a full-scale experiment. In order to be validated, however, the model results needed to be verified against some simulation results, run under simplified conditions, but with the highest possible number of nodes. To achieve this, a dedicated simulator has been developed using Erlang, a programming language dedicated to parallel and distributed applications, which allow for the simulation of extreme systems, with a number of nodes beyond one million, in the fastest way achievable, by fully exploiting the multicore architecture of modern machines. The simulator instantiates a lightweight thread for each node, and the communication are rendered by message passing between the different node threads, thus keeping the simulation conditions as close as possible to a real world behavior.

5.6. CCN-TV Omnet++ simulator

Participants: Vincenzo Ciancaglini [contact], Riccardo Loti, Luigi Liquori.

CCN-TV-SIM is a software, based on the network simulation framework Omnet++, which simulates a real time video broadcast system over content-centric networks. The system is able to manage multiple streams of video at different rates, using real video traces, simulate different caching policies, different channels being transmitted concurrently, background network traffic, and different channel switch rates. Furthermore it can exploits network topologies taken from real networks, like the Deutsche Telecom network, or the Geant.

5.7. Java implementation of the OGP protocol and the experiment controller

Participants: Giang Ngo Hoang [contact], Luigi Liquori.

OGP-Experiment contains Java implementation of the OGP protocol (OGP stands for overlay gateway protocol) which is used for inter-routing between heterogeneous overlay networks, and a Java implementation of the experiment controller, which is responsible for scheduling, managing and monitoring the statistics of the experiments. The software supports experiments in churn and no-churn environments. Performance metrics of the OGP protocol, such as the latency, the successful rate of data lookup and the traffic generated by a peer are reported. The experiments are performed on the Grid 5000 platform. Heterogeneous overlays which are connected by OGP can be easily plugged into the software.

5.8. Java implementation of the Synapse protocol and the experiment controller

Participants: Hoang Giang Ngo [contact], Luigi Liquori.

Synapse-Experiment contains Java implementation of the Synapse overlay interconnection protocol and Java implementation of the experiment controller which is responsible for scheduling, managing and monitoring the statistics of the experiments. The software supports experiments in churn and no-churn environments. Performance metrics of the Synapse protocol, such as the latency, the successful rate of data looking up and the traffic generated by a peer are reported. The experiments are performed on the Grid 5000 platform.

5.9. Reputation Computation Engine for Social Web Platforms

Participant: Thao Nguyen [contact].

Among the three components of a Trust and Reputation System, information gathering is most dependent on the application system, followed by the decision support component and then by the building of a robust Reputation Computation Engine and an experimental GUI, showing how bad users are segregated by the engine. To simulate the working of the reputation engine, we set up a population of Nu users, providing the same service, and undertaking Nt transactions. In each transaction, a random consumer is assigned to request the service. Other users will then be candidate providers for this request. When a user plays the role of a consumer, his behavior is modeled in the raterType attribute. Three types of raters include HONEST, DISHONEST and COLLUSIVE. HONEST raters share their personal experience honestly, i.e. Rr = Ep. DISHONEST raters provide ratings 0:5 different from their true estimation, i.e. Rr = Ep +- 0:5. COLLUSIVE raters give the highest ratings (Rr = 1) to users in their collusion and the lowest ratings (Rr = 0) to the rest. Similarly, when a user acts as a provider, he can be one of the following types of providers: GOOD, NORMAL, BAD, or GOODTURNBAD. This type is denoted in providerType attribute. The QoS of the service provided by a BAD, NORMAL, or GOOD provider has a value in the interval (0; 0:4], (0:4; 0:7], or (0:7; 1] respectively. A GOODTURNBAD provider will change the QoS of his service when 50% of Nt transactions have been done in the simulation. To get a transaction done, a consumer obtains a list of providers, computes reputation scores for them, chooses a provider to perform the transaction, updates his private information, and publishes his rating for the provider. The quality of service that the consumer will experience depends on the providerType of the chosen provider. The difference between the consumer's rating for the provider and his observation depends on the consumer's raterType.

To run a simulation, the user must specify 10 parameters as described above: Simulation(Nu, Nt, %G, %N, %B, %GTB, %H, %D, %C, %dataLost). The simulator has been published in [22].

In 2013, the simulator has been improved and made more robust: it would one of the output of the Ph.D. work of Thao Nguyen whose defense is envisaged in the first half of 2014.

5.10. Ariwheels

Participants: Luigi Liquori [contact for the *Ariwheels* simulator], Claudio Casetti [Politecnico di Torino, Italy], Diego Borsetti [Politecnico di Torino, Italy], Carla-Fabiana Chiasserini [Politecnico di Torino, Italy], Diego Malandrino [Politecnico di Torino, Italy, contact for the *Ariwheels* client].

../../../../projets/lognet/IMG/simtrust.png

Figure 5. A prototype graphical GUI for the Reputation Computation Engine

Ariwheels is an info-mobility solution for urban environments, with access points deployed at both bus stops (forming thus a wired backbone) and inside the buses themselves. Such a network is meant to provide connectivity and services to the users of the public transport system, allowing them to exchange services, resources and information through their mobile devices. *Ariwheels* is both:

- a protocol, based on *Arigatoni* and the publish/subscribe paradigm;
- a set of applications, implementing the protocol on the different types of nodes;
- a simulator, written in OMNET++ and recently ported to the ns2 simulator, see Fig 6 .

`../../../../projets/lognet/IMG/one.jpg`

Figure 6. The Ariwheels simulator in Omnet

See the web page http://www-sop.inria.fr/members/Luigi.Liquori/ARIGATONI/Ariwheels.htm and http://arigtt.altervista.org.

5.11. Arigatoni simulator

Participants: Luigi Liquori [contact], Raphael Chand [Université de Geneva, Switzerland].

../../../../projets/lognet/IMG/Simulateur-Arigatoni.jpg

Figure 7. The Arigatoni simulator

We have implemented in C++ (∼2.5K lines of code) the Resource Discovery Algorithm and the Virtual Intermittent Protocol of the Arigatoni Overlay Network. The simulator was used to measure the load when we issued n service requests at Global Computers chosen uniformly at random. Each request contained a certain number of instances of one service, also chosen uniformly at random. Each service request was then handled by the Resource Discovery mechanism of Arigatoni networks.

5.12. Synapse client

Participants: Laurent Vanni [contact], Luigi Liquori, Cédric Tedeschi, Vincenzo Ciancaglini.

In order to test our Synapse protocol [21] on real platforms, we have initially developed JSynapse, a Java software prototype, which uses the Java RMI standard for communication between nodes, and whose purpose is to capture the very essence of our Synapse protocol. It is a flexible and ready-to-be-plugged library which can interconnect any type of overlay networks. In particular, JSynapse fully implements a Chord-based inter-overlay network. It was designed to be a lightweight and easy-to-extend software. We also provided some practical classes which help in automating the generation of the inter-overlay network and the testing of specific scenarios. We have experimented with JSynapse on the Grid'5000 platform connecting more than 20 clusters on 9 different sites. Again, Chord was used as the intra-overlay protocol. See, http://www-sop.inria.fr/teams/lognet/synapse-net2012/.

5.13. Open Synapse client

Participant: Bojan Marinkovic [contact].

Opensynapse is an open source implementation of [21]. It is available for free under the GNU GPL. This implementation is based on Open Chord (v. 1.0.5) - an open source implementation of the Chord distributed hash table implementation by Distributed and Mobile Systems Group Lehrstuhl fuer Praktische Informatik Universitaet Bamberg, see http://www-sop.inria.fr/teams/lognet/synapse-net2012/.

Opensynapse is implemented on top of an arbitrary number of overlay networks. Inter-networking can be built on top of Synapse in a very efficient way. Synapse is based on co-located nodes playing a role that is reminiscent of neural synapses. The current implementation of Opensynapse in this precise case interconnects many Chord overlay networks. The new client currently can interconnect an arbitrary number of Chord networks. This implementation follows the notation presented in [20], and so, each new Chord network is called a *Floor*.

5.14. Husky interpreter

Participants: Marthe Bonamy [contact], Luigi Liquori.

`../../../../projets/lognet/IMG/husky.jpg`

Figure 8. Launching the Husky interpreter

Husky is a variable-less language based on lambda calculus and term rewriting systems. Husky is based on the version 1.1 of *Snake* It was completely rewritten in CAML by Marthe Bonamy, ENSL (new parser, new syntactic constructions, like, *e.g.*, guards, anti-patterns, anti-expressions, exceptions and parametrized pattern matching). In *Husky*, all the keywords of the language are ASCII-symbols. It could be useful for teaching basic algorithms and pattern-matching to children.

5.15. myTransport Gui

Participants: Laurent Vanni [contact], Vincenzo Ciancaglini, Liquori Liquori.

myTransport is a GUI built on top of the Synapse protocol and network. Its purpose is to be a proof of concept of the future service of info-mobility to be available in the myMed social Network, see Figure 9 . The GUI is written in Java and it is fully functional in the Nokia N800 Internet tablet devices. myTransport has been ported to the myMed social network.

5.16. myDistributed Catalog for Digitized Cultural Heritage

Participants: Vincenzo Ciancaglini [contact], Bojan Marinkovic [MISANU, Serbia], Liquori Liquori.

../../../../projets/lognet/IMG/mytransport.jpg

Figure 9. myTransport on the Nokia N800 Internet tablet

Peer-to-peer networks have emerged recently as a flexible decentralized solution to handle large amount of data without the use of high-end servers. We have implemented a distributed catalog built up on an overlay network called “Synapse”. The Synapse protocol allows interconnection of different overlay networks each of them being an abstraction of a “community” of virtual providers. Data storage and data retrieval from different kind of content providers (i.e. libraries, archives, museums, universities, research centers, etc.) can be stored inside one catalog. We illustrate the concept based on the Synapse protocol: a catalog for digitized cultural heritage of Serbia, see Figure 10 .

5.17. myStreaming P2P

Participants: Vincenzo Ciancaglini [contact], Rossella Fortuna [Politech Bari], Salvatore Spoto [Univ. Turin], Liquori Liquori, Luigi Alfredo Grieco [Politech Bari].

We have implemented, in Python, a fork of Goalbit http://goalbit.sourceforge.net, an open source video streaming platform peer-to-peer software streaming platform capable of distributing high-bandwidth live video content to everyone preserving its quality. We have aligned with the classical gossip-based distribution protocol a *DHT* that distribute contents according to a content-based strategy.

../../../../projets/lognet/IMG/mycatalog.png

Figure 10. myDistributed Catalog

MADYNES Project-Team

5. Software and Platforms

5.1. SecSIP

Participants: Abdelkader Lahmadi [contact], Olivier Festor.

SecSip [1] 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. Veto is a generic declarative language for attack patterns specification. SecSIP 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.

5.2. NDPMon

Participants: Isabelle Chrisment, Olivier Festor [contact].

The Neighbor Discovery Protocol Monitor (NDPMon) is an IPv6 implemention 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 [2]. Binary distributions are also available for .deb and .rpm based Linux flavors.

---

[1] http://secsip.gforge.inria.fr/doku.php

[2] http://www.freebsd.org/cgi/cvsweb.cgi/ports/net-mgmt/ndpmon/

5.3. AA4MM

Participants: Laurent Ciarletta [contact], Yannick Presse.

Vincent Chevrier (MAIA team, contact), and Benjamin Camus and Julien Vaubourg (MAIA team, LORIA) are contributors 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. The first implementation of the AA4MM meta-model was proposed in Julien Siebert's PhD [49] and written in Java. This version is currently being put into APP (Agence pour la protection des programmes).

This year, we have used this software in a strategic action with EDF R&D in the context of the simulation of smart-grids. Julien Vaubourg started a PhD on this project that is co-directed by Laurent Ciarletta and Vincent Chevrier.

5.4. MASDYNE

Participant: Laurent Ciarletta [contact].

This work was undertaken in a joint PhD Thesis between MAIA and Madynes Team. Vincent Chervrier (MAIA team, LORIA) has been director and co-advisor of this PhD and is correspondant for this software, which has been used by Tomas Navarrete (MAIA team, LORIA). Other contributors to this software were: Julien Siebert, 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 the MAIA team, as part as a modeling and simulation of ubiquitous networks effort.

MAESTRO Project-Team (section vide)

MESCAL Project-Team

5. Software and Platforms

5.1. Tools for cluster management and software development

Participant: Olivier Richard [correspondent].

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

KA-DEPLOY is an environment deployment toolkit that provides automated software installation and reconfiguration mechanisms for large clusters and light grids. The main contribution of KA-DEPLOY 2 toolkit is the introduction of a simple idea, aiming to be a new trend in cluster and grid exploitation: letting users concurrently deploy computing environments tailored exactly to their experimental needs on different sets of nodes. To reach this goal KA-DEPLOY must cooperate with batch schedulers, like OAR, and use a parallel launcher like TAKTUK (see below).

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

5.2. OAR: Batch scheduler for clusters and grids

Participant: Olivier Richard [correspondent].

The OAR project (see http://oar.imag.fr) focuses on robust and highly scalable batch scheduling for clusters and grids. Its main objectives are the validation of grid administration tools such as TAKTUK, the development of new paradigms for grid scheduling and the experimentation of various scheduling algorithms and policies.

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

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

5.3. CiGri: Computing resource Reaper

Participant: Olivier Richard [correspondent].

CiGri (see http://cigri.imag.fr/) is a middleware which gathers the unused computing resource from intranet infrastructure and makes it available for the processing of large set of tasks. It manages the execution of large sets of parametric tasks on lightweight grid by submitting individual jobs to each batch scheduler. It is s associated to the OAR resource management system (batch scheduler). Users can easily monitor and control their set of jobs through a web portal. CiGri provides mechanisms to identify job error causes, to isolate faulty components and to resubmit jobs in a safer context.

5.4. FTA: Failure Trace Archive

The Failure Trace Archive [11] is available at http://fta.inria.fr. Since Derrick Kondo left on sabbatical, the Failure Trace Archive has been migrated to University of Western Sidney, Australia (http://fta.scem.uws.edu.au/), which allows an easier management by his colleagues Bahman Javadi who was working as a post-doc in the MESCAL team while initializing the FTA.

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

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

5.5. SimGrid: simulation of distributed applications

Participants: Arnaud Legrand [correspondent], Lucas Mello Schnorr, Luka Stanisic, Augustin Degomme.

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

5.6. TRIVA: interactive trace visualization

Participants: Lucas Mello Schnorr [correspondent], Arnaud Legrand.

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

5.7. ψ and ψ^2: perfect simulation of Markov Chain stationary distributions

Participant: Jean-Marc Vincent [correspondent].

ψ and ψ^2 (see http://psi.gforge.inria.fr) are two software tools implementing perfect simulation of Markov Chain stationary distributions using *coupling from the past*. ψ starts from the transition kernel to derive the simulation program while ψ^2 uses a monotone constructive definition of a Markov chain.

5.8. GameSeer: simulation of game dynamics

Participant: Panayotis Mertikopoulos [correspondent].

Mathematica toolbox (graphical user interface and functions library) for efficient, robust and modular simulations of game dynamics.

5.9. Kameleon: environment for experiment reproduction

Participants: Olivier Richard [correspondent], Joseph Emeras.

Kameleon is a tool developed to facilitate the building and rebuilding of software environment. It helps the experimenter to manage his experiment's software environment which can include the operating system, libraries, runtimes, his applications and data. This tool is an element in the experimental process to obtain repeatable experiments and therefore reproducible results.

5.10. Platforms

5.10.1. Grid'5000

The MESCAL project-team is involved in development and management of Grid'5000 platform. The Digitalis and IDPot clusters are integrated in Grid'5000 as well as of CIMENT.

5.10.2. The ICluster-2, the IDPot and the new Digitalis Platforms

The MESCAL project-team manages a cluster computing center on the Grenoble campus. The center manages different architectures: a 48 bi-processors PC (ID-POT), and the center is involved with a cluster based on 110 bi-processors Itanium2 (ICluster-2) and another based on 34 bi-processor quad-core XEON (Digitalis) located at Inria. The three of them are integrated in the Grid'5000 grid platform.

More than 60 research projects in France have used the architectures, especially the 204 processors Icluster-2. Half of them have run typical numerical applications on this machine, the remainder has worked on middleware and new technology for cluster and grid computing. The Digitalis cluster is also meant to replace the Grimage platform in which the MOAIS project-team is very involved.

5.10.3. The Bull Machine

In the context of our collaboration with Bull the MESCAL project-team exploits a Novascale NUMA machine. The configuration is based on 8 Itanium II processors at 1.5 Ghz and 16 GB of RAM. This platform is mainly used by the Bull PhD students. This machine is also connected to the CIMENT Grid.

MOAIS Project-Team

5. Software and Platforms

5.1. KAAPI

Participants: Thierry Gautier [correspondant], Vincent Danjean, François Broquedis, Joao Ferreira Lima.

- ACM: D.1.3
- License: CeCILL
- OS/Middelware: Unix (Linux, MacOSX, ...)
- Programming language: C/C++, Fortran
- Characterization of Software : A-3 / SO-4 / SM-3 / EM-3 / SDL-4
- Own Contribution: DA-4 / CD-4 / MS-4 / TPM-4
- Additional information:

  Kaapi (http://kaapi.gforge.inria.fr, coordinator T. Gautier) is a middleware for high performance applications running on multi-cores/multi- processors as well as cluster or computational grid. Kaapi provides 1/ a very high level API based on macro data flow language; 2/ several scheduling algorithms for multi-threaded computations as well as for iterative applications for numerical simulation on multi-CPUs / multi-GPUs; 3/ fault-tolerant protocols. Publicly available at http://kaapi.gforge.inria.fr under CeCILL licence. Kaapi has won the 2008 Plugtest organized by Grid@Works. Kaapi provides ABI compliant implementations of Quark (PLASMA, Linear Algebra, Univ. of Tennesse) and libGOMP (GCC runtime for OpenMP). Direct competitors with 1/: Quark (UTK), OMPSs (UPC, BSC), OpenMP. Direct competitors with 2/: StarSs, StarPU (Inria RUNTIME). Direct competitors providing 3/: Charm++, MPI.

5.2. FlowVR

Participants: Bruno Raffin [correspondant MOAIS], Matthieu Dreher, Jérémy Jaussaud.

- ACM: D.1.3
- License: GPL and LGPL
- OS/Middelware: Unix (Linux, MacOSX, ...)
- Programming language: C/C++
- Characterization of Software : A-3 / SO-4 / SM-3 / EM-3 / SDL-4
- Own Contribution: DA-4 / CD-3 / MS-3 / TPM-4
- Additional information: FlowVR (http://flowvr.sf.net, coordinator B. Raffin) provides users with the necessary tools to develop and run high performance interactive applications on PC clusters and Grids. The main target applications include virtual reality, scientific visualization and in situ analytics. FlowVR enforces a modular programming that leverages software engineering issues while enabling high performance executions on distribued and parallel architectures. FlowVR is the reference backbone for Grimage. See also the web page http://flowvr.sf.net.

5.3. TakTuk - Adaptive large scale remote execution deployment

Participants: Guillaume Huard [correspondant], Pierre Neyron.

- Characterization of Software : A-2 / SO-3 / SM-5 / EM-3 / SDL-4
- Own Contribution: DA-4 / CD-4 / MS-4 / TPM-4
- Additional information:
  - web site: http://taktuk.gforge.inria.fr, Coordinator G. Huard
  - Objective of the software: TakTuk is a tool for deploying parallel remote executions of commands to a potentially large set of remote nodes. It spreads itself using an adaptive algorithm and sets up an interconnection network to transport commands and perform I/Os multiplexing/demultiplexing. The TakTuk mechanics dynamically adapt to environment (machine performance and current load, network contention) by using a reactive work-stealing algorithm that mixes local parallelization and work distribution.
  - Users community: TakTuk is a research open source project available in the Debian GNU/Linux distribution (package taktuk) used in lower levels of Grid5000 software architectures (nodes monitoring in OAR, environment diffusion in Kadeploy). The community is small : developers and administrators for large scale distributed platforms, but active.
  - Positioning: main competing tools are pdsh (but uses linear deployment) and gexec (not fault tolerant, requires installation), for more details : B. Claudel, G. Huard and O. Richard. TakTuk, Adaptive Deployment of Remote Executions. In Proceedings of the International Symposium on High Performance Distributed Computing (HPDC), 2009. TakTuk is the only tool to provide to deployed processes a communication layer (just like an MPIrun, but not tied to a specific environment) and synchronization capabilities.

5.4. Triva

Participant: Guillaume Huard [correspondant].

- Characterization of Software : A-2 / SO-4 / SM-5 / EM-3 / SDL-3
- Own Contribution: DA-4 / CD-3 / MS-3 / TPM-3
- Additional information:
  - web site: http://triva.gforge.inria.fr/, Coordinator, Lucas Schnorr
  - Objective of the software: Triva is an open-source tool used to analyze traces (in the pajé format) registered during the execution of parallel applications. The tool serves also as a sandbox to the development of new visualization techniques.
  - Users community: Research open source project, applications developers, especially parallel applications.
  - Positioning: Main competing tools are Vampir (classical 2D Gantt charts) and Tau (less advanced agregation techniques), more details in : A Hierarchical Aggregation Model to achieve Visualization Scalability in the analysis of Parallel Applications. Lucas Mello Schnorr, Guillaume Huard, Philippe Olivier Alexandre Navaux. Parallel Computing. Volume 38, Issue 3, March 2012.

5.5. OAR

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

- Characterization of Software : A-5 / SO-3 / SM-4 / EM-4 / SDL-5
- Own Contribution: DA-3 / CD-2 / MS-1 / TPM-1
- Additional information: OAR (http://oar.imag.fr, Coordinator O. Richard, Inria MESCAL) is a batch scheduler. The MOAIS team develops the central automata and the scheduling module that includes successive evolutions and improvements of the policy.OAR is used to schedule jobs both on the CiGri (Grenoble region) and Grid50000 (France) grids. CiGri is a production grid that federates about 500 heterogeneous resources of various Grenoble laboratories to perform computations in physics. MOAIS has also developed the distributed authentication for access to Grid5000.

5.6. SOFA

Participant: Bruno Raffin [correspondant].

- ACM: D.1.3
- Programming language: C/C++
- Characterization of Software : A-5 / SO-4 / SM-4 / EM-4 / SDL-5
- Own Contribution: DA-2 / CD-2 / MS-1 / TPM-1
- Additional information: SOFA (http://www.sofa-framework.org/, Coordinator F. Faure, Inria IMAGINE) is an Open Source framework primarily targeted at real-time simulation, with an emphasis on medical simulation. It is mostly intended for the research community to help develop newer algorithms, but can also be used as an efficient prototyping tool. Moais contributes to parallelization of kernel algorithms used in the simulation.

5.7. LinBox

Participants: Clément Pernet [correspondant], Thierry Gautier.

- Characterization of Software : A-3 / SO-4 / SM-2 / EM-3 / SDL-5
- Own Contribution: DA-4 / CD-3 / MS-3 / TPM-4
- Additional information:
  - web site: http://linalg.org
  - Objective of the software: LinBox is an open-source C++ template library for exact, high-performance linear algebra computations. It is considered as the reference library for numerous computations (such as linear system solving, rank, characteristic polynomial, Smith normal forms,...) over finite fields and integers with dense, sparse, and structured matrices.
  - The LinBox group is an international collaboration (USA: NCSU, UDel; Canada: U Waterloo, U Calgary; France: LIP, LIRMM, LJK and LIG). Articles related to the library have been published in the main Conferences of the area: ISSAC, ICMS. MOAIS contributes to its development and more specifically to its parallelization in the context of ANR HPAC project. It is currently experiencing a major change of design, to better integrate parallelism.
  - Users community: mostly researchers doing computational mathematics (number theory, cryptology, group theory, persistent homology. They use the library by either linking against it directly (the library is packaged in Debian, Fedora, etc) or withing the general purpose math software Sage (sagemath.org very broad diffusion) which includes LinBox as a kernel for exact linear algebra.

MYRIADS Project-Team

5. Software and Platforms

5.1. HOCL-tools

Contact: Cédric Tedeschi, `Cedric.Tedeschi@irisa.fr`

Status: Version 1.0 to be released in open source

License: TBD

Presentation: HOCL (Higher Order Chemical Language) is a chemical programming language based on the chemical metaphor presented before (see Section 3.5). It was developed for several years within the PARIS and Myriads teams. Within HOCL, following the chemical metaphor, computations can be regarded as chemical reactions, and data can be seen as molecules which participate in these reactions. If a certain condition is held, the reaction will be triggered, thus continuing until it gets inert: no more data can satisfy any computing conditions. To realize this program paradigm, a multiset is implemented to act as a chemical tank, containing necessary data and rules. An HOCL program is then composed of two parts: *chemical rule definitions* (reaction rules) and *multiset definition* (data). More specifically, HOCL provides the high order: reaction rules are molecules that can be manipulated like any other molecules. In other words, HOCL programs can manipulate other HOCL programs.

An HOCL compiler was developed using Java to execute some chemical programs expressed with HOCL. This compiler is based on the translation of HOCL programs to Java code. As a support for service coordination and service adaptation, we recently extended the HOCL compiler with the support of decentralized workflow execution. Works around the implementation of a distributed multiset gave birth to an underlying layer for this compiler, making it able to deploy HOCL programs transparently over large scale platforms. This last part is currently considered to be interfaced with the current HOCL compiler. All these features are planned to be released under the common name of *HOCL-tools*.

Active contributors (from Myriads project-team): Marko Obrovac, Cédric Tedeschi.

Impact: The compiler is used as a tool within the team to develop HOCL programs. The decentralized workflow execution support has been used extensively to produce results published and presented at several conferences. It is also used in the framework of the DALHIS [2] associated team.

5.2. Contrail Virtual Execution Platform (VEP)

Contact: Yvon Jégou, `Yvon.Jegou@inria.fr`

URL: http://project.inria.fr/vep/

Status: Version 2.1

License: BSD

Presentation: Virtual Execution Platform (VEP) [56] is a Contrail service that sits just above IaaS layer at the service provider end of the Contrail cloud federation. The VEP service provides a uniform interface for managing the whole lifecycle of elastic applications on the cloud and hides the details of the IaaS layer to the user. VEP applications are described in OVF (Open Virtualization Format) standard format. Resource usage is controlled by CEE (Constrained Execution Environment) rules which can be derived from SLAs (Service Level Agreement). The VEP service integrates a monitoring system where the major events about the application, mainly resource usage, are made available to the user.

The VEP service provides a RESTful interface and can be exploited directly by users on top of the provider IaaS. OpenNebula and OCCI-based IaaS interfaces are currently supported.

---

[2] http://project.inria.fr/dalhis

Active contributors (from Myriads project-team): Roberto Cascella, Florian Dudouet, Filippo Gaudenzi, Piyush Harsh, Yvon Jégou, Christine Morin.

Impact: VEP is part of Contrail software stack. Several Contrail partners experiment use cases on top of VEP. External users can experiment with it using the open testbed operated by Myriads team.

5.3. Snooze

Contact: Christine Morin, `Christine.Morin@inria.fr`

URL: http://snooze.inria.fr

Status: Version 2.1.1

License: GPLv2

Presentation: Snooze [26] [53] [5], [4], a novel Infrastructure-as-a-Service (IaaS) cloud management system, which is designed to scale across many thousands of servers and virtual machines (VMs) while being easy to configure, highly available, and energy efficient. For scalability, Snooze performs distributed VM management based on a hierarchical architecture. To support ease of configuration and high availability Snooze implements self-configuring and self-healing features. Finally, for energy efficiency, Snooze integrates a holistic energy management approach via VM resource (i.e. CPU, memory, network) utilization monitoring, underload/overload detection and mitigation, VM consolidation (by implementing a modified version of the Sercon algorithm [59]), and power management to transition idle servers into a power saving mode. Snooze is a highly modular software. It has been extensively evaluated on the Grid'5000 testbed using realistic applications.

Snooze is fully implemented from scratch in Java and currently comprises of approximately 15.000 lines of maintainable abstractions-based code. In order to provide a uniform interface to the underlying hypervisors and support transparent VM monitoring and management, Snooze integrates the *libvirt* virtualization library. Cassandra (since 2.0.0) can be used as base backend, providing reliability and scalability to the database management system. At a higher level Snooze provides its own REST API as well as an EC2 compatible API (since 2.1.0). It can thus be controlled from the command line (using the legacy client or an EC2 compatible tool), or from different langage libraries (libcloud, jcloud ...). Snooze also provides a web interface to control the system.

Active contributors (from Myriads team): Eugen Feller, Yvon Jégou, David Margery, Christine Morin, Anne-Cécile Orgerie, Matthieu Simonin.

Impact: Snooze has been used by students at LIFL, IRIT in France and LBNL in the US in the framework of internships. It has also been deployed and experimented at EDF R&D. Snooze entry won the 2nd prize of the scalability challenge at CCGrid2013. Finally, we know that it was experimented by external users from academia and industry as we received feed-back from them.

5.4. Resilin

Contact: Christine Morin, `Christine.Morin@inria.fr`

URL: http://resilin.inria.fr

Status: Version 1.0

License: GNU Affero GPL

Presentation: Resilin [31] is an open-source system for creating and managing MapReduce execution platforms over clouds. Resilin is compatible with the Amazon Elastic MapReduce (EMR) API, but it goes beyond Amazon's proprietary EMR solution in allowing users (e.g. companies, scientists) to leverage resources from one or more public and/or private clouds. This enables performing MapReduce computations over a large number of geographically-distributed and diverse resources. Resilin can be deployed across most of the open-source and commercial IaaS cloud management systems (e.g., OpenStack, OpenNebula, Amazon EC2). Once deployed, Resilin takes care of provisioning

Hadoop clusters and submitting MapReduce jobs, allowing users to focus on writing their MapReduce applications rather than managing cloud resources. Resilin is implemented in the Python language and uses the Apache Libcloud library to interact with IaaS clouds. Resilin has been evaluated on multiple clusters of the Grid'5000 experimentation testbed. The results show that Resilin enables the use of geographically distributed resources with a limited impact on MapReduce job execution time.

Active contributors (from the Myriads project-team): Ancuta Iordache, Christine Morin, Nikos Parlavantzas.

Impact: Resilin is being used in the MOAIS project-team at Inria Grenoble - Rhône Alpes.

5.5. Merkat

Contact: Nikos Parlavantzas, `Nikos.Parlavantzas@irisa.fr`

URL: http://www.irisa.fr/myriads/software/Merkat/

Status: Version 1.0

License: TBD

Presentation: Merkat is a market-based private PaaS (Platform-as-a-Service) system, supporting dynamic, fine-grained resource allocation and automatic application management[23], [22], [11]. Merkat implements a proportional-share auction that ensures maximum resource utilization while providing incentives to applications to regulate their resource usage. Merkat includes generic mechanisms for application deployment and automatic scaling. These mechanisms can be adapted to support diverse performance goals and application types, such as master-worker, MPI, or MapReduce applications. Merkat is implemented in Python and uses OpenNebula for virtual machine management. Experimental results on the Grid'5000 testbed show that using Merkat increases resource utilization and improves application performance. Merkat is currently being evaluated by EDF R&D using EDF high-performance applications.

Active contributors (from the Myriads team): Stefania Costache, Christine Morin, Nikos Parlavantzas.

Impact: Merkat has been integrated in EDF R&D portal providing access to internal computing resources and is currently used on a testbed at EDF R&D.

5.6. ConPaaS

Contact: Guillaume Pierre, `Guillaume.Pierre@irisa.fr`

URL: http://www.conpaas.eu/

Status: Version 1.3.1

License: BSD

Presentation: ConPaaS [60] is a runtime environment for hosting applications in the cloud. It aims at offering the full power of the cloud to application developers while shielding them from the associated complexity of the cloud. ConPaaS is designed to host both high-performance scientific applications and online Web applications. It automates the entire life-cycle of an application, including collaborative development, deployment, performance monitoring, and automatic scaling. This allows developers to focus their attention on application-specific concerns rather than on cloud-specific details.

Active contributors (from the Myriads team): Eliya Buyukkaya, Ancuta Iordache, Morteza Neishaboori, Guillaume Pierre, Yann Radenac, Dzenan Softic.

Impact: ConPaaS is recognized as one of the major open-source PaaS environments. It is being developed by teams in Rennes, Amsterdan, Berlin and Ljubljana. Technology transfer of ConPaaS technology is ongoing in the context of the MC-DATA EIT ICT Labs project.

5.7. Meryn

Contact: Nikos Parlavantzas, `Nikos.Parlavantzas@irisa.fr`

URL: http://www.irisa.fr/myriads/software/Meryn/

Status: Version 1.0

License: TBD

Presentation: Meryn is an open, SLA-driven PaaS architecture that supports cloud bursting and allows hosting an extensible set of application types. Meryn relies on a decentralized optimization policy that aims at maximizing the overall provider profit, taking into account the penalties incurred when quality guarantees are unsatisfied [24]. The current Meryn prototype is implemented in shell script, builds upon the Snooze VM manager software, and supports batch and MapReduce applications using respectively the Oracle Grid Engine OGE 6.2u7 and Hadoop 0.20.2 frameworks. Meryn is developed in the framework of Djawida Dib's PhD thesis.

Active contributors (from the Myriads team): Djawida Dib, Christine Morin, Nikos Parlavantzas.

Impact: Meryn is not yet distributed as open source.

OASIS Project-Team

5. Software and Platforms

5.1. ProActive

Participants: Françoise Baude, Denis Caromel, Ludovic Henrio, Fabrice Huet [correspondant], Bastien Sauvan.

ProActive (Proactive Parallel Suite) is a Java library (Source code under AGPL license) for parallel, distributed, and concurrent computing, also featuring mobility and security in a uniform framework. With a reduced set of simple primitives, ProActive provides a comprehensive API to simplify the programming of applications that are distributed on a Local Area Network (LAN), on cluster of workstations, Clouds, or on Internet Grids.

The library is based on an Active Object pattern that is a uniform way to encapsulate:

- a remotely accessible object,
- a thread,
- an actor with its own script,
- a server of incoming requests,
- a mobile and potentially secure agent.

and has an architecture to inter-operate with (de facto) standards such as Web Service, HTTP transport, ssh, Globus, etc.

ProActive is only made of standard Java classes, and requires **no changes to the Java Virtual Machine**, no preprocessing or compiler modification; programmers write standard Java code. Based on a simple Meta-Object Protocol, the library is itself extensible, making the system open for adaptations and optimisations. ProActive currently uses the RMI Java standard library as default portable transport layer, but others such as Ibis or HTTP can be used instead, in an adaptive way.

ProActive is particularly well-adapted for the development of applications distributed over the Internet, thanks to reuse of sequential code, through polymorphism, automatic future-based synchronisations, migration of activities from one virtual machine to another. The underlying programming model is thus innovative compared to, for instance, the well established MPI programming model.

In order to cope with the requirements of large-scale distributed and heterogeneous systems like the Grid, many features have been incorporated into ProActive, including support for many transport and job submission protocols, GCM component support, graphical visualization interface, object migration, distributed and non-functional exception handling, fault-tolerance and checkpointing mechanisms; file transfer capabilities, a job scheduler, a resource manager able to manage various hosting machines, support for JMX and OSGi capabilities, web service object exposition, an SCA personality, etc.

ProActive is a project of the former ObjectWeb, now OW2 Consortium. OW2 is an international consortium fostering the development of open-source middleware for cutting-edge applications: EAI, e-business, clustering, grid computing, managed services and more. For more information, refer to [39], [37] and to the web pages http://www.objectweb.org and http://proactive.inria.fr/.

ProActive management, distribution, support, and commercialisation is now ensured by the start-up company ActiveEon (http://www.activeeon.com), in the context of a collaboration with Inria and UNS.

This year, the OASIS team made the following extensions to the ProActive library:

- Implementations related to the multi-active object programming model: multi-active components, declarative request service priority.
- Extension of the support of non-functional aspects for component systems: scripting language for reconfiguration, interceptor components.

5.2. Vercors platform

Participants: Eric Madelaine, Ludovic Henrio, Bartlomiej Szejna, Alexandra Savu, Oleksandra Kulankhina, Dongqian Liu.

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

- We have finished conducting experiments within the Papyrus environment, aiming at the definition of a graphical specification environment combining some of the standard UML formalisms with a dedicated graphical formalism for the architecture of GCM components. We have concluded that Papyrus is not an appropriate environment for our purpose due to the fact that the software is very unstable and badly-documented.
- We have achieved this year a major port of the frontend of the Vercors, namely VCE, the Vercors Component Editor, that is now based on the Obeo Designer (Eclipse) platform (http://www.obeodesigner.com). The main motivation, and achievement of this port was to integrate editors for some existing UML formalisms (Class and State-machines) with our GCM architecture editor. The new version of Vercors Component Editor (VCE v.3) has an editor for the GCM Components diagrams integrated with the UML class and state-machines diagrams editors. It also includes a generator of the ADL v.2 specification of the GCM-based architecture and a diagrams validation module.

5.3. Open Simulation Architecture (OSA)

Participant: Olivier Dalle.

OSA stands for Open Simulation Architecture. OSA is primarily intended to be a federating platform for the simulation community: it is designed to favor the integration of new or existing contributions at every level of its architecture. The platform core supports discrete-event simulation engine(s) built on top of the ObjectWeb Consortium's Fractal component model. In OSA, the systems to be simulated are modeled and instrumented using Fractal components. In OSA, the event handling is mostly hidden in the controller part of the components, which alleviates noticeably the modeling process, but also eases the replacement of any part of the simulation engine. Apart the simulation engine, OSA aims at integrating useful tools for modeling, developing, experimenting, and analysing simulations. OSA is also a platform for experimenting new techniques and approaches in simulation, such as aspect oriented programming, separation of concerns, innovative component architectures, and so on.

5.4. BtrPlace

Participants: Fabien Hermenier, Vincent Kherbache, Huynh Tu Dang.

Btrplace (http://btrp.inria.fr) is an open source virtual machine (VM) placement algorithm for datacenters. BtrPlace has been designed to be extensible. It can be customized by plugins from third party developers to address new SLAs or optimization constraints. Its extensibility is possible thanks to a composable core reconfiguration algorithm implemented using Constraint Programming. BtrPlace is currently bundled with a catalog of more than 20 constraints to address performance, fault tolerance, isolation, infrastructure management or energy efficiency concerns. It is currently used inside the FSN project OpenCloudWare (http://opencloudware.org/) and the European project DC4Cities (http://dc4cities.eu/).

This year, the catalog of constraints has been augmented according to the needs expressed inside OpenCloudWare. It as also been upgraded to a support for *continuous constraints* [11], a safer restriction mode to ensure the constraints can be satisfied at any moment, even during a reconfiguration process. A significant effort has also been made to make BtrPlace more usable and visible thanks to frequent releases, software documentation, tutorials, and live demo.

5.5. EventCloud

Participants: Françoise Baude, Fabrice Huet, Laurent Pellegrino, Bastien Sauvan, Iyad Alshabani, Maeva Antoine, Amjad Alshabani, Justine Rochas, Michel Jackson de Souza.

EventCloud (http://eventcloud.inria.fr) is an open source middleware that aims to act as a distributed datastore for data fulfiling the W3C RDF specification (www.w3.org/RDF/). It allows storing and retrieving quadruples (RDF triples with context) through SPARQL but also managing events represented as quadruples. The EventCloud architecture is based on a structured P2P overlay network targetting high-performance elastic data processing. Consequently it aims to be deployed on infrastructures like grids, clouds, i.e. whose nodes acquisition and relinquishment can be dynamic and subject to a pay-per-use mode. Each node participating in the overlay networks constituting EventCloud instances is responsible for managing the storage of subsets of the events, and helps in matching potential looked up events and disseminating them in a collaborative manner. As such, each node is also potentially an event broker responsible for managing subscriptions and routing notifications.

The EventCloud middleware has been developed using the GCM/ProActive library embedding the most recent advances from the Multi-active Object model (see Section 6.1.1) and its implementation. Interactions from end user applications with an EventCloud instance can happen directly using Java APIs along the GCM/ProActive model, or they can be achieved through GCM interfaces exposed following the Web Services Notification specification. Web Services Notification (WSN) is a set of specifications from the OASIS consortium (www.oasis-open.org) that standardises the way Web Services interact using "Notifications" or "Events". They form the foundation for Event Driven Architectures built using Web Services.

The EventCloud middleware is currently used as a component within the platforms developed within ANR SocEDA (Section 7.1.1) and FP7 PLAY (Section 7.2.1.1) projects. A significant effort has also started to apply it in application domains from the BigData area, as in Intelligent Transportation Systems 7.2.2 .

PHOENIX Project-Team

5. Software and Platforms

5.1. DiaSuite: a Development Environment for Sense/Compute/Control Applications

Participants: Charles Consel [correspondent], Damien Martin-Guillerez, Milan Kabac, Paul Van Der Walt, Camille Manano, Adrien Carteron, Alexandre Spriet, Emilie Balland.

Despite much progress, developing a pervasive computing application remains a challenge because of a lack of conceptual frameworks and supporting tools. This challenge involves coping with heterogeneous devices, overcoming the intricacies of distributed systems technologies, working out an architecture for the application, encoding it in a program, writing specific code to test the application, and finally deploying it.

DIASUITE is a suite of tools covering the development life-cycle of a pervasive computing application:

- *Defining an application area.* First, an expert defines a catalog of entities, whether hardware or software, that are specific to a target area. These entities serve as building blocks to develop applications in this area. They are gathered in a taxonomy definition, written in the taxonomy layer of the DIASPEC language.
- *Designing an application.* Given a taxonomy, the architect can design and structure applications. To do so, the DIASPEC language provides an application design layer [31]. This layer is dedicated to an architectural pattern commonly used in the pervasive computing domain [27]. Describing the architecture application allows to further model a pervasive computing system, making explicit its functional decomposition.
- *Implementing an application.* We leverage the taxonomy definition and the architecture description to provide dedicated support to both the entity and the application developers. This support takes the form of a Java programming framework, generated by the DIAGEN compiler. The generated programming framework precisely guides the developer with respect to the taxonomy definition and the architecture description. It consists of high-level operations to discover entities and interact with both entities and application components. In doing so, it abstracts away from the underlying distributed technologies, providing further separation of concerns.
- *Testing an application.*DIAGEN generates a simulation support to test pervasive computing applications before their actual deployment. An application is simulated in the DIASIM tool, without requiring any code modification. DIASIM provides an editor to define simulation scenarios and a 2D-renderer to monitor the simulated application. Furthermore, simulated and actual entities can be mixed. This hybrid simulation enables an application to migrate incrementally to an actual environment.
- *Deploying a system.* Finally, the system administrator deploys the pervasive computing system. To this end, a distributed systems technology is selected. We have developed a back-end that currently targets the following technologies: Web Services, RMI, SIP and OSGI. This targeting is transparent for the application code. The variety of these target technologies demonstrates that our development approach separates concerns into well-defined layers.

This development cycle is summarized in the Figure 2 .

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

../../../../projets/phoenix/IMG/DevCycle.jpg

Figure 2. DiaSuite *Development Cycle*

5.1.1. DiaSpec: a Domain-Specific Language for Networked Entities

The core of the DiaSuite development environment is the domain specific language called DiaSpec and its compiler DiaGen:

- DiaSpec is composed of two layers:
  - The *Taxonomy Layer* allows the declaration of entities that are relevant to the target application area. An entity consists of sensing capabilities, producing data, and actuating capabilities, providing actions. Accordingly, an entity description declares a data source for each one of its sensing capabilities. As well, an actuating capability corresponds to a set of method declarations. An entity declaration also includes attributes, characterizing properties of entity instances. Entity declarations are organized hierarchically allowing entity classes to inherit attributes, sources and actions. A taxonomy allows separation of concerns in that the expert can focus on the concerns of cataloging area-specific entities. The entity developer is concerned about mapping a taxonomical description into an actual entity, and the application developer concentrates on the application logic.
  - The *Architecture Layer* is based on an architectural pattern commonly used in the pervasive computing domain [27]. It consists of context components fueled by sensing entities. These components process gathered data to make them amenable to the application needs. Context data are then passed to controller components that trigger actions on entities. Using an architecture description enables the key components of an application to be identified, allowing their implementation to evolve with the requirements (*e.g.,* varying light management implementations in a controller component to optimize energy consumption).
- DiaGen is the DiaSpec compiler that performs both static and runtime verifications over DiaSpec declarations and produces a dedicated programming framework that guides and eases the implementation of components. The generated framework is independent of the underlying distributed technology. As of today, DiaGen supports multiple targets: Local, RMI, SIP, Web Services and OSGI.

5.1.2. DiaSim: a Parametrized Simulator for Pervasive Computing Applications

Pervasive computing applications involve both software and integration concerns. This situation is problematic for testing pervasive computing applications because it requires acquiring, testing and interfacing a variety of software and hardware entities. This process can rapidly become costly and time-consuming when the target environment involves many entities.

To ease the testing of pervasive applications, we are developing a simulator for pervasive computing applications: DiaSim. To cope with widely heterogeneous entities, DiaSim is parameterized with respect to a DiaSpec specification describing a target pervasive computing environment. This description is used to generate with DiaGen both a programming framework to develop the simulation logic and an emulation layer to execute applications. Furthermore, a simulation renderer is coupled to DiaSim to allow a simulated pervasive system to be visually monitored and debugged. The simulation renderer is illustrated in Figure 3 .

`../../../../projets/phoenix/IMG/Diasim.jpg`

Figure 3. A screenshot of the DIASIM *simulator*

5.2. DiaSuiteBox: an Open Orchestration Platform

Participants: Julien Bruneau [correspondent], Damien Martin-Guillerez, Quentin Enard, Charles Consel.

The DiaSuiteBox platform runs an open-ended set of applications leveraging a range of appliances and web services. Our solution consists of a dedicated development environment, a certifying application store, and a lightweight runtime platform. This solution is based on the DIASUITE project.

DiaSuiteBox platform architecture The DiaSuiteBox platform can be embedded in a small plug-computer or deployed in the cloud. Thanks to the application store and the developer community, the platform is fed by a full offer of new innovative applications. During the submission process, an application is automatically analyzed and checked in order to be certified. The user is ensured the behavior of its applications are innocuous and correct with respect to the provided information. Finally, DiaSuiteBox provides an extensible software architecture. This allows the easily connect new device technologies to the platform. For example, the support for new wireless communication technologies such as Zigbee, Z-Wave or Sigfox can be easily added to the DiaSuiteBox platform.

More details can be found on the web page http://diasuitebox.inria.fr.

The iQSpot startup uses DiaSuiteBox as a software platform to ease the management of Smart Buildings. In this project, the DiaSuiteBox platform is first used to host building management functionalities such as lighting management, heating/ventilating/air conditioning management, energy efficiency monitoring. It is also used to host software drivers that allow the building management functionalities to interact with the connected devices deployed in buildings. These devices can use wired communication technologies such LonWorks, BACNet or KNX, as well as wireless communication technologies such as Z-Wave or Zigbee.

5.3. School+ Apps: Assistive tablet applications for school Inclusion

Participants: Charles Consel [correspondent], Charles Fage, Damien Martin-Guillerez, Camille Manano, Hélène Sauzéon.

College+ is a package of 7 applications. Three applications are assistive applications, guiding the child doing specific tasks. Three others are training applications made as serious games, addressing specific skills. The last application is a meta-application, comprising a link to the three training applications, with an access to statistics of their usage. For each application, data are separated from the design, meaning that every element of each application (pictures, texts, settings, etc.) can be changed at any time. Each application records a log file every interactions performed by the child.

Assistive applications :

- *Routines application*. This application shows a list of tasks, with a short description. After clicking the starting button, a specific slideshow is shown ; it decomposes a task into steps. For each step, a text and a picture can be displayed. Thumbnail of previous and next steps are also displayed. This application guides the child through classroom situations: entering classroom, taking school materials out of a backpack, writing notes, handling agenda, leaving the classroom.
- *Communication application*. With the same design, the assistance provided by this application targets to communicating situations inside the classroom. The application covers four scenarios addressing two interaction situations (initiating and answering the interaction) and two types of interlocutors (professor and classmate). For each scenario, different slideshows guide the child, depending on the goal of the interaction.
- *Emotion Regulation application*. This application aims to assist the child to self-regulate his/her emotions. Four simplified emoticons are proposed to the child to choose from: anger, sadness, joy and fear. Then, (s)he selects a level of intensity via a thermometer with a scale from 1 to 4. In response, the application delivers different multimedia contents according to the level selected to help the child regulate his/her emotions. Typically, a text (breathing instructions) are shown at level 1, pictures at level 2, a video at level 3 and another text at level 4.

`../../../../projets/phoenix/IMG/assistivesapplications.png`

Figure 4. Assistive applications

../../../../projets/phoenix/IMG/trainingapplications.png

Figure 5. Assistive applications

Training applications: These three applications are serious games with increasing levels of difficulties, reachable after a ratio of good answers has been attained.

- *Emotion Recognition application: pictures*. In this application, the child is instructed to identify a specific emotion among 4 pictures showing different people exhibiting an emotion. Seven emotions are involved in this application: joy, sadness, fear, anger, surprise, disgust and neutral. The emotion to be recognized is displayed together with its simplified emoticon. The type of pictures changes with the difficulty level: level 1 contains pictures of unfamiliar people and level 2 contains pictures of friends and relatives of the child.
- *Emotion Recognition application: videos*. In this application, the child is presented with a fragment of an animated cartoon. At some point, the video stops and the child is asked to identify the emotion of the character. Four emotions are involved in this application: joy, sadness, fear and anger. Videos are slowed down, with a speed percentage that can be changed at each level. Videos change with difficulty level: level 1 contains videos of a very basic cartoon (only one cartoon character drawn by basic form un-textured), level 2 contains a video of more sophisticated cartoons and level 3 contains movies with actors.
- *Attention Training*. In this application, the child is presented a picture of a face and asked to make eye contact with it. Second, a symbol appears briefly in the eyes of the character. Third, the child is asked to identify the symbol shown in the previously displayed picture, to make sure he kept eye contact. The speed at which the symbol appears and disappears is changed according to the difficulty level. Types of pictures also change with the level : level 1 contains pictures of faces and level 2 contains pictures of classroom situations.

RAP Project-Team (section vide)

REGAL Project-Team

4. Software and Platforms

4.1. Coccinelle

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

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

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

Coccinelle is freely available at http://coccinelle.lip6.fr under a GPL v2 license.

4.2. SwiftCloud

Participants: Mahsa Najafzadeh, Burcu Külahçioglu Özkan, Marc Shapiro [correspondent], Marek Zawirski.

Cloud computing infrastructures improve latency and provide high availability by geo-replicating data at different locations across the world. This improves user experience, which is important for services such as social networks, online shops and games. Nevertheless, the distance to the closest data centre is still far from optimal for many users.

SwiftCloud is the first system to bring geo-replication all the way to the client machine, in order to provide the best possible latency and availability. This raises two main challenges. One, how to provide, efficiently, convenient programming guarantees, including access to consistent data in read-write transactions, and ensuring session guarantees. Two, to continue providing these guarantees, despite failures that require a client to switch to a different data centre.

Our research report [61] presents the design of SwiftCloud and the algorithms it uses to achieve the desired properties, while aiming for efficiency and for scalability to large numbers of clients. Our evaluation confirms that its programming model is practical, and that its performance and fault tolerance objectives are met.

SwiftCloud is supported by the ConcoRDanT ANR project (Section 7.1.6), by a Google European Doctoral Fellowship, and by the new FP7 grant SyncFree (Section 7.2.1.1).

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

4.3. JESSY

Participants: Masoud Saeida Ardekani, Marc Shapiro [correspondent].

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

1. The framework provides a fair, apples-to-apples comparison between transactional protocols;
2. By replacing plugs-ins, developers can use Jessy to understand bottlenecks in their protocols;
3. This in turn enables the improvement of existing protocols; and
4. Given a protocol, Jessy allows to evaluate the cost of ensuring various degrees of dependability.

Articles related to Jessy were published in an Inria research report [60], in the Symp. on Reliable Distr. Sys. (SRDS) [43] and in the Euro-Par conference [42] Jessy is supported by a UPMC PhD scholarship to Masoud Saeida Ardekani, and by the ConcoRDanT ANR project (Section 7.1.6).

Jessy is freely available on github under http://Github.com/msaeida/jessy under an Apache license.

4.4. Java and .Net runtimes for LLVM

Participants: Koutheir Attouchi, Harris Bakiras, Bertil Folliot, Julia Lawall, Gilles Muller, Thomas Preud'Homme, Gaël Thomas [correspondent].

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

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

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

RMOD Project-Team

5. Software and Platforms

5.1. Moose

Participants: Stéphane Ducasse [correspondant], Muhammad Bhatti, Andre Calvante Hora, Nicolas Anquetil, Anne Etien, Guillaume Larcheveque, Tudor Gîrba [University of Bern].

Web: http://www.moosetechnology.org/

The platform. Moose is a language-independent environment for reverse- and re-engineering complex software systems. Moose provides a set of services including a common meta-model, metrics evaluation and visualization, a model repository, and generic GUI support for querying, browsing and grouping. The development of Moose began at the Software Composition Group in 1997, and is currently contributed to and used by researchers in at least seven European universities. Moose offers an extensible meta-described metamodel, a query engine, a metric engine and several visualizations. Moose is currently in its fourth major release and comprises 55,000 lines of code in 700 classes.

The RMoD team is currently the main maintainer of the Moose platform. There are 200 publications (journal, international conferences, PhD theses) based on execution or use of the Moose environment.

The first version running on top of Pharo (Moose 4.0) was released in June 2010. In 2013, three releases of Moose where done: 4.7 to 4.9. The current focus is Moose 5.0, which is running on Pharo3 and will be released together with Pharo3 in spring 2014.
Here is the self-assessment of the team effort following the grid given at http://www.inria.fr/institut/organisation/instances/commission-d-evaluation.

- **(A5)** Audience : 5 – Moose is used by several research groups, a consulting company, and some companies using it in ad-hoc ways.
- **(SO4)** Software originality : 4 – Moose aggregates the last results of several research groups.
- **(SM4)** Software Maturity : 4 – Moose is developed since 1996 and got two main redesign phases.
- **(EM4)** Evolution and Maintenance : 4 – Moose will be used as a foundation of our Synectique start up so its maintenance is planned.
- **(SDL4)** Software Distribution and Licensing : 4 – Moose is licensed under BSD
- **(OC)** Own Contribution : (Design/Architecture)DA-4, (Coding/Debugging)-4, (Maintenance/Support)-4, (Team/Project Management)-4

5.2. Pharo

Participants: Marcus Denker [correspondant], Damien Cassou, Stéphane Ducasse, Esteban Lorenzano, Damien Pollet, Igor Stasenko, Camillo Bruni, Camille Teruel, Clément Bera.

Web: http://www.pharo.org/

The platform. Pharo is a new open-source Smalltalk-inspired language and environment. It provides a platform for innovative development both in industry and research. By providing a stable and small core system, excellent developer tools, and maintained releases, Pharo's goal is to be a platform to build and deploy mission critical applications.

The first stable version, Pharo 1.0, was released in 2010. The development of Pharo accelerated in 2011 and 2012: Versions 1.2 to 1.4 have been released (with more than 2400 closed issues). In 2013, Pharo 2.0 was released. The development cycle will now be one major release per year, with Pharo3 to be released in March 2014.

In 2012, RMoD organized the first *Pharo Conference* during two days in May with 60 participants, the second Pharo conference was held in Bern, Switzerland in 2013.

Additionally, in November 2012 RMoD launched the Pharo Consortium (http://consortium.pharo.org/) and the Pharo Association (http://association.pharo.org). Over 10 companies are now paying members of the Consortium.

RMoD is the main maintainer and coordinator of Pharo.
Here is the self-assessment of the team effort following the grid given at http://www.inria.fr/institut/organisation/instances/commission-d-evaluation.

- **(A5)** Audience: 5 – Used in many universities for teaching, more than 25 companies.
- **(SO3)** Software originality : 3 – Pharo offers a classical basis for some aspects (UI). It includes new frameworks and concepts compared to other Smalltalk implementations.
- **(SM4)** Software Maturity: 4 – Bug tracker, continuous integration, large test suites are on place.
- **(EM4)** Evolution and Maintenance: 4 – Active user group, consortium and association had just been set up.
- **(SDL4)** Software Distribution and Licensing: 4 – Pharo is licensed under MIT.
- **(OC5)** Own Contribution: (Design/Architecture) DA-5, (Coding/Debugging) CD-5, (Maintenance/Support) MS-5, (Team/Project Management) TPM-5

5.3. Fuel

Participants: Martin Dias [Correspondant], Mariano Martinez-Peck.

Web: http://rmod.lille.inria.fr/web/pier/software/fuel

Objects in a running environment are constantly being born, mutating their status and dying in the volatile memory of the system. The goal of serializers is to store and load objects either in the original environment or in another one. Fuel is a general-purpose serializer based on four principles: (1) speed, through a compact binary format and a pickling algorithm which obtains the best performance on materialization; (2) good object-oriented design, without any special help from the virtual machine; (3) specialized for Pharo, so that core objects (such as contexts, block closures and classes) can be serialized too; (4) flexible about how to serialize each object, so that objects are serialized differently depending on the context.

Since Pharo 2.0, Fuel is part of the standard distribution.
Here is the self-assessment of the team effort following the grid given at http://www.inria.fr/institut/organisation/instances/commission-d-evaluation.

- **(A4)** Audience: 4 – Large audience software, usable by people inside and outside the field with a clear and strong dissemination, validation, and support action plan.
- **(SO3)** Software originality : 3.
- **(SM4)** Software Maturity: 4 – Bug tracker, continuous integration, large test suites are on place.
- **(EM4)** Evolution and Maintenance: 4.
- **(SDL4)** Software Distribution and Licensing: 4 – Fuel is licensed under MIT.
- **(OC5)** Own Contribution: (Design/Architecture) DA-5, (Coding/Debugging) CD-5, (Maintenance/Support) MS-5, (Team/Project Management) TPM-5

5.4. Athens

Participant: Igor Stasenko [Correspondant].

Athens is a vector graphics framework for Pharo. Athens is now part of Pharo since version 3 as a technology Preview. We plan to make Athens the default graphics framework with Pharo4 in 2015.

5.5. Citezen

Participants: Damien Pollet [Correspondant], Stéphane Ducasse.

Web: http://people.untyped.org/damien.pollet/software/citezen/

Citezen is a suite of tools for parsing, validating, sorting and displaying BibTeX databases. This tool suite is integrated within the Pier Content Management System (CMS) and both are implemented on top of Pharo. Citezen aims at replacing and extending BibTeX, in Smalltalk; ideally, features would be similar to BibTeX, CrossTeX, and CSL.

5.6. Handles

Participant: Jean-Baptiste Arnaud [Correspondant].

Web: http://jeanbaptiste-arnaud.eu/handles/

An Handle is a first-class reference to a target object. Handles can alter the behavior and isolate the state of the target object. Handles provide infrastructure to automatically create and wrap new handles when required. A real-time control of handles is possible using a special object called metaHandle.

5.7. Hazelnut

Participants: Guillermo Polito [Correspondant], Benjamin Van Ryseghem, Nicolas Paez, Igor Stasenko.

Web: http://rmod.lille.inria.fr/web/pier/software/Seed

Traditionally, Smalltalk-based systems are not bootstrapped because of their ability to evolve by self-modification. Nevertheless, the absence of a bootstrap process exposes many problems in these systems, such as the lack of reproducibility and the impossibility to reach certain evolution paths. Hazelnut is a tool that aims to introduce a bootstrap process into these systems, in particular Pharo.

5.8. LegacyParsers

Participants: Muhammad Bhatti [Correspondant], Nicolas Anquetil, Guillaume Larcheveque, Esteban Lorenzano, Gogui Ndong.

As part of our research on legacy software and also for the Synectique company), we started to define several parsers for old languages like Cobol for example. This work is important to help us validate our meta-model and tools against a larger range of existing technologies and to discover the limits of our approach. From our initial results, and the in-depth understanding that it gave us, we are formulating new research objectives in meta-model driven reverse engineering. This work is also important for the spin-off company, as being able to work with such technologies is fundamental.

5.9. Mate

Participants: Marcus Denker [Correspondant], Clement Bera, Camillo Bruni.

Mate is the future research-oriented virtual machine for Pharo. Its goal is to serve as a prototype for researchers to experiment with. As a result, the design of Mate is very simple to understand. As of today, Mate consists of an AST interpreter, a new object memory layout, and a simple garbage collector.

5.10. NativeBoost

Participant: Igor Stasenko [Correspondant].

Web: http://code.google.com/p/nativeboost/

NativeBoost is a Smalltalk framework for generating and running machine code from the language side of Pharo. As part of it comes a foreign function interface that enables calling external C functions from Smalltalk code with minimal effort.

5.11. Nabujito

Participants: Camillo Bruni [Correspondant], Marcus Denker.

Nabujito is an experimental Just In Time compiler implemented as a Smalltalk application, based on NativeBoost, that does not require changes in the virtual machine.

5.12. Nautilus

Participants: Benjamin Van Ryseghem [Correspondant], Stéphane Ducasse, Igor Stasenko, Camillo Bruni, Esteban Lorenzano.

Nautilus is a new source code browser based on the latest infrastructure representations. Its goal is mainly to replace the current system browser that was implemented in the 80s and that doesn't provide optimal tools for the system as it has evolved.

5.13. Spec

Participants: Benjamin Van Ryseghem [Correspondant], Stéphane Ducasse, Johan Fabry.

Spec is a programming framework for generating graphical user interfaces inspired by VisualWorks' Subcanvas. The goal of Spec is to tackle the lack of reuse experienced in existing tools. Spec serves as a pluggable layer on top of multiple lower-level graphical frameworks. Many improvements have been noticed in Pharo after the introduction of Spec in terms of speed or number of lines of code while we re-implemented existing tools using Spec.

ROMA Team

5. Software and Platforms

5.1. MUMPS

Participants: Patrick Amestoy, Alfredo Buttari, Jean-Yves L'Excellent [correspondent], Wissam M. Sid-Lakhdar, Bora Uçar.

MUMPS (for *MUltifrontal Massively Parallel Solver*) see http://mumps-solver.org is a software package for the solution of large sparse systems of linear equations. It implements a direct method, the so called multifrontal method; it is a parallel code capable of exploiting distributed-memory computers as well as multithreaded libraries; its main originalities are its numerical robustness and the wide range of functionalities available.

The latest public release is MUMPS 4.10.0 (May 2011).

The development of MUMPS was initiated by the European project PARASOL (Esprit 4, LTR project 20160, 1996-1999), whose results and developments were public domain. Since then, MUMPS has been supported by CERFACS, CNRS, ENS Lyon, INPT(ENSEEIHT)-IRIT, Inria, and University of Bordeaux. Following a contractual agreement signed by those institutes, the next release of MUMPS will be distributed under the Cecill-C license; a technical committee was also defined, currently composed of Patrick Amestoy, Abdou Guermouche, and Jean-Yves L'Excellent.

In the context of an ADT project (Action of Technological Development), Maurice Brémond (from Inria "SED" service in Grenoble) also worked part-time on the project, in particular on visualization tools helping researchers to analyze the behaviour of a parallel MUMPS execution.

More information on MUMPS is available on http://mumps-solver.org. See also Section 6.20 of this report.

RUNTIME Project-Team

5. Software and Platforms

5.1. Common Communication Interface

Participant: Brice Goglin.

- The *Common Communication Interface* aims at offering a generic and portable programming interface for a wide range of networking technologies (Ethernet, InfiniBand, ...) and application needs (MPI, storage, low latency UDP, ...).
- CCI is developed in collaboration with the *Oak Ridge National Laboratory* and several other academics and industrial partners.
- CCI is in early development and currently composed of 19 000 lines of C.
- http://www.cci-forum.org

5.2. Hardware Locality

Participants: Brice Goglin, Samuel Thibault.

- *Hardware Locality* (HWLOC) is a library and set of tools aiming at discovering and exposing the topology of machines, including processors, cores, threads, shared caches, NUMA memory nodes and I/O devices.
- It builds a widely-portable abstraction of these resources and exposes it to the application so as to help them adapt their behavior to the hardware characteristics.
- HWLOC targets many types of high-performance computing applications [2], from thread scheduling to placement of MPI processes. Most existing MPI implementations, several resource managers and task schedulers already use HWLOC.
- HWLOC is developed in collaboration with the OPEN MPI project. The core development is still mostly performed by Brice GOGLIN and Samuel THIBAULT from the RUNTIME team-project, but many outside contributors are joining the effort, especially from the OPEN MPI and MPICH2 communities.
- HWLOC is composed of 30 000 lines of C.
- http://runtime.bordeaux.inria.fr/hwloc/

5.3. Network Locality

Participant: Brice Goglin.

- *Netloc Locality* (HWLOC) is a library that extends hwloc to network topology information by assembling hwloc knowledge of server internals within graphs of inter-node fabrics such as Ethernet or Infiniband.
- HWLOC targets the same challenges as hwloc but focuses on a wider spectrum by enabling cluster-wide solutions such process placement.
- HWLOC is developed in collaboration with University of Wisconsin in LaCrosse and Cisco, within the OPEN MPI project.
- NETLOC is composed of 15 000 lines of C.
- http://netloc.org

5.4. KNem

Participant: Brice Goglin.

- KNEM (*Kernel Nemesis*) is a Linux kernel module that offers high-performance data transfer between user-space processes.
- KNEM offers a very simple message passing interface that may be used when transferring very large messages within point-to-point or collective MPI operations between processes on the same node.
- Thanks to its kernel-based design, KNEM is able to transfer messages through a single memory copy, much faster than the usual user-space two-copy model.
- KNEM also offers the optional ability to offload memory copies on INTEL I/O AT hardware which improves throughput and reduces CPU consumption and cache pollution.
- KNEM is developed in collaboration with the MPICH2 team at the Argonne National Laboratory and the OPEN MPI project. These partners already released KNEM support as part of their MPI implementations.
- KNEM is composed of 8 000 lines of C. Its main contributor is Brice GOGLIN.
- http://runtime.bordeaux.inria.fr/knem/

5.5. Open-MX

Participant: Brice Goglin.

- The OPEN-MX software stack is a high-performance message passing implementation for any generic ETHERNET interface.
- It was developed within our collaboration with Myricom, Inc. as a part of the move towards the convergence between high-speed interconnects and generic networks.
- OPEN-MX exposes the raw ETHERNET performance at the application level through a pure message passing protocol.
- While the goal is similar to the old GAMMA stack [42] or the recent iWarp [41] implementations, OPEN-MX relies on generic hardware and drivers and has been designed for message passing.
- OPEN-MX is also wire-compatible with Myricom MX protocol and interface so that any application built for MX may run on any machine without Myricom hardware and talk other nodes running with or without the native MX stack.
- OPEN-MX is also an interesting framework for studying next-generation hardware features that could help ETHERNET hardware become legacy in the context of high-performance computing. Some innovative message-passing-aware stateless abilities, such as multiqueue binding and interrupt coalescing, were designed and evaluated thanks to OPEN-MX [5].
- Brice GOGLIN is the main contributor to OPEN-MX. The software is already composed of more than 45 000 lines of code in the Linux kernel and in user-space.
- http://open-mx.org/

5.6. StarPU

Participants: Olivier Aumage, Andra Hugo, Nathalie Furmento, Raymond Namyst, Marc Sergent, Samuel Thibault, Pierre-André Wacrenier.

- STARPU permits high performance libraries or compiler environments to exploit heterogeneous multicore machines possibly equipped with GPGPUs or Xeon Phi processors.
- STARPU offers a unified offloadable task abstraction named codelet.In case a codelet may run on heterogeneous architectures, it is possible to specify one function for each architectures (e.g. one function for CUDA and one function for CPUs).

- STARPU takes care to schedule and execute those codelets as efficiently as possible over the entire machine. A high-level data management library enforces memory coherency over the machine: before a codelet starts (e.g. on an accelerator), all its data are transparently made available on the compute resource.
- STARPU obtains portable performances by efficiently (and easily) using all computing resources at the same time.
- STARPU also takes advantage of the heterogeneous nature of a machine, for instance by using scheduling strategies based on auto-tuned performance models.
- STARPU can also leverage existing parallel implementations, by supporting *parallel tasks*, which can be run concurrently over the machine.
- STARPU provides *scheduling contexts* which can be used to partition computing resources. Scheduling contexts can be dynamically resized to optimize the allocation of computing resources among concurrently running libraries.
- STARPU provides integration in MPI clusters through a lightweight DSM over MPI.
- STARPU provides a scheduling platform, which makes it easy to implement and experiment with scheduling heuristics
- STARPU comes with a plug-in for the GNU Compiler Collection (GCC), which extends languages of the C family with syntactic devices to describe STARPU's main programming concepts in a concise, high-level way.
- STARPU provides a scheduling platform, which makes it easy to implement and experiment with scheduling heuristics
- http://runtime.bordeaux.inria.fr/StarPU/

5.7. NewMadeleine

Participant: Alexandre Denis.

- NEWMADELEINE is communication library for high performance networks, based on a modular architecture using software components.
- The NEWMADELEINE optimizing scheduler aims at enabling the use of a much wider range of communication flow optimization techniques such as packet reordering or cross-flow packet aggregation.
- NEWMADELEINE targets applications with irregular, multiflow communication schemes such as found in the increasingly common application conglomerates made of multiple programming environments and coupled pieces of code, for instance.
- It is designed to be programmable through the concepts of optimization *strategies*, allowing experimentations with multiple approaches or on multiple issues with regard to processing communication flows, based on basic communication flows operations such as packet merging or reordering.
- The reference software development branch of the NEWMADELEINE software consists in 90 000 lines of code. NEWMADELEINE is available on various networking technologies: Myrinet, Infiniband, Quadrics and ETHERNET. It is developed and maintained by Alexandre DENIS.
- http://runtime.bordeaux.inria.fr/newmadeleine/

5.8. PadicoTM

Participant: Alexandre Denis.

- PadicoTM is a high-performance communication framework for grids. It is designed to enable various middleware systems (such as CORBA, MPI, SOAP, JVM, DSM, etc.) to utilize the networking technologies found on grids.
- PadicoTM aims at decoupling middleware systems from the various networking resources to reach transparent portability and flexibility.
- PadicoTM architecture is based on software components. Puk (the PadicoTM micro-kernel) implements a light-weight high-performance component model that is used to build communication stacks.
- PadicoTM component model is now used in NEWMADELEINE. It is the cornerstone for networking integration in the projects “LEGO” and “COOP” from the ANR.
- PadicoTM is composed of roughly 60 000 lines of C.
- PadicoTM is registered at the APP under number IDDN.FR.001.260013.000.S.P.2002.000.10000.
- http://runtime.bordeaux.inria.fr/PadicoTM/

5.9. MAQAO

Participants: Denis Barthou, Olivier Aumage, Tamara Meunier.

- MAQAO is a performance tuning tool for OpenMP parallel applications. It relies on the static analysis of binary codes and the collection of dynamic information (such as memory traces). It provides hints to the user about performance bottlenecks and possible workarounds.
- MAQAO relies on binary codes for Intel x86 and ARM architectures. For x86 architecture, it can insert probes for instrumention directly inside the binary. There is no need to recompile. The static/dynamic approach of MAQAO analysis is the main originality of the tool, combining performance model with values collected through instrumentation.
- MAQAO has a static performance model for x86 and ARM architectures. This model analyzes performance of the codes on the architectures and provides some feed-back hints on how to improve these codes, in particular for vector instructions.
- The dynamic collection of data in MAQAO enables the analysis of thread interactions, such as false sharing, amount of data reuse, runtime scheduling policy, ...
- MAQAO is in the European FP7 project ”MontBlanc” and in the Samsung GRO project ”Gepetto”.
- http://www.maqao.org/

5.10. QIRAL

Participants: Denis Barthou, Olivier Aumage.

- QIRAL is a high level language (expressed through LaTeX) that is used to described Lattice QCD problems. It describes matrix formulations, domain specific properties on preconditionings, and algorithms.
- The compiler chain for QIRAL can combine algorithms and preconditionings, checking validity of the composition automatically. It generates OpenMP parallel code, using libraries, such as BLAS.
- This code is developped in collaboration with other teams participating to the ANR PetaQCD project.

5.11. TreeMatch

Participants: Emmanuel Jeannot, Guillaume Mercier, François Tessier.

- TREEMATCH is a library for performing process placement based on the topology of the machine and the communication pattern of the application.
- TREEMATCH provides a permutation of the processes to the processors/cores in order to minimize the communication cost of the application.
- Important features are : the number of processors can be greater than the number of applications processes ; it assumes that the topology is a tree and does not require valuation of the topology (e.g. communication speeds) ; it implements different placement algorithms that are switched according to the input size.
- Some core algorithms are parallel to speed-up the execution.
- TREEMATCH is integrated into various software such as the Charm++ programming environment as well as in both major open-source MPI implementations: Open MPI and MPICH2.
- TREEMATCH is available at: http://treematch.gforge.inria.fr.

SCORE Team

4. Software and Platforms

4.1. Rivage

Participants: Claudia-Lavinia Ignat, Stéphane Martin [contact].

Rivage (Real-tIme Vector grAphic Group Editor) is a real-time collaborative graphical editor. Several users can edit at the same time and in real-time a graphical document, user changes being immediately seen by the other users. The editor relies on a peer-to-peer architecture where users can join and leave the group at any time. Each user has a copy of the shared document and user changes on the document copies are merged in real-time by using a CRDT (Commutative Replicated Data Type) algorithm. The code is available at https://github.com/stephanemartin/rivage.

4.2. Replication Benchmarker

Participants: Pascal Urso [contact], Mehdi Ahmed-Nacer, Stéphane Martin, Gérald Oster.

The Replication Benchmarker is a performance evaluation framework for optimistic replication mechanisms used in collaborative applications. It contains a library of implementation of several CRDT (Commutative Replicated Data Type) and OT (Operational Transformation) algorithms for different data types: text, set, trees. The framework is able to evaluate the performance of comparable algorithms on different corpus of events traces. These events traces can be produced randomly according to different parameters, can be extracted from real real-time editing session that have been recorded, or can be automatically extracted from distributed version control repositories such as the one produced with Git. Performances of the algorithms are measured in term of execution time, memory footprint and merge result quality (compared to manual merge history stored in git repositories). The source code of this evaluation framework is available at https://github.com/score-team/replication-benchmarker.

4.3. BeGoood

Participant: Gérôme Canals.

BeGoood is a generic system for managing non-regression tests on knowledge-bases. BeGoood allows to define test plans in order to monitor the evolution of knowledge-bases. Any system answering queries by providing results in the form of set of strings can be tested with BeGoood. BeGoood has been developed following a REST architecture and is independent of any application domain. BeGoood is a part of the Kolflow infrastructure and is available at https://github.com/kolflow.

SOCRATE Project-Team

5. Software and Platforms

5.1. WSnet

Socrate is an active contributor to WSnet (http://wsnet.gforge.inria.fr/) a multi-hop wireless network discrete event simulator. WSnet was created in the ARES team and it is now supported by the D-NET team of Inria Rhône-Alpes.

5.2. Wiplan

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

5.3. FloPoCo

The purpose of the open-source FloPoCo project is to explore the many ways in which the flexibility of the FPGA target can be exploited in the arithmetic realm. FloPoCo is a generator of operators written in C++ and outputting synthesizable VHDL automatically pipelined to an arbitrary frequency. In 2013, a CORDIC-based arctangent was written in Socrate.

Among the known users of FloPoCo are U. Bristol,U. Cape Town, U.T. Cluj-Napoca, Imperial College, U. Essex, U. Madrid, U. P. Milano, T.U. Muenchen, T. U. Kaiserslautern, U. Paderborn, CalTech, U. Pernambuco, U. Perpignan, U. Tohoku, U. Tokyo, Virginia Tech U. and several companies.

Web page: http://flopoco.gforge.inria.fr/

TRISKELL Project-Team

5. Software and Platforms

5.1. Kermeta

Participants: Didier Vojtisek [correspondant], Olivier Barais, Arnaud Blouin, Benoit Combemale, Fabien Coulon, Thomas Degueule, François Fouquet, David Mendez Acuna, Clément Guy, Jean-Marc Jézéquel.

Nowadays, object-oriented meta-languages such as MOF (Meta-Object Facility) are increasingly used to specify domain-specific languages in the model-driven engineering community. However, these meta-languages focus on structural specifications and have no built-in support for specifications of operational semantics. Integrated with the industrial standard Ecore and aligned with the OMG standard EMOF 2.0, the Kermeta language consists in a extension to these meta languages to support behavior definition. The language adds precise action specifications with static type checking and genericity at the meta level. Based on object-orientation and aspect orientation concepts, the Kermeta language adds model specific concepts.

It is used in several use cases:

- to give a precise semantic of the behavior of a metamodel which then can be simulated.
- to act as a model transformation language.
- to act as a constraint language.

The development environment built for the Kermeta language provides an integrated workbench based on Eclipse. It offers services such as : model execution, text editor (with syntax highlighting, code autocompletion), additional views and various import/export transformations.

Thanks to Kermeta it is possible to build various frameworks dedicated to domain specific metamodels. Those frameworks are organised into MDKs (Model Development Kits). For example, Triskell proposes MDKs to work with metamodels such as Java5, UML2, RDL (requirements), Ecore, Traceability,...

After a first refactoring of Kermeta in 2011 to ease the integration of EMF and to focus on a fully compiled mode, we did a new refactoring of Kermeta in 2013 to leverage on xTend. The Kermeta action language is now defined as an extension of xTend proposing model-specific features (e.g., model type, containment, opposite) and an open class mechanism for aspect weaving. The main objective of this new refactoring was to benefit from the model-non-specific features of xTend (including the basics of the action language and its respective tooling such as editor, type checker and compiler), and to focus in our development on the innovative solutions for MDE.

Especially, in addition to an xTend extension dedicated to model manipulation, we started to integrate in Kermeta various facilities to support a software language engineering (slicing, pruning, reuse, variability management...).

Moreover, while this version of Kermeta is a DSML development workbench that provide good support for developing independent DSMLs, little or no support is provided for integrated use of multiple DSMLs. The lack of support for explicitly relating concepts expressed in different DSMLs makes it very difficult for developers to reason about information spread across models describing different system aspects.

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

- APP: IDDN.FR.001.420009.000.S.P.2005.000.10400
- Version: 2.0.1
- Programming language: Java, Scala, Kermeta

Main competitors:

- XMF-Mosaic is developed by Ceteva and is now open-source since 2008.
- GME is a large scale Meta-Modeling Environment developed at Vanderbilt University (ISIS project) since 2002.
- MOFLON is a Metamodeling Framework with Graph Transformations, developed by A. Schuerr's group (TU-Darmstadt) since 2008.
- XCore is a recent (2011) Eclipse project supported by Itemis/Macro Modelling that provides a single operational surface syntax for Ecore.
- Many QVT inspired model transformation tools focused on model transformations.

Main innovative features:

Kermeta was one of the first solutions to offer an operational semantics on top of EMOF. It still proposes several unique features that cannot be found in the tools presented above, such as:

- aspect weaving at the metamodel level allows fast prototyping of a wide variety of tools;
- model typing allows a safe model polymorphism (e.g., reuse of algorithms and transformations accross diffrent metamodels), as well as language inheritance, evolution and interoperability.

Impact:

Kermeta is already quite well used by the community as a research platform for trying MDE ideas both in the academic community and in corporate R&D. Many softwares tools are built on top of Kermeta either within the Triskell team, within other Inria teams or in other companies and research institutes:

- The following tools have been built within the Triskell team : K-CVL (implementation of the OMG CVL standard), Kompren (model slicing tool), Malai, Pramana. Kermeta is also used in all the collaborative projects Triskell is involved with, and is the catalyst of many collaborations in industrial contracts.
- The following tools have been built using Kermeta (or use some transformations written in Kermeta) in other Inria teams:
  - Gecos (CAIRN): C compiler infrastructure following the Model Driven Engineering. It leverages the Eclipse Modeling Framework and uses Eclipse as an underlying infrastructure. Consequently, the grammar of the source languages and the intermediate representations become metamodels, and the compilation passes become model transformations.
  - Timesquare (AOSTE) is a language based on the formal Clock Constraint Specification Language (CCSL), which allows the manipulation of logical time.
  - Polychrony (ESPRESSO) is a toolset for a polychronous data-flow language (Signal)
- The following tools have been built using Kermeta outside of Inria:
  - Modhel'x (Supelec) is a framework for simulating multi-formalism models.
  - RAM (Mc Gill University) Reusable Aspect Models is an aspect-oriented multi-view modeling approach that integrates class diagram, sequence diagram and state diagram AOM techniques.

Since 2008, we invested a large effort to transfer these concepts in industry and the standardization bodies. Especially, we have initiated some collaborations with the Eclipse Foundation and OMG to include some Kermeta concepts (model typing, static introduction, ECORE/OCL/Kermeta composition, etc.) in the MXF project proposal [1] of the Eclipse Modeling Project.

According to google scholar [2], the Kermeta platform was used or cited in more than 800 papers. It has been downloaded about 1000 times per year since 2006[3].

---

[1] cf. http://www.eclipse.org/proposals/mxf

[2] http://scholar.google.fr/scholar?q=kermeta+model

[3] according to the unique visitors count on the Kermeta update site. Cf. http://kermeta.org/awstats.pl?month=all&year=2010&output=main&config=kermeta.org

5.2. Kevoree

Participants: Olivier Barais [correspondant], François Fouquet, Erwan Daubert, Jean-Émile Dartois, Johann Bourcier, Noël Plouzeau, Maxime Tricoire, Francisco-Javier Acosta Padilla, Jacky Bourgeois, Mohamed Boussaa, Antonio de Mattos, Thomas Degueule, Inti Gonzalez Herrera, Tam Le Nhan, Ivan Paez Anaya.

Kevoree is an open-source models@runtime platform [4] to properly support the dynamic adaptation of distributed systems. Models@runtime basically pushes the idea of reflection [85] one step further by considering the reflection layer as a real model that can be uncoupled from the running architecture (e.g. for reasoning, validation, and simulation purposes) and later automatically resynchronized with its running instance.

Kevoree has been influenced by previous work that we carried out in the DiVA project [85] and the Entimid project [86]. With Kevoree we push our vision of models@runtime [84] farther. In particular, Kevoree provides a proper support for distributed models@runtime. To this aim we introduced the *Node* concept to model the infrastructure topology and the *Group* concept to model semantics of inter node communication during synchronization of the reflection model among nodes. Kevoree includes a *Channel* concept to allow for multiple communication semantics between remote*Components* deployed on heterogeneous nodes. All Kevoree concepts (Component, Channel, Node, Group) obey the object type design pattern to separate deployment artifacts from running artifacts. Kevoree supports multiple kinds of very different execution node technology (e.g. Java, Android, MiniCloud, FreeBSD, Arduino, ...).

Kevoree is distributed under the terms of the LGPL open source license.

Main competitors:

- the Fractal/Frascati eco-system [5].
- SpringSource Dynamic Module [6]
- GCM-Proactive [7]
- OSGi [8]
- Chef [9]
- Vagran [10]

Main innovative features:

- distributed models@runtime platform (with a distributed reflection model and an extensible models@runtime dissemination set of strategies).
- Support for heterogeneous node type (from Cyber Physical System with few resources until cloud computing infrastructure).
- Fully automated provisioning model to correctly deploy software modules and their dependencies.
- Communication and concurrency access between software modules expressed at the model level (not in the module implementation).

Impact:

A tutorial have been performed at the Middleware conference in december 2013.

Several European projects leveraging the Kevoree platform have recently been accepted. Besides we are currently developing a testbed named DAUM. This testbed is developed since mid 2011 to experiment with Kevoree in real life situations. More precisely, DAUM is a highly dynamic pervasive system that mixes wireless smart sensors, user interaction devices such as digital pads, and distributed data servers in a cloud.

---

[4] http://www.kevoree.org

[5] http://frascati.ow2.org

[6] http://docs.spring.io/osgi/docs/1.2.1/reference/html/

[7] http://proactive.inria.fr/

[8] http://www.osgi.org

[9] http://wiki.opscode.com/display/chef/Deploy+Resource

[10] http://vagrantup.com/

The current specialization of DAUM is a distributed tactical information and decision system for firefighters. This application includes individual sensors in the personal protective equipment of firefighters, embedded computation nodes that are fully reconfigurable in real time and over the air, distributed monitoring servers in trucks, and personal computers for information access and decision making. The DAUM platform is used internally to try research results on distributed *models@runtime*. DAUM is used externally to prepare and support cooperation activities with other research teams (the Myriads Inria team is a partner of DAUM) and with potential industrial partners.

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

- Version: 1.0
- Programming language: Java, Scala, Kermeta, Kotlin, Javascript

5.3. FAMILIAR

Participants: Mathieu Acher [correspondant], Olivier Barais, Guillaume Bécan, Aymeric Hervieu, Julien Richard-Foy, Sana Ben Nasr, Edward Mauricio Alferez Salinas, João Ferreira Filho, Didier Vojtisek, Benoit Baudry.

Modeling and reasoning about configuration options is crucial for the effective management of configurable software systems and product lines. The FAMILIAR project provides dedicated languages, APIs, and comprehensive environments for that purpose. Specifically, FAMILIAR provides support for feature models (by far the most popular notation). There are more than 20 years of research [75] and the formalism of feature models is widely used in the industry [76]. FAMILIAR (for FeAture Model scrIpt Language for manIpulation and Automatic Reasoning) provides a scripting language for importing, exporting, composing, decomposing, editing, configuring, computing "diffs", refactoring, reverse engineering, testing, and reasoning about (multiple) feature models. For interoperability, many bridges with existing feature modeling languages are implemented. All these operations can be combined to realize complex variability management tasks: extraction of feature models from software artifacts [23], product line evolution [35], management of multiple models [34], model-based validation of SPLs [49], large scale configuration of feature models [68], etc. The level of maturity of the Familiar platform is TRL 3 (New technology tested Prototype built and functionality demonstrated through testing over a limited range of operating conditions. These tests can be done on a scaled version if scalable)

Main competitors:

- FAMA
- TVL
- Clafer
- pure::variants

Main innovative features:

- reverse engineering of variability models from multiple kinds of artefacts
- composition of multiple variability models (e.g., for combining different sources of variability)
- slicing of variability model (e.g., for scheduling a configuration process in different steps)
- connection with the Common Variability Language (CVL)

Impact:

The results are connected to the CVL standardization initiative. From a research perspective, FAMILIAR helps to support all the research activity on variability modeling (e.g., design of new operators, benchmarking). Several tutorials have been performed at SPLC (the major conference in software product lines), at ECOOP, at CIEL and MODELS in 2012 and 2013. FAMILIAR is also used in the context of teaching activities. From an industrial perspective, the languages and tools have already been applied in practical contexts in different application domains (medical imaging, video surveillance, system engineering, web configurators, etc.) and for various purposes. This platform is also used for supporting the transfer activity with company such as Thales or Kereval. FAMILIAR is currently involved in different research projects (in the Merge Itea project, in the MOTIV project, in the VaryMDE project).

FAMILIAR is distributed under the terms of the LGPL and EPL open source license.

See also the web page http://familiar-project.github.com.

- Version: 1.2
- Programming language: Java, Scala

URBANET Team

5. Software and Platforms

5.1. WSNet

UrbaNet is an active contributor to WSnet (http://wsnet.gforge.inria.fr/), a discrete event simulator dedicated to large scale wireless networks developed and maintained by members of Inria and CITI lab. A major part of this contribution is represented by the implementation of state of the art protocols for medium access control and routing.

The WSNet simulation results obtained following this process are sometimes used as an input for another part of our development effort, which consists in prototype software based on the combination of CPLEX and AMPL for solving mixed integer linear programming problems with column generation.

5.2. TAPASCologne vehicular mobility dataset

Based on the data made available by the Institute of Transportation Systems at the German Aerospace Center (ITS-DLR), the dataset aims at reproducing, with a high level of realism, car traffic in the greater urban area of the city of Cologne, Germany. To that end, different state-of-art data sources and simulation tools are brought together, so to cover all of the specific aspects required for a proper characterization of vehicular traffic:

- The street layout of the Cologne urban area is obtained from the OpenStreetMap (OSM) database;
- The microscopic mobility of vehicles is simulated with the Simulation of Urban Mobility (SUMO) software;
- The traffic demand information on the macroscopic traffic flows across the Cologne urban area (i.e., the O/D matrix) is derived through the Travel and Activity PAtterns Simulation (TAPAS) methodology;
- The traffic assignment of the vehicular flows described by the TAPASCologne O/D matrix over the road topology is performed by means of Gawron’s dynamic user assignment algorithm.

The resulting synthetic trace of the car traffic in a the city of Cologne covers a region of 400 square kilometers for a period of 24 hours, comprising more than 700.000 individual car trips. More information is available on the project website at http://kolntrace.project.citi-lab.fr/ .

ALICE Project-Team

4. Software and Platforms

4.1. Vorpaline

Participants: Dobrina Boltcheva, Bruno Lévy, Thierry Valentin.

Vorpaline is an automatic surfacic and volumetric mesh generation software, distributed with a commercial license. Vorpaline is based on the main scientific results stemming from projets GoodShape and VORPALINE, funded by the European Research Council, about optimal quantization, centroidal Voronoi diagrams and fast/parallel computation of Voronoi diagrams in high-dimension space. The current version (1.0) provides functionalities such as isotropic/adaptive/anisotropic surface re-meshing, tolerant surface re-meshing, mesh repair and mesh decimation. Next versions will provide functionalities such as constrained surface meshing (2.0), quad-dominant surface meshing (3.0) and hex-dominant volume meshing (4.0).

4.2. IceSL

Participants: Jérémie Dumas, Jean Hergel, Sylvain Lefebvre.

In the new software IceSL, we propose to exploit recent advances in GPU and Computer Graphics to accelerate the slicing process of objects modelled via a CSG [1] language. Our target are open source low cost *fused deposition modeling* printers such as RepRaps.

Our approach first inputs a CSG description of a scene which can be composed of both meshes and analytic primitives. During display and slicing the CSG model is converted on the fly into an intermediate representation enabling fast processing on the GPU. Slices can be quickly extracted, and the tool path is prepared through image erosion. The interactive preview of the final geometry uses the exact same code path as the slicer, providing an immediate, accurate visual feedback.

IceSL is the recipient software for our ERC research project “ShapeForge”, led by Sylvain Lefebvre.

../../../../projets/alice/IMG/IceSL-modeled-trivase.jpg

../../../../projets/alice/IMG/IceSL-prin

Figure 1. Left. A two-colored vase is modeled in IceSL. Right. An early printed result.

4.3. Graphite

Participants: Dobrina Boltcheva, Samuel Hornus, Bruno Lévy, David Lopez, Romain Merland, Jeanne Pellerin, Nicolas Ray.

---

[1] Constructive Solid Geometry

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. Graphite is one of the common software platforms used in the frame of the European Network of Excellence AIMShape .

Graphite and its research-plugins are actively developed and extended. The latest version was released on January 2nd, 2014 and has been downloaded 135 times as of January 29.

4.4. GraphiteLifeExplorer

Participant: Samuel Hornus.

GLE is a 3D modeler, developed as a plugin of Graphite, dedicated to molecular biology. It is developed in cooperation with the Fourmentin Guilbert foundation and has recently been renamed "GraphiteLifeExplorer". Biologists need simple spatial modeling tools to help in understanding the role of the relative position of objects in the functioning of the cell. In this context, we develop a tool for easy DNA modeling. The tool generates DNA along any user-given curve, open or closed, allows fine-tuning of atoms position and, most importantly, exports to PDB (the Protein Daba Bank file format).

The development of GLE is currently on hold, but it is still downloaded (freely) about twice a day (1600 downloads to date). A paper describing it was published in the broad journal PLOS One [12].

4.5. OpenNL - Open Numerical Library

Participants: 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 [5] 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 Saboret, 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.

4.6. 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 Méditerrannée and the ALICE / Inria Nancy Grand-Est teams.

ALPAGE Project-Team

5. Software and Platforms

5.1. Syntax

Participants: Pierre Boullier [correspondant], Benoît Sagot.

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

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

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

5.2. DyALog

Participant: Éric Villemonte de La Clergerie [maintainer].

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

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

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

The current release handles logic programs, DCGs (*Definite Clause Grammars*), FTAGs (*Feature Tree Adjoining Grammars*), FTIGs (*Feature Tree Insertion Grammars*) and XRCGs (*Range Concatenation Grammars* with logic arguments). Several extensions have been added to most of these formalisms such as intersection, Kleene star, and interleave operators. Typed Feature Structures (TFS) as well as finite domains may be used for writing more compact and declarative grammars [101].

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

DYALOG is largely used within ALPAGE to build parsers but also derivative softwares, such as a compiler of Meta-Grammars (cf. 5.3). It has also been used for building FRMG, a parser from a large coverage French TIG/TAG grammar derived from a Meta-Grammar. This parser has been used for the Parsing Evaluation campaign EASy, the two Passage campaigns (Dec. 2007 and Nov. 2009), cf. [99], [100], and very large amount of data (700 millions of words) in the SCRIBO project. New results concerning FRMG are described in 6.11 .

A new statistical dependency parser, based on a shift-reduce algorithm, was also developed in 2013 within the DYALOG system (see 6.12).

DYALOG and other companion modules are available on Inria GForge.

5.3. Tools and resources for Meta-Grammars

Participant: Éric Villemonte de La Clergerie [maintainer].

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

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

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

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

The current version of `mgcomp` has been used to compile a wide coverage Meta-Grammar FRMG (version 2.0.1) to get a grammar of around 200 TAG trees [12]. Without the use of guarded nodes and regular operators, this grammar would have more than several thousand trees and would be almost intractable. FRMG has been packaged and is freely available.

To ease the design of meta-grammars, a set of tools have been implemented, mostly by Éric Villemonte De La Clergerie, and collected in MGTOOLS (version **2.2.2**). This package includes a converter from a compact format to a XML pivot format, an Emacs mode for the compact and XML formats, a graphical viewer interacting with Emacs and XSLT stylesheets to derive HTML views.

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

5.4. The Bonsai PCFG-LA parser

Participants: Marie-Hélène Candito [correspondant], Djamé Seddah, Benoît Crabbé.

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

Alpage has developed as support of the research papers [60], [53], [54], [11] a statistical parser for French, named Bonsai, trained on the French Treebank. This parser provides both a phrase structure and a projective dependency structure specified in [4] as output. This parser operates sequentially: (1) it first outputs a phrase structure analysis of sentences reusing the Berkeley implementation of a PCFG-LA trained on French by Alpage (2) it applies on the resulting phrase structure trees a process of conversion to dependency parses using a combination of heuristics and classifiers trained on the French treebank. The parser currently outputs several well known formats such as Penn treebank phrase structure trees, Xerox like triples and CONLL-like format for dependencies. The parsers also comes with basic preprocessing facilities allowing to perform elementary sentence segmentation and word tokenisation, allowing in theory to process unrestricted text. However it is believed to perform better on newspaper-like text. See 6.12 for recent work and results involving Bonsai.

The parser is available under a GPL license.

5.5. The MICA parser

Participants: Benoît Sagot [correspondant], Pierre Boullier.

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

MICA (Marseille-Inria-Columbia- AT&T) is a freely available dependency parser [48] currently trained on English and Arabic data, developed in collaboration with Owen Rambow and Daniel Bauer (Columbia University) and Srinivas Bangalore (AT&T). MICA has several key characteristics that make it appealing to researchers in NLP who need an off-the-shelf parser, based on Probabilistic Tree Insertion Grammars and on the SYNTAX system. MICA is fast (450 words per second plus 6 seconds initialization on a standard high-end machine) and has close to state-of-the-art performance (87.6% unlabeled dependency accuracy on the Penn Treebank).

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

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

5.6. Alpage's linguistic workbench, including SxPipe

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

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

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

SxPipe [80] is a modular and customizable chain aimed to apply to raw corpora a cascade of surface processing steps. It is used

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

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

- Support of new languages, and most notably German (although this is still at a very preliminary stage of development;
- Analysis of unknown words, in particular in the context of the ANR project EDyLex and of the collaboration with *viavoo*; this involves in particular (i) new tools for the automatic pre-classification of unknown words (acronyms, loan words...) (ii) new morphological analysis tools, most notably automatic tools for constructional morphology (both derivational and compositional), following the results of dedicated corpus-based studies (see 6.2 for new results);
- Development of new local grammars for detecting new types of entities and improvement of existing ones, in the context of the PACTE project (see 6.7 for new results).

5.7. MElt

Participants: Benoît Sagot [correspondant], Pierre Magistry.

MElt is a part-of-speech tagger, initially developed in collaboration with Pascal Denis (Magnet, Inria — then at Alpage), which was trained for French (on the French TreeBank and coupled with the Le*fff*), also trained on English [63], Spanish [69], Italian [94], German, Dutch, Polish, Kurmanji Kurdish [104] and Persian [89], [90]. It is state-of-the-art for French.

It is now able to handle noisy corpora (French and English only).

MElt also includes a lemmatization post-processing step.

A specific effort has been made towards the usability of MElt by linguists. In particular, a training session has been organized, and a user guide has been written.

Moreover, a preliminary version of MElt which accepts input DAGs has been developed.

MElt is distributed freely as a part of the Alpage linguistic workbench.

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

Participants: Benoît Sagot [correspondant], Laurence Danlos.

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

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

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

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

Alexina also hosts *Aleda* [98], [88] a large-scale entity database currently developed for French but under development for English, Spanish and German, extracted automatically from Wikipedia and Geonames. It is used among others in the SxPipe processing chain and its NP named entity recognition, as well as in the NOMOS named entity linking system.

5.9. The free French wordnet WOLF

Participants: Benoît Sagot [correspondant], Sarah Beniamine.

The WOLF (Wordnet Libre du Français) is a wordnet for French, i.e., a lexical semantic database. The development of WOLF started in 2008 [82], [83]. At this time, we focused on benefiting from available resources of three different types: general and domain-specific bilingual dictionaries, multilingual parallel corpora and Wiki resources (Wikipedia and Wiktionaries). This work was achieved in a large part in collaboration with Darja Fišer (University of Ljubljana, Slovenia), in parallel with the development of a free Slovene wordnet, sloWNet. However, it was also impacted by specific collaborations, e.g., on adverbial synsets [84].

In 2013, a beta version of a new version of WOLF (version 1.0b2) was published, which integrates and extends the various efforts performed and published somewhat independently in 2012.

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

5.10. OGRE (Optimized Graph Rewriting Engine)

Participants: Corentin Ribeyre [correspondant], Djamé Seddah, Éric Villemonte de La Clergerie, Marie-Hélène Candito.

OGRE (Optimized Graph Rewriting Engine) is a graph rewriting system specifically designed for manipulating linguistic trees and graphs [78]. It relies on a rule specification language for expressing graph rewriting patterns. The transformation is performed in two steps:

1. First, the system performs simple transformations following the rewriting patterns;
2. Second, constraints can be applied on edges, which applies transformations depending on their environment that are propagated while all constraints are satisfied.

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

Web site: http://www.corentinribeyre.fr/projects/view/OGRE

5.11. Automatic construction of distributional thesauri

Participants: Marie-Hélène Candito [correspondant], Enrique Henestroza Anguiano.

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

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

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

5.12. Tools and resources for time processing

Participant: Laurence Danlos [correspondant].

Alpage developed the *French TimeBank*, a freely-available corpus annotated with ISO-TimeML-compliant temporal information (dates, events and relations between events) [1].

5.13. LexViz

Participants: Mikael Morardo [maintainer], Éric Villemonte de La Clergerie.

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

5.14. Mgwiki

Participants: Paul Bui Quang [maintainer], Éric Villemonte de La Clergerie.

In the context of Inria ADT Mgwiki, Paul Bui Quang has developed a linguistic wiki that may used to discuss linguistic phenomena with the possibility to add annotated illustrative sentences. The work is essentially devoted to the construction of an instance for documenting and discussing FRMG, with the annotations of the sentences automatically provided by parsing them with FRMG. This instance also offers the possibility to parse small corpora with FRMG and an interface of visualization of the results.

Another instance was deployed for managing the annotation guide for the Deep version of the Sequoia treebank, confirming the potential of the notion of linguistic wiki

5.15. NewsProcess

Participants: Éric Villemonte de La Clergerie [maintainer], Damien Nouvel.

NewsProcess is an HTTP-based service that may be used to process AFP news through the Alpage Processing Chain, in order to extract information, in particular citations. The chain has been completed to track the emergence of new words in the news.

In the context on ANR EdyLex, a new version of NewsProcess has been designed for processing AFP news wires and extracting information about unknown words (see 6.2)

5.16. System EasyRef

Participant: Éric Villemonte de La Clergerie [maintainer].

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

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

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

EASYREF is maintained under Inria GForge.

AVIZ Project-Team

5. Software and Platforms

5.1. Graph Cuisine

Participants: Évelyne Lutton [correspondant], Benjamin Bach, André Spritzer, Jean-Daniel Fekete.

`../../../../projets/aviz/IMG/GraphCuisine.png`

Figure 1. GraphCuisine interface showing one real graph (right), the measures extracted from it (bottom circles), several graphs with similar measures (left) and one of them selected (middle).

GraphCuisine lets users steer an Evolutionary Algorithm (EA) to create random graphs that match user-specified measures. Generating random graphs with particular characteristics is crucial for evaluating graph algorithms, layouts and visualization techniques. Current random graph generators provide limited control of the final characteristics of the graphs they generate. The situation is even harder when one wants to generate random graphs similar to a given one, all-in-all leading to a long iterative process that involves several steps of random graph generation, parameter changes, and visual inspection. Our system follows an approach based on interactive evolutionary computation. Fitting generator parameters to create graphs with pre-defined measures is an optimization problem, while assessing the quality of the resulting graphs often involves human subjective judgment. GraphCuisine has been proved to be able to generate graphs that mimic a given real-world network. http://www.aviz.fr/Research/Graphcuisine

5.2. Histomages

Participants: Fanny Chevalier, Pierre Dragicevic [correspondant], Christophe Hurter.

Histomages is an image editor based on a new interaction model that considers histogram views as spatial rearrangements of image pixels. Users can select pixels on image histograms as they would select image regions and directly manipulate them to adjust their colors. Histomages are affected by other image tools such as paintbrushes. We explored some possibilities offered by this interaction model, and discussed the four key principles behind it as well as their implications for the design of feature-rich software in general. http://www.aviz.fr/histomages/.

5.3. Gliimpse

Participants: Pierre Dragicevic [correspondant], Stéphane Huot, Fanny Chevalier.

Gliimpse is a quick preview technique that smoothly transitions between document markup code (HTML, LaTeX,...) and its visual rendering. This technique allows users to regularly check the code they are editing in-place, without leaving the text editor. This method can complement classical preview windows by offering rapid overviews of code-to-document mappings and leaving more screen real-estate. A proof-of-concept editor can be downloaded for free at http://www.aviz.fr/gliimpse/.

5.4. The Obvious Toolkit

Participants: Pierre-Luc Hémery, Jean-Daniel Fekete [correspondant].

Information Visualization, Java, Toolkit

The Obvious Toolkit is a new Interactive Graphics Toolkit written in Java to facilitate the interoperability between Information Visualization toolkits and components (Fig. 4).

The Obvious Toolkit is an abstraction layer above visualization toolkits. Currently, it connects the most popular toolkits in Java: Prefuse, the InfoVis Toolkit, Improvise, JUNG, as well as other libraries such as the Java Database Communication Toolkit (JDBC) and two Machine-Learning toolkits: Weka and RapidMiner.

It is meant to provide an abstraction layer for information visualization application builders so that they can postpone their choice of a concrete toolkit to use. When faced with the final choice, application builders can use one of the toolkits or connect all of them dynamically to Obvious. A paper on Obvious was presented at the IEEE Visual Analytics Science and Technology conference (VAST 2011) [61]. Obvious is available at http://code.google.com/p/obvious.

5.5. GeneaQuilts

Participants: Jean-Daniel Fekete [correspondant], Pierre Dragicevic, Anastasia Bezerianos, Julie Bae, Ben Watson.

../../../../projets/aviz/IMG/hiistograms-teaser.png

Figure 2. Example of sky enhancement with Histomages: (a) the image is duplicated and its pixels rearranged into a lightness histogram ; (b) bright pixels are selected with the rubber-band selection tool; (c) all pixels are rearranged into a hue histogram and yellow pixels are filtered out with the subtract selection brush (bottom). Missing pixels are added with the add selection brush on the image (top); (d) the sky is enhanced by resizing the selection on the saturation histogram.

../../../../projets/aviz/IMG/gliimpse.png

Figure 3. Gliimpse: A detail of the animation between an article and its LaTeX source code.

../../../../projets/aviz/IMG/obvious.png

Figure 4. The Obvious toolkit showing the same graph with a Prefuse and an IVTK rendering.

GeneaQuilts [2] is a new genealogy exploration software that allows genealogists and historians to visualize and navigate in large genealogies of up to several thousand individuals (Fig. 5). The visualization takes the form of a diagonally-filled matrix, where rows are individuals and columns are nuclear families. The GeneaQuilts system includes an overview, a timeline, search and filtering components, and a new interaction technique called Bring & Slide that allows fluid navigation in very large genealogies. The tool has been featured in several InfoVis and genealogy Websites and the website has been visited over 9000 times. It has been integrated in commercial and open-source implementations (4 to date). See also the web page http://www.aviz.fr/geneaquilts/.

5.6. Diffamation

Participants: Fanny Chevalier, Pierre Dragicevic [correspondant], Anastasia Bezerianos, Jean-Daniel Fekete.

Animation, Edit histories, Wikipedia, Revision Control

The Diffamation system [3] allows rapid exploration of revision histories such as Wikipedia or subversion repositories by combining text animated transitions with simple navigation and visualization tools. Diffamation can be used for example to get a quick overview of the entire history of a Wikipedia article or to see what has happened to one's contributions. Diffamation complements classical diff visualizations: once moments of interest have been identified, classical diff visualizations can come in useful to compare two given revisions in detail.

The Diffamation revision exploration system has been presented at the plenary session of the Ubuntu Developer Summit. It is available at http://www.aviz.fr/diffamation/.

5.7. The InfoVis Toolkit

Participant: Jean-Daniel Fekete [correspondant].

Information Visualization, Java, Toolkit

../../../../projets/aviz/IMG/geneaquilts.png

Figure 5. The genealogy of the Simpsons family (left) and of the Greek Pantheon (right), produced by the GeneaQuilts software.

`../../../../projets/aviz/IMG/diffamation.png`

Figure 6. Screenshot the Diffamation system during a transition: (a) the document view, (b) the overview scrollbar and (c) the timeline.

The InfoVis Toolkit [60] is an Interactive Graphics Toolkit written in Java to facilitate the development of Information Visualization applications and components.

The InfoVis Toolkit implements several visualization techniques, as well as interaction techniques related. It has been used for teaching the Information Visualization course (Masters level, Univ. of Paris-Sud) and is the basis for all AVIZ contracts. It is our main development platform for information visualization; most of our Information Visualization prototypes rely on it. It is available at http://ivtk.sourceforge.net.

In the forthcoming years, it will be superseded by extensions of the Obvious Toolkit (see section 5.4).

5.8. GraphDice

Participants: Jean-Daniel Fekete [correspondant], Pierre Dragicevic, Niklas Elmqvist, Anastasia Bezerianos.

GraphDice [1] is a visualization system for exploring multivariate networks (Fig. 7). GraphDice builds upon our previous system ScatterDice (best paper award at the IEEE InfoVis 2008 conference) [59]: it shows a scatter plot of 2 dimensions among the multiple ones available and provides a very simple paradigm of 3D rotation to change the visualized dimensions. The navigation is controlled by a scatter plot matrix that is used as a high-level overview of the dataset as well as a control panel to switch the dimensions.

While ScatterDice works on any tabular dataset (e. g., CSV file), the GraphDice system show networks using a node-link diagram representation as a scatter plot with links drawn between connected nodes. For more information, see the web page at http://graphdice.gforge.inria.fr.

../../../../projets/aviz/IMG/graphdice.png

Figure 7. Screenshot the GraphDice system.

AYIN Team

5. Software and Platforms

5.1. Deposits

- The software SAMD (Semi-Automatic Melanoma Detection) V1.0 was deposited with the APP in December 2013. It has been tested on public databases.
- The software SAAD (Semi-Automatic Acne Detection) V1.0 was deposited with the APP in December 2013. It has been tested on public databases as well as on data sets provided by CHU Nice and Galderma.

DAHU Project-Team (section vide)

DREAM Project-Team

5. Software and Platforms

5.1. Introduction

The pieces of software described in this section are prototypes implemented by members of the project. Any interested person should contact relevant members of the project.

5.2. QTempIntMiner: quantitative temporal sequence mining

QTEMPINTMINER (Quantitative Temporal Interval Miner) is a data mining (cf. 3.2.2) software that implements several algorithms presented in [59] and [3].

The software is mainly implemented in Matlab. A standalone application is now avalaible. It uses the Mixmod toolbox [44] to compute multi-dimensional Gaussian distributions. The main features of QTEMPINTMINER are:

- a tool for generating synthetic noisy sequences of temporal events,
- an implementation of the QTEMPINTMINER, QTIAPRIORI and QTIPREFIXSPAN algorithms,
- a graphical interface that enables the user to generate or import data set and to define the parameters of the algorithm and that displays the extracted temporal patterns.
- a sequence transformer to process long sequences of temporal events. Long sequences are transformed into a database of short temporal sequences that are used as input instances for the available algorithms.

The following website gives many details about the algorithms and provides the latest stable implementation of QTEMPINTMINER: http://www.irisa.fr/dream/QTempIntMiner/.

5.3. Sacadeau: qualitative modeling and decision-aid to preserve the water quality from pollutants as herbicides

SACADEAU is an environmental decision software (cf. 4.3) that implements the SACADEAU transfer model.The SACADEAU simulation model couples two qualitative models, a transfer model describing the pesticide transfer through the catchment and a management model describing the farmer decisions. Giving as inputs a climate file, a topological description of a catchment, and a cadastral repartition of the plots, the SACADEAU model simulates the application of herbicides by the farmers on the maize plots, and the transfer of these pollutants through the catchment until the river. The two main simulated processes are the runoff and the leaching. The output of the model simulation is the quantity of herbicides arriving daily to the stream and its concentration at the outlets. The originality of the model is the representation of water and pesticide runoffs with tree structures where leaves and roots are respectively up-streams and down-streams of the catchment.

The software allows the user to see the relationships between these tree structures and the rules learnt from simulations (cf. 3.2.1). A more elaborated version allows to launch simulations and to learn rules on-line. This year, we have developed this new version by enabling access to two recommendation action algorithms.The user can choose different parameters (set of classification rules from which actions will be built, parameters concerning action feasibility, etc) before asking for action recommending process, and then easily visualize the characteristics of situations to improve (polluted ones) compared with the different recommended actions. The software is mainly in Java.

The following website is devoted to the presentation of the SACADEAU: http://www.irisa.fr/dream/SACADEAU/. See also [10] for a presentation.

5.4. Ecomata

EcoMata is a tool-box for qualitative modeling and exploring ecosystems and for aiding to design environmental guidelines.We have proposed a new qualitative approach for ecosystem modeling (cf. 4.3) based on timed automata (TA) formalism combined to a high-level query language for exploring scenarios.

To date, EcoMata is dedicated to ecosystems that can be modeled as a collection of species (prey-predator systems) under various human pressures and submitted to environmental disturbances. It has two main parts: the Network Editor and the Query Launcher. The Network Editor let a stakeholder describe the trophic food web in a graphical way (the species icons and interactions between them). Only few ecological parameters are required and the user can save species in a library. The number of qualitative biomass levels is set as desired. An efficient algorithm generates automatically the network of timed automata. EcoMata provides also a dedicated window to help the user to define different fishing pressures, a nice way being by using chronograms. In the Query Launcher, the user selects the kind of query and the needed parameters (for example the species biomass levels to define a situation). Results are provided in a control panel or in files that can be exploited later. Several additional features are proposed in EcoMata: building a species library, import/export of ecosystem model, batch processing for long queries, etc. EcoMata is developed in Java (Swing for the GUI) and the model-checker called for the timed properties verification is UPPAAL.

The following website is devoted to the presentation of ECOMATA: http://oban.agrocampus-ouest.fr:8080/ecomata.

5.5. Paturmata

Paturmata is a tool-box for qualitative modeling and exploring agrosystems, specifically management of herd based on pasture [6]. The system is modelled using a hierarchical hybrid model described in timed automata formalism.

In PaturMata software, users can create a pasture system description by entering herds and plots information. For each herd, the only parameter is the number of animals. For each plot, users should enter the surface, the density, the herb height, the distance to the milking shed, a herb growth profile and an accessibility degree .

Users then specify pasturing and fertilization strategies. Finally, users can launch a pasture execution. PaturMata displays the results and a detailed trace of pasture. Users can launch a batch of different strategies and compare the results in order to find the best pasture strategy.

PaturMata is developed in Java (Swing for the GUI) and the model-checker that is called for the timed properties verification is UPPAAL.

Another feature which will be soon added to PaturMata is strategy synthesis. Users choose a pasture configuration or a type of pasture configuration and PaturMata proposes the best pasture and fertilization strategy in order to minimize the pasture procedure cost and use of nitrogen fertilizer.

5.6. ManageYourself

ManageYourself is a collaborative project between Dream and the Telelogos company aiming at monitoring smartphones from a stream of observations made on the smartphone state (cf. 3.2.3).

Today’s smartphones are able to perform calls, as well as to realize much more complex activities. They are small computers. But as in computers, the set of applications embedded on the smartphone can lead to problems. The aim of the project ManageYourself is to monitor smartphones in order to avoid problems or to detect problems and to repair them.

The ManageYourself application includes three parts :

- A monitoring part which triggers preventive rules at regular time to insure that the system is working correctly, e.g. *if the memory is full then delete the tmp directory*. This part is always running on the smartphone.

- A reporting part which records regularly the state of the smartphone (the memory state - free vs allocated -, the connection state, which applications are running, etc.). This part also is always running on the smartphone. The current state is stored in a report at regular period and is labeled *normal*. When an application or the system bugs, the current buggy state is stored in a report and is labeled *abnormal*. At regular timestamps, all the reports are sent to a server where the learning process is executed.
- A learning part which learns new bug rules from the report dataset. This part is executed offline on the server. Once the bug rules are learnt, human experts translates them into preventive rules which are downloaded and integrated in the monitoring part of the smartphones.

The following website is devoted to the presentation of MANAGEYOURSELF: http://www.irisa.fr/dream/ManageYourself/Site/ManageYourself.html.

5.7. GeoImageRMP: a RapidMiner extension to georeferenced data

RapidMiner is one of the most used frontend for data mining, modelling and analysis. RapidMiner enables the user to design data processing tool chains interactively. A tool chain is a flow chart of processing tools represented by boxes in the interface. This software is easily extendable by designing Plugins. The GeoImageRMP plugin is a plugin dedicated to the design of tool chains to process georeferenced images (raster and vector images) [18]. It is a practical and useful respond to the analytic tasks of georeferenced data. This is the first plugin that is interested in including georeferenced data in RapidMiner and although the only user-friendly tool to create and compare geoereferenced data tool chains. It benefits from the large amount of data processing tools that are already implemented in RapidMiner (classification, clustering, frequent pattern mining, etc.). One of the main aims of this plugin is to quickly prototype machine learning tools chain for remote sensing classification task. The GeoImageRMP plugin provide several new processing boxes:

- georeferenced data import/export: create and export dataset that can be processed by standard RapidMiner tool box from/to standard geospatial format (GeoTiff, Shapefiles)
- geospatial sampling method: based on multi-heterogeneous layers of georeferenced data, the sampling method can be transets, random, equidistant, from punctual layer.
- georeferenced data transformation tools: a set of tools dedicated to the manipulation of our new data structures (coordinates, SRS, etc.)
- visualization tool

The following website is devoted to the presentation of GEOIMAGERMP: http://geoimagermp.gforge.inria.fr/.

5.8. A plugin for visualizing and editing spatial graphs in QGis

Spatial graphs are accurate representations of spatial information through spatial objects linked by relationships (spatial or not). This representation is suited to the modeling and analysis of spatial information by computer processing (data mining, search for shortest paths, etc.). While Geographic information System (GIS), such as QGis, offers the possibility to visualize and manage georeferenced information, the use of spatial graph suffers from the lack of tools to facilitate the construction and integrated visualization.

We developed a QGis plugin for the visualization and the interactive construction of spatial graphs. QGis is the most used open source GIS. This plugin introduces a new type of layer: GraphLayer [16]. These new layers can be integrated into any GIS projects. They offer rich functionality for visualization and interactive editing.

5.9. Odisseptale: a software for implementing and evaluating sanitary event detectors in cattle

Odisseptale is a software for implementing disease detectors using monitoring of data provided by sensors placed on calves or cows. Sensors record streams of data such as body temperature, physical activity, feeding behavior, etc. These data are transmitted regularly to a monitoring software that aims to detect if a noticeable change has occurred on the data streams. Several detectors can be simultaneously active and each contribute to the final decision (detection of a disease). Two kinds of detectors have been implemented: a generic detector based on adaptive CUSUM and a symbolic pattern-based detector. Odisseptale provides also facilities for parameter setting and performance evaluation. This year, the software has been re-implemented in Python for enhanced portability and dissemination.

E-MOTION Project-Team

4. Software and Platforms

4.1. PROTEUS Software

Participants: Amaury Nègre, Juan Lahera-Perez.

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

The latest developments have been focuses on the robotics simulator. This simulator is based on the simulation and 3D rendering engine “mgEngine“ (http://mgengine.sourceforge.net/) embedded with the physics engine “bullets physics” (http://bulletphysics.org) for realistic robot dynamic simulation.

We also worked on the interface with the robotics middleware “ROS“ (http://www.ros.org) in order to offer interoperability with many robotics applications.

The simulator is now fully integrated with the robotics middleware "ROS" (http://www.ros.org) which allow interoperability with a large set of robotics applications and visualization tools. This software is developed in C++ and the simulator operates with the Lua scripting language. The simulation software is used in the ANR Proteus (http://www.anr-proteus.fr), as a simulation engine for the PROTEUS Toolkit.

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

4.2. AROSDYN

Participants: Igor Paromtchik, Mathias Perrollaz, Alexandros Makris, Amaury Nègre, Christian Laugier.

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

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

Another important property of this software is a large part of the computation task executed on GPU. As the processing of stereo image and the computaion in the BOF can be highly parallelized, we run these tasks on the GPU to improve the time performance. The GPU calculation is based on CUDA library and is carried out in an independent thread.

Furthermore, thanks to the design of the software, we can easily add new models to it and let them work together. The fast detection and tracking algorithm (FCTA) and the Gaussian process based collision assessment algorithm are added into this framework. The sofware is implemented on the Lexus car. In 2012, a demand for deposing the GPU BOF software to the APP is in progress.

4.3. Embedded Perception

Participants: Mathias Perrollaz, Amaury Nègre, Christian Laugier.

The method for computing occupancy grids from a stereoscopic sensor, developped in the e-motion team, has been implemented on GPU, using NVIDIA CUDA. This allows a real time implementation and an online processing within the Lexus experimental platform.

The program has been deposed to the APP in 2012, under the reference: IDDN.FR.001.270004.000.S.P.2012.000.10800

4.4. Bayesian Occupancy Filter

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

- Version: 1
- Patent: 0552736 (2005), 0552735 (2005)
- Programming language: C/C++

4.5. PROBT

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

EXMO Project-Team

5. Software and Platforms

5.1. Alignment API

Participants: Jérôme Euzenat [Correspondent], Jérôme David, Nicolas Guillouet, Armen Inants, Luz Maria Priego-Roche.

We have designed a format for expressing alignments in a uniform way [1]. The goal of this format is to share available alignments on the web. It should help systems using alignments, e.g., mediators, translators, to take advantage of any alignment algorithm and it will help alignment algorithms to be used in many different tasks. This format is expressed in RDF, so it is freely extensible.

The API itself [1] is a JAVA description of tools for accessing the common format. It defines five main interfaces (OntologyNetwork, Alignment, Cell, Relation and Evaluator) and proposes the following services:

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

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

- a base implementation of the interfaces with all useful facilities;
- a library of sample matchers;
- a library of renderers (XSLT, RDF, SKOS, SWRL, OWL, C-OWL, SPARQL);
- a library of evaluators (various generalisation of precision/recall, precision/recall graphs);
- a flexible test generation framework which allows for generating evaluation datasets;
- a library of wrappers for several ontology API;
- a parser for the format.

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

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

The Alignment API is used in the Ontology Alignment Evaluation Initiative data and result processing (§6.1.1). It is also used by more than 30 other teams worldwide.

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

5.2. The OntoSim library

Participants: Jérôme David [Correspondent], Jérôme Euzenat.

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

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

OntoSim is based on an ontology interface allowing for using ontology parsed with different APIs. It is written in Java and is available, under the LGPL licence, at http://ontosim.gforge.inria.fr.

FLOWERS Project-Team

5. Software and Platforms

5.1. Perception Tools

Participants: David Filliat [correspondant], Natalia Lyubova, Louis-Charles Caron, Alexander Gepperth.

5.1.1. Perception Abstraction Engine

Participants: David Filliat [correspondant], Natalia Lyubova.

PAE (Perception Abstraction Engine) is a C++ library developed to provide a uniform interface to existing visual feature detector such as SIFT, SURF, MSER, superpixels, etc... Its main goal is to be able to use these various feature detectors in a "bag of feature" approach for applications such as robot localisation and object recognition. Several approach are also implemented for the visual vocabularies, in particular the fast incremental vocabularies developed in the team.

The library provide common C++ interfaces to feature detectors, visual features and visual vocabularies. A factory approach make it possible to change the feature detectors and visual vocabularies types and parameters through configuration strings, without the need to recompile. Some applications are also included in the library, in particular topological robot localization (room recognition) and visual object recognition. An Urbi interface is also provided for these modules.

5.1.2. Incremental object discovery

Participants: Natalia Lyubova [correspondant], David Filliat.

This software makes it possible to detect, model and recognize objects in a scenario of interaction between a humanoid robot and a human teacher. It is based either on standard images, or on the kinect camera to take advantage of the depth information. The software is written in C++ and relies mainly on PAE and OpenCV.

The software implements several modules: candidate object segmentation based on motion information, keypoint-based object tracking, incremental object model construction integrating multiple features (keypoints + superpixels) and object categorisation based on mutual information with robot motors (making it possible to segment robot parts, objects and humans). Based on all these modules, it is possible for the robot to learn objects shown by a human partner and to improve the objects models by manipulating them when they are put in front of the robot.

5.1.3. Object recognition from a 3-D point cloud

Participants: Louis-Charles Caron [correspondant], Alexander Gepperth, David Filliat.

This software scans the 3-D point cloud of a scene to find objects and match them against a database of known objects. The process consists in 3 stages. The segmentation step finds the objects in the point cloud, the feature extraction computes discriminating properties to be used in the classification stage for object recognition.

The segmentation is based on simple assumptions about the geometry of an indoor scene. Successive RANSACs are used to find large planes, which correspond to the floor, ceiling and walls. The cloud is stripped from the points belonging to these planes. The remaining points are clustered, meaning that close-by points are considered to form a single object.

Objects are characterized by their shape and color. Color histograms and SIFT features are computed, using the PAE library, to capture the visual appearance of the objects. Their shape is encoded by computing thousands of randomly chosen SURFLET features to construct a relative frequency histrogram.

`../../../../projets/flowers/IMG/flowchart.png`

Figure 1. System Overview of the Incremental object discovery Software.

An early classification is done using each of the 3 features separately. For the color features a bag of words approach (from PAE) is used. For the shape feature, the minimum squared distance between the object's histogram and that of all objects in the database is calculated. Classification scores are then fused by a feed-forward neural network to get the final result [81].

5.1.4. *PEDDETECT: GPU-accelerated person detection demo*

Participant: Alexander Gepperth [correspondant].

PEDDETECT implements real-time person detection in indoor or outdoor environments. It can grab image data directly from one or several USB cameras, as well as from pre-recorded video streams. It detects mulitple persons in 800x600 color images at frame rates of >15Hz, depending on available GPU power. In addition, it also classifies the pose of detected persons in one of the four categories "seen from the front", "seen from the back", "facing left" and "facing right". The software makes use of advanced feature computation and nonlinear SVM techniques which are accelerated using the CUDA interface to GPU programming to achieve high frame rates. It was developed in the context of an ongoing collaboration with Honda Research Institute USA, Inc.

5.1.5. *A Python OptiTrack client*

Participant: Pierre Rouanet [correspondant].

This python library allows you to connect to an OptiTrack from NaturalPoint (http://www.naturalpoint.com/optitrack/). This camera permits the tracking of 3D markers efficiently and robustly. With this library, you can connect to the Motive software used by the OptiTrack and retrieve the 3D position and orientation of all your tracked markers directly from python.

5.2. Datasets

5.2.1. *Choreography dataset 1 and 2*

Participants: Olivier Mangin [correspondant], Haylee Fogg.

These databases contain choreography motions recorded through a kinect device. In the first dataset, these motions have a combinatorial structure: from a given set of primitive dance motions, choreographies are constructed as simultaneous execution of some of these primitive motions. Primitive dance motions are chosen from a total set of 48 motions and are spanned over one or two limbs, either the legs (e.g. walk, squat), left or right arm (e.g. wave hand, punch) or both arms (e.g. clap in hands, paddle). Complex choreographies are produced as the simultaneous demonstration of two or three of these primitive motion: either one for legs and one for both arm, or one for legs and one for each arm. The dataset has been used in the experiments from [104] for studying learning techniques allowing to identify dictionaries of motion primitives, and is publicly available at https://flowers.inria.fr/choreography_database.html.

The second dataset only contains choreographies composed of a single motion. It contains 110 records of each gesture from a set of 10 simple gestures and was used in the experiments from [53]. The dataset is publicly available at https://flowers.inria.fr/choreo2.

5.2.2. *Development on NoFish Platform*

Participants: Paul Fudal [correspondant], Sao Mai Nguyen.

NoFish (fig. 2) platform is a setup used by Mai Nguyen to perform severals experiments following her PhD work on social learning and intrinsic motivation. The setup consists to an ErgoRobot (fig. 3) with a fishing rod attached to the tip and a cap made with a red juggling ball. The robot is plugged on an ethernet power-switch used to turn it off if something wrong happens with the robot (for example : the robot try to go in a position he cannot reach and its motors are forcing too much). Tracking the cap is made with Full HD camera on the ceiling. At last, a video-projector prints informations on the floor (fig. 4) which helps for interactions between humans and the robots.

../../../../projets/flowers/IMG/NoFishPlatform.jpg

Figure 2. Illustration of the NoFishPlatform

../../../../projets/flowers/IMG/ErgoNoFish.jpg

Figure 3. Illustration of NoFish ErgoRobot

../../../../projets/flowers/IMG/NoFishFloor.jpg

Figure 4. NoFish cap and informations printed on the floor

Controlling the robot is made using URBI [1] scripts allowing different control level from single motor control to pre-programmed primitive (for example : reseting the robot or make it going to its starting position). URBI is also used to perform action through the power-switch if the robot must be turned off.

Tracking the cap is made by a program written using OpenCV and keep up to date the cap's coordinates by sending them to URBI through a network socket.

An other program written in JAVA and Processing [2] allows to print informations on the floor; it consists of a server receiving through a network socket texts and shapes informations to print on the floor and a JAVA object which can be used with compatible software. This component where really useful for giving simple and direct information during an experiment and also during interaction between a human and the robot for socially guided experiments.

The main program is written in Matlab and includes all the explained previous components and softwares. It also includes a forward kinematic calculation module used to ensure a movement sent can be safely be played by the robot to avoid it breaking himself; this module gives step by step informations of what will happen when the robot will play the movement like, for example, if the fishing cap will touch the floor at the end or not which permits to keep the robot safe and speed up experiments by ignoring dangerous moves for the robot and useless ones for the deployed algorithm.

This setup were used by Sao Mai Nguyen to run experiments during her thesis on social learning and intrinsic motivation

5.3. Learning algorithms

5.3.1. KidLearn

Participants: Manuel Lopes [correspondant], Benjamin Clement, Pierre-Yves Oudeyer, Didier Roy.

The KidLearn software provides and Intelligent Tutoring System that optimizes teaching sequences based on the estimated level of each particular student [65]. We implemented a Game of Money that allows students, ages 7-8, to learn how to use money. It includes 3 main components: i) a webserver that handles the requests and stores the experiments in a databased; ii) a GUI that provides the interface for the game; and iii) the optimization software.

Graphical interfaces in ITS can have unwanted side effects. For this reason, the interface was entirely designed with the help of a didactician, with several specific design choices motivated by pedagogical, motivational and attention requirements. For example, the interface, shown in Figure 5 . is such that:

- display is as clear and simple as possible;
- there is no chronometer, so that students are not put under time pressure;
- coins and banknotes have realistic visual appearance, and their relative sizes are respected;
- display of prices use visual encodings commonly used in shops;
- the zone for receiving money is automatically cleared in case of error after the student submits it;
- automatic snapping of money and tokens icons in the reception zone, and automatic visual arrangement;
- text quantity is kept to minimum;

Four principal regions are defined in the graphical interface, as shown in Figure 5 . The first is the wallet location where users can pick and drag the money to drop them on the repository location to make for the correct price. The object and the price are present in the object location, where the price can the written and/or spoken depending on the parameterization of the activity. The information location is using to display information for the learners such as extra clues when they make a mistake (for which they have to press the light bulb) and feedback. In order to improve the pedagogical success of the activity, the correct solution is presented automatically to the students if they fail to compose the correct price after 3 trials.

---

[1] http://www.gostai.com/products/jazz/urbi/

[2] http://processing.org

`../../../../projets/flowers/IMG/interf1.png` `../../projets/flowers/IMG/interf2.png`

Figure 5. Example interface with four principal regions of interaction: wallet location, repository, object and price, and feedback location.

5.3.2. RLPark - Reinforcement Learning Algorithms in JAVA

Participant: Thomas Degris [correspondant].

RLPark is a reinforcement learning framework in Java. RLPark includes learning algorithms, state representations, reinforcement learning architectures, standard benchmark problems, communication interfaces for three robots, a framework for running experiments on clusters, and real-time visualization using Zephyr. More precisely, RLPark includes:

- Online Learning Algorithms: Sarsa, Expected Sarsa, Q-Learning, On-policy and off-policy Actor-Critic with normal distribution (continuous actions) and Boltzmann distribution (discrete action), average reward actor-critic, TD, TD(λ), GTD(λ), GQ(λ), TDC
- State Representations: tile coding (with no hashing, hashing and hashing with mumur2), Linear Threshold Unit, observation history, feature normalization, radial basis functions
- Interface with Robots: the Critterbot, iRobot Create, Nao, Puppy, Dynamixel motors
- Benchmark Problems: mountain car, swing-up pendulum, random walk, continuous grid world

An example of RLpark running an online learning experiment on a reinforcement learning benchmark problem is shown in Figure 6 .

RLPark was started in spring 2009 in the RLAI group at the university of Alberta (Canada) when Thomas Degris was a postdoc in this group. RLPark is still actively used by RLAI. Collaborators and users include Adam White, Joseph Modayil and Patrick Pilarski (testing) from the University of Alberta.

RLPark has been used by Richard Sutton, a professor and iCORE chair in the department of computing science at the University of Alberta, for a demo in his invited talk *Learning About Sensorimotor Data* at the Neural Information Processing Systems (NIPS) 2011 [3]. Patrick Pilarski used RLPark for live demos on television (Breakfast Television Edmonton, CityTV, June 5th, 2012) and at TEDx Edmonton on Intelligent Artificial Limbs[4]. So far, RLPark has been used in more than a dozens of publications (see http://rlpark.github.com/publications.html for a list).

---

[3] http://webdocs.cs.ualberta.ca/~sutton/Talks/Talks.html#sensorimotor

[4] http://www.youtube.com/watch?v=YPc-Ae7zqSo

RLPark has been ported to C++ by Saminda Abeyruwan, a student of the University of Miami (United States of America). The Horde architecture in RLPark has been optimized for GPU by Clément Gehring, a student of the McGill University in Montreal (Canada).

Future developments include the implementation of additional algorithms (the Dyna architecture, back propagation in neural networks, ...). A paper is under review for the JMLR Machine Learning Open Source Software. Documentation and tutorials are included on the RLPark web site [5]. RLPark is licensed under the open source Eclipse Public License.

../../../../projets/flowers/IMG/ZephyrVectorViews.png

Figure 6. An example of an experiment in RLPark. Zephyr displays two views of a learned weight vector, an animation of the problem, the current policy distribution learned by the algorithm and the reward obtained by the algorithm. Videos are available at: http://rlpark.github.com.

5.3.3. DMP-BBO Matlab library

Participant: Freek Stulp [correspondant].

The dmp_bbo (Black-Box Optimization for Dynamic Movement Primitives) Matlab library is a direct consequence of the insight that black-box optimization outperforms reinforcement learning when using policies represented as Dynamic Movement Primitives. It implements several variants of the PI^{BB} algorithm for direct policy search. It is currently being used and extended by several FLOWERS members (Manuel Lopes, Clément Moulin-Frier) and external collaborators (Jonas Buchli, Hwangbo Jemin of ETH Zurich). This code was used for the following publications: [130], [127], [128].

---

[5] http://rlpark.github.com

5.3.4. Self-calibration BCI - Matlab library

Participants: Jonathan Grizou [correspondant], Iñaki Iturrate, Luis Montesano, Manuel Lopes, Pierre-Yves Oudeyer.

The Matlab software implements the algorithms described in [45]. It allows a robot to be instructed a new task by a human using communicative signals initially totally unknown to the robot. It is currently extended and improved in the context of EEG-based brain-machine interfaces (BMIs) [44].

It results in a BCI based control of sequential tasks with feedback signals that do not require any calibration process. As a by-product, the method provides an unsupervised way to train a decoder with the same performance of state-of-the-art supervised classifiers, while keeping the system operational and solving, with a lower performance during the first steps, the unknown task. The algorithm has been tested with online experiments (fig. 7), showing that the users were able to guide from scratch an agent to a desired position.

`../../../../projets/flowers/IMG/Evo_likelihood.jpg`

Figure 7. Results from the online BCI experiment for identifying the task. Evolution of the probability of the taught task for each subject and run

To improve the efficiency of the algorithm, we introduced a new planning method that uses the uncertainty in the signal-target estimation. This planner is inspired by exploration methods with exploration bonuses that allow guiding to reduce the uncertainty in an efficient way. We showed that trying to follow the best hypothesis does not explore the space significantly to reduce uncertainty and thus identify the correct task. Only through an approach that plans how to reduce the uncertainty multiple steps ahead are we sure that the agent will reach states that can only be explained by the correct hypothesis.

`../../../../projets/flowers/IMG/Avg_step_scratch.jpg`

Figure 8. Comparison between different exploration methods. Planning wrt. uncertainty in noth task and signal space is the most efficient method

5.3.5. PROPRE: simulation of developmental concept formation using PYTHON

Participant: Alexander Gepperth [correspondant].

This simulation software implements the algorithms described in [86], [83]. It is available online under the URL www.gepperth.net/downloads.html. The simulation is implemented in PYTHON for easy use, yet the time-critical core functions are written in C.

5.3.6. pyStreamPlayer: synchronized replay of multiple sensor recordings and supplementary data

Participant: Alexander Gepperth [correspondant].

This Python software is intended to facilitate the application of machine learning algorithms by avoiding to work directly with an embodied agent but instead with data recorded in such an agent. Assuming that non-synchronous data from multiple sensors (e.g., camera, Kinect, laser etc.) have been recorded according to a flexible format defined by the pyStreamPlayer architecture, pyStreamPlayer can replay these data while retaining the exact temporal relations between different sensor measurements. As long as the current task does not involve the generation of actions, this software allows to process sensor data as if it was coming from an agent which is usually considerably easier. At the same time, pyStreamPlayer allows to replay arbitrary supplementary information such as, e.g., object information, as if it was coming from a sensor. In this way, supervision information can be stored and accessed together with sensory measurements using an unified interface. pyStreamPlayer has been used to facilitate real-world object recognition tasks, and several of the major databases in this field (CalTech Pedestrian database, HRI RoadTraffic traffic objects database, CVC person database, KITTI traffic objects database) have been converted to the pyStreamPlaer format and now serve as a source of training and test data for learning algorithms.

pyStreamPlayer has been integrated into a ROS node as well, allowing th replay and transmission across networks of distributed processes.

5.3.7. *Multimodal: framework around the NMF algorithm for multimodal learning*

Participant: Olivier Mangin [correspondant].

The pyhton code provides a minimum set of tools and associated libraries to reproduce the experiments on [53], together with the choreography datasets. The code, publicly available at https://github.com/omangin/multimodal, under the new BSD license, is primarily intended for reproduction of the mulimodal learning experiment mentioned above. It is also expected that the public availability of the code encourages further experimentation by other scientists with data coming from other domains, thus increasing both the impact of the aforementioned publication and the knowledge on the algorithm behaviors. The nonnegative matrix factorization algorithm used in the experiments will also soon be included as a third party project to http://scikit-learn.org. Finally the code is currenlty being used by other members of the team and is expected to play an important role in further collaborations.

5.3.8. *Tools for curiosity-driven learning on a robotic arm*

Participants: Pierre Rouanet [correspondant], Clément Moulin-Frier.

This library is intended to provide tools for experimenting how curiosity model can facilitate learning of complex tasks such as manipulating objects. First, it provides high-level access to a robotic arm made of dynamixel motors: forward and inverse kinematics, demonstrations recording. Then it wraps the IMLE [77] library which we used for incremental and online learning of the sensorimotor mappings of the robot. Finally, it implements curiosity-driven learning based on the maximization of the learning progress. This modelling is based on recent works by Moulin-Frier and Oudeyer [54], [56], [55] proposing a probabilistic algorithmic architecture unifying various principles of developmental robotics such as motor babbling, goal babbling and curiosity-driven exploration. This architecture has already been successfully applied to model infant speech acquisition in the previously cited papers.

5.4. Software Platforms

5.4.1. *Meka robot plateform enhancement and maintenance*

Participants: Antoine Hoarau [ADT Engineer Since Nov. 2012], Freek Stulp [Supervisor], David Filliat [Supervisor].

Autonomous human-centered robots, for instance robots that assist people with disabilities, must be able to physically manipulate their environment. There is therefore a strong interest within the FLOWERS team to apply the developmental approach to robotics in particular to the acquisition of sophisticated skills for manipulation and perception. ENSTA-ParisTech has recently acquired a Meka (cf. 9) humanoid robot dedicated to human-robot interaction, and which is perfectly fitted to this research. The goal of this project is to install state-of-the-art software architecture and libraries for perception and control on the Meka robot, so that this robot can be jointly used by FLOWERS and ENSTA. In particular, we want to provide the robot with an initial set of manipulation skills.
The goal is to develop a set of demos, which demonstrate the capabilities of the Meka, and provide a basis on which researchers can start their experiments.

The platform is evolving as the sofware (Ubuntu, ROS, our code) is constantly updated and requires some maintenance so less is needed for later. A few demos were added, as the hand shaking demo, in which the robot detects people via kinect and initiates a hand shake with facial expressions. This demo has been used to setup a bigger human robot interaction experiment, currently tested on subjects at Ensta (cf. 10). The stacking cups demo is more of a manipulation and vision demo : the robot detects its cups by their shape and color, and tries to make a tower with it (cf. 12). This demo has required to manually update the old pr2 tabletop object detector to the new ROS version, create a tool to semi-automatically calibrate the extrinsicts parameters of the kinect, and different set of tools to catch objects from different angles (waiting for moveit to be fully integrated). Finally, we've seen that the robot itself also needs some maintenance; some components broke (a finger tendon), a welding got cold (in the arm) and a few cables experienced fatigue (led matrix and cameras) (cf. 11).
I've also given a talk at Humanoid 2013 at Altanta and the University of Texas at Austin for my participation on [61].

`../../../../projets/flowers/IMG/meka_robot.jpg`

Figure 9. The Meka robot plateform acquired by ENSTA ParisTech

5.4.2. Experiment platform for multiparameters simulations

Participants: Fabien Benureau, Paul Fudal.

../../../../projets/flowers/IMG/meka_hand_shake_demo.png

Figure 10. Hand shake demo visualized on Rviz (ROS)

../../../../projets/flowers/IMG/meka_repair.jpg

Figure 11. Maintenance is required on the robot

../../../../projets/flowers/IMG/meka_tower.png

Figure 12. The stacking cup demo (video available at http://cogrob.ensta-paristech.fr)

Simulations in robotics have many shortcommings. At the same time, they offer high customizability, rapidity of deployment, abscence of failure, consistency across time and scalability. In the context of the PhD work of Fabien Benureau, it was decided to investigate hypothesis first in simulation before moving to real hardware. In order to be able to test a high number of different hypothesis, we developped a software platform that would scale to the computing ressource available.

We designed simple continuous simulations around a of-the-shelf 2D physic engine and wrote a highly modular platform that would automatically deploy experiments on cluster environments, with proper handling of dependencies; our work investigate transfer learning, and some experiments's input data is dependent of the results of another.

So far, this platform and the university cluster has allowed to conduct thousands of simulations in parallel, totaling more than 10 years of simulation time. It has led us to present many diverse experiments in our published work [40], each repeated numerous times. It has allowed us to conduct a multi-parameter analysis on the setup, which led to new insights, which are being presented in a journal article to be submitted in the beginning of this year.

Because of its high modularity, this platform is proving to be highly flexible. We are currently adaptating it to a modified, cluster-ready, version of the V-REP simulator. Those simulations will serve to back ones on similar real-world hardware that are currently setup.

We have released the platform and the complete experiments code when we published the results of [40], allowing to reproduce the results of the paper, and will continue to do so with each published work.

5.4.3. PyPot

Participants: Pierre Rouanet [correspondant], Matthieu Lapeyre.

PyPot is a framework developed to make it easy and fast to control custom robots based on dynamixel motors. This framework provides different levels of abstraction corresponding to different types of use. More precisely, you can use PyPot to:

1. directly control robotis motors through a USB2serial device,
2. define the structure of your particular robot and control it through high-level commands,
3. define primitives and easily combine them to create complex behavior.

PyPot has been entirely written in Python to allow for fast development, easy deployment and quick scripting by non-necessary expert developers. It can also benefits from the scientific and machine learning libraries existing in Python. The serial communication is handled through the standard library and thus allows for rather high performance (10ms sensorimotor loop). It is crossed-platform and has been tested on Linux, Windows and Mac OS.

PyPot is part of the Poppy project (http://www.poppy-project.org) and has been released under an open source license GPL V3. More details are available on PyPot website: https://github.com/poppy-project/pypot

5.5. Experimental Setups

5.5.1. Experimental Platform for User Study of Curiosity-driven Exploration

Participants: Pierre Rouanet [correspondant], Jonathan Grizou, Brice Miard, Julie Golliot.

This platform has been developed to investigate curiosity-driven behaviors and more precisely how humans explore new sensori-motor spaces. It consists in several simple games where users control a 2D/3D shape with the movements of their body. They have to discover the mapping between their movements and a shape displayed on the screen and learn how to make the controlled shape match the target one (fig 13).

`../../../../projets/flowers/IMG/ESM.png`

Figure 13. A screenshot representing the game interface as seen by the user.

The software is entirelly written in Python. It includes a Kinect wrapper allowing the access of 3D position of tracked skeleton joints. It provides a framework for creating new games based on the 2D drawing library (pygame). It also includes a web server used to display game instructions, cut-scene videos and questionnaire.

5.6. Visualization Tools

5.6.1. Zephyr - Realtime Visualization in JAVA

Participant: Thomas Degris [correspondant].

Zephyr is a software to visualize numeric variables and data structure in real time and at different time scale. Zephyr is practical because it requires only minimal changes in the code: it uses Java reflexivity to automatically detect variables in the code to monitor and data structure with an associated dedicated view. Zephyr can easily be extended with new plugins because it is based on the popular Eclipse Rich Client Platform. Consequently, Zephyr takes advantage of an already existing and fully operational Eclipse plugins for many of its functionalities. Finally, Zephyr is distributed with a Java python virtual machine named Jython and a lisp implementation named Clojure. An example of a Zephyr screen is shown in Figure 14 .

Zephyr was started in fall 2009 in the RLAI group at the university of Alberta (Canada) when Thomas Degris was a postdoc in this group. Zephyr is still actively used by RLAI. Users include Adam White, Joseph Modayil and Patrick Pilarski from the University of Alberta. Zephyr has been registered on the Eclipse marketplace since October 2011. Documentation about Zephyr is included on its website: http://zephyrplugins.github.com. Zephyr is licensed under the open source Eclipse Public License.

5.6.2. Experimental Setups for User Study of Alignment in Asymmetric Interactions

Participants: Jonathan Grizou [correspondant], Chloé Rozenbaum, Manuel Lopes, Katharina Rohlfing, Pierre-Yves Oudeyer.

Figure 14. Left: Zephyr showing the different steps of a video processing pipeline in real-time. Right: Zephyr showing different data structure and variables of a reinforcement learning agent at different time scale. A video is available at: http://zephyrplugins.github.com.

This platform has been developed to investigate alignement in asymmetric interactions. We consider a remote construction task, where one user (user A) knows what to build but do not have access to the construction site while its partner (user B) is at the site but do not know what to do. By constraining the communicative channel between the two partners, we study how, and if, they will agree on a similar set of signals to convey information and what type of information they tend to produce.

The experimental setup consist of box with button, a video recording system and two screens. User A can send signals to user B by pressing buttons (fig. 15). Signals are displayed on a screen (fig. 15) at user B side. User A is not aware of what is displayed on user B screen, neither user B is aware of the relation between button presses and screen events. The video of user B construction scene is streamed to a screen at user B side.

The task consist of bulding arbitrary construction (fig. 15) using colored toy bricks (fig. 15).

Figure 15. Three examples of sign displayed on the learner screen; The box and the button use as an interface for the teacher to communicate with the learner; Examples of construction presented to the teacher.

5.7. Hardware

5.7.1. Poppy Platform

Participants: Matthieu Lapeyre [correspondant], Pierre Rouanet, Jonathan Grizou, Pierre-Yves Oudeyer [supervisor].

5.7.1.1. Main goals :

No current platform (Nao [87], Darwin Op [88], Nimbro Op [126], HRP-2, ...) does offer both a adapted morphology in the sense of allowing physical interaction (safe, compliant, playful) and optimized for walking. So to explore these challenges we have decided to build a new bio-inspired humanoid robotic platform, called Poppy, which provides some of the software and hardware features needed to explore both social interaction and biped locomotion for personal robot. It presents the following main features to make it an interesting platform to study how the combination of morphology and social interaction can help the learning:

- Design inspired from the study of the anatomy of the human body and its bio-mechanic
- Dynamic and reactive: we try to keep the weight of the robot as low as possible (geometry of the pieces and smaller motors)
- Social interaction: screen for communication and permits physical interaction thanks to compliance
- Study of the morphology of the leg to improve the biped walking
- Practical platform: low cost, ease of use and easy to reproduce

5.7.1.2. Overview :

Poppy platform (Figure 16) is a humanoid, it is 84cm tall for 3 kg. It has a large sensor motors space including 25 dynamical motors (MX-28 and AX-12), force sensors under its feet and some extra sensors in the head: 2 HD-wide angle-cameras, stereo-micros and an inertial central unit (IMU 9DoF) plus a large LCD Screen (4 inch) for visual communication (e.g. emotions, instructions or debug). The mechanical parts were designed and optimized to be as light as possible while maintaining the necessary strength. For this, the choice of a lattice beam structure manufactured with 3Dprinting polyamide was used.

The poppy morphology is designed based on the actual human body. We have deeply studied the biomechanics of the human body and have extracted some interesting features for humanoid robotics. This inspiration is expressed in the whole structure (e.g. the limb proportions) and in particular in the trunk and legs.

`../../../../projets/flowers/IMG/poppy_reel.png`

Figure 16. a. Global view of the Poppy platform. b. Zoom on legs design

Poppy uses the bio-inspired trunk system introduced by Acroban [101]. These five motors allow it to reproduce the main changes brought by the human spine. This feature allows the integration of more natural and fluid motion while improving the user experience during physical interactions. In addition, the spine plays a fundamental role in bipedal walking and postural balance by actively participating in the balancing of the robot.

The legs were designed to increase the stability and agility of the robot during the biped walking by combining bio-inspired, semi-passive, lightweight and mechanical-computation features. We will now describe two examples of this approach:

The architecture of the hips and thighs of Poppy uses biomechanical principles existing in humans. The human femur is actually slightly bent at an angle of about 6 degrees. In addition, the implantation of the femoral head in the hip is on the side. This results in a reduction of the lateral hip movement needed to move the center of gravity from one foot to another and a decrease in the lateral falling speed. In the case of Poppy, the inclination of its thighs by an angle of 6 degrees causes a gain of performance of more than 30% for the two above mentioned points.

Another example is Poppy's feet. Poppy has the particularity of having small feet compared to standard humanoids. It has humanly proportioned feet (ie about 15% of its total size).It is also equipped with compliant toes joints (see Figure 17 .a). We believe that this feet involve two keys features to obtain a human-like and efficient walking gait. However, that raises problems regarding balance because the support polygon is reduced. We decided to add pressure sensors under each foot in order to get accurate feedback of the current state of the robot (see Figure 17 .b).

`../../../../projets/flowers/IMG/pieds.png`

Figure 17. Poppy feet use actual children shoes combine with a compliant feet, toes (a.) and pressure sensors (b.)

5.7.1.3. Open source release :

To allow the distribution in the robotic community, we have decided to make Poppy an open platform. So the software and the hardware are open source. They have each a repository. The PyPot library is under GPLV3 license and is available on a bitbucket repository (https://github.com/poppy-project/pypot). The hardware, Solidworks files and STL needed to print the robot are available under a Creative Commons BY+SA+NC license on a private GrabCAD Workbench repository. People can request access to the hardware repository on the Poppy website (http://www.poppy-project.org/open-platform/).

The platform is currently under beta-testing meaning that we let the community grows little by little to ensure a good support of interesting projects.

For now, there are about 200 people on the GrabCAD project. They have access to all files needed to print the robot. There is also about 60 beta testers. They have access to a private section on the website with documentation and a forum for support.

Several of them are already doing a great work both by reporting bugs and managing to build the robot outside the lab. We are trying to work closely with them as they are a great source of feedback to improve the platform before a more wide distribution.

5.7.1.4. Impact in the community

Poppy has been release open source the 15/10/2013. The announcement has been done by Pierre Yves Oudeyer during the Lift 2013 conference. To prepare this event, the website and a overview video were made. The video, accessible here (http://vimeo.com/76917854) has reached about 40K views until now.

A part of the audience are technology-enthusiast people (about 50% on grabCAD), interested by the fact Poppy is 3D printable and so, highly customizable. More interesting, we received a large number of beta request from various applications domains (see 18) around the world (see 19).

`../../../../projets/flowers/IMG/poppy_charts.png`

Figure 18. The 60 selected beta testers profiles chart.

5.7.1.4.1. Education:

We receive a lot of request from Fablab around the world (France, USA, New-Zeland, South Africa,...). All have great educational projects for teaching robotics, mechanics and computer science to children. Some of them are close beta testers and we have weekly interactions. In the same topic, several schools, engineering schools and universities showed interest to use Poppy as experimental support. One of our close beta tester is a Bordeaux high school. They are planning to use Poppy as support for mechanics, computer sciences but also architecture or philosophy.

5.7.1.4.2. Art:

A current art project is under construction. A residence with an artist, a dancer and us will take place on the 24/02/2014 to the 05/03/2014. It comes to artistically invest humanoid robotics and thereby examine the relationship of the body to the digital world. The encounter between art and science generates potential new ideas for both disciplines who find themselves at the crossroads of questions relating to the gesture, movement and body. The Poppy Project is focused on morphological and motor aspects in a context human-robot interface. The look of an artist and the movements of a dancer are testing this interface with an unprecedented and direct manere.

`../../../../projets/flowers/IMG/poppy_locations.png`

Figure 19. Where are located the beta testers?

5.7.1.4.3. Research:

A large number of researchers showed interest in the platform. Most of them are interested to use it as a experimental tool. They want to address challenges such as balance and walking control, use of force controlled motors, explore human-robot interactions. On this last topic, the Bristol robotic lab target to use tele-operation to investigate what factors are important in terms of appearance and behaviour for credible and trustworthy interaction. For this purpose, they will develop functional hands for gesture and grasping task.

5.7.1.5. Next step:

We are currently working with beta testers on several improvements to make Poppy more accessible, more easy to use and more polyvalent. Two internship students will arrive on the project to work on the embedded electronic and on the feet design. We are targeting to release the final version during the summer 2014. Also we are thinking about the creation of a "Poppy pack" including all necessary components and tools to easily build the robot.

Another very important point is the community management. We are currently adding new collaborative tools on the Poppy website. These tools are tested with beta testers. The challenge is to offer the good tools to provide an efficient support to future users and to encourage people to contribute. This work is done in collaboration with Stephane Ribas (D2T inria Grenoble).

GRAPHIK Project-Team

5. Software and Platforms

5.1. Cogui

Participants: Alain Gutierrez, Michel Leclère, Marie-Laure Mugnier, Michel Chein, Madalina Croitoru.

Cogui (http://www.lirmm.fr/cogui) is a tool for building and verifying knowledge bases. It is a freeware written in Java (version 1.6). Currently, it supports Conceptual Graphs and import/export in RDFS and Datalog+/-. This year, the following features have been developed:

- we have introduced the concept of scripted rule which associates more fluidly the editable graphical objects with scripts that perform operations on knowledge graphs. These features have been tested and improved in various projects this year (see *e.g.* Qualinca in Section 8.1 or CTFC in Section 7.2).
- we have implemented an interface for quick and assisted creation of graphs. It is based upon Datalog+/- language and provides tools for automatic completion.
- finally, default conceptual graphs rules were implemented in Cogui. An editing interface is available as well as the operation to find all extensions of a rule set. This feature is required by the CTFC project (see Section 7.2).

5.2. Cogui/Capex

Participants: Alain Gutierrez, Patrice Buche, Awa Diattara, Jérôme Fortin.

Cogui/Capex is a platform for expert knowledge management. It has been developed in order to propose a simple and useful interface to applicative domain experts. This will allow us to validate the integration of our theoretical tools into a real-world application and strengthen GraphIK's involvement in agronomy applications (see the projects with CTFC in Section 7.2 and Panzani in Section 7.3).

5.3. Alaska

Participants: Bruno Paiva Lima Da Silva, Jean-François Baget, Madalina Croitoru.

Alaska (http://alaska.bplsilva.com/) is a java library dedicated to the storage and querying of large knowledge bases. It intends to be the foundation layer of our OBDA (Ontology Based Data Access) software developments. It has been built, first as part of a master thesis, and now of Bruno Paiva Lima da Silva' PhD (that will be defended in Jan. 2014).

In Alaska, facts and queries are defined via a generic interface that favors a logical view of these objects. Implementations of this interface allow for the storage of facts w.r.t. different storage paradigms and systems (e.g., relational databases *MySQL* and *Sqlite*; triple stores *Sesame* and graph databases *Neo4J*, *DEX*, *HyperGraphDB* and *OrientDB*). For the time being, we can store 10^7 to 10^8 atoms. In the same way, logical queries can be evaluated through different methods, be it the native querying mechanism of the considered database (*e.g.SPARQL* or *SQL*), or specifically designed algorithms (from a simple backtrack to a full constraint solver based upon *Choco*http://www.emn.fr/z-info/choco-solver/ for hard problem instances). Note that all these methods provide the same answer set to queries.

This library already allows for testing our OBDA algorithms on large instances. The ADT Quasar (that will start in March 2014) will involve the integration of Alaska with other tools developped in the team (see also Section 5.4), and its improvement from a research library to a distributable tool.

5.4. Tools for Rule-Based Reasoning

Participants: Mélanie König, Michel Leclère, Marie-Laure Mugnier, Swan Rocher, Michaël Thomazo.

Kiabora has been designed to analyze an existential rule base (see Section 6.1) and determine if it allows for finite query answering, i.e., if any conjunctive query evaluated over any fact base while taking this set of rules into account will be answered in a finite time. This year, we fixed some bugs and added some specific options. In addition, a presentation and a demo of Kiabora were made at RR 2013 [38].

Besides, the algorithms presented in [38], [37], [41] were implemented and let to experiments. These algorithms are still under development since new improvements have to be integrated.

HYBRID Project-Team

5. Software and Platforms

5.1. OpenViBE

Participants: Anatole Lécuyer [contact], Jozef Legény, Jussi Lindgren.

OpenViBE is a free and open-source software platform devoted to the design, test and use of Brain-Computer Interfaces (BCI). The platform consists of a set of software modules that can be integrated easily and efficiently to design BCI applications. The key features of OpenViBE software are its modularity, its high-performance, its portability, its multiple-users facilities and its connection with high-end/VR 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 AGPL licence, and it was officially released in June 2009. Since then, the OpenViBE software has already been downloaded more than 12000 times, and it is used by numerous laboratories, projects, or individuals worldwide. The OpenViBE software is supported and improved in the frame of OpenViBE-NT project (section 8.2.9). More information, downloads, tutorials, videos, documentations are available on the OpenViBE website.

5.2. GVT

Participants: Bruno Arnaldi, Valérie Gouranton [contact], Florian Nouviale, Thomas Lopez.

The aim of GVT software (Generic Virtual Training) is to offer personalizable VR training sessions for industrial equipments. The main features of GVT software are the safeness offered by VR training (as opposed to trainind in risky real conditions), the optimization of the learning process, the creation of dedicated scenarios runnable on multiple hardware configurations: laptop or desktop computer, immersive room, distribution over network, etc. The current kernel of the GVT platform is divided into two main elements that rely on innovative models we have proposed: LORA (Language for Object-Relation Application) and STORM (Simulation and Training Object-Relation Model) models. With GVT behavioral engine, the objects of the virtual world expose behavorial capacities through the use of STORM. Then, the GVT scenario engine is used to determine the next steps of the procedure for a trainee, and its state evolves as the trainees achieve some actions, the scenario being written in LORA. As for today, a commercialized version of GVT, which includes a pedagogical engine developed at CERV laboratory, proposes training on individual procedures. In CORVETTE (section 8.2.1) and SIFORAS (section 8.2.6) projects, new features based on GVT are being designed, such as interactive, collaborative and physicalized actions, actors knowledge management, dialog using natural langage.

5.3. Collaviz

Participants: Thierry Duval [contact], Thi Thuong Huyen Nguyen.

The aim of Collaviz software (collaborative interactive visualization) is to allow to design, deploy and share collaborative virtual environments (CVE). Collaviz allows VR developpers to concentrate on the behavior of virtual objects that can be shared between users in a CVE. Indeed, Collaviz provides a software architecture that hides the network programmation details of the distribution and the synchronization of the content of the CVE, and that facilitates the coupling with the 3D graphics API used for rendering. Collaviz is written mainly in Java and is runnable on multiple hardware configurations: laptop or desktop computer, immersive room, mobile devices. The PAC-C3D software architecture of Collaviz makes it possible to use various 3D APIs for graphic rendering: Java3D, jReality, jMonkeyEngine, OpenSG, Unity3D (work in progress) and Havok Anarchy (work in progress), and also to use various physical engines such as jBullet and SOFA. The distribution over the network can be achieved using TCP or HTTP. An on-going collaboration with Triskell team intends to extend Collaviz using a Model Driven Engineering approach in order to provide high-level tools to generate a large part of java code of virtual objects.

IMAGINE Project-Team

4. Software and Platforms

4.1. MyCorporisFabrica

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

../../../../projets/imagine/IMG/mycf_pic.png

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

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

4.2. SOFA

Participants: François Faure, Ali Hamadi Dicko, Armelle Bauer, Olivier Carré, Matthieu Nesme, Romain Testylier, Moreno Trlin.

../../../../projets/imagine/IMG/sofa_pic.png

Figure 2. SOFA is an open source simulator for physically based modeling.

SOFA is a C++ library primarily targeted at medical simulation research. Based on an advanced software architecture, it allows to (1) create complex and evolving simulations by combining new algorithms with algorithms already included in SOFA; (2) modify most parameters of the simulation – deformable behavior, surface representation, solver, constraints, collision algorithm, etc. – by simply editing an XML file; (3) build complex models from simpler ones using a scene-graph description; (4) efficiently simulate the dynamics of interacting objects using abstract equation solvers; and (5) reuse and easily compare a variety of available methods.

SOFA is gaining momentum. A start-up based on SOFA, InSimo, has been created in Strasbourg by Inria people, and one of our former engineers, François Jourdes, has been hired. A SOFA-specific workshop was co-located with conference Vriphys'13 in Lille, with 50 attendants and the participation of several companies including CAE (a Canadian world leader in simulation), Haption, BASF, InSimo and others.

4.3. Expressive

Participants: Marie-Paule Cani, Amaury Jung, Mohamed-Galal Koraa, Maxime Quiblier, Cédric Zanni, Antoine Begault.

../../../../projets/imagine/IMG/convol.png

Figure 3. GUI and Example of implicit surface and modeled with the Expressive platform.

Expressive is a new C++ library developed to gather and share the models and algorithms developed within the ERC Expressive project. It enables us to make our latest research results on new creative tools; typically high level models together with intuitive, sketching or sculpting interfaces - soon available to the rest of the group and easily usable in our industrial partnerships. Its most developed part is Convol, a library dedicated implicit surfaces; and more particularly to the sub-classes of convolution surfaces and other integral surfaces along skeletons. Convol incorporates all the necessary material for constructive implicit modeling: skeleton-based convolution and SCALIS primitives, with closed form solution for the field values and gradient whenever possible; a variety of blending operators; and several methods for tessellating an implicit surface into a mesh, and for refining the later in highly curved regions. The creation of new geometry can be performed by direct manipulation of skeletal primitives or through sketch-based modeling.

IN-SITU Project-Team

5. Software and Platforms

5.1. WILD Platform

Participants: Michel Beaudouin-Lafon [correspondant], Olivier Chapuis, Stéphane Huot, Romain Primet, Amani Kooli, Monireh Sanaei, Gabriel Tezier, Jonathan Thorpe.

WILD (Wall-Size Interaction with Large Datasets) is InSitu's experimental ultra-high-resolution interactive platform for studying collaborative interaction and the visualization of very large datasets [2] (Figure 1 . It features a wall-sized display with thirty-two 30" LCD screens, i.e. a 5m50 x 1m80 (18' x 6') wall displaying 20 480 x 6 400 = 131 million pixels, powered by a 16-computer cluster and two front-end computer. The platform also features a camera-based motion tracking system supporting interaction with the wall as well as within the surrounding space, a multitouch table and various mobile devices. WILD provides a unique experimental environment for interactive visualization and is part of the DIGISCOPE Equipment of Excellence. In addition to using WILD for our research, we have also developed software architectures and toolkits that enable developers to run applications on such multi-device, cluster-based systems.

Figure 1. The WILD platform (clockwise): wall display, motion tracking system, display cluster, interactive table.

5.2. jBricks

Participants: Stéphane Huot [correspondant], Mathieu Nancel, Romain Primet.

jBricks (Figure 2) is a Java toolkit that integrates a high-quality 2D graphics rendering engine based on ZVTM [46] and a versatile input configuration module (based on ICon [42] and FlowStates 5.4) into a coherent framework, enabling the exploratory prototyping of interaction techniques and rapid development of post-WIMP applications running on cluster-driven interactive visualization platforms such as wall-sized displays. The goal of this framework is to ease the development, testing and debugging of interactive visualization applications. It also offers an environment for the rapid prototyping of novel interaction techniques and their evaluation through controlled experiments.

- ACM: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)
- Software benefit: See E. Pietriga, S. Huot, M. Nancel, R. Primet, Rapid Development of User Interfaces on Cluster-Driven Wall Displays with jBricks, EICS '11: Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems, pages 185-190, June 2011
- OS/Middleware: Java (Linux, Mac OS X, Windows)
- Required library or software: several, managed through Maven
- Programming language: Java

../../../../projets/in-situ/IMG/jbricks.png

Figure 2. jBricks applications running on the WILD platform (32 tiles for a total resolution of 20 480 × 6 400 pixels). (a) Zoomed-in visualization of the North-American part of the world-wide air traffic network (1 200 airports, 5 700 connections) overlaid on NASA's Blue Marble Next Generation images (86 400 × 43 200 pixels) augmented with country borders ESRI shapefiles. (b) Panning and zooming in Spitzer's Infrared Milky Way (396 032 × 12 000 pixels). (c) Controlled laboratory experiment for the evaluation of mid-air multi-scale navigation techniques.

5.3. The SwingStates Toolkit

Participants: Caroline Appert [correspondant], Michel Beaudouin-Lafon.

SwingStates [36] is a library that adds state machines and a graphical canvas to the Java Swing user interface toolkit. It was motivated by the lack of widely disseminated toolkits that support advanced interaction techniques and the observation that HCI research toolkits are little used outside the lab. By extending the popular Java Swing toolkit rather than starting from scratch, the goal is to facilitate the dissemination and adoption of SwingStates by practitioners.

SwingStates uses *state machines* to specify interaction. It provides programmers with a natural syntax to specify state machines and reduces the potential for an explosion of the number of states by allowing multiple state machines to work together or separately. SwingStates can be used to add new interaction techniques to existing Swing widgets, e.g. to select buttons and checkboxes by crossing rather than clicking. It can also be used with the SwingStates canvas (see below) and to control high-level dialogues.

SwingStates also provides a powerful *canvas widget*. The canvas can contain any Java2D shape, including geometric shapes, images, text strings and even Swing widgets. Shapes can be manipulated individually or collectively, through *tags*. An intensive use of polymorphism allows to apply almost any command to a tag: the command is then applied to all objects with this tag. Tags are also used in conjunction with state machines, to specify transitions that occur only on objects with a given tag. For example, pie menus can be implemented by creating a canvas in the overlay layer of any Swing application (Figure 3).

`../../../../projets/in-situ/IMG/PieMenu.png`

`../../../../projets/in-situ/IMG/JoystickTextField.png`

Figure 3. A numeric text field whose value can be set by a joystick-like interaction (left) and a semi-transparent menu to change the background color of Swing widgets (right)

SwingStates tightly integrates state machines, the Java language and the Swing toolkit to provide programmers with a natural and powerful extension to their natural programming environment. SwingStates is available at http://swingstates.sf.net under the GNU Lesser General Public License (LGPL).

- ACM: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)
- Software benefit: See C. Appert and M. Beaudouin-Lafon (2008) SwingStates: Adding State Machines to Java and the Swing Toolkit. Software: Practice and Experience, 38(11):1149 - 1182.
- OS/Middleware: Mac OS X, Linux, Windows
- Required library or software: Java virtual machine
- Programming language: Java

5.4. The FlowStates Toolkit

Participants: Caroline Appert [correspondant], Michel Beaudouin-Lafon, Stéphane Huot.

FlowStates [37], is a new toolkit to program advanced interaction techniques which require non standard input (e.g., two different mice that act independently, a joystick, a tablet, etc.). It is built on top of two existing toolkits: SwingStates [36] and ICon [42].

`../../../../projets/in-situ/IMG/flowstates.png`

Figure 4. State machines and data flow in FlowStates

With FlowStates the developer can program interaction logic using state machines like SwingStates does but does not restrict the set of possible input channels to Java AWT standard input (a single couple <mouse, keyboard>). The state machines just have to define the virtual input events that are required to trigger their transitions so that FlowStates turns these machines into ICon devices which can be plugged to any physical input channels (Figure 4). An ICon device is a data flow building block that has input and output slots in order to be connected to other devices in the simple graphical environment provided by ICon. State machines can also send out events which appear as output slots in the data flow model.

With FlowStates we showed how two models for programming interaction (state machines and data flow) can be fully integrated to offer a huge power of expression. The explicit decision to not set strict limits between the roles of each model makes this hybrid approach highly flexible, the developer setting himself the limit between the two according to his needs and habits.

FlowStates is available at http://www.lri.fr/~appert/FlowStates/.

- ACM: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)
- Software benefit: See C. Appert, S. Huot, P. Dragicevic and M. Beaudouin-Lafon (2009) FlowStates: Prototypage d'applications interactives avec des flots de données et des machines à états. In Proceedings of IHM 2009. ACM, pages 119-128.
- OS/Middleware: Mac OS X, Linux, Windows
- Required library or software: ICon, Java virtual machine
- Programming language: Java

5.5. TouchStone

Participants: Caroline Appert [co-correspondant], Michel Beaudouin-Lafon, Wendy Mackay [co-correspondant].

TouchStone [8] is a platform for designing, running and analyzing the results of controlled experiments (Figure 5). While it focuses on experiments comparing interaction techniques, it can be used in a wide variety of contexts.

../../../../projets/in-situ/IMG/touchstone.png

Figure 5. The architecture of the Touchstone platform

The *Touchstone Design platform* allows an experimenter to specify the factors, levels and measures in a controlled experiment, supports blocking and counterbalancing of trials and calculates how long it will take to run the experiment. Experimenters can compare the trade-offs between different experiment designs. The platform produces an XML file that serves as a protocol for the experiment and can be used as input to the Run platform.

The *Touchstone Run platform* provides a framework for implementing and running an experiment and collecting performance data. The flexible plug-in architecture supports various input devices and interaction techniques. The XML script from the Design platform can be run directly or edited to accommodate needs from specific experiments.

Log data from the Run platform can be analyzed by standard statistics tools such as JMP, R or Excel. In future, we hope to create a more elaborate *Touchstone Analysis platform* that will generate analysis scripts based on the output of the Design platform.

Members of InSitu use Touchstone for a variety of experiments and Students in the Research Masters (M2R Interaction) have been using it to design and implement experiments since 2011. Touchstone is available at http://code.google.com/p/touchstone-platforms/ under a BSD License.

- ACM: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)
- Software benefit: See W. Mackay, C. Appert, M. Beaudouin-Lafon, O. Chapuis, Y. Du, JD. Fekete and Y. Guiard (2007) TouchStone: Exploratory Design of Experiments. In Proceedings of ACM CHI 2007 Conference on Human Factors and Computing Systems. ACM, pages 1425-1434.
- OS/Middleware: Mac OS X, Linux, Windows
- Required library or software: Java virtual machine
- Programming language: Java

5.6. Metisse

Participant: Olivier Chapuis [correspondant].

Metisse [40] is a window system that facilitates the design, implementation and evaluation of innovative window management techniques. The system is based on a compositing approach, making a clear distinction between the rendering and the interactive compositing processes. The Metisse server is a modified X server that supports both input and output redirection. The default compositor is a combination of a slightly modified version of FVWM, a standard window manager, with an interactive viewer application called *FvwmCompositor*.

FvwmCompositor uses OpenGL to display windows, which offers a rich graphics model well adapted to the exploration of new window management techniques. Texture mapping, for example, makes it possible to transform the window shapes in real-time (Figure 6 , left). Alpha blending makes it easy to create translucent objects and shadows. Scaling, rotation and translation can also be used to position windows in 2D$\frac{1}{2}$ or 3D (Figure 6 , middle and right). Input redirection makes it still possible to interact with applications no matter the visual transformations applied to the windows. It also makes it possible to adapt, reconfigure or re-combine existing graphical interfaces [48]. This year we used again Metisse to implement novel desktop interaction techniques [4].

- Web: http://insitu.lri.fr/metisse/
- ACM: H.5.2 [User Interfaces]: Windowing systems
- Software benefit: see [40], [48], [41], [44] and [4].
- License: GPL
- Type of human computer interaction: Graphique
- OS/Middleware: X Window et Mac OS X
- Required library or software: OpenGL via nucleo [1] and some usual C/C++ libraries
- Programming language: * C/C++

---

[1] http://interaction.lille.inria.fr/~roussel/projects/nucleo/index.html

../../../../projets/in-situ/IMG/metisse-folding.png
../../../../projets/in-situ/IMG/metisse-tabletop.png
../../../../projets/in-situ/IMG/met

Figure 6. Sample window management techniques implemented with Metisse: extended paper metaphor (left), interactive table configuration that allows to duplicate and rotate windows (middle) and zoomable 3D desktop (right).

5.7. The Substance Middleware

Participants: Michel Beaudouin-Lafon [correspondant], Clemens Klokmose, Tony Gjerlufsen, James Eagan, Clement Pillias.

Substance is a middleware based on a novel programming paradigm called *data-oriented programming* and was designed to facilitate the development of multi-surface interactive applications [45]. Such applications are distributed by nature as they involve a varying number of display and interaction surfaces that are controlled by different computers. For example, our WILD room includes a 32-monitor display wall driven by 16 computers plus a front-end, a multi-touch table, various mobile devices such as iPodTouch and iPads, and the laptops that the users of the room may bring with them. We want to support seamless interaction techniques across these surfaces, such as the pick-and-drop technique pioneered by Rekimoto [47].

Data-oriented programming consists of attaching functionality to a tree data structure through *facets* attached to the individual nodes of the tree. Facets can be added and removed dynamically, and notified of changes in the tree. Substance supports two powerful ways to share nodes and facets: mounting, where access to the shared tree is managed through remotely, and replication, where the shared tree is replicated at each site and synchronized.

Substance has been used to create two full-scale applications (Figure 7): a generalized Canvas that can display and manage graphics, PDF files, image files and other content (through an extensible content manager) across surfaces spanning multiple displays and computers; SubstanceGrise, which uses multiple instances of the Anatomist/BrainVISA application to display coordinated 3D imagery of many brains in parallel on the WILD wall and control from a physical model of the brain.

Substance is available at http://substance-env.sourceforge.net/ under a GNU GPL 3.0 licence.

- ACM: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)
- Software benefit: See T. Gjerlufsen, C. Klokmose, J. Eagan, C. Pillias and M. Beaudouin-Lafon (2011) Shared Substance: Developing Flexible Multi-Surface Applications. In CHI '11: Proceedings of the 29th international conference on Human factors in computing systems. ACM, pages 3383-3392.
- OS/Middleware: Mac OS X, Linux
- Required library or software: several, managed by Python install
- Programming language: Python

../../../../projets/in-situ/IMG/Canvas.png

../../../../projets/in-situ/IMG/Substance

Figure 7. The Canvas (left) and SubstanceGrise (right) applications developed with Substance. (©CNRS-Phototheque - Cyril FRESILLON for SubstanceGrise).

5.8. Scotty

Participants: Michel Beaudouin-Lafon [correspondant], James Eagan, Wendy Mackay.

The goal of Scotty is to support *malleable interfaces*, i.e. interfaces that can be modified at run-time in ways not anticipated by the designers [43]. Scotty is a toolkit that allows a programmer to extend an existing Mac OS X application without access to its source code. Scotty provides the following abstractions: hooks to alter the appearance of windows and widgets, event funnels to alter their behavior, glass sheets to overlay graphics and add new interaction methods, dynamic code loading and object proxies to redefine and extend existing objects. Scotty also provides a higher-level interface based on instrumental interaction [38]. Scotty currently runs on Mac OS X for applications written with the Cocoa user interface framework.

../../../../projets/in-situ/IMG/iphone.png

../../../../projets/in-situ/IMG/toolglass

Figure 8. Using Scotty to teleport a window of a Mac OS X application onto an iPhone (left) and to create a toolglass in the Pages word processor (right).

Scotty has been used to create a number of extensions (Figure 8). *Scribbler* is a generic extension that uses glass sheets to allow handwritten annotations of any Cocoa window. *Teleportation* is another generic extension that can teleport and resize the content of any Cocoa window onto another computer, including an iPhone or iPad. The user can interact with the teleported content as if it was on the original computer. It was used to create a content provider for the Substance Canvas (see above), making it possible to display any application running on a laptop onto the WILD wall display and/or table. When vector-based content is available, e.g., for text, Scotty provides smooth rescaling without the typical pixelation apparent when enlarging bitmap images. Finally *Stylesheet* is an extension to the Pages word processor that provides a semi-transparent toolglass for specifying the styles of paragraphs.

Scotty is available at http://insitu.lri.fr/Projects/Scotty under a GNU GPL 3.0 licence.

- ACM: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)
- Software benefit: See J. Eagan, W. Mackay and M. Beaudouin-Lafon (2011) Cracking the Cocoa Nut: User Interface Programming at Runtime. In UIST 2011: Proceedings of the 24th ACM Symposium on User Interface Software and Technology. ACM, pages 225-234.
- OS/Middleware: Mac OS X
- Required library or software: none
- Programming language: Objective-C, Python

LAGADIC Project-Team

5. Software and Platforms

5.1. ViSP: a visual servoing and tracking software library

Participants: Fabien Spindler [correspondant], Aurélien Yol, Eric Marchand, François Chaumette.

Since 2005, we develop and release under the terms of the GPLv2 licence, ViSP, an open source library available from http://team.inria.fr/lagadic/visp/visp.html. It allows fast prototyping of visual tracking and visual servoing tasks. ViSP was designed to be independent with the hardware, to be simple to use, expandable and cross-platform.

ViSP allows to design vision-based tasks for eye-in-hand and eye-to-hand visual servoing that contains the most classical visual features that are used in practice. It involves a large set of elementary positioning tasks with respect to various visual features (points, segments, straight lines, circles, spheres, cylinders, image moments, pose,...) that can be combined together, and image processing algorithms that allows tracking of visual cues (dots, segments, ellipses,...) or 3D model-based tracking of known objects. Simulation capabilities are also available. ViSP and its full functionalities are presented in Fig. 1 and described in [6].

This year, we continued our efforts to improve the software by increasing the compatibility with exotic platforms, fixing issues, and by introducing an hybrid scheme in the model-based tracker to take advantage of texture. We also improved the documentation by providing tutorials covering the main capabilities of the software. Two releases were produced, one in February downloaded 1000 times and the other in July downloaded 730 times. With the help of the community, the last release was also packaged for Ubuntu 13.10. A new template tracker developed during A. Dame's Ph.D. was recently introduced and will be available in the next release.

Concerning ROS community, all the existing packages in “`vision_visp`” ROS stack (see http://www.ros.org/wiki/vision_visp) were updated and ported to catkin build system. To ease ViSP usage in the ROS framework, the last release was packaged for ROS.

ViSP is used in research labs in France, USA, Japan, Korea, India, China, Lebanon, Italy, Spain, Portugal, Hungary, Canada. For instance, it is used as a support in graduate courses at IFMA Clermont-Ferrand, University of Picardie in Amiens, Télécom Physique in Strasbourg and ESIR in Rennes.

5.2. DESlam software

Participant: Patrick Rives [correspondant].

The DESlam (Dense Egocentric Slam) software developed in collaboration with Andrew Comport from I3S in Sophia Antipolis was registered to the APP (“Agence de Protection des Programmes”) (IDDN.FR.001.320001.000.S.P.2012.000.21000). This software proposes a full and self content solution to the dense Slam problem. Based on a generic RGB-D representation valid for various type of sensors (stereovision, multi-cameras, RGB-D sensors...), it provides a 3D textured representation of complex large indoors or outdoors environments and it allows to localize in real time (45Hz) a robot or a person carrying out a mobile camera.

5.3. Robot vision platforms

Participant: Fabien Spindler [correspondant].

We exploit two industrial robotic systems built by Afma Robots in the nineties to validate our researches in visual servoing and active vision. The first one is a Gantry robot with six degrees of freedom, the other one is a cylindrical robot with four degrees of freedom (see Fig. 2). These robots are equipped with cameras. The Gantry robot allows also to embed grippers on its end-effector.

`../../../../projets/lagadic/IMG/visp1.png`

`../../../../projets/lagadic/IMG/visp2.png`

`../../../../projets/lagadic/IMG/afma.png`

Figure 2. Lagadic robotics platforms for vision-based manipulation

Three papers published by Lagadic in 2013 enclose results validated on this platform.

5.4. Medical robotics platforms

Participants: Fabien Spindler [correspondant], Alexandre Krupa.

This testbed is of primary interest for researches and experiments concerning ultrasound visual servoing applied to positioning or tracking tasks described in Section 6.4 .

This platform is composed by two Adept Viper six degrees of freedom arms (see Fig. 3). Ultrasound probes connected either to a SonoSite 180 Plus or an Ultrasonix SonixTouch imaging system can be mounted on a force torque sensor attached to each robot end-effector.

We started experimentation to validate needle detection and tracking under ultrasound imaging (see Section 6.4.1).

This year, two papers enclose experimental results obtained with this platform.

`../../../../projets/lagadic/IMG/viper-twin.png`

Figure 3. Lagadic medical robotics platforms. On the right Viper S850 robot arm equipped with a SonixTouch 3D ultrasound probe. On the left Viper S650 equipped with a tool changer that allows to attach a classical camera.

5.5. Mobile robotics platforms

Participants: Fabien Spindler [correspondant], Erwan Demairy, Marie Babel, Patrick Rives.

5.5.1. Indoors mobile robots

For fast prototyping of algorithms in perception, control and autonomous navigation, the team uses Hannibal in Sophia Antipolis, a cart-like platform built by Neobotix (see Fig. 4 .a), and a Pioneer 3DX from Adept in Rennes (see Fig. 4 .b) as well as a Robotino from Festo. These platforms are equipped with various sensors needed for Slam purposes, autonomous navigation and sensor-based control.

Moreover, to validate the researches in personally assisted living topic (see 6.3.4), we bought in Rennes a six wheel electric wheelchair from Penny and Giles Drives Technology (see Fig. 4 .c). The control of the wheelchair is performed using a plug and play system between the joystick and the low level control of the wheelchair. Such a system let us acquire the user intention through the joystick position and control the wheelchair by applying corrections to its motion. The wheelchair has been fitted with cameras to perform the required servoing for assisting handicapped people. Moreover, to ensure the direct security of the user, seven infrared proximity sensors have been installed all around the wheelchair.

Note that three papers exploiting the indoors mobile robots were published this year.

5.5.2. Outdoors mobile robots

The team exploits also Cycab urban electrical cars (see Figs. 4 .d and 4 .e). Two vehicles in Sophia Antipolis and one in Rennes are instrumented with cameras and range finders to validate researches in the domain of intelligent urban vehicle. Cycabs were used as experimental testbeds in several national projects.

Three papers published by Lagadic in 2013 enclose experimental results obtained with these outdoors mobile robots.

5.5.3. Technological Development Action (ADT) P2N

The ADT P2N aims at sharing existing and in development codes between the Lagadic and E-Motion teams in the field of autonomous navigation of indoors robots. These codes are also used in the platforms involved in the large-scale initiative action PAL (Personnally Assisted Living, see Section 8.2.6). This year, the most notable activities for this ADT have been to:

- adapt a navigation module developed by E-Motion to the mobile platform used at Sophia-Antipolis;
- make the SLAM module developed by Lagadic usable by the E-Motion navigation module;
- port the code on the wheelchairs used in PAL;
- develop the core architecture running under ROS supporting the different sensors and platforms available in Sophia-Antipolis.

`../../../../projets/lagadic/IMG/Hannibal.png`
`../../../../projets/lagadic/IMG/pioneer.jpg`
`../../../../projets/lagadic/IMG/fau`

(a) (b) (c)

`../../../../projets/lagadic/IMG/Cycab_Rennes.png`
`../../../../projets/lagadic/IMG/Cycab_Sop`

(d) (e)

Figure 4. a) Hannibal platform, b) Pioneer P3-DX robot, c) six wheel electric wheelchair, d) Cycab available in Rennes, e) one of the Cycabs available in Sophia Antipolis.

LEAR Project-Team

5. Software and Platforms

5.1. Large-scale image classification

Participants: Matthijs Douze [correspondant], Zaid Harchaoui, Florent Perronnin [XRCE], Cordelia Schmid.

JSGD is the implementation of a Stochastic Gradient Descent algorithm used to train linear multiclass classifiers. It is biased towards large classification problems (many classes, many examples, high-dimensional data). It can be used on the ImageNet large scale classification challenge. It uses several optimization techniques, both algorithmic (scale factors to spare vector multiplications, vector compression with product quantizers) and technical (vector operations, multithreading, improved cache locality). It has Python and Matlab interfaces. It is distributed under a Cecill licence. Project page: http://lear.inrialpes.fr/src/jsgd.

5.2. Fisher vector image representation

Participants: Matthijs Douze [correspondant], Hervé Jégou [TEXMEX Team Inria Rennes], Cordelia Schmid.

We developed a package that computes Fisher vectors on sparse or dense local SIFT features. The dense feature extraction was optimized, so that they can be computed in real time on video data. The implementation was used for several publications and in our submisssion to the Trecvid 2013 MED task. We provide a binary version of the local descriptor implementation, and the Fisher implementation is integrated in the Yael library, with Python and Matlab interface, see http://lear.inrialpes.fr/src/inria_fisher.

5.3. Video descriptors

Participants: Clement Leray, Dan Oneata, Cordelia Schmid [correspondant], Heng Wang, Jakob Verbeek.

We have developed and made on-line available software for video description based on dense trajectories and motion boundary histograms. The trajectories capture the local motion information of the video. A state-of-the-art optical flow algorithm enables a robust and efficient extraction of the dense trajectories. Descriptors are aligned with the trajectories and based on motion boundary histograms (MBH) which are robust to camera motion. This year we have further developed this software to increase its robustness and scalability to large datasets. Most importantly, we have added a robust background stabilization technique, which allows to remove camera motion. This has shown to significantly improve the performance. Furthermore, we have improved the efficiency of the approach. For example, we avoid writing the raw MBH descriptors to disk, but rather aggregate them directly into a signature for the complete video using Fisher vectors. This allowed us to use these descriptors on the 4,000 hour video dataset of the TrecVid 2013 MED task as well as on the 3500 hours of AXES broadcast videos.

5.4. SPArse Modeling Software (SPAMS)

Participants: Julien Mairal [correspondant], Jean-Paul Chieze [WILLOW Project-Team], Jean Ponce [WILLOW Project-Team], Francis Bach [SIERRA Project-Team].

SPAMS v2.4 was released as open-source software in December 2013 (v1.0 was released in September 2009). It is an optimization toolbox implementing algorithms to address various machine learning and signal processing problems involving

- Dictionary learning and matrix factorization (NMF, sparse PCA, ...);
- Solving medium-scale sparse decomposition problems with LARS, coordinate descent, OMP, SOMP, proximal methods;
- Solving large-scale sparse estimation problems with stochastic optimization;
- Solving structured sparse decomposition problems (sparse group lasso, tree-structured regularization, structured sparsity with overlapping groups,...).

The software and its documentation are available at http://spams-devel.gforge.inria.fr/.

This year, we added new functionalities to the toolbox. A graphical tool for visualizing dictionaries was developed by Jean-Paul Chieze, and stochastic optimization tools corresponding to the papers [24], [23] were added for dealing with large-scale sparse estimation problems.

5.5. FlipFlop: Fast Lasso-based Isoform Prediction as a Flow Problem

Participants: Elsa Bernard [Institut Curie, Ecoles des Mines-ParisTech], Laurent Jacob [CNRS, LBBE Laboratory], Julien Mairal [correspondant], Jean-Philippe Vert [Institut Curie, Ecoles des Mines-ParisTech].

FlipFlop is an open-source software, implementing a fast method for de novo transcript discovery and abundance estimation from RNA-Seq data [36]. It differs from classical approaches such as Cufflinks by simultaneously performing the identification and quantitation tasks using a penalized maximum likelihood approach, which leads to improved precision/recall. Other softwares taking this approach have an exponential complexity in the number of exons of a gene. We use a novel algorithm based on network flow formalism, which gives us a polynomial runtime. In practice, FlipFlop was shown to outperform penalized maximum likelihood based softwares in terms of speed and to perform transcript discovery in less than 1/2 second for large genes.

FlipFlop 1.0.0 is a user friendly bioconductor R package. It is freely available on the Bioconductor website under a GPL licence: http://bioconductor.org/packages/release/bioc/html/flipflop.html.

5.6. DeepFlow

Participants: Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, Cordelia Schmid.

We developed a package for the "deep flow" algorithm [31]. "Deep flow" combines a standard variational framework with a our new matching algorithm "deep matching". The code for "deep matching" is in python and the code for "deep flow" in C. Both of them are available on-line at http://lear.inrialpes.fr/src/deepmatching. Note that the run time is a few seconds per images pair, which is less than for most other methods.

5.7. Object category localization

Participants: Ramazan Cinbis, Matthijs Douze, Cordelia Schmid, Jakob Verbeek.

We developed an object category localization system based on a Fisher vector representation over densly extracted local SIFT descriptors [18]. To improve the robustness with respect to background clutter in the detection windows we developed an approximate object segmentation method that is used to weigh the contribution of local SIFT descriptors. Our system achieves state-of-the-art localization performance as measured on the PASCAL VOC 2007 and 2010 datasets. The system is developed in both C, python, and Matlab. The system will be released in early 2014.

LINKS Team

4. Software and Platforms

4.1. QuiX-Tool Suite

Participants: Joachim Niehren [correspondant], Denis Debarbieux, Tom Sebastian.

The QuiX-Tool Suite provides tools to process XML streams and documents. The QuiX-Tool Suite is based on early algorithms: query answers are delivered as soon as possible and in all practical cases at the earliest time point. The QuiX-Tool Suite provides an implementation of the main XML standart over streams. XPath, XSLT, XQuery and XProc are W3C standarts while Schematron is an ISO one. The QuiX-Tool suite is developed in the Inria transfer project QuiXProc in cooperation with Innovimax. It includes among the others existing tools such as FXP and QuixPath, along with new tools, namely X-Fun. Both, a free and a professional version are available. The ownership of QuiX-Tool Suite is shared between Inria and Innovimax. The main application of QuiX-Tool Suite is its usage in QuiXProc, an professional implementation of the W3C pipeline language XProc owned by Innovimax.

The QuiXPath language is a large fragment of XPath with full support for the XML data model. The QuiXPath library provides a compiler from QuiXPath to FXP, which is a library for querying XML streams with a fragment of temporal logic.

The X-Fun language is a functional language for defining transformations between XML data trees, while providing shredding instructions. X-Fun can be understood as an extension of Frisch's XStream language with output shredding, while pattern matching is replaced by tree navigation with XPath expressions. The QuiX-Tool suite includes QuiXSLT, which is a compiler from XSLT into a fragment of X-Fun, which can be considered as the core of XSLT. It also provides QuiXSchematron, which is a compiler from Schematron to X-Fun, and QuiXQuery, which is a compiler from XQuery to X-Fun.

See also the web page https://project.inria.fr/quix-tool-suite/.

- Version: QuixPath v2.0.0
- Version: X-Fun v0.5.0
- Version: QuiXSLT v0.5.0
- Version: QuiXSchematron v1.0.0

4.2. SmartHal

Participants: Joachim Niehren [correspondant], Antoine Ndione.

SmartHal is a better tool for querying the HAL bibliography database, while is based on Haltool queries. The idea is that a Haltool query returns an XML document that can be queried further. In order to do so, SmartHal provides a new query language. Its queries are conjunctions of Haltool queries (for a list of laboratories or authors) with expressive Boolean queries by which answers of Haltool queries can be refined. These Boolean refinement queries are automatically translated to XQuery and executed by Saxon. A java application for extraction from the command line is available. On top of this, we have build a tool for producing the citation lists for the evaluation report of the LIFL, which can be easily adapter to other Labs.

See also the web page http://smarthal.lille.inria.fr/.

- Version: SmartHal v1.0.0

MAGNET Team

5. Software and Platforms

5.1. CoRTex

Participants: Pascal Denis [correspondent], David Chatel.

CoRTex is a LGPL-licensed Python library for Noun Phrase coreference resolution in natural language texts. This library contains implementations of various state-of-the-art coreference resolution algorithms, including those developped in my own research, such as [3]. In addition, it provides a set of APIs and utilities for text pre-processing, reading the main annotation formats (ACE, CoNLL and MUC), and performing evaluation based on the main evaluation metrics (MUC, B-CUBED, and CEAF). As such, CoRTex provides benchmarks for researchers working on coreference resolution, but it is also of interest for developers who want to integrate a coreference resolution within a larger platform. This project is hosted on Inria gforge: https://gforge.inria.fr/projects/cortex/.

5.2. JProGraM

Participant: Antonino Freno [correspondent].

JProGraM is a GPL-licensed Java library for machine learning and statistical analysis over graphs and through graphs. Supported models for vectorial data include e.g. Bayesian networks, Markov random fields, Gaussian mixtures, kernel density estimators, and neural networks, whereas random graph tools include small-world networks, preferential-attachment, exponential random graphs, and spectral models (as well as subgraph sampling algorithms). One strong point of the library is the extensive support for continuous random variables. JProGraM integrates implementations for the recent results in [20] and [21]. For more information, see the associated webpage at http://researchers.lille.inria.fr/~freno/JProGraM.html.

MAGRIT Project-Team

5. Software and Platforms

5.1. Software and Platforms

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://qt.digia.com

[2] http://www.openscenegraph.org/projects/osg

MAIA Project-Team

5. Software and Platforms

5.1. AA4MM

Participants: Vincent Chevrier [correspondant], Benjamin Camus, Julien Vaubourg.

Laurent Ciarletta (Madynes team, LORIA) is a collaborator and correspondant for this software. Yannick Presse (Madynes team, LORIA) is collaborator 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. The first implementation of the AA4MM meta-model was proposed in Julien Siebert's PhD [65] and written in Java. A newer version with more coupling models is currently submitted to the APP (Agence pour la protection des programmes).

This year, we used this software in a strategic action with EDF R&D in the context of the simulation of smart-grids.

5.2. MASDYNE

Participants: Vincent Chevrier [correspondant], Tomas Navarrete.

This work was undertaken in the PhD Thesis of Julien Siebert, a joint thesis between MAIA and Madynes Team. Laurent Ciarletta (Madynes team, LORIA) has been co-advisor of this PhD and correspondant for this software.

Other contributors to this software were: 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. It has been updated by Tomas Navarrete with new functionalities for the simulation of scenarii.

5.3. FiatLux

Participant: Nazim Fatès [correspondant].

FiatLux is a discrete dynamical systems simulator that allows the user to experiment with various models (for example 1D and 2D cellular automatas, moving agents on cellular automatas) and to perturb them. Its main feature is to allow users to change the type of updating, for example from a deterministic parallel updating to an asynchronous random updating. FiatLux has a Graphical User Interface and can also be launched in a batch mode for the experiments that require statistics.

In 2013, FiatLux was officially registered by the Agence pour la protection des programmes (APP).A new release is available under the CeCILL licence on the FiatLux website : fiatlux.loria.fr

5.4. Cart-o-matic

Participants: Olivier Simonin [correspondant], François Charpillet, Antoine Bautin, Nicolas Beaufort.

Philippe Lucidarme (Université d'Angers, LISA) is a collaborator and the coordinator of the Cart-o-matic project.

Cart-o-matic is a software platform for (multi-)robot exploration and mapping tasks. It has been developed by Maia members and LISA (Univ. Angers) members during the robotics ANR/DGA Carotte challenge (2009-2012). This platform is composed of three softwares tools which are protected by software copyrights (through the Agence pour la Protection des Programmes): Slam-o-matic a SLAM algorithm developed by LISA members, Plan-o-matic a robot trajectory planning algorithm developed by Maia and LISA members, and Expl-o-matic a distributed multi-agent strategy for multi-robot exploration developed by Maia members (which is based on algorithms proposed in the PhD Thesis of Antoine Bautin). Cf. illustration at Cart-o-matic.

The purchase of Cart-o-matic by some robotics companies is underway.

MANAO Team

4. Software and Platforms

4.1. EIGEN

Participants: G. Guennebaud, D. Nuentsa

Keywords : Linear algebra

Efficient numerical computation is central to many computer science domains. In particular, in computer graphics, space transformations and local regressions involve dense linear algebra, data interpolation and differential equations require sparse linear algebra, while more advanced problems involve non-linear optimization or spectral analysis. On the one hand, solutions such as MatLab are limited to prototyping. On the other hand, optimized libraries coming from the HPC (high performance computing) world are often tedious to use and more adapted for very large problems running on clusters. Moreover, all these solutions are very slow at handling very small but numerous problems which often arise in computer graphics, vision, or robotics. As a result, researchers of these domains used to waste a lot of time at either implementing their own half cooked solution, or dealing with dozens of complex to use libraries.

The objective of Eigen is to fill this gap by proposing an easy to use, efficient, and versatile C++ mathematical template library for linear algebra and related algorithms. In particular it provides fixed and dynamic size matrices and vectors, matrix decompositions (LU, LLT, LDLT, QR, eigenvalues, etc.), sparse matrices with iterative and direct solvers, some basic geometry features (transformations, quaternions, axis-angles, Euler angles, hyperplanes, lines, etc.), some non-linear solvers, automatic differentiations, etc. Thanks to expression templates, Eigen provides a very powerful and easy to use API. Explicit vectorization is performed for the SSE, AltiVec and ARM NEON instruction sets, with graceful fallback to non-vectorized code. Expression templates allow to perform global expression optimizations, and to remove unnecessary temporary objects.

Eigen is already a well established library with about 30k unique visitors of the website per month. Eigen is co-developed and maintained with a couple of other researchers and occasional contributors spread over the world. Its development started in 2008, and the last release is the 3.2 version in July 2013. Eigen has been supported by Inria through an ADT started in January 2012, and that ended in September 2013. This year, Eigen received the **"high-quality software in geometry processing award"** from the Symposium on Geometry Processing 2013 which was held in Genova, Pisa.

Facts:

- Web: http://eigen.tuxfamily.org/
- License: MPLv2

4.2. PatateLib

Participants: N. Mellado, G. Ciaudo, G. Guennebaud, P. Barla

Keywords : multi-scale analysis, material appearance, vector graphics, expressive rendering, 2D animation

Patate is a header only C++/CUDA library for graphics applications released under the MPL license.

It provides a collection of Computer Graphics techniques that incorporate the latest innovations from Inria research teams working in the field. It strives for efficiency and ease-of-use by focusing on low-level core operators and key algorithms, organised in modules that each tackle a specific set of issues. The central goal of the library is to drastically reduce the time and efforts required to turn a research paper into a ready-to-use solution, for both commercial and academic purposes.

Each module is initially developped by a few persons, usually those who have authored the corresponding research papers. An engineer, Gautier Ciaudo, has been recruited via the ADT program to perform unit tests, bug tracking, and make examples. Our first module provides efficient methods for the fitting and analysis of point-clouds in arbitrary dimensions. It may be used for varied purposes such as curvature computation, surface reconstruction, scale-space analysis, image processing, and sketch vectorization. More modules will be developped in 2014.
Facts:

- Web: http://patate.gforge.inria.fr/html/index.html
- License: MPLv2

4.3. PFSTools

Participant: I. Ihrke

Keywords : high dynamic range image processing, merging, calibration and tone-mapping

The `pfstools` package is a set of command line programs for reading, writing, manipulating and viewing high-dynamic range (HDR) images and video frames. All programs in the package exchange data using a simple generic high dynamic range image format, `pfs`, and they use unix pipes to pass data between programs and to construct complex image processing operations.

`pfstools` come with a library for reading and writing pfs files. The library can be used for writing custom applications that can integrate with the existing `pfstools` programs. It offer also a good integration with high-level mathematical programming languages, such as MATLAB or GNU Octave. `pfstools` can be used as the extension of MATLAB or Octave for reading and writing HDR images or simply to store effectively large matrices. The `pfstools` package is an attempt to integrate the existing high dynamic range image formats by providing a simple data format that can be used to exchange data between applications. It is accompanied by the `pfscalibration` and `pfstmo` packages.

The `pfscalibration` package provides an algorithm for the photometric calibration of cameras and for the recovery of high dynamic range (HDR) images from the set of low dynamic range (LDR) exposures. Maintenance of the pfscalibration package is done by Ivo Ihrke since January 2011. A major update to make the software compatible with current digital SLR cameras and their raw file formats, especially for measurement purposes, has been performed. A new set of MATLAB scripts has been developed for improved calibration performance. It is intended to merge these new procedures into the existing software.

The `pfstmo` package contains the implementation of seven state-of-the-art tone mapping operators suitable for convenient processing of both static images and animations.

The software received wider interest of the Open Source community and third party contributors prepared installation packages which are included in several Linux distributions including Debian, Fedora and Suse.
Facts:

- Web: http://pfstools.sourceforge.net/
- License: GPL

MAVERICK Project-Team

5. Software and Platforms

5.1. Introduction

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

5.2. Gratin

Participant: Romain Vergne [contact].

Gratin is a node-based compositing software for creating, manipulating and animating 2D and 3D data. It uses an internal direct acyclic multi-graph and provides an intuitive user interface that allows to quickly design complex prototypes. Gratin has several properties that make it useful for researchers and students. (1) it works in real-time: everything is executed on the GPU, using OpenGL, GLSL and/or Cuda. (2) it is easily programmable: users can directly write GLSL scripts inside the interface, or create new C++ plugins that will be loaded as new nodes in the software. (3) all the parameters can be animated using keyframe curves to generate videos and demos. (4) the system allows to easily exchange nodes, group of nodes or full pipelines between people. In a research context, Gratin aims at facilitating the creation of prototypes, testing ideas and exchanging data. For students, Gratin can be used to show real-time demos/videos, or help learning how to program with the GPU. Gratin has already been used for creating new computer graphics tools but also for designing perceptual experiments. Most of the work published by R. Vergne was done with Gratin.

5.3. PlantRad

Participant: Cyril Soler [contact].

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

5.4. High Quality Renderer

Participant: Cyril Soler [contact].

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

5.5. MobiNet

Participants: Fabrice Neyret [contact], Joëlle Thollot.

`../../../../projets/maverick/IMG/screenshot-gratin.jpg`

Figure 2. Gratin interface.

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

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

5.6. Freestyle

Freestyle is a software for Non-Photorealistic Line Drawing rendering from 3D scenes (Figure 3). It is designed as a programmable interface to allow maximum control over the style of the final drawing: the user "programs" how the silhouettes and other feature lines from the 3D model should be turned into stylized strokes using a set of programmable operators dedicated to style description. This programmable approach, inspired by the shading languages available in photorealistic renderers such as Pixar's RenderMan, overcomes the limitations of integrated software with access to a limited number of parameters and permits the design of an infinite variety of rich and complex styles. The system currently focuses on pure line drawing as a first step. The style description language is Python augmented with our set of operators. Freestyle was developed in the framework of a research project dedicated to the study of stylized line drawing rendering from 3D scenes. This research has lead to two publications [25], [26].

In 2008, Freestyle get a new life, completely outside Maverick or Inria: it was the basis of one of the 6 *Google Summer of Code* projects awarded to the *Blender Foundation* [1]! The goal of the project was to integrate Freestyle to the well known free 3D modeler *Blender*, as its standard NPR line-drawing renderer. Maxime Curioni (under the mentoring of Jean-Luc Peurière from the *Blender Foundation*), is currently making the integration. First beta versions are publicly available, and tested by enthusiasts around the web.

5.7. Diffusion Curves

Participant: Joëlle Thollot [contact].

We provide an implementation of the vector drawing tool described in our Diffusion Curves Siggraph paper [2] (Figure 4). This prototype is composed of the Windows binary, along with the required shader programs (ie. in source code). The software is available for download at http://artis.imag.fr/Publications/2008/OBWBTS08 for free, for non-commercial research purposes.

5.8. VRender: vector figures

Participant: Cyril Soler [contact].

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

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

---

[1] http://www.blender.org/

../../../../projets/maverick/IMG/freestyle.jpg

Figure 3. Stylized plane using Freestyle.

../../../../projets/maverick/IMG/diffusion_curves_ui.jpg

Figure 4. Diffusion curves freely downloadable demo.

VRender is also responsible for the vectorial snapshot feature of the QGLViewer library. VRender is released under the LGPL licence and is freely available for download at http://artis.imag.fr/Software/VRender.

5.9. ProLand

Participants: Fabrice Neyret [contact], Eric Bruneton.

Now available at http://proland.inrialpes.fr/ in double licencing GPL/commercial.

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

5.10. GigaVoxels

Participants: Fabrice Neyret [contact], Goswami Prashant, Sinoir Jérémy, Cyril Crassin, Pascal Guehl, Eric Heitz.

Soon available at http://gigavoxels.inrialpes.fr/index.htm in double licencing GPL/commercial.

Gigavoxel is a software platform initiated from the PhD work of Cyril Crassin, and currently funded by the ANR CONTINT RTIGE (Figure 5). The goal of this platform is the real-time quality rendering of very large and very detailed scenes which couldn't fit memory. Performances permit showing details over deep zooms and walk through very crowdy scenes (which are rigid, for the moment). The principle is to represent data on the GPU as a Sparse Voxel Octree which multiscale voxels bricks are produced on demand only when necessary and only at the required resolution, and kept in a LRU cache. User defined producer lays accross CPU and GPU and can load, transform, or procedurally create the data. Another user defined function is called to shade each voxel according to the user-defined voxel content, so that it is user choice to distribute the appearance-making at creation (for faster rendering) or on the fly (for storageless thin procedural details). The efficient rendering is done using a GPU differential cone-tracing using the scale corresponding to the 3D-MIPmapping LOD, allowing quality rendering with one single ray per pixel. Data is produced in case of cache miss, and thus only whenever visible (accounting for view frustum and occlusion). Soft-shadows and depth-of-field is easily obtained using larger cones, and are indeed cheaper than unblurred rendering. Beside the representation, data management and base rendering algorithm themself, we also worked on realtime light transport, and on quality prefiltering of complex data. Ongoing researches are addressing animation. GigaVoxels is currently used for the quality real-time exploration of the detailed galaxy in ANR RTIGE. This work led to several publications and several licences have been sold to companies.

../../../../projets/maverick/IMG/gigavoxel.jpg

Figure 5. GigaVoxels freely downloadable demo.

MIMETIC Project-Team

5. Software and Platforms

5.1. HPTS++: Hierarchical Parallel Transition System ++

Participant: Fabrice Lamarche [contact].

APP deposit number: IDDN.FR.001.290017.000.S.P.2003.000.10400

HPTS++ is a platform independent toolkit to describe and handle the execution of multi-agent systems. It provides a specific object oriented language encapsulating C++ code for interfacing facilities and a runtime kernel providing automatic synchronization and adaptation facilities.

The language provides functionalities to describe state machines (states and transitions) and to inform them with user specific C++ code to call at a given point during execution. This language is object oriented and supports concepts such as polymorphism and inheritance (state machines and user defined C++ classes). The compilation phase translates a state machine in a C++ class that can be compiled separately and linked through static or dynamic libraries. The runtime kernel includes a scheduler that handles parallel state machines execution and that provides synchronization facilities such as mutual exclusion on resources, dead lock avoidance, notions of priorities and execution adaptation in accordance with resources availability.

HPTS++ also provides a task model. Thanks to this model, the user can describe primitive behaviors through atomic tasks and combine them with operators (e.g. sequence, parallelism, loops, alternatives). Theses operators are fully dynamic. Hence they can be used at runtime to rapidly create complex behaviors.

5.2. MKM: Manageable Kinematic Motions

Participants: Richard Kulpa [contact], Franck Multon.

APP deposit numbers: IDDN.FR.000.290016.001.S.A.2003.000.41200, IDDN.FR.001.290016.001.S.A.2003.000.41200 et IDDN.FR.002.290016.001.S.A.2003.000.41200

We have developed a framework for animating human-like figures in real-time, based on captured motions. This work was carried-out in collaboration with the M2S Laboratory (Mouvement, Sport, Santé) of the University Rennes 2.

In this software, we propose a morphology-independent representation of the motion that is based on a simplified skeleton which normalizes the global postural informations. This formalism is not linked to morphology and allows very fast motion retargetting and adaptation to geometric constraints that can change in real-time This approach dramatically reduces the post production time and allows the animators to handle a general motion library instead of one library per avatar.

The framework provides an animation library which uses the motions either obtenained from our off-line tool (that tranforms standard formats into our morphology-independent representation) or parameterized models in order to create complete animation in real-time. Several models are proposed such as grasping, orientation of the head toward a target. We have also included a new locomotion model that allows to control the character directly using a motion database.

In order to create realistic and smooth animations, MKM uses motion synchronization, blending and adaptation to skeletons and to external constraints. All those processes are performed in real-time in an environment that can change at any time, unpredictably.

All these features have been used to anticipate and control the placement of footprints depending on high level parameters. This link between control and behavior levels will be used for reactive navigation in order to have realistic motion adaptations as well as to deal with constrained environments.

5.3. TopoPlan: Topological Planner and Behaviour Library

Participant: Fabrice Lamarche [contact].

APP deposit numbers: FR.001.480016.00.S.P.2008.000.41200

TopoPlan (Topological Planner) is a toolkit dedicated to the analysis of a 3D environment geometry in order to generate suitable data structures for path finding and navigation. This toolkit provides a two step process: an off-line computation of spatial representation and a library providing on-line processes dedicated to path planning, environmental requests...

TopoPlan is based on an exact 3D spatial subdivision that accurately identifies floor and ceiling constraints for each point of the environment. Thanks to this spatial subdivision and some humanoid characteristics, an environment topology is computed. This topology accurately identifies navigable zones by connecting 3D cells of the spatial subdivision. Based on this topology several maps representing the environment are extracted. Those maps identify obstacle and step borders as well as bottlenecks. TopoPlan also provides a runtime library enabling the on-line exploitation of the spatial representation. This library provides several algorithms including roadmap-based path-planning, trajectory optimization, footprint generation, reactive navigation and spatial requests through customizable spatial selectors.

TopoPlan behavior is a library built on top of TopoPlan and MKM providing several behaviors described thanks to the HPTS++ task model. Its goal is to provide a high level interface handling navigation and posture adaptation within TopoPlan environments. Provided behaviors include:

- A behavior handling fully planned navigation toward an arbitrary destination. This behavior precisely handles footprint generation within constrained environments such as stairs for instance.
- A behavior controlling an MKM humanoid to follow a trajectory specified by the user.
- A behavior controlling MKM to follow a list of footprints given by the user.
- A behavior adapting the humanoid posture to avoid collision with ceiling. This behavior runs in parallel of all other behaviors and adapts humanoid motion when needed without any user intervention.
- A behavior handling reactive navigation of virtual humans. This behavior plan a path to a given target and follows the path while avoiding collisions with other navigating entities.

Those behaviors have been built using the HPTS++ task model. Thus, they can be easily combined together or with other described behaviors through task operators.

MINT Project-Team

5. Software and Platforms

5.1. LibGINA

Participant: Laurent Grisoni [correspondant].

This library has been developped within the context of the ADT GINA, for one of the installation that have been made in collaboration with Le Fresnoy national studio (Damassama, Léonore Mercier). This library is currently being posted as APP, and has been used by Idées-3com small company, in the context of our join I-lab program. This library allows for use of gesture for command, and is able to handle strong variability into recognized patterns.

Current version: version 1.0

Software characterization: A-2 SO-3 SM-2-up EM-3 SDL-3 OC-DA4-CD4-MS2-TPM4

5.2. 3D interaction using mobile phone

Participants: Samuel Degrande [correspondant], Laurent Grisoni.

This work has been achieved in the context of the Idées-3com I-lab. In this context a module, that allows to use any android based smartphone to control an Explorer module for navigation and interaction with VRML-based content. This module was used as a basis by Idées-3com in their commercial product this year.

Current version: version 1.0

Software characterization: A-2 SO-3 SM-2-up EM-2-up SDL-3 OC-DA4-CD4-MS2-TPM4

5.3. tIO (tactile input & output)

Participants: Marc-Antoine Dupré, Nicolas Roussel [correspondant], Takashi Miyaki.

tIO is a library designed to facilitate the implementation of doubly tactile interaction techniques (tactile input coupled with tactile feedback) based on the STIMTAC technology. Supporting all current STIMTAC prototypes, it makes it easy to move the system pointer of the host computer according to motions detected on them and adapt their vibration amplitude based on the color of the pointed pixel or the nature of the pointed object. The library includes a set of `Qt` demo applications that illustrate these two different approaches and makes it easy to "augment" existing `Qt` applications with tactile feedback. It also makes it possible to supplement or substitute tactile feedback with basic auditory feedback synthesized using `portaudio` (friction level is linearly mapped to the frequency of a sine wave). This not only facilitates the development and documentation of tactile-enhanced applications but also makes it easier to demonstrate them to a large audience.

Software characterization: A2, SO3-up, SM-2, EM2, SDL1.

5.4. libpointing

Participants: Géry Casiez [correspondant], Damien Marchal, Nicolas Roussel.

Libpointing is a software toolkit that provides direct access to HID pointing devices and supports the design and evaluation of pointing transfer functions [2]. The toolkit provides resolution and frequency information for the available pointing and display devices and makes it easy to choose between them at run-time through the use of URIs. It allows to bypass the system's transfer functions to receive raw asynchronous events from one or more pointing devices. It replicates as faithfully as possible the transfer functions used by Microsoft Windows, Apple OS X and Xorg (the X.Org Foundation server). Running on these three platforms, it makes it possible to compare the replicated functions to the genuine ones as well as custom ones. The toolkit is written in C++ with Python and Java bindings available. It is publicly available under the GPLv2 license.

Web site: http://libpointing.org/

Software characterization: A3, SO3, SM-2, EM2, SDL4

5.5. Platform PIRVI

MINT is associated to the CPER-CIA (2007-2013), and participates to the PIRVI platform (Framework for Computer Human Animation, Virtual Reality and Images, handled by F. Aubert, co-animated by F. Aubert and D. Marchal), which aims at promoting research achieved by participant research teams (6 research teams, among which MINT), as well as encouraging collaborations with regional economical tissue on the knowledge fields covered within the associated research teams. The PIRVI allows these research teams to share a Virtual-Reality Room and various mid-size research equipments : multitouch tables, cameras (depth, infrared, ...), interactive devices (force-feedback, multitouch, smartphones...), a configurable multitouch wall. This dissemination activity has been supported with a regional contract 500 Keuros.

MORPHEO Team

5. Software and Platforms

5.1. Platforms

5.1.1. The Grimage platform

The Grimage platform is an experimental multi-camera platform dedicated to spatio-temporal modeling including immersive and interactive applications. It hosts a multiple-camera system connected to a PC cluster, as well as visualization facilities including head mounted displays. This platform is shared by several research groups, most proeminently Moais, Morpheo and Perception. In particular, Grimage allows challenging real-time immersive applications based on computer vision and interactions between real and virtual objects, Figure 1 . Note that the Grimage platform will be replaced by the Kinovis platform that will exhibit a larger acquisition space and better acquisition facilities.

`../../../../projets/morpheo/IMG/grimage.jpg`

Figure 1. Platform: the Grimage acquisition.

5.1.2. Kinovis

Kinovis (http://kinovis.inrialpes.fr/) is a new multi-camera acquisition project that was was selected within the call for proposals ”Equipements d’Excellence” of the program “Investissement d’Avenir˝ funded by the French government. The project involves 2 institutes: the Inria Grenoble Rhône-Alpes, the université Joseph Fourier and 4 laboratories: the LJK(laboratoire Jean Kuntzmann - applied mathematics), the LIG (laboratoire d’informatique de Grenoble - Computer Science), the Gipsa lab (Signal, Speech and Image processing) and the LADAF (Grenoble Hospitals - Anatomy). The Kinovis environment will be composed of 2 complementary platforms. A first platform located at the Inria Grenoble will have a 10mx10m acquisition surface and will be equipped with 60 cameras. It is the evolution of the Grimage platform previously described towards the production of better models of more complex dynamic scenes. A second platforms located at Grenoble Hospitals, within the LADAF anatomy laboratory, will be equipped with both color and X-ray cameras to enable combined analysis of internal and external shape structures, typically skeleton and bodies of animals. Installation works of both platforms started in 2013 and should be finished in 2014. Members of Morpheo are highly involved in this project. Edmond Boyer is coordinating this project and Lionel Reveret is in charge of the LADAF platform.

../../../../projets/morpheo/IMG/kinovis1.png

../../../../projets/morpheo/IMG/kinovis2.png

Figure 2. Kinovis platforms: on the left the Inria platform; on the right Grenoble Hospital platform.

5.1.3. Multicamera platform for video analysis of mice behavior

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

5.2. Software packages

5.2.1. LucyViewer

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

5.2.2. Ethomice

Ethomice http://morpheo.inrialpes.fr/people/reveret/ethomice/ is a motion analysis software to characterize motor behavior of small vertebrates such as mice or rats. From a multiple views video input, a biomechanical model of the skeleton is registered. Study on animal model is the first important step in Biology and Clinical research. In this context, the analysis of the neuro-motor behaviour is a frequent cue to test the effect of a gene or a drug. Ethomice is a platform for simulation and analysis of the small laboratory animal, such as rat or mouse. This platform links the internal skeletal structure with 3D measurements of the external appearance of the animal under study. From a stream of multiple views video, the platform aims at delivering a three dimensional analysis of the body posture and the behaviour of the animal. The software was developed by Lionel Reveret and Estelle Duveau. An official APP repository has been issued this year.

../../../../projets/morpheo/IMG/ethomice.png

Figure 3. Ethomice: Experimental platform for video analysis of mice behavior.

5.3. Databases

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

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

OAK Project-Team

5. Software and Platforms

5.1. Amada

Name: Amada (https://team.inria.fr/oak/amada/)

Contact: Jesús Camacho-Rodríguez (jesus.camacho-rodriguez[at]inria.fr))

Other contacts: Ioana Manolescu (ioana.manolescu[at]inria.fr), Dario Colazzo (dario.colazzo[at]dauphine.fr), François Goasdoué (fg[at]irisa.fr)

Presentation: A platform for Web data management in the Amazon cloud.

5.2. FactMinder

Name: FactMinder (http://tripleo.saclay.inria.fr/xr/demo/)

Contact: Julien Leblay (julien.leblay[at]inria.fr)

Other contacts: Stamatis Zampetakis (stamatis.zampetakis[at]inria.fr), François Goasdoué (fg[at]irisa.fr), Ioana Manolescu (ioana.manolescu[at]inria.fr)

Presentation: A system for archiving, annotating, and querying semantic-rich Web content.

5.3. Nautilus Analyzer

Name: Nautilus Analyzer (http://nautilus.saclay.inria.fr/)

Contact: Melanie Herschel (melanie.herschel[at]lri.fr)

Other contacts: n.a.

Presentation: A tool for analyzing and debugging SQL queries using why-provenance and why-not provenance.

5.4. RDFViewS

Name: RDFViewS (http://tripleo.saclay.inria.fr/rdfvs/)

Contact: Konstantinos Karanasos (kkaranasos[at]gmail.com)

Other contacts: François Goasdoué (fg[at]irisa.fr), Julien Leblay (julien.leblay[at]gmail.com), and Ioana Manolescu (ioana.manolescu[at]inria.fr)

Presentation: A storage tuning wizard for RDF applications.

5.5. ViP2P

Name: ViP2P (Views in Peer-to-Peer, http://vip2p.saclay.inria.fr)

Contact: Ioana Manolescu (ioana.manolescu[at]inria.fr)

Other contacts: Jesús Camacho-Rodríguez (jesus.camacho-rodriguez[at]inria.fr)

Presentation: A P2P platform for disseminating and querying XML and RDF data in large-scale distributed networks.

5.6. WARG

Name: WARG (https://team.inria.fr/oak/warg/)

Contact: Alexandra Roatiş (alexandra.roatis[at]lri.fr))

Other contacts: Ioana Manolescu (ioana.manolescu[at]inria.fr), Dario Colazzo (dario.colazzo[at]dauphine.fr), François Goasdoué (fg[at]irisa.fr)

Presentation: A platform for specifying and exploiting warehouses of RDF data.

5.7. XUpOp

Name: XUpOp (XML Update Optimization)

Contact: Dario Colazzo (dario.colazzo[at]dauphine.fr)

Other contacts: Nicole Bidoit (bidoit[at]lri.fr), Mohamed Amine Baazizi (baazizi[at]lri.fr)

Presentation: A general purpose type-based optimizer for XML updates.

5.8. XUpIn

Name: XUpIn (XML Update Independence)

Contact: Federico Ulliana (Federico.Ulliana[at]lri.fr)

Other contacts: Dario Colazzo (colazzo[at]lri.fr), Nicole Bidoit (bidoit[at]lri.fr)

Presentation: An XML query-update independence tester.

5.9. XUpTe

Name: XUpTe (XML Update for Temporal Documents)

Contact: Dario Colazzo (dario.colazzo[at]dauphine.fr)

Other contacts: Nicole Bidoit (bidoit[at]lri.fr), Mohamed-Amine Baazizi (amine.baazizi[at]gmail.com)

Presentation: A type-based optimizer for representing and updating XML temporal data.

5.10. XPUQ

Name: XPUQ (XML Partitioning for Updates and Queries)

Contact: Dario Colazzo (dario.colazzo[at]dauphine.fr)

Other contacts: Nicole Bidoit (bidoit[at]lri.fr)

Presentation: A static analyzer and partitioner for XML queries and updates.

ORPAILLEUR Project-Team

5. Software and Platforms

5.1. Generic Symbolic KDD Systems

5.1.1. The Coron Platform

Participants: Jérémie Bourseau [contact person], Aleksey Buzmakov, Victor Codocedo, Adrien Coulet, Amedeo Napoli, Yannick Toussaint.

Keywords: data mining, frequent itemset, closed itemset, generator, association rule, rare itemset

The Coron platform [117], [101] 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 (see http://coron.loria.fr). The Coron-base component includes a complete collection of data mining algorithms for extracting itemsets such as frequent itemsets, closed itemsets, generators and rare itemsets. In this collection we can find APriori, Close, Pascal, Eclat, Charm, and, as well, original algorithms such as ZART, Snow, Touch, and Talky-G. 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].

Keywords: skyline, skycube

This program implements the algorithms described in a research paper published at VLDB 2010 [111]. 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].

Keywords: Hidden Markov Models, stochastic process

The system CarottAge is based on Hidden Markov Models of second order and provides a non supervised temporal clustering algorithm for data mining and 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. CarottAge is also used for mining hydromorphological data. Actually a comparison was performed with three other algorithms classically used for the delineation of river continuum and CarottAge proved to give very interesting results for that purpose [102].

CarottAge is freely available under GPL license (see http://www.loria.fr/~jfmari/App/).

5.2.2. The ARPEnTAge system

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

Keywords: Hidden Markov Models, stochastic process

ARPEnTAge [1] (for *Analyse de Régularités dans les Paysages: Environnement, Territoires, Agronomie* is a software based on stochastic models (HMM2 and Markov Field) for analyzing spatio-temporal data-bases [107]. 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. It performs a Time-Space clustering of a landscape based on its time dynamic Land Uses (LUS). Displaying tools and the generation of Time-dominant shape files have also been defined.

ARPEnTAge is freely available (GPL license) 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. In these practical applications, CarottAge and ARPEnTAge aim at building a partition –called the hidden partition– in which the inherent noise of the data is withdrawn as much as possible. The estimation of the model parameters is performed by training algorithms based on the Expectation Maximization and Mean Field theories. The ARPEnTAge 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).

ARPEnTAge and CarottAge were used for mining decision rules in a territory showing environmental issues. They provide a way of visualizing the impact of farmers decision rules in the landscape and revealing new extra hidden decision rules [116].

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 [83]. The IntelliGO measure is available on line (http://plateforme-mbi.loria.fr/intelligo/) to be used 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

KNIME (for “Konstanz Information Miner”) is an open-source visual programming environment for data integration, processing, and analysis. KNIME 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.

Various KNIME nodes were developed for supporting relational data mining using the ALEPH program (http://www.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/aleph.pl). These nodes include a data preparation node for defining a set of first-order predicates from a set of relation schemes 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.

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

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

---

[1] http://www.loria.fr/~jfmari/App/

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 version of the software was declared 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. MODIM was also used for organizing data about non ribosomal peptide syntheses. 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.

Keywords: classification-based reasoning, case-based reasoning, decision knowledge management, knowledge edition, knowledge base maintenance, semantic portal

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 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) is 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 (see [17] where an an extension of Kasimir for multi-viewpoint case-based reasoning is presented).

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. Actually, the mining process in CabamakA is based on a frequent close itemset extraction module from the Coron platform (see §5.1.1).

The Oncologik system [12] is a collaborative editing tool aiming at facilitating the management of medical guidelines (http://www.oncologik.fr/). Based on a semantic wiki, it allows the acquisition of formalized decision knowledge. Oncologik also includes a graphical decision tree editor called KcatoS.

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

Participants: Valmi Dufour-Lussier, Emmanuelle Gaillard, Laura Infante Blanco, Florence Le Ber, Jean Lieber, Amedeo Napoli, Emmanuel Nauer [contact person].

Keywords: knowledge acquisition, ontology engineering, semantic annotation, case-based reasoning, hierarchical classification, text mining

Taaable [69] 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 such as 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, were 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 since 2008 at http://taaable.fr. A new version of Taaable was implemented using Tuuurbine, a generic ontology-guided CBR engine based on semantic web technologies (see Section 5.4.3). BeGoood (see Section 5.4.4), a generic system for managing non-regression tests on knowledge bases, is also plugged for acquiring test sets. When the Taaable system returns answers to a query, the user may evaluate the relevance of the answers. Currently, user feedback is collected using BeGoood and will be used in the future to run tests when the knowledge exploited by the CBR system evolves. The objective is to ensure that the knowledge base evolution does not affect the quality of answers given by the CBR system.

5.4.3. Tuuurbine: a generic ontology guided case-based inference engine

Participants: Laura Infante Blanco, Jean Lieber, Emmanuel Nauer [contact person].

Keywords: case-based reasoning, inference engine, knowledge representation, ontology engineering, semantic web

The experience acquired since 5 years with the Taaable system conducted to the creation of a generic cased-based reasoning system, whose reasoning procedure is based on a domain ontology. This new system, called Tuuurbine (http://tuuurbine.loria.fr/), takes into account the retrieval step, the case base organization, but also an adaptation procedure which is not addressed by other generic case-based reasoning tools. Moreover, Tuuurbine is built over semantic web standards allowing to be connected to the web of data. The domain knowledge is represented in an RDF store, which can be interfaced with a semantic wiki, for collaborative edition and management of the knowledge involved in the reasoning system (cases, ontology, adaptation rules). The development of Tuuurbine was supported by an Inria ADT funding until October 2013.

5.4.4. BeGoood: a generic system for managing non-regression tests on knowledge-bases

Participants: Laura Infante Blanco, Emmanuel Nauer [contact person].

Keywords: tests, non-regression, knowledge evolution

BeGoood [67] is a system allowing to define test plans, independent of any application domain, and usable for testing any system answering queries by providing results in the form of sets of strings. BeGoood provides all the features usually found in test systems, such as tests, associated queries, assertions, and expected result sets, test plans (sets of tests) and test reports. The system is able to evaluate the impact of a system modification by running again test plans and by evaluating the assertions which define whether a test fails or succeeds. The main components of BeGoood are (1) the “test database” that stores every test artifacts, (2) the “remote query evaluator” which evaluates test queries, (3) the “assertion engine” which evaluates assertions over the expected and effective query result sets. and finally (4) the “REST API” which offers the test functionalities as web services.

BeGoood is available under a AGPL license on github [2]. BeGoood is used to manage the non-regression of the Taaable system (see Section 5.4.2) when the knowledge base used by the CBR system is modified.

5.4.5. Revisor: a library of revision operators and revision-based adaptation operators

Participants: Valmi Dufour-Lussier, Alice Hermann, Florence Le Ber, Jean Lieber [contact person], Emmanuel Nauer, Gabin Personeni.

Keywords: belief revision, adaptation, revision-based adaptation, case-based reasoning, inference engines, knowledge representation

Revisor is a library of inference engines dedicated to belief revision and to revision-based adaptation for case-based reasoning [60]. It is open source, under a GPL license and available on the web (http://revisor.loria.fr/). It gathers several engines developed during the previous years, for various knowledge representation formalisms (propositional logic—with or without the use of adaptation knowledge [65]—conjunction of linear constraints, and qualitative algebras [3]). Some of these engines are already used in the Taaable system. Current developments on Revisor aim at defining new engines in other formalisms.

---

[2] https://github.com/kolflow/begoood

PANAMA Project-Team

5. Software and Platforms

5.1. MPTK: the Matching Pursuit Toolkit

Participants: Rémi Gribonval [contact person], Jules Espiau de Lamaestre.

The Matching Pursuit ToolKit (MPTK) is a fast and flexible implementation of the Matching Pursuit algorithm for sparse decomposition of monophonic as well as multichannel (audio) signals. MPTK is written in C++ and runs on Windows, MacOS and Unix platforms. It is distributed under a free software license model (GNU General Public License) and comprises a library, some standalone command line utilities and scripts to plot the results under Matlab. This software has been registered at the APP (Agence de Protection des Programmes).

http://mptk.gforge.inria.fr, http://mptk.irisa.fr

5.2. FASST: a Flexible Audio Source Separation Toolbox

Participants: Nancy Bertin, Frédéric Bimbot.

Emmanuel Vincent [contact person]

FASST is a Flexible Audio Source Separation Toolbox, designed to speed up the conception and automate the implementation of new model-based audio source separation algorithms.

FASST is currently being developed jointly with the PAROLE team in Nancy and the TEXMEX team in Rennes through an Inria funded ADT (Action de Développement Technologique). The first implementation is in Matlab. http://bass-db.gforge.inria.fr/fasst/

5.3. NACHOS: Nearfield Acoustic HOlography with Sparse regularization

Participants: Nancy Bertin [contact person], Rémi Gribonval.

The software and associated database were developed within the ANR ECHANGE project, with the participation of Gilles Chardon, Laurent Daudet, François Ollivier and Antoine Peillot.

NACHOS (Nearfield Acoustic HOlography with Sparse regularization) is a downloadable companion software for the journal paper [79], distributed to comply with the "reproducible research" principle. It performs the reconstruction of operational deflection shapes of a vibrating structure, from acoustic measurements of the generated sound field. The software consists in Matlab source code, and automatically downloads the needed database. It allows to reproduce all results and figures of the paper, and to experiment some additional settings. It is distributed under GPL 3.0 license. Inter Deposit Digital Numbers: IDDN.FR.001.420023.000.S.P.2013.000.31235 (NACHOSDB) % IDDN.FR.001.420023.000.S.P.2013.000.31235 (NACHOS).

http://echange.inria.fr/nah.

PAROLE Project-Team

5. Software and Platforms

5.1. WinSnoori

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 WinSnoori 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 current version of WinSnoori is available on http://www.winsnoori.fr.

5.2. JSnoori

JSnoori is written in Java and uses signal processing algorithms developed within WinSnoori software with the double objective of being a platform independent signal visualization and manipulation tool, and also for designing exercises for learning the prosody of a foreign language. JSnoori thus focused the calculation of F0, the forced alignment of non native English uttered by French speakers and the correction of prosody parameters (F0, rhythm and energy). Since phonetic segmentations and annotations play a central role in the derivation of diagnosis concerning the realization of prosody by learners, several tools have been incorporated to segment and annotate speech. In particular, a complete phonetic keyboard is available, several kinds of annotation can be used (phonemes, syllables and words) and forced alignment can exploit variants to cope with non native accents. In addition, JSnoori offers real time F0 calculation which can be useful from a pedagogical point of view.

5.3. Xarticulators

Xarticulators software is intended to delineate contours of speech articulators in X-ray images, to construct articulatory models and to synthesize speech from X-ray films. This software provide tools to track contours automatically, semi-automatically or by hand, to make the visibility of contours easier, to add anatomical landmarks to speech articulators and to synchronize images together with the sound.

It also enables the construction of adaptable linear articulatory models from the X-ray images.

This year we particularly worked on the possibility of synthesizing speech from X-ray images. We thus substantially improved algorithms used to compute the centerline of the vocal tract in order to segment the vocal tract into elementary tubes approximating the propagation of a one-dimensional wave. We also developed time patterns used to synthesize sequences of voiceless consonants and vowels (VCV). In addition we also added the possibility of processing digitized manual delineation results made on sheet of papers in the seventies.

5.4. SUBWEB

We published in 2007 a method which allows to align sub-titles comparable corpora [94]. 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].

---

[2] http://wikitalc.loria.fr/dokuwiki/doku.php?id=operations:subweb

5.5. ANTS

The aim of the Automatic News Transcription System (ANTS) is to transcribe radio or TV shows. ANTS is composed of several stages. The first processing steps aim at splitting the audio stream into homogeneous segments of a manageable size and at identifying the segment characteristics in order to allow the use of specific algorithms or models according to the nature of the segment. This includes broad-band/narrow-band speech segmentation, speech/music classification, speaker segmentation and clustering, detection of silences/breathing segments and generally speaker gender classification.

Each segment is then decoded using a large vocabulary continuous speech recognition engine, either the Julius engine or the Sphinx engine. The Julius engine 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. The Sphinx engine processes the speech input segment in a single forward pass using a trigram language model.

Further processing passes are usually run in order to apply unsupervised adaptation processes on the feature computations (VTLN: vocal tract length normalization) and/or on the model parameters (MLLR: maximum likelihood linear regression), or to use speaker adaptive training (SAT) based models. Moreover decoding results of both systems can be efficiently combined for improved decoding performance.

The latest version which relies on a perl script exploits the multiple CPUs available on a computer to reduce the processing time, and runs on both a stand alone linux machine and on the cluster.

5.6. CoALT

CoALT (Comparing Automatic Labeling Tools) compares two automatic labelers or two speech-text alignment tools, ranks them and displays statistics about their differences. The main feature of our software is that a user can define its own criteria for evaluating and comparing two speech- text alignment tools. With CoALT, a user can give more importance to either phoneme labels or phoneme boundaries because the CoALT elastic comparison algorithm takes into account time boundaries. Moreover, by providing a set of phonetic rules, a user can define the allowed discrepancies between the automatic labeling result and the hand-labeling one.

5.7. TTS SoJA

TTS SoJA (Speech synthesis platform in Java) is a software for text-to-speech synthesis. 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 the 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 launch 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

5.8. JCorpusRecorder

JCorpusRecorder is a software for the recording of audio corpora. It provides a easy tool to record with a microphone. The audio input gain 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 displaying a textual/visual/audio context of the sentence to pronounce. This software is suitable for recording sentences with information to guide the speaker. The sentences can be presented randomly.

The software is now developed in Java (since 2013). It is currently used for the recording of sentences in several projects (including IFCASL).

5.9. VisArtico

VisArtico is intended to visualize articulatory data acquired using an articulograph [97]. It is intended for researchers that need to visualize data acquired from the articulograph with no excessive processing. It is well adapted to the data acquired using the AG500 and AG501 (developed by Carstens Medizinelektronik GmbH), and the articulograph NDI Wave, developed by Northern Digital Inc.

The software allows displaying the positions of the sensors that are simultaneously animated with the speech signal. It is possible to display the tongue contour and the lips contour. The software helps to find the midsagittal plane of the speaker and find the palate contour. In addition, VisArtico allows labeling phonetically the articulatory data.

All this information is very useful to researchers working in the field of speech production, as phoneticians for instance. VisArtico provides several possible views: (1) temporal view, (2) 3D spatial view and (3) 2D midsagittal view. In the temporal view, it is possible to display different articulatory trajectories in addition to the acoustic signal and eventually labels. The midsagittal view can display the tongue contour, the jaw, the lips and the palate.

VisArtico provides several tools to help to improve the quality of interpreting the data. It is a cross-platform software as it is developed in JAVA and does not need any additional external library or framework. It was tested and worked on Windows, Mac OS, and Linux. It should work on any system having JAVA installed. VisArtico is freely distributed via a dedicated website http://visartico.loria.fr.

5.10. FASST

The Flexible Audio Source Separation Toolbox (FASST) is a toolbox for audio source separation (http://bass-db.gforge.inria.fr/fasst/). It aims to become the reference software for research and applications of audio source separation. Its unique feature is the possibility for users to specify easily a suitable algorithm for their use case thanks to the general modeling and estimation framework. Besides, it forms the basis of most of our current research in audio source separation, some of which may be incorporated into future versions of the software.

PERCEPTION Team

5. Software and Platforms

5.1. Mixed camera platform

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

`../../../../projets/perception/IMG/room-inward.jpg`

Figure 1. The mixed multi-camera system composed of four TOF-stereo sensor units.

5.2. Audiovisual robot heads

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

For more information on POPEYE and on NAO please visit https://team.inria.fr/perception/popeye/ and https://team.inria.fr/perception/nao/.

../../../../projets/perception/IMG/nao-new.png

../projets/perception/IMG/popeye-

Figure 2. Left: The consumer humanoid robot NAO is equipped with a binocular-binaural head specially designed for human-humanoid interaction; Right: The binocular-binaural robot head POPEYE equipped with a four degrees of freedom stereo camera pair and with a dummy head.

POTIOC Team

5. Software and Platforms

5.1. OpenViBE

Participants: Fabien Lotte [local correspondant], Alison Cellard [engineer].

As part of our research work on BCI, we contribute to the development of the OpenViBE [2] software, which is an open source platform dedicated to the design, evaluation and use of BCI for real and virtual applications. OpenViBE development is led by Inria, and Potioc is one of the Inria team contributing to its evolution. Moreover, Potioc is involved in the Inria ADT (Technological Development Action) OpenViBE-NT that is dedicated to the development of OpenViBE together with 3 other Inria teams (Hybrid, Athena, Neurosys).

5.2. Drile

Participant: Florent Berthaut.

As part of the research on Virtual Reality for Musical Performance, notably the Drile system, various software pieces are being developped and made available to the community. These software pieces are the following:

- Pure-Data external to access data from the Virtual Reality Peripheral Network : https://github.com/scrime/vrpd
- Drile: http://hitmuri.net/index.php/Research/Drile

5.3. 3DCityTestbed

Participants: Jacek Jankowski, Thomas Hulin.

As part of the research on the "Villes transparentes" project in collaboration with Mappy (Solocal group) and Vectuel - VirtuelCity, we develop a software platform dedicated to the design and evaluation of innovative interaction techniques for the navigation in urban 3D environments. This code is not publicly available yet.

5.4. PapARt

Participant: Jérémy Laviole [Main developer].

As part of his thesis work, Jérémy Laviole has developed a software suite for PapARt : Paper Augmented Reality Toolkit. This suite enables the calibration of depth cameras such as the Kinect with a planar surface and with a videoprojection. It also enables the detection of finger touch on the planar surface. This system can be extended to 3D objects. Many external devices are compatible, such as pen tablets (Wacom) and LEAP Motion. The code runs on Linux, Mac OS and Windows, and is made for Processing [3]. This code is not publicly available yet.

---

[2] http://openvibe.inria.fr

[3] http://www.processing.org

Prima Project-Team

4. Software and Platforms

4.1. OMiSCID Middleware for Distributed Multimodal Perception

Participants: Rémi Barraquand, Amaury Nègre, Patrick Reignier, Dominique Vaufreydaz [correspondant].

Middleware, Distributed perceptual systems

OMiSCID is lightweight middleware for dynamic integration of perceptual services in interactive environments. This middleware abstracts network communications and provides service introspection and discovery using DNS-SD (*DNS-based Service Discovery* [31]). Services can declare simplex or duplex communication channels and variables. The middleware supports the low-latency, high-bandwidth communications required in interactive perceptual applications. It is designed to allow independently developed perceptual components to be integrated to construct user services. Thus our system has been designed to be cross-language, cross-platform, and easy to learn. It provides low latency communications suitable for audio and visual perception for interactive services.

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

The OMiSCID graphical user interface (GUI) is an extensible graphical application that facilitates analysis and debugging of service oriented applications. The core functionality of this GUI is to list running services, their communication channels and their variables. This GUI is highly extensible and many modules (i.e. plugins) have been created by different members of the team: figure 2 shows an example of some of these modules. OMiSCID GUI is based on the Netbeans platform and thus inherits from its dynamic installation and update of modules.

4.2. Pal-Gate

Participants: Rémi Barraquand, Amaury Nègre, Dominique Vaufreydaz [correspondant].

Middleware, Distributed perceptual systems, Robotic Operating System (ROS), IPL PAL

A part of our efforts in the PAL project has been put toward developing a solution that would ease the integration of our multi-partners' software components. We refer to this solution as PALGate.

The design of PALGate results from the obvious observation that, within the PAL project, each partner must be considered as an ecosystem characterized, among other things, by 1) its software culture e.g. its curiosity and knowledge about software concepts, software architectures and design patterns, programatic languages, etc.; 2) its resources, e.g. its manpower, its possession or not of an experimental platform; 3) its competences and fields of research and expertise; 4) its habits e.g. its uses of a particular programming language, (c/c++, Java, Python) and computing platforms (OSx, Linux, Windows, Android, etc.), its adoption or not of a dedicated technology to interconnect software components (OSGi, OMiSCID, MPI, PVM, etc.); and 5) its particular needs and constraints e.g. requirement of a hard real-time system, mobility, etc.

For it to be widely accepted, PALGate is therefore designed to be ecologic and pragmatic. Ecologic in the sense that the solution does not perturb the ecology of each ecosystem [1], pragmatic in the sense that setting up this solution did not require an heavy development effort, also because it was targetted to PAL and is taking as much as possible advantage of existing solutions.

---

[1] namely, if a partner is used to Java and OSGi, deploying PALGate will not affect this in any way nor engender an heavy effort to interface it.

`../../../../projets/prima/IMG/omiscid-gui.png`

Figure 2. OMiSCID GUI showing a list of running services and some modules for service interconnections, variable plotting, live video stream display and variable control

For developing PALGate we introduced a novel concept: software gate. Unlike software components/services which can be instantiated, a software gate is only a concept, it is defined as an ecologic and hermetic interface between different ecosystems. A software gate is characterized by the subset of functionalities it exposes to other gates, where the functionalities it exposes are provided by the software components/services of its belonging ecosystem. A software gate is hermetic in the sense that only a selected subset of functionalities of an ecosystem are exposed but also because it propagates only filtered information exposed by other gates into its ecosystem. The last characteristic of a software gate is that it makes explicit to other gates the communication mechanisms it uses.

While a software gate is only conceptual, PALGate is an implementation of a gate oriented middleware. PALGate uses ROS to support the basic communication between gates. Within PALGate, each ecosystem is associated to only one software gate. Practically, PALGate 1) is a ROS stack containing gates definition 2) is a set of conventions (e.g. stack organization, package/node/topic/service names, namespaces, etc.) 3) it provides dedicated tools to ease the integration and its usage by partners. A software gate in PALGate is a ROS package containing definition of ROS types (i.e. msgs and srvs types), but also exposed ROS communication channels (i.e. topics and RPCs).

With this architecture each partner has to provide PALGate with a package containing the definition of its gate. Then in order a) to expose functionalities out of their ecosystem and b) to propagate information into their ecosystem, each partner must create ROS nodes. These ROS nodes let each partner interface their ecosystem through ROS topics and ROS services without having to change anything about their architecture. For instance if a partner is using Java and OSGi, it can create nodes in ROS Java that will expose/register functionalities through ROS services, publish/subscribe information using ROS topics.

4.3. EmoPRAMAD

Participants: Claudine Combe, Dominique Vaufreydaz [correspondant].

Affective computing,

Within the Pramad project, we want to offer a full affective loop between the companion robot and the elderly people at home. This affective loop is necessary within the context of everyday interaction of elderly and the companion robot. A part of this loop is to make the robot express emotions in response to the emotional state of the user. To do that, we need to test our working hypothesis about the visual representation of emotions with the 3D face of robot. EmoPRAMAD is an evaluation tool designed to conduct comparative studies between human faces and the 3D faces expressing a defined set of emotions.

The evaluation conducted though EmoPRAMAD concerns both unimodal (facial only) and bimodal conditions (facial/sound). The emotions set is composed of 4 basic emotions (joy, fear, anger, sadness) and a neutral state. While experimenting, the software collects several parameters in order to evaluate more than correctness of the answers: time to respond, length of mouse moves, etc. Experimentation is still in progress at Inria in Grenoble, University Pierre and Marie Currie and Broca Hospital in Paris. A set of 235 participants from 14 to 88 years old was already recorded.

4.4. Detection and Tracking of Pedestrians in INRETS Intelligent Urban Spaces Platform

Participants: Claudine Combe, James Crowley [correspondant], Lukas Rummelhard.

Visual detection and tracking of pedestrians, Intelligent Urban Space

The project ANR-07-TSFA-009-01 CIPEBUS ("Carrefour Intelligent - Pole d'Echange - Bus) has been proposed by INRETS-IFSTTAR, in collaboration with Inria, Citilog, Fareco, and the city of Versaille. The Objective of the CIPEBUS project is to develop an experimental platform for observing activity in a network of urban streets in order to experiment with techniques for optimizing circulation by context aware control of traffic lights.

../../../../projets/prima/IMG/EmoPramad.png

Figure 3. EmoPRAMAD interfaces with a human face and a 3D face from our virtual agent.

../../../../projets/prima/IMG/Cipebus-pedestrians.png

Figure 4. Cipebus: pedestrian tracking system.

Within CipeBus, Inria has developed a real time multi-camera computer vision system to detect and track people using a network of surveillance cameras. The CipeBus combines real time pedestrian detection with 2D and 3D Bayesian tracking to record the current position and trajectory of pedestrians in an urban environment under natural view conditions. The system extends the sliding window approach to use a half-octave Gaussian Pyramid to explore hypotheses of pedestrians at different positions and scales. A cascade classifier is used to determine the probability that a pedestrian can be found at a particular position and scale. Detected pedestrians are then tracked using a particle filter.

The resulting software system has been installed and tested at the INRETS CipeBus platform and is currently used for experiments in controlling the traffic lights to optimize the flow of pedertrians and public transportation while minimizing the delay imposed on private automobiles.

4.5. Multisensor observation of human activity for integrated energy and comfort management

Participants: Claudine Combe, James Crowley [correspondant], Lucas Nacsa, Amaury Nègre, Lukas Rummelhard.

multimodal tracking of human activity

As part of Inria's contribution of ICTLabs Action TSES - Smart Energy Systems, we have constructed a system that integrates information from multiple environmental sensor to detect and track people in indoor environments. This system, constructed as part of activity 11831 Open SES Experience Labs for Prosumers and New Services, has been released to ICTLabs partners in June 2012. It has also been used for construction of a smart spaces testbed at Schneider Electric.

This software, named MultiSensor activity tracker, integrates information from multiple environmental sensors to keep track of the location and activity of people in a smart environment. This model is designed to be used by a home energy broker that would work in conjunction with a smart grid to manage the energy consumption of home appliances, balancing the needs of inhabitants with opportunities for savings offered by electricity rates. This database will also be used for by advisor services that will offer advice to inhabitants on the consequences to energy consumption and energy cost that could potentially result from changes to lifestyle or home energy use.

Work in this task draws from earlier result from a number of development projects at Inria. In the ANR Casper project Inria created Bayesian tracking system for human activity using a voxel based occupancy grid. Within the INRA ADT PAL project, Inria is creating methods for plug and play installation of visual and acoustic sensors for tracking human activity within indoor environments.

While a voxel based Bayesian tracker has served well for a number of applications, a number of limitations have been observed. For example, under certain circumstances, the sensor data can provide contradictory or ambiguous data about the location and activities of people. Resolving such cases required the Bayesian tracker to choose between a numbers of competing hypotheses, potentially resulting in errors. Several members of our group have argued that an alternative integration approach based on the use of a Particle filter would solve these problems and provide a more reliable tracking system. This task has been undertaken to evaluate this hypothesis. The system configured and optimized for detecting and tracking people within rooms using multiple calibrated cameras. The system currently uses corner mounted cartesian cameras, ceiling mounted cameras with wide angle lenses and panoramic cameras placed on tables. Cameras may be connected and disconnected while the component is running, but they must be pre-calibrated to a common room reference frame. We are currently experimenting with techniques for Bayesian estimation of camera parameters for auto-calibration. Cameras may be connected dynamically.

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

`../../../../projets/prima/IMG/3DSmartRoom.jpg`

Figure 5. The 3D tracker integrates observations from multiple sensors

4.6. Stereo Viewfinder

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

Stereoscopy, Auto-calibration, Real-time video processing, Feature matching

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

4.7. Tracking Focus of Attention for Large Screen Interaction

Participants: Rémi Barraquand, Claudine Combe, James Crowley [correspondant], Varun Jain, Sergi Pujades-Rocamora, Lukas Rummelhard.

Embedded Detection and Tracking of Faces for Attention Estimation.

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

The objective of this task is to leverage recent technological advances in real time face detection developed for cell phones and mobile computing to provide a low-cost real time visual sensor for observing users of large multi-touch interactive displays installed in public spaces.

People generally look at things that attract their attention. Thus it is possible to estimate the subject of attention by estimating where people look. The location of visual attention is manifested by a region of space known as the horopter where the optical axis of the two eyes intersect. However estimating the location of attention from human eyes is notoriously difficult, both because the eyes are small relative to the size of the face, and because eyes can rotate in their socket with very high accelerations. Fortunately, when a human attends to something, visual fixation tends to remain at or near that subject of attention, and the eyes are relaxed to a symmetric configuration by turning the face towards the subject of attention. Thus it is possible to estimate human attention by estimating the orientation of the human face.

We have constructed an embedded software system for detecting, tracking and estimating the orientation of human faces. This software has been designed to be embedded on mobile computing devices such as laptop computers, tablets and interactive display panels equipped with a camera that observes the user. Noting the face orientation with respect to the camera makes it possible to estimate the region of the display screen to which the user is attending.

The system uses a Bayesian Particle filter tracker operating on a Scale invariant Gaussian pyramid to provide integrated tracking and estimation of face orientation. The use of Bayesian tracking greatly improves both the reliability and the efficiency for face detection and orientation estimation. The scale invariant Gaussian pyramid provides automatic adaptation to image scale (as occurs with a change in camera optics) and makes it possible to detect and track faces over a large range of distances. Equally important the Gaussian Pyramid provides a very fast computation of a large number of image features that can be used by a variety of image analysis algorithms.

An similar software was released in 2007 using face color rather than appearance. The system SuiviDeCiblesCouleur located individuals in a scene for video communications. FaceStabilsationSystem renormalised the position and scale of images to provide a stabilised video stream. SuiviDeCiblesCouleur has been declared with the APP "Agence pour la Protection des Programmes" under the Interdeposit Digital number IDDN.FR.001.370003.000.S.P.2007.000.21000.

4.8. Visual Emotion Recognition for Health and Well Being.

Participants: Rémi Barraquand, Claudine Combe, James Crowley [correspondant], Varun Jain, Sergi Pujades-Rocamora, Lukas Rummelhard.

Visual Emotion Recognition

People express and feel emotions with their face. Because the face is the both externally visible and the seat of emotional expression, facial expression of emotion plays a central role in social interaction between humans. Thus visual recognition of emotions from facial expressions is a core enabling technology for any effort to adapt ICT to improve Health and Wellbeing.

Constructing a technology for automatic visual recognition of emotions requires solutions to a number of hard challenges. Emotions are expressed by coordinated temporal activations of 21 different facial muscles assisted by a number of additional muscles. Activations of these muscles are visible through subtle deformations in the surface structure of the face. Unfortunately, this facial structure can be masked by facial markings, makeup, facial hair, glasses and other obstructions. The exact facial geometry, as well as the coordinated expression of muscles is unique to each individual. In additions, these deformations must be observed and measured under a large variety of illumination conditions as well as a variety of observation angles. Thus the visual recognition of emotions from facial expression remains a challenging open problem in computer vision.

Despite the difficulty of this challenge, important progress has been made in the area of automatic recognition of emotions from face expressions. The systematic cataloging of facial muscle groups as facial action units by Ekman [41] has let a number of research groups to develop libraries of techniques for recognizing the elements of the FACS coding system [33]. Unfortunately, experiments with that system have revealed that the system is very sensitive to both illumination and viewing conditions, as well as the difficulty in interpreting the resulting activation levels as emotions. In particular, this approach requires a high-resolution image with a high signal-to-noise ratio obtained under strong ambient illumination. Such restrictions are not compatible with the mobile imaging system used on tablet computers and mobile phones that are the target of this effort.

As an alternative to detecting activation of facial action units by tracking individual face muscles, we propose to measure physiological parameters that underlie emotions with a global approach. Most human emotions can be expressed as trajectories in a three dimensional space whose features are the physiological parameters of Pleasure-Displeasure, Arousal-Passivity and Dominance-Submission. These three physiological parameters can be measured in a variety of manners including on-body accelerometers, prosody, heart-rate, head movement and global face expression.

The PRIMA Group at Inria has developed robust fast algorithms for detection and recognition of human faces suitable for use in embedded visual systems for mobile devices and telephones. The objective of the work described in this report is to employ these techniques to construct a software system for measuring the physiological parameters commonly associated with emotions that can be embedded in mobile computing devices such as cell phones and tablets.

A revised software package has recently been released to our ICTlab partners for face detection, face tracking, gender and age estimation, and orientation estimation, as part of ICTlabs Smart Spaces action line. This software has been declared with the APP "Agence pour la Protection des Programmes" under the Interdeposit Digital number IDDN.FR.001.370003.000.S.P.2007.000.21000.

A software library, named PrimaCV has been designed, debugged and tested, and released to ICTLabs partners for real time image acquisition, robust invariant multi-scale image description, highly optimized face detection, and face tracking. This software has been substantially modified so as to run on an mobile computing device using the Tegra 3 GPU.

REVES Project-Team

4. Software and Platforms

4.1. Multi-View Image-Based Rendering and Relighting Suite

Participants: Clement Riant, Sylvain Duchêne, Pierre-Yves Laffont, Adrien Bousseau, George Drettakis.

We have designed and implemented a set of libraries for handling multi-view image-based rendering and relighting algorithms. These constitute the basis for the software developed for the EU projects VERVE and CR-PLAY.

4.1.1. RID: Rich Intrinsic-image Decomposer

We developed a software platform to perform rich intrinsic decomposition methods from photographs of outdoor scenes, as described in [13]. It includes main scripts and functions in Matlab for treatment of the input data, interfaces to software for multi-view reconstruction (Bundler, PMVS) and meshing from point clouds (method developed by Julie Digne, a postdoc in the GEOMETRICA project team). We then interface software for image matting using the Matting Laplacian, and User-Assisted Intrinsic Images. The system also includes an interface with Adobe Photoshop, for visualization and demonstration of our results in end-user image editing software. The method performs the computation of sun, sky and indirect lighting received at 3D points of an automatically reconstructed scene, using a modified version of the PBRT stochastic raytracer. Finally, there is a scene calibration module and an OpenGL viewer.

4.1.2. ROSSE: Relighting Outdoor Scenes with Shadow Editing

This software package includes a set of modules for processing point clouds and meshes produced by automatic multi-view stereo computer vision solutions. It includes all file management, point cloud and mesh handling, as well as ray-tracing using the Intel Embree ray tracer to computer illumination properties on the mesh. An interactive viewer is also included. A new intrinsic image approach is included as well as a module for relighting and shadow movement, based on an image-driven approach to moving cast shadows.

4.1.3. SWARPI: Superpixel Warp for Image-based rendering

Depth Synthesis and Warped-Based Superpixel Image-Based Rendering. This software package is the implementation of the publication [12]. The main software consists of two components: the depth synthesis step and the image-based rendering step:

a) The depth synthesis step is a Matlab package that reads 3D points coming from an automated 3D reconstruction pipeline, together with images and calibrated cameras, and produces the superpixel decomposition and the depth synthesis algorithm. The current version uses the open source packages bundler and PMVS (GPL v3 license), but other 3D reconstruction approaches could be used instead. b) The rendering step is in C++, and takes the result of the first step as input to allow interactive navigation. The code uses multi-pass deferred shading with geometry shaders (OpenGL 4.0 or above) to perform the rendering.

In addition to the implementation of [12], we have developed a Matlab interface for manual depth correction ("depth painting") An APP (Agency for the Protection of Programs) registration of this software is pending.

4.2. APF: state-of-the-art 3D audio library

Participants: Adrien David, George Drettakis.

This work was performed in collaboration with Jean-Christophe Lombardo of the DREAM group (i.e., the research support development group of our Inria center). REVES has several audio research publications over the last 10 years, which correspond to a class of functionalities such as clustering, masking, progressive processing etc.. The first component is the masking or culling algorithm, which aims at removing all the inaudible audio sources from a virtual scene based on perceptual metrics. The second component, called clustering, aims at grouping audio sources that are spatially close to each other and premix them to a representative cluster source, so that all spatialization related processing can be applied only on the representative premixed source [9]. Other audio topics were also considered and developed, like progressive and scalable frequency domain mixing, sound propagation, scalable reverberation, modal sound synthesis and contact sounds generation [1].

In order to maintain all the knowledge in the group and re-use these technologies in the Immersive Space, a previous young engineer (David Grelaud) wrote a fully documented audio library (APF) which gathers about 10 audio publications and 1 US patent. APF is a cross-platform, object oriented C++ API available on GForge. All the code has been re-implemented and a completely new software architecture resulted in a twofold increase in the speed of our algorithms. APF runs in the Immersive Space and uses the tracking system to spatialize virtual audio sources around the listener. It can also exploit personal Head Related Transfer Functions (HRTF).

We have implemented a network communications layer to create an audio rendering server on a separate machine, and the library is fully integrated into the osgVR platform.

APF has been critical in establishing collaborations in the context of various grant proposals (EU and national).

4.3. GaborNoise Software

Participants: Ares Lagae, George Drettakis.

We proposed a new procedural noise function last year, Gabor noise [6]. In the context of this project, we have developed a software package, which includes a CPU reference implementation of the 2D noise, and a complete GPU implementation of the 2D noise, surface noise, and 3D noise. This software package has been filed for APP protection and is in the process of being transferred to industrial partners.

This work is a collaboration with Sylvain Lefebvre, former member of the team, now in the ALICE project-team, Inria Nancy - Grand Est.

4.4. Gabor Noise By Example

Participant: George Drettakis.

In collaboration with B. Galerne, S. Lefebvre and A. Lagae (KU Leuven) we have released to code for the 2012 SIGGRAPH paper Gabor Noise By Example (see http://www-sop.inria.fr/reves/Basilic/2012/GLLD12/). This includes a matlab code for the analysis and C++/cuda code for the synthesis.

SÉMAGRAMME Project-Team

5. Software and Platforms

5.1. Leopar

Participants: Bruno Guillaume [correspondent], Guy Perrier, Tatiana Ekeinhor.

5.1.1. Software description

Leopar is a parser for natural languages which is based on the formalism of Interaction Grammars [40]. 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 $f \longleftarrow v$ by a dual positive feature-value pair $f \longrightarrow v$. 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 developped 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-2 or of sets of description trees associated to some word in the linguistic resources.

One of the difficulties with symbolic parsing is that several solution can be produced for a single sentence and we want te be able to rank them. Tatiana Ekeinhor, during her second year Master Intership (from February to June 2013), implemented a ranker based on statistical techniques. Using the Sequoia TreeBank as a training corpus, she obtained an improvement of the system compared to the handcrafted rules.

5.2. ACG Development Toolkit

Participants: Sylvain Pogodalla [correspondent], Philippe de Groote.

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 ACG Development toolkit development effort is part of the POLYMNIE project (see Section 7.2.1.1). It will support the experimentation and evaluation parts of the project.

The current version of the ACG development toolkit prototype [1] 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.

The prototype now enables the transformation from the object terms to the abstract terms. The parsing algorithm follows [43]'s method which 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.

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 has been adapted to produce not only yes/no answer to queries, but also all the proofs of the answers to the queries. The next steps concern optimization and efficiency. 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 [correspondent], Guy Perrier.

Graph rewriting, Interface syntaxe-sémantique

Grew is a Graph Rewriting tools dedicated to applications in NLP. It is freely-available (from the page http://grew.loria.fr) 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.
- Rules can be parametrized by lexical information.
- Filters can be used at the output of each module to control the structure produced are well-formed.

---

[1] Available 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 or for demonstrations).

The Grew software was used for several kind of applications manipulating syntactic and/or semantic graph representations. It was used to build DMRS semantic representation from syntactic dependency trees in the French TreeBank [51].

More recently, it was used in the project "Deep Syntax Annotation of the Sequoia French Treebank". First, it was used as a pre-annotation tool and; second, it is used to detect ill-formed structures that don't fit the annotation guide requirement.

5.4. Other developments

Participants: Bruno Guillaume [correspondent], Maxime Amblard [correspondent].

Concordancer, Dependencies, Graphical tools Other peripheral developments of the team are available either as web service of as downloadable code:

- A concordancer named CONDOR which is usable online: http://condor.loria.fr. With Condor, it is possible to search for all inflexions (given by a lexicon) of some lemma; it is possible to search for a couple of lemmas to find collocations.
- A program (named DEP2PICT) to build graphical representations (PNG, SVG or PDF) of dependency structures. It is presented in http://dep2pict.loria.fr; it is usable online http://dep2pict.loria.fr/demo.
- a management chain of the transcriptions of interviews for the SLAMproject. including the production of a full anonymized randomized version of the resources.
- A program which use Distagger and propose different analyze of the repartition of disfluencies.

SIROCCO Project-Team

5. Software and Platforms

5.1. Visual Fixation Analysis

Participant: Olivier Le Meur [contact person].

From a set of fixation data and a picture, the software called Visual Fixation Analysis extracts from the input data a number of features (fixation duration, saccade length, orientation of saccade...) and computes an human saliency map. The software can also be used to assess the degree of similarity between a ground truth (eye fixation data) and a predicted saliency map. This software is dedicated to people working in cognitive science and computer vision. This software has been registered at the APP (Agence de Protection des Programmes).

5.2. Hierarchical super-resolution based inpainting

Participant: Olivier Le Meur [contact person].

From an input binary mask and a source picture, the software performs an examplar-based inpainting. The method is based on the combination of multiple inpainting applied on a low resolution of the input picture. Once the combination has been done, a single-image super-resolution method is applied to recover the details and the high frequency in the inpainted areas. This software is dedicated to people working in image processing and post production. This software is being registered at the APP (Agence de Protection des Programmes).

5.3. Salient object extraction

Participants: Zhi Liu, Olivier Le Meur [contact person].

This software detects salient object in an input picture in an automatic manner. The detection is based on super-pixel segmentation and contrast of histogram. This software is dedicated to people working in image processing and post production. This software is being registered at the APP (Agence de Protection des Programmes).

5.4. loss concealment algorithm using examplar-based video inpainting

Participants: Ronan Le Boulch, Mounira Ebdelli, Christine Guillemot, Olivier Le Meur [contact person].

This software recovers regions of a video sequence which can be lost after transmission over a network with no guarantee of quality of service. Motion information of impaired areas is first interpolated from the motion vectors of known areas. An examplar-based video inpainting method is then used to fill in the corrupted areas. This software is being registered at the APP (Agence de Protection des Programmes).

5.5. Standardization

Participants: Christine Guillemot, Laurent Guillo [contact person].

In the continuity of the ADT Picovin-P, we have in 2013, pursued our activities of standardization in the area of multi-view plus depth video coding. We in particular followed the standardization activities within the Joint Collaborative Team on 3D Video Coding Extension (JCT-3V). JCT-3V aims at developing 3D extensions for video codecs, which are AVC (ATM) or HEVC (HTM) based. We have pursued the developments of our proposal related to inter-view motion vector prediction, leading to a joint proposal with Qualcomm and Mediatek which has been adopted in the standard in July 2013.

SMIS Project-Team

5. Software and Platforms

5.1. Introduction

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

For a decade, we have developed successive prototypes addressing different application domains, introducing different technical challenges and relying on different hardware platforms. PicoDBMS was our first attempt to design a full-fledged DBMS embedded in a smart card [39] [27]. Chip-Secured Data Access (C-SDA) embedded a reduced SQL query engine and access right controller in a secure chip and acted as an incorruptible mediator between a client and an untrusted server hosting encrypted data [34]. Chip-Secured XML Access (C-SXA) was an XML-based access rights controller embedded in a smart card [35]. Prototypes of C-SXA have been the recipient of the e-gate open 2004 Silver Award and SIMagine 2005 Gold award, two renowned international software contests. The next subsections detail the two prototypes we are focusing on today.

5.2. PlugDB engine

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

More than a stand-alone prototype, PlugDB is part of a complete architecture dedicated to a secure and ubiquitous management of personal data. PlugDB aims at providing an alternative to a systematic centralization of personal data. To meet this objective, the PlugDB architecture lies on a new kind of hardware device called Secure Portable Token (SPT). Roughly speaking, a SPT combines a secure microcontroller (similar to a smart card chip) with a large external Flash memory (Gigabyte sized). The SPT can host data on Flash (e.g., a personal folder) and safely run code embedded in the secure microcontroller. PlugDB engine is the cornerstone of this embedded code. PlugDB engine manages the database on Flash (tackling the peculiarities of NAND Flash storage), enforces the access control policy defined on this database, protects the data at rest against piracy and tampering, executes queries (tackling low RAM constraint) and ensures transaction atomicity. Part of the on-board data can be replicated on a server (then synchronized) and shared among a restricted circle of trusted parties through crypto-protected interactions. PlugDB engine has been registered at APP (Agence de Protection des Programmes) in 2009 [28] and a new version is registered each year. The underlying Flash-based indexing system has also been patented by Inria and Gemalto [40]. It has been demonstrated in a dozen of national and international events including JavaOne and SIGMOD. It is being experimented in the field to implement a secure and portable medical-social folder helping the coordination of medical care and social services provided at home to dependent people. The next step in our agenda is to put this software in open-source so that students and communities of developers can complement it and develop innovative privacy-by-design applications. In 2012, we have ported PlugDB-engine on a new hardware platform to 1) become completely independent from Gemalto, 2) have a plug-and-play implementation on Android, 3) serve as a basement to port it on other custom hardware implementations. We have already discussed with hardware companies located in "Ile-de-France" to produce new hardware tokens to host future versions of PlugDB-engine. Link: http://www-smis.inria.fr/_DMSP/home.php.

5.3. uFLIP Benchmark

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

It is amazingly easy to produce meaningless results when measuring flash devices, partly because of the peculiarity of flash memory, but primarily because their behavior is determined by layers of complex, proprietary, and undocumented software and hardware. uFLIP is a component benchmark for measuring the response time distribution of flash IO patterns, defined as the distribution of IOs in space and time. uFLIP includes a benchmarking methodology which takes into account the particular characteristics of flash devices. The source code of uFLIP, available on the web (700 downloads, 4000 distinct visitors), was registered at APP in 2009 [32]. It has been demonstrated at SIGMOD.

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

STARS Project-Team

5. Software and Platforms

5.1. SUP

`../../../../projets/stars/IMG/SUP-architecture.jpg`

Figure 5. Tasks of the Scene Understanding Platform (SUP).

SUP is a Scene Understanding Software Platform written in C and C++ (see Figure 5). SUP is the continuation of the VSIP platform. SUP is splitting the workflow of a video processing into several modules, such as acquisition, segmentation, etc., up to activity recognition, to achieve the tasks (detection, classification, etc.) the platform supplies. Each module has a specific interface, and different plugins implementing these interfaces can be used for each step of the video processing. This generic architecture is designed to facilitate:

1. integration of new algorithms in SUP;
2. sharing of the algorithms among the Stars team.

Currently, 15 plugins are available, covering the whole processing chain. Several plugins are using the Genius platform, an industrial platform based on VSIP and exploited by Keeneo.

Goals of SUP are twofold:

1. From a video understanding point of view, to allow the Stars researchers sharing the implementation of their work through this platform.
2. From a software engineering point of view, to integrate the results of the dynamic management of vision applications when applied to video analytics.

5.2. ViSEval

ViSEval is a software dedicated to the evaluation and visualization of video processing algorithm outputs. The evaluation of video processing algorithm results is an important step in video analysis research. In video processing, we identify 4 different tasks to evaluate: detection, classification and tracking of physical objects of interest and event recognition.

The proposed evaluation tool (ViSEvAl, visualization and evaluation) respects three important properties:

- To be able to visualize the algorithm results.
- To be able to visualize the metrics and evaluation results.
- For users to easily modify or add new metrics.

The ViSEvAl tool is composed of two parts: a GUI to visualize results of the video processing algorithms and metrics results, and an evaluation program to evaluate automatically algorithm outputs on large amount of data. An XML format is defined for the different input files (detected objects from one or several cameras, ground-truth and events). XSD files and associated classes are used to check, read and write automatically the different XML files. The design of the software is based on a system of interfaces-plugins. This architecture allows the user to develop specific treatments according to her/his application (e.g. metrics). There are 6 interfaces:

1. The video interface defines the way to load the images in the interface. For instance the user can develop her/his plugin based on her/his own video format. The tool is delivered with a plugin to load JPEG image, and ASF video.
2. The object filter selects which objects (e.g. objects far from the camera) are processed for the evaluation. The tool is delivered with 3 filters.
3. The distance interface defines how the detected objects match the ground-truth objects based on their bounding box. The tool is delivered with 3 plugins comparing 2D bounding boxes and 3 plugins comparing 3D bounding boxes.
4. The frame metric interface implements metrics (e.g. detection metric, classification metric, ...) which can be computed on each frame of the video. The tool is delivered with 5 frame metrics.
5. The temporal metric interface implements metrics (e.g. tracking metric,...) which are computed on the whole video sequence. The tool is delivered with 3 temporal metrics.
6. The event metric interface implements metrics to evaluate the recognized events. The tool provides 4 metrics.

The GUI is composed of 3 different parts:

1. The widows dedicated to result visualization (see Figure 6):
   - Window 1: the video window displays the current image and information about the detected and ground-truth objects (bounding-boxes, identifier, type,...).
   - Window 2: the 3D virtual scene displays a 3D view of the scene (3D avatars for the detected and ground-truth objects, context, ...).
   - Window 3: the temporal information about the detected and ground truth objects, and about the recognized and ground-truth events.
   - Window 4: the description part gives detailed information about the objects and the events,
   - Window 5: the metric part shows the evaluation results of the frame metrics.
2. The object window enables the user to choose the object to be displayed (see Figure 7).
3. The multi-view window displays the different points of view of the scene (see Figure 8).

`../../../../projets/stars/IMG/ViSEvAlLegend.jpg`

Figure 6. GUI of the ViSEvAl software

`../../../../projets/stars/IMG/interfaceparam.jpg`

Figure 7. The object window enables users to choose the object to display

`../../../../projets/stars/IMG/multicamera.jpg`

Figure 8. The multi-view window

The evaluation program saves, in a text file, the evaluation results of all the metrics for each frame (whenever it is appropriate), globally for all video sequences or for each object of the ground truth.

The ViSEvAl software was tested and validated into the context of the Cofriend project through its partners (Akka,...). The tool is also used by IMRA, Nice hospital, Institute for Infocomm Research (Singapore),... The software version 1.0 was delivered to APP (French Program Protection Agency) on August 2010. ViSEvAl is under GNU Affero General Public License AGPL (http://www.gnu.org/licenses/) since July 2011. The tool is available on the web page : http://www-sop.inria.fr/teams/pulsar/EvaluationTool/ViSEvAl_Description.html

5.3. Clem

The *Clem Toolkit* [68](see Figure 9) is a set of tools devoted to design, simulate, verify and generate code for LE [19] [82] programs. LE is a synchronous language supporting a modular compilation. It also supports automata possibly designed with a dedicated graphical editor.

Each LE program is compiled later into lec and lea files. Then when we want to generate code for different backends, depending on their nature, we can either expand the lec code of programs in order to resolve all abstracted variables and get a single lec file, or we can keep the set of lec files where all the variables of the main program are defined. Then, the *finalization* will simplify the final equations and code is generated for simulation, safety proofs, hardware description or software code. Hardware description (Vhdl) and software code (C) are supplied for LE programs as well as simulation. Moreover, we also generate files to feed the NuSMV model checker [65] in order to perform validation of program behaviors.

`../../../../projets/stars/IMG/clem-toolkit.jpg`

Figure 9. The Clem Toolkit

TEXMEX Project-Team

5. Software and Platforms

5.1. Software

When applicable, we provide the IDDN is the official number, which is obtained when registering the software at the APP (Agence de Protection des Programmes).

5.1.1. New Software

5.1.1.1. DeCP-Index

Participants: Laurent Amsaleg [Correspondent], Gylfi Gudmundsson, Diana Moise, Denis Shestakov.

DeCP-Index is a Map-Reduce oriented implementation of the vectorial quantization scheme developped during the PhD of Gylfi Gudmundsson. It is in Java.

First APP deposit: IDDN.FR.001.500011.000.S.P.2013.000.40000

5.1.1.2. DeCP-Scripts

Participants: Laurent Amsaleg [Correspondent], Gylfi Gudmundsson, Diana Moise, Denis Shestakov.

DeCP-Scripts is a series of script for installing, configuring and deploying the Map Reduce framework over the grid infrastructure.

First APP deposit: IDDN.FR.001.500012.000.S.P.2013.000.40000

5.1.1.3. *SVM

Participants: François Poulet [correspondent], Thanh Nghi Doan.

Soon available from the home page of the Inria Parallel Benchmark Suite http://www.irisa.fr/alf/index.php?option=com_content&view=article&id=82&Itemid=&lang=fr.

*SVM include a set of parallel and incremental SVM classifiers for large scale classification tasks on GPU, CPU or cluster / Grid.

5.1.2. Main software started before 2012

5.1.2.1. Peyote

Participants: Sébastien Campion, Jonathan Delhumeau [correspondent], Hervé Jégou.

Peyote is a framework for Video and Image description, indexation and nearest neighbor search. It can be used as-is by a video-search or image-search front-end with the implemented descriptors and search modules. It can also be used via scripting for large-scale experimentation. Finally, thanks to its modularity, it can be used for scientific experimentation on new descriptors or indexation methods. Peyote is used in the AABOT software.

First APP deposit: IDDN.FR.001.4200008.000.S.P.2012.000.20900.

5.1.2.2. Aabot

Participant: Jonathan Delhumeau.

AABOT is a tool to facilitate annotation of large video databases. It's primary design focus has been for the annotation on commercials in two 6-month long TV databases. The software keeps a database of already annotated commercials and suggests when it finds a new probable instance. It also validates user annotations by suggesting similar existing commercials if it finds any which are similar by name or content. The user can then confirm the creation of new commercials or accept the correction if he was mistaken.

AABOT is accessed via a web-browser. It is mostly used by uploading and downloading an annotation file. An interactive HTML5 interface is also available when some user feedback is needed (during validation). It uses Peyote as an description / indexing engine.

First APP deposit: IDDN.FR.001.4200010.000.S.P.2012.000.20900.

5.1.2.3. Pqcodes

Participant: Hervé Jégou [correspondent].

Jointly maintained with Matthijs Douze, Inria/LEAR.

Pqcodes is a library which implements the approximate k nearest neighbor search method of [88] based on product quantization. This software has been transferred to two companies (in August 2011 and May 2012, respectively).

The current version registered at the APP is IDDN.FR.001.220012.001.S.P.2010.000.10000.

5.1.2.4. Yael

Participant: Hervé Jégou [correspondent].

Jointly maintained with Matthijs Douze, from Inria/LEAR.

Yael is a C/python/Matlab library providing (multi-threaded, Blas/Lapack, low level optimization) implementations of computationally demanding functions. In particular, it provides very optimized functions for k-means clustering and exact nearest neighbor search. The library has been downloaded about 2,000 times in 2013.

The current version registered at APP is IDDN.FR.001.220014.001.S.P.2010.000.10000.

5.1.2.5. BonzaiBoost

Participant: Christian Raymond [correspondent].

Available at http://bonzaiboost.gforge.inria.fr/.

BonzaiBoost stands for boosting over small decisions trees. BonzaiBoost is a general purpose machine-learning program based on decision tree and boosting for building a classifier from text and/or attribute-value data. Currently one configuration of BonzaiBoost is ranked first on http://mlcomp.org a website which propose to compare several classification algorithms on many different datasets

5.1.2.6. Irisa_Ne

Participant: Christian Raymond [correspondent].

IRISA_NE is a couple of Named Entity tagger, one of them is based on CRF and the other HMM. It is dedicated to automatic transcriptions of speech. It does not take into account uppercase or punctuation and has no concept of sentences. However, they also manage texts with punctuation and capitalization.

5.1.2.7. IRISA News Topic Segmenter (irints)

Participants: Guillaume Gravier [correspondent], Pascale Sébillot, Anca-Roxana Simon.

This software is dedicated to unsupervised topic segmentation of texts and transcripts. The software implements several of our research methods and is particularly adapted for automatic transcripts. It provides topic segmentation capabilities virtually for any word-based language, with presets for French, English and German. The software has been licensed to several of our industrial partners.

5.1.3. Other softwares

- BAG OF COLORS: describe images based on color
- I-DESCRIPTION: IDDN.FR.001.270047.000.S.P.2003.000.21000
- ASARES: symbolic machine learning system to infer corpus-specific morpho-syntactic and semantic patterns from descriptions of pairs of linguistic elements found in a corpus in which the components are linked by a given semantic relation IDDN.FR.001.0032.000.S.C.2005.000.20900
- ANAMORPHO: detects morphological relations between words in many languages IDDN.FR.001.050022.000.S.P.2008.000.20900
- DIVATEX: audio/video frame server IDDN.FR.001.320006.000.S.P.2006.000.40000

- NAVITEX: video annotation tool IDDN.FR.001.190034.000.S.P.2007.000.40000
- TELEMEX: web service that enables TV and radio stream recording
- VIDSIG: small and robust video signature (64 bits per image)
- VIDSEG: multimodal video segmentation IDDN.FR.001.250009.000.S.P.2009.000.40000
- ISEC: web application used as graphical interface for content-based image search engines
- GPU-KMEANS: k-means algorithm on GPU
- CORRESPONDENCE ANALYSIS: factorial correspondence analysis (FCA) for image retrieval.
- GPU CORRESPONDENCE ANALYSIS: GPU implementation of CORRESPONDENCE ANALYSIS
- CAVIZ: interactive graphical tool to display and extract knowledge from the results of a CORRESPONDENCE ANALYSIS on images
- KIWI: keyword extraction from texts and ASR transcripts
- TOPIC SEGMENTER: topic segmentation of texts and ASR transcripts.
- S2E: automatic discovery of audiovisual structuring events in videos.
- 2PAC: builds classes of words of similar meanings (“semantic classes”) IDDN.FR.001.470028.000.S.P.2006.000.40000
- FAESTOS: Fully Automatic Extraction of Sets of keywords for TOpic characterization and Spotting IDDN.FR.001.470029.000.S.P.2006.000.40000
- FISHNET: automatic Web pages grabber associated with a specific theme
- MATCH MAKER: semantic relation extraction by statistical methods.
- IRISAPHON: grapheme to phoneme conversion
- PYTHON-GEOHASH: implementation of the geometric hashing algorithm [99]
- AVSST: automatic video stream structuring tool (detection of repetitions, classification program/inter-program, EPG alignment) with GUI
- TVSEARCH: content-based retrieval search engine to search and propagate manual annotation such as advertisement in a TV corpora.
- SAMUSA: multimedia content speech/music segmentation
- KERTRACK: visual graphical interface for tracking visual targets based on particle filter tracking or mean-shift.
- MOZAIC2D: spatio-temporal mosaic based on dominant motion compensation.
- BABAZ: audio database management system with an audio-based search function IDDN.FR.001.010006.000.S.P.2012.000.10000
- PIMPY: Python module and binders for multimedia content indexing

5.2. Demonstration: Texmix

Participants: Sébastien Campion [correspondent], Guillaume Gravier.

Structuring a collection of news shows requires some level of semantic understanding of the content in order to segment shows into their successive stories and to create links between stories in the collection, or between stories and related resources on the Web. Spoken material embedded in videos, accessible by means of automatic speech recognition, is a key feature to semantic description of video contents. We have developed multimedia content analysis technology combining automatic speech recognition, natural language processing and information retrieval to automatically create a fully navigable news portal from a collection of video files.

In 2013, we extended the Texmix demonstration to include transcript-free summarization using word discovery.

See the demo at http://texmix.irisa.fr.

5.3. Experimental platform

Participants: Laurent Amsaleg, Sébastien Campion [correspondent], Patrick Gros, Pascale Sébillot.

Until 2005, we used various computers to store data and to carry out experiments. In 2005, we began work to specify and set-up dedicated equipment to experiment on very large collections of data. During 2006 and 2007, we specified, bought and installed our first complete platform. It is organized around a very large storage capacity (155TB), and contains 4 acquisition devices (for Digital Terrestrial TV), 3 video servers, and 15 computing servers partially included in the local cluster architecture (IGRIDA). A dedicated website has been developed in 2009 to provide a user support. It contains useful information such as references of available and ready to use software on the cluster, list of corpus stored on the platform, pages for monitoring disk space consumption and cluster loading, tutorials for best practices and cookbooks for treatments of large datasets. In 2010, we have acquired a new large memory server with 144GB of RAM which is used for memory demanding tasks. The previous server dedicated to this kind of jobs (acquired in 2008) has been upgraded to 96GB of RAM. In 2012, we extended our storage capacity to 215TB and expanded our computing resources with two new large memory servers with 256GB of RAM for each of them. Both have their own HPC storage of 12TB. This year our backbone network was fully upgraded in order to connect each element of the platform with a 10GB/s bandwith.

A new distributed file system architecture was design and will be implement in 2014.

The platform is funded by a joint effort of Inria, INSA Rennes and University of Rennes 1.

5.4. Web services

Participant: Sébastien Campion [correspondent].

This year after a first prototyping of web service where each one of our algorithm was deployed on it's own server, we decided to develop a second version more centralized and named All|Go. All|Go was designed, developed and deployed in order to save resources unnecessarily locked and painful maintenance tasks.

Available at http://allgo.irisa.fr, All|Go currently host five TexMex web services (Samusa, Otis, Termex, Nero, VidSeg).

All|Go infrastructure is based on the Ruby On Rails (ROR) framework for the web "frontoffice" part. ROR enable to create and run task with an HTML or XML, JSON API. SideKiq schedule each job on several nodes. Finally, thanks to the new linux container technology named Docker, applications are configured and deployed on agnostic nodes, inside their container. Container must be seen as very light virtual machine. All our application are stored in a private registry. Data are shared with the NFS protocol. A automation software named Puppet manage infrastructure throughout its lifecycle, from provisioning and configuration to orchestration and reporting.

TITANE Team

5. Software and Platforms

5.1. CGAL, the Computational Geometry Algorithms Library

Participants: Pierre Alliez, Clement Jamin, Florent Lafarge, Sven Oesau, David Bommes.

CGAL is a C++ library of geometric algorithms and data structures. Our team is involved in several ongoing implementations: parallelization of mesh generation and triangulations, shape detection in unstructured point sets, geodesic distances on surface meshes and barycentric coordinates (in collaboration with Dmitry Anisimov). Pierre Alliez is a member of the CGAL Editorial Board.

5.2. APP deposits

5.2.1. MeshMantics

Participants: Yannick Verdie, Florent Lafarge, Pierre Alliez.

MeshMantics is a software for segmenting 2-manifold surface meshes in an urban context. Four classes of interest are considered: ground, vegetation, roof and facades.

TYREX Team

5. Software and Platforms

5.1. XML Reasoning Solver

Participants: Pierre Genevès, Nabil Layaïda, Nils Gesbert, Manh-Toan Nguyen.

The XML Reasoning Solver is a tool for the static analysis of XPath queries and XML schemas based on the latest theoretical advances [9]. It allows automated verification of properties that are expressed as logical formulas over trees. A logical formula may for instance express structural constraints or navigation properties (like e.g. path existence and node selection) in finite trees.

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

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

5.1.1. Extensions for CSS

We have introduced the first system capable of statically verifying properties of a given cascading style sheet (CSS) over the whole set of documents to which this stylesheet applies [8]. The system is composed of a set of parsers for reading the CSS and schema files (XML Schema, Relax NG, or DTD) together with a text file corresponding to the problem description as a logical formula. We have developed a compiler that translates CSS files into their logical representations. Then, the solver takes the overall problem formulation and checks it for satisfiability.

5.1.2. XQuery IDE

We have started the development of an XQuery IDE with a web interface. This prototype integrates static analyses performed by the solver inside a development environment suited for XQuery programmers.

5.2. ClaireCourseMaker Library

Participants: Nicolas Hairon, Cécile Roisin.

The goal of the ClaireCourseMakeris to provide straightforward editing tools for structuring, annotating and timeline-based authoring of continuous content such as audio or video. Even if it can be used for any content, it is mainly devoted to synchronize pedagogical material (video, slides, chaptering, etc.) in order to provide rich media online courses à la MOOC. The underlying technology is standard-based and uses the open source JavaScript Popcorn library and Popcorn Maker web application by Mozilla.

The result is a wysiwyg web-based authoring tool which benefits from all the generic features of Popcorn and the specific services that cope with chaptering and synchronization needs.

ClaireCourseMaker is the direct follow-up tool of the Timesheet library which is a cross-browser JavaScript implementation for scheduling the dynamic behavior of HTML5 content that can be described with declarative SMIL markup (SMIL Timing and Synchronization, SMIL Timesheets).

ClaireCourseMaker is developed in collaboration with the OpenClassrooms company in the context of the Claire project (see section 7.1.1).

5.3. Mobile Audio Language

Participants: Yohan Lasorsa, Jacques Lemordant.

5.3.1. MAUDL library

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

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

MAUDL prevents audio information overwhelming through categorization at the declarative level and the use of priority queues at the execution level. This takes account of speed when walking, and of rapid hand gestures when interrogating the environment, for example. MAUDL can be used as an authoring time interchange file format for interactive mobile applications or as a runtime file format that is actually loaded through the web and played directly in the device. MAUDL is a cue-oriented interactive audio system, where audio services are requested using named events and the system's response to each event is determined by the audio artist.

The library has been implemented in C++ and now supports different mobile operating systems such as Android and iOS. MAUDL has been widely used in the first demonstrator of the VENTURI project consisting of a mobile augmented reality game.

5.3.2. 3D Audio Pointer

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

As the 3D audio pointer is based on MAUDL this technology is also available for both platforms, iOS and Android. It has been used by Metaio for the micro-navigation which is part of the second VENTURI demonstrator. The use case was to help a visually impaired person to find a box on a shelf with computer vision and 3D audio rendering.

5.4. PDRTrack

Participants: Jacques Lemordant, Mathieu Razafimahazo.

PDRTrack is a localization utility running on iOS or Android smartphones used for recording and playing data sets (accelerometer, gyroscope, barometer and magnetometer values) to find out the effect of different pedometer and map matching parameters and their result on localization accuracy. This application uses the PDR library, written in C++ and developed by the team, which provides the user's location in real time based on the interpretation of sensor readings. Three main modules have been designed to build this localization system:

- a pedometer that estimates the distance the user has walked and his speed
- a motion manager that enables data set recording and simulation but also the creation of virtual sensors or filters (e.g gyroscope drift compensation, linear acceleration, altimeter)
- a map-matching algorithm that provides a new location based on a given OpenStreetMap file description and the current user's trajectory

The PDR library has been shared to the VENTURI consortium for the first part of the second year demonstrator: guiding a visually impaired person from Fondazione Bruno Kessler's bus stop to the building entrance. Others partners have used this localization system for retrieving a scale factor needed for the computer vision part (i.e SLAM).

5.5. Interactive eXtensible Engine (IXE)

Participants: Yohan Lasorsa, Jacques Lemordant, David Liodenot, Thibaud Michel, Mathieu Razafimahazo.

GPS navigation systems, when used in an urban environment, are limited in precision and can only give instructions at the level of the street and not of the pavement. GPS is also limited to outdoor navigation and requires some delicate transitioning system when switching to another positioning system to perform indoor navigation.

IXE is an open source urban pedestrian navigation system based on Inertial Measurement Units (IMU) and running on mobile phones with onboard geographic data and a routing engine. With IXE, the distinction between indoor and outdoor is blurred as an IMU-based location engine can run indoor and outdoor. IXE allows augmented reality queries on customized embedded geographical data. Queries on route nodes or POIs, on ways and relations are predefined for efficiency and quality of information.

Following the web paradigm, IXE is a browser for XML documents describing navigation networks: by using the micro-format concept, one can define inside OpenStreetMap a complex format for pedestrian navigation networks allowing navigation at the level of pavements or corridors. The big advantage of doing this instead of defining new XML languages is that we can use the standard OpenStreetMap editor JOSM to create navigation networks in a short amount of time.

The purpose of the IXE browser is to read these OSM documents and to generate from them visible or audible navigation information. IXE works on any mobile phone running under iOS or Android. Its heart is composed of three engines, one for dead-reckoning navigation, one for interactive audio and the last one for Augmented Reality visual information, allowing quick reconfiguration for extremely varied applications.

IXE can be used for accessible navigation allowing independent living for people with disabilities.

IXE Android is an enhanced version of our iOS navigation demonstrator. It uses our latest work on the localization positioning system such as PDR, GPS, user and NFC. This application is based on predetermined walks described in a XML format extending OpenStreetMap for navigation purpose, everybody can create and share their walks. In order to create a new walk, the author has to follow specifications described in part 6.3.1. We added some functionalities to the open source application Java OpenStreetMap Editor to enhance walk authoring for the IXE app.

- IXE-iOS
- IXE-Android

WILLOW Project-Team

5. Software and Platforms

5.1. SPArse Modeling Software (SPAMS)

SPAMS v2.4 was released as open-source software in May 2013 (v1.0 was released in September 2009, v2.0 in November 2010). It is an optimization toolbox implementing algorithms to address various machine learning and signal processing problems involving

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

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

5.2. Local dense and sparse space-time features

This is a package with Linux binaries implementing extraction of local space-time features in video. We are preparing a new release of the code implementing highly-efficient video descriptors described in Section 6.4.3 . Previous version of the package was released in January 2011. The code supports feature extraction at Harris3D points, on a dense space-time grid as well as at user-supplied space-time locations. The package is publicly available at http://www.di.ens.fr/~laptev/download/stip-2.0-linux.zip.

5.3. Automatic Mining of Visual Architectural Elements

The code on automatic mining of visual architectural elements (v4.5) described in (Doersch *et al.* SIGGRAPH 2012) has been publicly released online in January 2013 (earlier version v4.3 was released in December 2012 and v3.0 was released in September 2012) at http://graphics.cs.cmu.edu/projects/whatMakesParis/paris_sigg_release_v4.5.tar.gz.

5.4. Joint learning of actors and actions in video

This is a package of Matlab code implementing the multi-view face processing pipeline and joint learning of actors and actions in movies described in (Bojanowski *et al.* ICCV 2013 [2]. The package was last updated in December 2013 and is available at http://www.di.ens.fr/willow/research/actoraction/.

5.5. Visual Place Recognition with Repetitive Structures

Open-source release of the software package for visual localization in urban environments has been made publicly available. The software package implements newly developed method [9] for representing visual data containing repetitive structures (such as building facades or fences), which often occur in urban environments and present significant challenge for current image matching methods. The software is available at http://www.di.ens.fr/willow/research/repttile/download/repttile_demo_ver02.zip.

WIMMICS Project-Team

5. Software and Platforms

5.1. Corese

Participants: Olivier Corby [correspondant], Alban Gaignard, Fabien Gandon.

Corese (COnceptual REsource Search Engine) is a Semantic Web Factory. It enables users to load and process RDFS schemas, RDF data and query and update the graph base thus created by using the SPARQL 1.1 Query & Update Language (figure 1).

Furthermore, Corese query language integrates original features such as approximate search, extended Property Path, SQL or XPath. It provides a SPARQL based pretty printing language for RDF graphs and a SPARQL based Inference Rule Language for RDF. Corese also provides distributed federated query processing, thanks to a collaboration with Alban Gaignard and Johan Montagnat from CNRS I3S.

Corese is a Semantic Web Factory that enables us to design and develop Semantic Web applications; it is available for download. In the past, Corese received two software development grants (ADT) from Inria and in 2013 we have a new grant for two more years. Corese is registered at the APP and in 2007 we decided to distribute it as open source software under license CeCILL-C.

Corese is used and has been used in more than 60 applications, 24 PhD Thesis and is used for education by several institutions. It has been used in European projects such as Ontorule, Palette, SevenPro, SeaLife and in ANR projects such as Kolflow, Ginseng, Neurolog, VIP, ISICIL, e-WOK Hub. Corese is the Semantic Web engine of Discovery Hub and of the Semantic Web Import Plugin for Gephi visualization.

The work on Corese was published in [2], [1], [3], [4].

Web page: http://wimmics.inria.fr/corese

5.2. Semantic Web Import Plugin for Gephi visualization

Participants: Erwan Demairy, Fabien Gandon, Olivier Corby.

The SemanticWebImport [2] plugin is intended to allow the import of semantic data into Gephi open graph vizualisation platform (figure 2). Gephi is an interactive visualization and exploration platform for all kinds of networks and complex systems, dynamic and hierarchical graphs. The imported data are obtained by processing a SPARQL request on the semantic data. The data can be accessed following three manners:

1. by accessing local RDF & RDFS files and using the embedded Corese engine to apply the SPARQL request;
2. by accessing a remote REST SPARQL endpoint. In that case, the SPARQL request is applied remotely and the graph is built locally by analyzing the result sent by the endpoint;
3. by accessing a remote SOAP SPARQL endpoint. As for the REST endpoint, the resulting graph is built from the result returned by the endpoint.

The software is released under version 1.0. It has received a development grant (ADT) from Inria.

Web pages: http://wiki.gephi.org/index.php/SemanticWebImport
https://gforge.inria.fr/projects/segviz-public

5.3. Datalift Linked Open Data Platform

Participants: Luca Costabello, Fabien Gandon, Serena Villata.

The Datalift platform aims at easing and automating publication of raw structured datasets on the Web of data. The platform proposes an extensible architecture and comes with modules enabling: data selection, schema selection and mapping; format and vocabulary conversion; storage, publication querying and access control; interlinking with other sources; visualization. The latest version of the code is maintained on the public forge of Inria [3].

---

[2] http://wiki.gephi.org/index.php/SemanticWebImport

../../../../projets/wimmics/IMG/CoreseCapture.png

Figure 1. Corese

`../../../../projets/wimmics/IMG/GephiCapture.png`

Figure 2. Gephi

Web page: http://www.datalift.org

5.4. Question Answering Wikiframework-based System

Participant: Elena Cabrio.

The QAKiS system (figure 3) implements question answering over DBpedia. QAKiS allows end users to submit a query to an RDF triple store in English and obtain the answer in the same language, hiding the complexity of the non-intuitive formal query languages involved in the resolution process. At the same time, the expressiveness of these standards is exploited to scale to the huge amounts of available semantic data. Its major novelty is to implement a relation-based match for question interpretation, to convert the user question into a query language (e.g. SPARQL). English, French and German DBpedia chapters are the RDF data sets to be queried using a natural language interface.

Web page: http://www.qakis.org

5.5. French Chapter of DBpedia

Participants: Julien Cojan, Fabien Gandon.

DBpedia is an international crowd-sourced community effort to extract structured information from Wikipedia and make this information available on the semantic Web as linked open data. The DBpedia triple stores then allow anyone to solve sophisticated queries against Wikipedia extracted data, and to link the different data sets on these data. The French chapter of DBpedia was created and deployed by Wimmics and is now an online running platform providing data to several projects such as: QAKIS, Izipedia, zone47, Sépage, HdA Lab., JocondeLab, etc.

The platform can be found at: http://www.dbpedia.fr.

It is part of the Semanticpedia convention: http://www.semanticpedia.org/.

5.6. Semantic Wiki

Participants: Pavel Arapov, Michel Buffa.

WikiNEXT is a semantic wiki prototype (figure 4) written in JavaScript, from database to server and client code. It is not in competition with wikis like Semantic Media Wiki, but more a test bed for new ideas. Every wiki page is an application that keeps a Web Socket open with the server, enabling incremental saves or collaborative editions using Google wave like algorithms. Using JavaScript on the whole chain of operations avoids data transformation from/to different formats like in traditional approaches (Objects, JSON/XML, and SQL). WikiNEXT uses JavaScript distributed objects and includes an IDE to write JS applications within wiki pages.

Web page: http://wikinext.gexsoft.com

5.7. ISICIL

Participants: Nicolas Delaforge, Fabien Gandon [resp.].

In the context of the ISICIL ANR project, we have developped a Semantic Web server which provides core services to manage simple tagging of resources (internal or from the Web) and to assist the semantic enrichment of the folksonomy of our communities of users. This server's implementation is based on the ISICIL main framework. The tagging model combines already existing ontologies such as SIOC [4], SCOT, and Newman's Tag Ontology[5]. SRTag, the model of folksonomy enrichment, is based on a named graph mechanism in order to maintain diverging statements made between tags using SKOS (for thesaurus like relation between tags) or SCOT (for spelling variant relations), and is shown in figure 5 .

---

[3] https://gforge.inria.fr/projects/datalift/

[4] http://sioc-project.org

[5] http://www.holygoat.co.uk/owl/redwood/0.1/tags

../../../../projets/wimmics/IMG/QAKISCapture.png

Figure 3. QAKIS

../../../../projets/wimmics/IMG/WikiNextCapture.png

Figure 4. WikiNEXT

`../../../../projets/wimmics/IMG/srtag.png`

Figure 5. Folksonomy enrichment model

Web page: https://gforge.inria.fr/projects/isicil/

The code is now being refactored by the company Mnemotix, a SCOOP created as a spin-off of the project: http://www.mnemotix.com.

5.8. ZONE-project

Participant: Christophe Desclaux.

ZONE-project provides a new, innovative way to follow news (figure 6). At its core, the system is aggregating news items from various RSS feeds. Using the power of Semantic Web we are able to efficiently tag & annotate each news.Those tags are the basis of filters. Filters allow users to see only news that are relevant. For instance users can retrieve all news containing a tag, or on the contrary never see news containing specific tags. Basically it means that each user can create custom news feeds according to his interests. Though it may be tedious for John Doe to build its own filters, thus it will be possible to exchange filters with other users, or read specific news feeds built by other users. This will enable users to create news group feed focused on specific topics such as technology, heath, industry, transport, agriculture, communication, environment, etc. This project won the Inria BoostYourCode 2012 contest which was created in order to promote free & open source software.

Web page: http://www.zone-project.org/

`../../../../projets/wimmics/IMG/ZoneReadorCapture.png`

Figure 6. ZoneReador

ZENITH Project-Team

5. Software and Platforms

5.1. WebSmatch (Web Schema Matching)

Participants: Emmanuel Castanier, Rémi Coletta, Patrick Valduriez [contact].

URL: http://websmatch.gforge.inria.fr/

In the context of the Action de Développement Technologique (ADT) started in october 2010, WebSmatch is a flexible, open environment for discovering and matching complex schemas from many heterogeneous data sources over the Web. It provides three basic functions: (1) metadata extraction from data sources; (2) schema matching (both 2-way and n-way schema matching), (3) schema clustering to group similar schemas together. WebSmatch is being delivered through Web services, to be used directly by data integrators or other tools, with RIA clients. Implemented in Java, delivered as Open Source Software (under LGPL) and protected by a deposit at APP (Agence de Protection des Programmes). WebSmatch is being used by Datapublica and CIRAD to integrate public data sources.

5.2. SON (Shared-data Overlay Network)

Participants: Esther Pacitti, Didier Parigot [contact], Patrick Valduriez.

URL: http://www-sop.inria.fr/teams/zenith/SON

SON is an open source development platform for P2P networks using web services, JXTA and OSGi. SON combines three powerful paradigms: components, SOA and P2P. Components communicate by asynchronous message passing to provide weak coupling between system entities. To scale up and ease deployment, we rely on a decentralized organization based on a DHT for publishing and discovering services or data. In terms of communication, the infrastructure is based on JXTA virtual communication pipes, a technology that has been extensively used within the Grid community. Using SON, the development of a P2P application is done through the design and implementation of a set of components. Each component includes a technical code that provides the component services and a code component that provides the component logic (in Java). The complex aspects of asynchronous distributed programming (technical code) are separated from code components and automatically generated from an abstract description of services (provided or required) for each component by the component generator.

5.3. P2Prec (P2P recommendation service)

Participants: Esther Pacitti [contact], Didier Parigot, Maximilien Servajean.

URL: http://p2prec.gforge.inria.fr

P2Prec is a recommendation service for P2P content sharing systems that exploits users social data. To manage users social data, we rely on Friend-Of-A-Friend (FOAF) descriptions. P2Prec has a hybrid P2P architecture to work on top of any P2P content sharing system. It combines efficient DHT indexing to manage the users FOAF files with gossip robustness to disseminate the topics of expertise between friends. P2Prec is implemented in java using SON.

5.4. ProbDB (Probabilistic Database)

Participants: Reza Akbarinia [contact], Patrick Valduriez.

URL: http://probdb.gforge.inria.fr

ProbDB is a probabilistic data management system to manage uncertain data on top of relational DBMSs. One of the main features of the prototype is its portability; that means with a minimum effort it can be implemented over any DBMS. In ProbDB, we take advantage of the functionalities provided by almost all DBMSs, particularly the query processing functions. It is implemented in Java on top of PostgreSQL.

5.5. Pl@ntNet-mobile

Participants: Vera Bakic, Souheil Selmi, Hervé Goëau, Alexis Joly [contact].

URL: http://goo.gl/CpSrr3

Pl@ntNet-mobile is an image sharing and retrieval application for the identification of plants built in the continuity of the former web application Pl@ntNet-Identify [2] (presented in last year activity report). It is developed in the context of the Pl@ntNet project that involves four French research organisations (Inria, Cirad, INRA, IRD) and the members of Tela Botanica social network. The key feature of this free app is to help identifying plant species from photographs, through a server-side visual search engine based on several results of ZENITH team on content-based information retrieval. Since its first release in March 2013 on the apple store, the application was downloaded by around 80K users in about 150 countries (between 200 and 2000 active users daily with peaks occurring during the week-ends). The collaborative training set that allows the content-based identification is continuously enriched by the users of the application and the members of Tela Botanica social network. At the time of writing, it includes about 80K images covering more than 3500 French plant species about 2/3 of the whole French flora (this is actually the widest identification tool built anytime).

5.6. Pl@ntNet-DataManager

Participants: Mathias Chouet [contact], Alexis Joly.

Pl@ntNet-DataManager [3] is a software dedicated to managing and sharing distributed heterogeneous botanical data. It is developed jointly by Zenith, the AMAP UMR team (CIRAD) and Telabotanica non profit organization. It allows scientists to define data structures dedicated to their own datasets, and share parts of their structures and data with collaborators in a decentralized way. Pl@ntNet DataManager offers innovative features like partial or complete P2P synchronization between distant databases (master-master), and a user friendly data structure editor. It also provides full text search, querying, CSV import/export, SQL export, image management, and geolocation. DataManager is built on NoSQL technology (CouchDB database), Javascript (Node.js), HTML5 and CSS3, and may be deployed on a server or run on a local machine (standalone version for Linux, Windows, Mac). It is being used by researchers and engineers of the Pl@ntNet Project (CIRAD, INRA, Inria, IRD, Tela-Botanica) to manage taxonomical referentials, herbarium data and geolocated plant observations.

5.7. SnoopIm

Participants: Julien Champ [contact], Alexis Joly, Pierre Letessier.

URL: http://otmedia.lirmm.fr/

SnoopIm is a content-based search engine allowing to discover and retrieve small visual patterns or objects in large collections of pictures and to derive statistics from them (frequency, visual cover, size variations, etc.). It is implemented in Javascript on top of a C++ library developed in collaboration with INA [4]. The software is used at INA by archivists and sociologists in the context of the Transmedia Observatory project. It is also being experimented in several contexts including a logo retrieval application set up in collaboration with the French Press Agency, an experimental plant identification tool mixing textual and visual information retrieval (in the context of the Pl@ntNet project) and a research project on high-throughput analysis of root architecture images.

---

[2] http://identify.plantnet-project.org

[3] http://data.plantnet-project.org/

[4] http://www.ina-sup.com/

5.8. SciFloware

Participants: Dimitri Dupuis, Didier Parigot [contact], Patrick Valduriez.

URL: http://www-sop.inria.fr/members/Didier.Parigot/pmwiki/Scifloware

SciFloware is an action of technology development (ADT Inria) with the goal of developing a middleware for the execution of scientific workflows in a distributed and parallel way. It capitalizes on our experience with SON and an innovative algebraic approach to the management of scientific workflows. SciFloware provides a development environment and a runtime environment for scientific workflows, interoperable with existing systems. We will validate SciFloware with workflows for analyzing biological data provided by our partners CIRAD, INRA and IRD.

COPRIN Project-Team

5. Software and Platforms

5.1. Introduction

Software development is an essential part of the research done by COPRIN since a large part of our methods can only be validated experimentally (both for our numerical experiments and in robotics). Software developments follow various directions:

1. interval arithmetic: although we do not plan to work in this very specialized area (we generally rely on existing packages) interval arithmetic is an important part of our interval analysis algorithms and we may have to modify the existing packages so as to deal, in particular, with multi-precision and arithmetic extensions
2. interval analysis libraries: we daily use the ALIAS library that has been designed in the project and is still under development. A long term work is to develop a generic programming framework that allows for modularity and flexibility, with the objectives of testing new functionalities easily and building specific solvers by a simple juxtaposition of existing modules
3. interface to interval analysis: in our opinion interval analysis software must be available within general purpose scientific software (such as `Maple`, `Mathematica`, `Scilab`) and not only as a stand-alone tool. Indeed most end-users are reluctant to learn a new programming language just to solve problems that are only small elements of a more general problem. Furthermore interval analysis efficiency may benefit from the functionalities available in the general purpose scientific software.

5.2. Interval analysis libraries

5.2.1. ALIAS

Participants: David Daney, Jean-Pierre Merlet [correspondant], Odile Pourtallier.

The `ALIAS` library (*Algorithms Library of Interval Analysis for Systems*), whose development started in 1998, is a collection of procedures based on interval analysis for systems solving and optimization.

`ALIAS` is made of two parts:

- `ALIAS-C++`: the C++ library (87 000 code lines) which is the core of the algorithms
- `ALIAS-Maple`: the `Maple` interface for `ALIAS-C++` (55 000 code lines). This interface allows one to specify a solving problem within `Maple` and get the results within the same `Maple` session. The role of this interface is not only to generate the C++ code automatically, but also to perform an analysis of the problem in order to improve the efficiency of the solver. Furthermore, a distributed implementation of the algorithms is available directly within the interface.

Although these libraries are intended to be used within the project-team they can be freely downloaded as a library file (but the user may introduce its own code in several part of the package) and has been used for example at LIRMM and IRCCyN.

IMARA Project-Team

5. Software and Platforms

5.1. MELOSYM

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

MELOSYM is the acronym for “**M**odélisation de l’**E**nvironnement et **LO**calisation en temps réel pour un **SY**stème **M**obile autonome ou pas, fondé sur des données du capteur laser”. This is a SLAM based algorithm for the environment mapping and vehicle localization in real-time using laser data. The particularity of the algorithm is its hierarchical approach that improves the accuracy of the system and speeds up the computations. Version 3 is under edition. It runs now in standalone mode without the use of RTMaps software libraries.

- Version: V2

5.2. Stereoloc-3D

Participants: Benjamin Lefaudeux, Fawzi Nashashibi [correspondant].

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

- Version: V1

5.3. Fuzzy logic tool

Participant: Joshué Pérez Rastelli [correspondant].

A fuzzy logic module has been implemented to translate human knowledge to driverless control processes, considering risk/warning situation. Fuzzy logic techniques have been widely implemented in different industrial process in the last decade. For this reason, many libraries, mainly developed in C++, are easily found in the literature. The goal is to achieve the autonomous driving of the vehicle using simple sentences defined in a rule base. Then, it is just necessary to define the input and output membership functions. Two modules based on fuzzy logic libraries were created. One of them was developed in order to compare the classic controller of a previous work with a fuzzy controller to improve the lateral control tracking previously developed. Moreover, another module to warn speed references at intersections with traffic lights was done in the framework of the project CoDrive. The idea is that the vehicle is able to know at which speed it must travel to avoid abrupt braking and save fuel.

5.4. Dynamic path generation

Participant: Joshué Pérez Rastelli [correspondant].

An algorithm for dynamic path generation in urban environments is presented, taking into account structural and sudden changes in straight and bend segments (e.g. roundabouts and intersections). The results present some improvements in path generation (previously hand plotted) considering parametric equations and continuous-curvature algorithms, which guarantees a comfortable lateral acceleration. This work is focused in a smooth and safe path generation using road and obstacle detection information. Finally, some simulation results show a good performance of the algorithm using different ranges of urban curves.

5.5. V2ProVu

Participants: Pierre Merdrignac, Oyunchimeg Shagdar [correspondant].

A Java-based software is developed to enable direct Wi-Fi communications between devices, especially between vehicle on-board communication devices and pedestrian hand-held devices (e.g., tablets). The software includes an algorithm that calculates vehicle-to-pedestrian collision risk and GUI, for hazard alarming.

5.6. Network Selector

Participant: Oyunchimeg Shagdar [correspondant].

An OSGi based software is developed under the scope of SCORE@F project. The software has the functionality of switching between Geo- and IP-networks in vehicular communications allowing e.g., Cooperative Awareness Messages (CAM) as well as Decentralized Event Notification Messages (DENM) being able distributed over one or both of the Geo- and IP-networks.

5.7. FAC-CM

Participant: Manabu Tsukada [correspondant].

An OSGi based software is developed under the scope of SCORE@F project. The software allows information exchange between Facilities and Management entities of ITS stations (e.g., vehicle on-board communication device).