Activity Report 2013

Section Software

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.

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 Ziggie 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 Ziggie platform is part of the BoWI project funded by CominLabs. Ziggie 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.

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/.

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-operate 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.

<p style="text-align:center">Hycomes Team</p>

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).

<p style="text-align: center; color: red;">**S4 Project-Team**</p>

5. Software and Platforms

5.1. 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 [8]. 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).

5.2. Synet: A General Petri-Net Sythesis Toolbox

Participant: Benoît Caillaud.

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

Synet is a software tool for the synthesis of bounded and unbounded Petri-nets, based on the theory of regions [21]. It can synthesize Petri-nets from automata or regular expressions and can be configured by command-line options to synthesize nets modulo graph isomorphism or language equality. Petri nets computed by Synet can be displayed using the GraphViz 2D graph layout software, or saved to a file for further transformation and analysis.

The tool actually implements two linear-algebraic synthesis methods: a first method uses the simplex algorithm and the second one is based on the computation of extremal rays of polyhedral cones, using Chernikova's algorithm [23]. Both methods imply that the input graphs are given by extension. Nevertheless, Synet yields good performances on many practical use-cases and is the only tool supporting unbounded net synthesis.

The main application of Synet is the synthesis of communicating distributed protocols and controllers [20]. Synthesis is constrained to produce so-called distributables nets [22], a class of nets that can be turned into networks of communicating automata by automated methods. This allows to divide the synthesis problem in two steps: Given the specification of a protocol as a finite automaton, (i) synthesize (if it exists) a distributable net, and then (ii) derive a network of communicating automata from the distributable net. While the second step is automatic and straightforward, the first step is in essence a computer assisted design task, where the distributed Petri-net synthesis algorithm helps the designer to refine the protocol specification into a graph isomorphic to the marking graph of a distributable net.

<div align="center">**SUMO Team**</div>

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 developed 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 redistributed. 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.

ASPI Project-Team (section vide)

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)

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].

<p style="text-align:center">**FLUMINANCE Project-Team**</p>

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.

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/

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 Bucharest, 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)

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.

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.
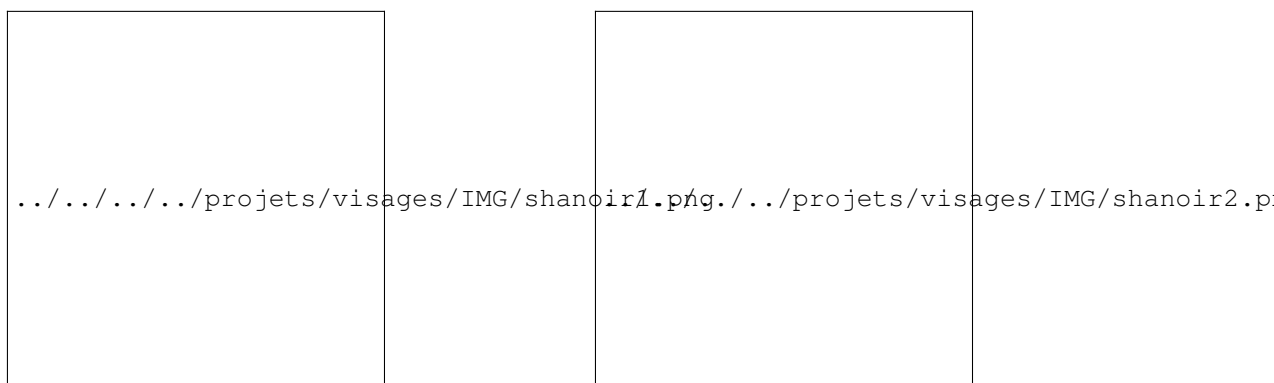- Programming language: Java
- Documentation: see the website

../../../../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/
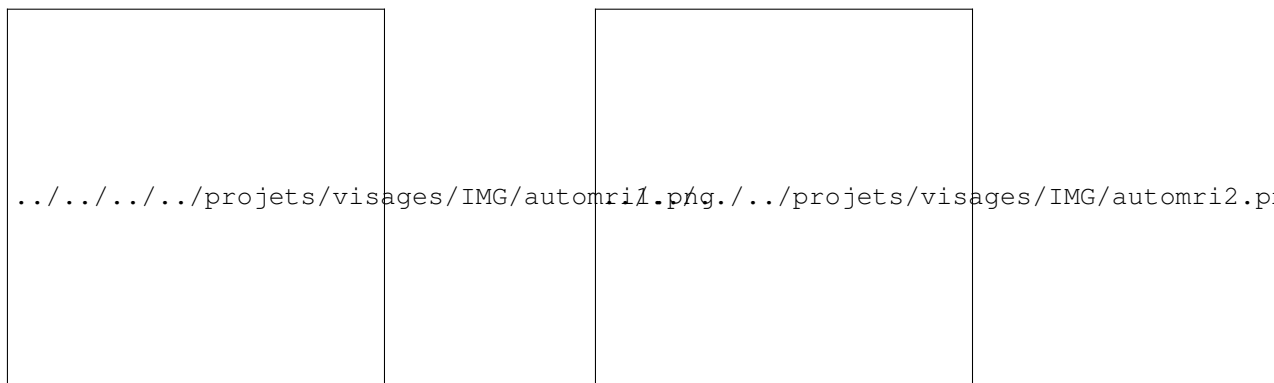

Figure 3. Illustrations of results obtained with autoMRI: Conjunction map showing areas of hypoperfusion and hypometabolism in semantic dementia (right), Detection of relaxometry defect in an MS patient (left).

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++

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++

<p style="text-align:center">**ACES Project-Team**</p>

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.

<p align="center" style="color:red">ASAP Project-Team</p>

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

CIDRE Project-Team

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.

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`.

<p style="text-align:center; color:red;">**KERDATA Project-Team**</p>

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 Open-Nebula 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.

<h2 style="text-align:center; color:red">MYRIADS Project-Team</h2>

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.

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

<p align="center" style="color:red">DREAM Project-Team</p>

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 geoerenced 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.

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.

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.
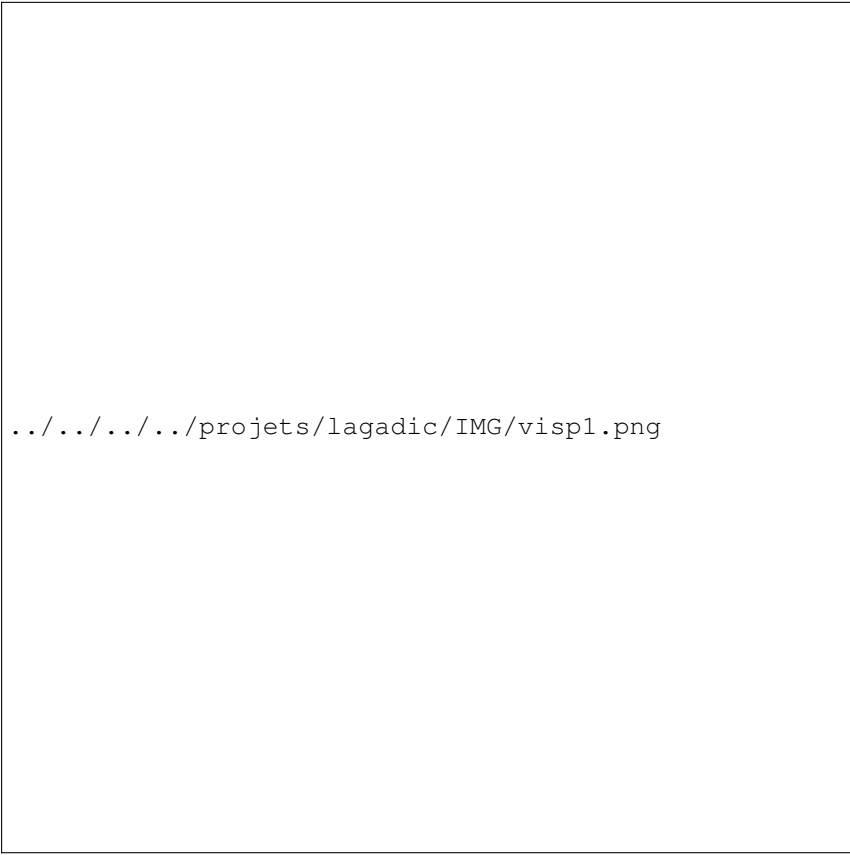
5.2. DESlam software

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/pioneero.png ../../../../projets/lagadic/IMG/fau

(a)   (b)   (c)

../../../../projets/lagadic/IMG/Cycab_Rennes.png ../../../../projets/lagadic/IMG/Cycab_Sop

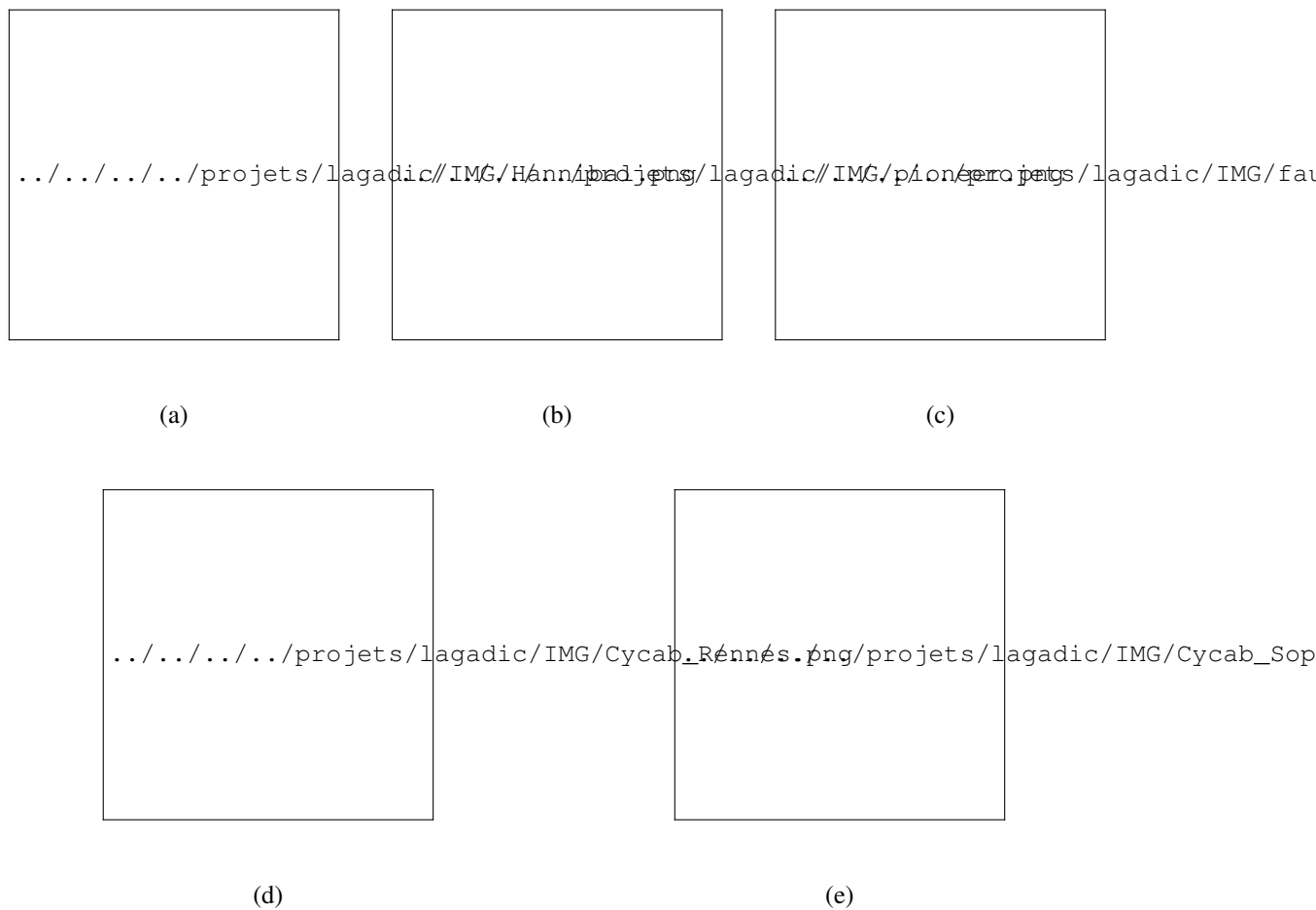(d)   (e)

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.

<p style="text-align:center; color:red;">**MIMETIC Project-Team**</p>

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.

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.

<p style="text-align:center">**SIROCCO Project-Team**</p>

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.

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 CORRE-SPONDENCE 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.