



RESEARCH CENTER
Rennes - Bretagne-Atlantique

FIELD

Activity Report 2014

Section Software

Edition: 2015-03-24

1. ALF Project-Team	4
2. ASAP Project-Team	7
3. ASCOLA Project-Team	11
4. ASPI Project-Team (section vide)	14
5. ATLANMOD Project-Team	15
6. CAIRN Project-Team	21
7. CELTIQUE Project-Team	26
8. CIDRE Project-Team	28
9. DIONYSOS Project-Team	29
10. DIVERSE Project-Team	30
11. DREAM Project-Team	33
12. DYLISS Project-Team	36
13. ESTASYS Exploratory Action	40
14. FLUMINANCE Project-Team	42
15. GENSCALE Project-Team	44
16. HYBRID Project-Team	46
17. HYCOMES Team	47
18. I4S Project-Team	48
19. IPSO Project-Team (section vide)	50
20. KerData Project-Team	51
21. LAGADIC Project-Team	54
22. LINKMEDIA Project-Team	61
23. MIMETIC Project-Team	62
24. MYRIADS Project-Team	63
25. PANAMA Project-Team	67
26. SAGE Project-Team	68
27. SERPICO Project-Team	71
28. SIROCCO Project-Team	75
29. SUMO Project-Team	76
30. TACOMA Team	77
31. TASC Project-Team	78
32. TEA Project-Team	81
33. VISAGES Project-Team	85

ALF Project-Team

5. New Software and Platforms

5.1. Panorama

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

Among the prototypes developed in the project, this section reports only the softwares that had significant revisions in 2014. Among the softwares available from the project website and not reported here, **ATMI** <http://www.irisa.fr/alf/atmi>, a microarchitecture temperature model for processor simulation, **STiMuL** <http://www.irisa.fr/alf/stimul>, a temperature model for steady state studies, **ATC** <http://www.irisa.fr/alf/atc>, an address trace compressor, and **HAVEGE** <http://www.irisa.fr/alf/havege> an unpredictable random number generator.

5.2. TPCalc

Participant: Pierre Michaud.

microarchitecture simulation

TPCalc is a throughput calculator for microarchitecture studies concerned with multi-program workloads consisting of sequential programs. Because microarchitecture simulators are slow, it is difficult to simulate throughput experiments where a multicore executes many jobs that enter and leave the system. The usual practice of measuring instantaneous throughput on independent coschedules chosen more or less randomly is not a rigorous practice because it assumes that all the coschedules are equally important, which is not always true. TPCalc can compute the average throughput of a throughput experiment without actually doing the throughput experiment. The user first defines the workload heterogeneity (number of different job types), the multicore configuration (number of cores and symmetries). TPCalc provides a list of base coschedules. The user then simulates these coschedules, using some benchmarks of his/her choice, and feeds back to TPCalc the measured execution rates (e.g., instructions per cycle or instructions per second). TPCalc eventually outputs the average throughput. Several throughput metrics are available, corresponding to different workload assumptions. These metrics are described in our ACM TACO paper, a collaboration with Ghent University [15].

TPCalc is an open-source software written in C++. It runs on Unix-based systems (Linux, OS X ...). It is available for download at <http://www.irisa.fr/alf/downloads/michaud/tpcalc.html>.

5.3. Heptane

Participants: Isabelle Puaut, Damien Hardy.

WCET estimation

Status: Registered with APP (Agence de Protection des Programmes). Available under GNU General Public License v3, with number IDDN.FR.001.510039.000.S.P.2003.000.10600.

The aim of Heptane is to produce upper bounds of the execution times of applications. It is targeted at applications with hard real-time requirements (automotive, railway, aerospace domains). Heptane computes WCETs using static analysis at the binary code level. It includes static analyses of microarchitectural elements such as caches and cache hierarchies.

For more information, please contact Damien Hardy or Isabelle Puaut.

5.4. 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, with number IDN.FR.001.450006.000.S.P.2011.000.10800. Current version is 2.2, released March 2013.

Tiptop has been integrated in major Linux distributions, such as Fedora, Debian, Ubuntu.

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 preferred setup, as well as custom expressions.

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

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

5.5. Padrone

Participants: Erven Rohou, Emmanuel Riou.

Performance, profiling, dynamic optimization

Status: Ongoing development, early prototype. Registered with APP (Agence de Protection des Programmes).

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 instrumental to the PhD developments of Nabil Hallou.

Padrone is written in C.

For more information, please contact Erven Rohou.

5.6. Barra

Participant: Sylvain Collange.

GPU simulator

Other Contributors : David Defour (Université de Perpignan), Alexandre Kouyoumdjian (Inria), Elie Gedeon (ENS Lyon), Fabrice Mouhartem (Inria)

Status : APP registration in progress. Available under the new BSD License

Research on throughput-oriented architectures demands accurate and representative models of GPU architectures in order to be able to evaluate new architectural ideas, explore design spaces and characterize applications. The Barra project ⁰ is a simulator of the NVIDIA Tesla GPU architecture.

Barra builds upon knowledge acquired through micro-benchmarking, in order to provide a baseline model representative of industry practice. The simulator provides detailed statistics to identify optimization opportunities and is fully customizable to experiment ideas of architectural modifications. Barra incorporates both a functional model and a cycle-level performance model.

Visit <http://barra.gforge.inria.fr/> or contact Sylvain Collange.

⁰http://gforge.inria.fr/plugins/mediawiki/wiki/barra/index.php/Main_Page

ASAP Project-Team

5. New Software and Platforms

5.1. MediEgo: A recommendation solution for webmasters

Participants: Jacques Falcou, Arnaud Jégou, Xavier Lucas, 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. The MEDIEGO recommendation engine is built in collaboration with Sébastien Campion. We have demonstrated the software at the conference Le Web 14 (9-11 Dec 2014) in collaboration with France Television.

5.2. AllYours-P2P: A distributed news recommender (former WhatsUp)

Participants: Heverson Borba Ribeiro, Raziél Carvajal Gomez, Davide Frey, Arnaud Jégou, Anne-Marie Kermarrec.

Contact:	Davide Frey
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

Within the context of the AllYours EIT/ICT-Labs project, we refined the implementation of WhatsUp into the Peer-to-Peer AllYours application. AllYours-P2P is a peer-to-peer based news recommender system that organizes users into an implicit social network based on their explicit opinions. In AllYours-P2P the recommendation process is collaboratively performed by connected users. Every user runs a symmetric piece of software responsible for storing user interests and calculating the affinity between a user and its neighborhood. The local computed similarity then is used to keep virtual connections to other users whose interests are alike and remove connection to the ones that are not. As a result, an interest-based overlay is built and users converge to groups of similar interests within which news are disseminated. The AllYours-P2P software 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 back-end is written in Java, while the user interface comprises HTML and Javascript code. AllYours-P2P is currently available in three different platforms: Mac OSx (10.5 or later), Windows (Vista and Windows 7) and Linux (Ubuntu 10.4 or later). We have tested AllYours-p2p in a real life environment with a set of invited users in Italy from Sep to Nov 2014. These test were a part of joint project between ASAP Team and its Italian partner Trentorise.

Currently the implementation of AllYours-p2p includes approximately 21K lines of code.

5.3. HyRec: A hybrid recommender system

Participants: Davide Frey, Anne-Marie Kermarrec.

Contact: Davide Frey
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. This software has been developed in collaboration with Antoine Boutet (Univ. Saint-Étienne) and Rhicheck Patra (EPFL).

5.4. 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. Currently the implementation of GossipLib includes approximately 9K lines of code, and is used in several projects by ASAP, including HEAP, AllYours-P2P, and Behave.

5.5. YALPS: A library for p2p applications

Participants: Heverson Borba Ribeiro, Davide Frey, Arnaud Jégou, 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. The implementation of YALPS includes approximately 16K lines of code, and is used in several projects by ASAP, including HEAP, AllYours-P2P, and Behave. This work was done in collaboration with Maxime Monod (EPFL).

5.6. 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 led 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.

5.7. Brow2Brow: Browser-to-browser serverless toolboxes

Participants: Raziel Carvajal Gomez, Davide Frey, Anne-Marie Kermarrec.

Brow2Brow is an "Action de Développement Technologique", i.e. a collaborative development project that aims at providing a middleware and software library for browser-to-browser applications. Brow2Brow involves the ASAP team as well as the DICE Team from Inria Grenoble (Antenne de Lyon). The project seeks to provide an alternative to the current model followed by Web2.0 applications by exploiting the recently introduced WebRTC standard. Existing Web 2.0 applications collect data on browsers and send it to servers that store and process it. The goal of Brow2Brow is to provide an alternative approach where browsers can themselves proceed to collaborative data processing. This will make it possible avoid data concentration at a single server. The project has resulted so far in the development of WebGC, a library for gossip-based applications on browsers.

5.8. WebGC: Web-based Gossip Communication

Participants: Raziel Carvajal Gomez, Davide Frey, Anne-Marie Kermarrec.

Contact:	Raziel Carvajal Gomez, Davide Frey
License:	Not-yet released
Presentation:	Library for Gossip protocols within Web Browsers
Status:	Ongoing development

WebGC is a library for gossip-based communication between web-browsers. It has been developed in collaboration with Mathieu Simonin in the context of the Brow2Brow ADT project. WebGC builds on the recent WebRTC standard as well as on PeerJS, an open-source project that provides primitives for data transfer on top of WebRTC.

The library currently includes the implementation of two peer sampling protocols, CYCLON and the generic peer-sampling protocol from [7], as well as a clustering protocol [1]. All protocols implement a common GOSSIPPROTOCOL “interface”—since Javascript does not natively support interfaces, we adopt the interface pattern. A COORDINATOR makes it possible to stack these protocols on top of each other to implement applications.

ASCOLA Project-Team

5. New Software and Platforms

5.1. btrCloud (and Entropy)

Participants: Jean-Marc Menaud [correspondent], Guillaume Le Louët, 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 centers. 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, on the long run, or in order to predict future trends. This feature includes boards for performance evaluation and analysis as well as trends estimation.

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 Cloud 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 [27][26].

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. 8.3) 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>.

In 2014, we have used and extended the tool in order to investigate accountability properties of service-based applications, see Sec. 6.3.

5.5. SimGrid/VMPlaces

Participants: Takahiro Hirofuchi, Adrien Lebre [correspondent], Jonathan Pastor, Flavien Quesnel, Mario Südholt.

Simulation, Virtualization, Cloud computing, VM placement

SimGrid is a toolkit for the simulation of algorithms executed on large-scale distributed systems. Developed for more than a decade, it has been used in a large number of studies described in more than 100 publications. In 2013, ASCOLA with the support of the SimGrid core-developers, designed and implemented additional capabilities, in particular the Virtual Machine abstraction, enabling to address Cloud Computing related concerns.

Developed, first, in an experimental repository, the integration of these extensions into the master branch of SimGrid has been achieved during Summer 2014. The principal role of ASCOLA is now to ensure the maintenance of this portion of the code with respect to the evolutions of the SimGrid toolkit (such as for instance the recent port of the SimGrid kernel in C++).

Although the virtualization extensions are recent, several projects leveraging them have been already proposed.⁰ Among them, ASCOLA is working on dedicated framework to evaluate and compare VM placement algorithms. Entitled VMPlaces, this framework is composed of two major components: the injector and the VM placement algorithm. The injector is the generic part of the framework (i.e. the one you can directly use) while the VM placement algorithm is the part you want to study (or compare with available algorithms). Currently, the VMPlaceS is released with three algorithms:

- Entropy, a centralized approach using a constraint programming approach to solve the placement/reconfiguration VM problem
- Snooze, a hierarchical approach where each manager of a group invokes Entropy to solve the placement/reconfiguration VM problem. Note that in the original implementation of Snooze, it is using a specific heuristic to solve the placement/reconfiguration VM problem. As the sake of simplicity, we have simply reused the entropy scheduling code.
- DVMS, a distributed approach that dynamically partitions the system and invokes Entropy on each partition.

SimGrid is available at <http://simgrid.gforge.inria.fr>.

VMPlaces is available at <http://beyondtheclouds.github.io/VMPlaceS/>

⁰The list of the projects is available at : <http://simgrid.gforge.inria.fr/contrib/clouds-sg-doc.html>

ASPI Project-Team (section vide)

ATLANMOD Project-Team

5. New 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 [48]. 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. 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 has been developed showing how transformations can be developed without taking into account concurrency concerns, and such a transformation engine can automatically parallelize operations improving execution times.

Aligned with this research line we propose in recent works an approach to automatically parallelize the computation of model transformation using Cloud infrastructures. For this, we take advantage of a well-known distributed programming model: *MapReduce*. In this sense, we introduce an algorithm aligning both execution semantics of ATL and MapReduce. Based on this, a new prototype tool has been developed⁰ showing in several experiments the scalability of the solution.

5.2. MoDisco (Model Discovery)

URL: <http://www.eclipse.org/MoDisco/>

⁰https://github.com/atlanmod/ATL_MR

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 ARTIST⁰, 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://atlanmod.github.io/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. Since the very first implementation of the Collaboro toolset was released, it has evolved to provide support to both Eclipse-based and web-based clients.

The Eclipse-based client has been developed as a plugin in the platform while the web-based client includes two components: (1) the server-side part, which offers a set of services to access to the main functionalities of Collaboro; and the client-side part, which allows both end-users and developers to take part of the DSML development process from their browsers. The server-side component has been developed as a Java web application which uses a set of Servlets providing the required services. On the other hand, the client-side component has been developed as an AngularJS-enabled website and provides.

⁰<http://www.artist-project.eu/>

The Collaboro clients provide access to the following features:

- 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.
- 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 incorporating 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: <https://github.com/atlanmod/emfviews>

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 has been initially developed in the context of the TEAP industrial project <http://www.teap-project.org/> that ended in November 2014, 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 transformation using dedicated technologies (e.g. Eclipse ATL).
- Report generation from views, etc.

The EMF Views prototype is currently being re-used and further developed, in a (meta)model extension context this time, within the ongoing MoNoGe industrial project. The objective of this present work is to propose a simple base generic (meta)model extension mechanism, relying on EMF Views capabilities, that could be deployed in different scenarios where (meta)model extension is required (e.g. metamodel evolution, model integration, etc.)

A presentation of EMF Views took place at EclipseCon 2014⁰, held in San Francisco, California, U.S.A

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 (MDT⁰) 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 solver⁰ 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 [45] 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.

⁰<https://www.eclipsecon.org/na2014/session/modeling-symposium>

⁰<http://www.eclipse.org/modeling/mdt/?project=ocl>

⁰<http://eclipseclp.org/>

5.8. NeoEMF

URL: <http://www.neo4emf.com>

NeoEMF (a relaunch of the tool formerly known as Neo4EMF) is an open source software distributed under the terms of the Eclipse Public License that provides a backend-agnostic persistence solution for big, complex and highly interconnected EMF models. NeoEMF is a model repository and persistence framework allowing on-demand loading, storage, and unloading of large-scale EMF models.

NeoEMF is designed to allow the easy integration of custom backends depending on user needs. By default, NeoEMF is bundled with out-of-the-box support for graph databases (based on the blueprints API⁰ and key-value stores (based on MapDB⁰). Blueprints is an abstraction layer for graph storages that allows changing the actual database used without affecting the application code. The blueprints-based back-end allows the integration of NeoEMF and Neo4j—among other databases—providing in NeoEMF the full set of features already implemented in Neo4EMF. MapDB is an efficient key-value store that provides concurrent Maps, Sets and Queues backed by disk storage or off-heap memory.

In terms of performance, NeoEMF 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 provided through the following features:

- Lazy-loading mechanism. Model objects are loaded on demand while needed. In its basic configuration, model objects act as a proxy that occupy little memory, and fields are only retrieved when accessed.
- Caching. NeoEMF relies on database caches to retrieve EObjects, but in some situation this is not enough. For this reason, the architecture on NeoEMF allows the easy implementation of domain-specific cache strategies based on the decorator pattern.
- Auto-commit. In back-ends in which transaction data is stored on the heap, it is possible to use the auto-commit feature to split large transaction into several small ones.
- Dirty saving. The dirty saving feature is an step forward on the auto-commit strategy. It allows to safely handle big transactions by splitting them into small ones by saving partial changes made on models to disk. In case of transaction failure or cancellation, the partial model changes can be reverted and the model is restored to its original state.

A session about NeoEMF took place at eclipseCon France 2014⁰, held in Toulouse, France.

Works are still going over NeoEMF (within the context of the project ITM Factory - FUI14) to provide more utilities such as backend-aware query languages (which allows improving performance by taking advantage of the backend built-in query languages), concurrent access, model distribution, and other Ecore utilities.

5.9. GitHub Label Analyzer

URL: <http://atlanmod.github.io/gila/>

Reporting bugs, asking for new features and in general giving any kind of feedback is the easiest way to contribute to an Open-Source Software (OSS) project. In GitHub, the largest code hosting service for OSS, this feedback is typically expressed as new issues for the project managed by an issue-tracking system available in each new project repository. Among other features, the issue tracker allows creating and assigning labels to issues with the goal of helping the project community to better classify and manage those issues (e.g., facilitating the identification of issues for top priority components or candidate developers that could solve them). Nevertheless, as the project grows a manual browsing of the project issues is no longer feasible.

⁰<https://github.com/tinkerpop/blueprints/>

⁰<https://github.com/jankotek/MapDB>

⁰<https://www.eclipsecon.org/france2014/session/neo4emf-when-big-models-are-no-longer-issue>

We believe that visualization techniques could be applied here to overcome this challenge. In particular, we have created GiLA, a tool to better understand how labels are being used in GitHub projects, with the aim of providing more insights into how such projects are being managed. GiLA provides three visualizations addressing three different viewpoints, specifically:

- V1 Label usage, which helps to identify the most used labels and which ones are commonly used together.
- V2 User involvement, which allows discovering the most active and knowledgeable users around each label.
- V3 Typical Label timeline, which provides some insights about how issues under that label evolve over time (e.g., time to be treated).

The tool can be used to explore these viewpoints on all the original projects (i.e., projects that are not a fork of a previous project) in GitHub. We believe that the results favour not only a better comprehension of the project but also help in its advancement, e.g., by helping to quickly identify experts on a particular topic/label.

CAIRN Project-Team

5. New 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.

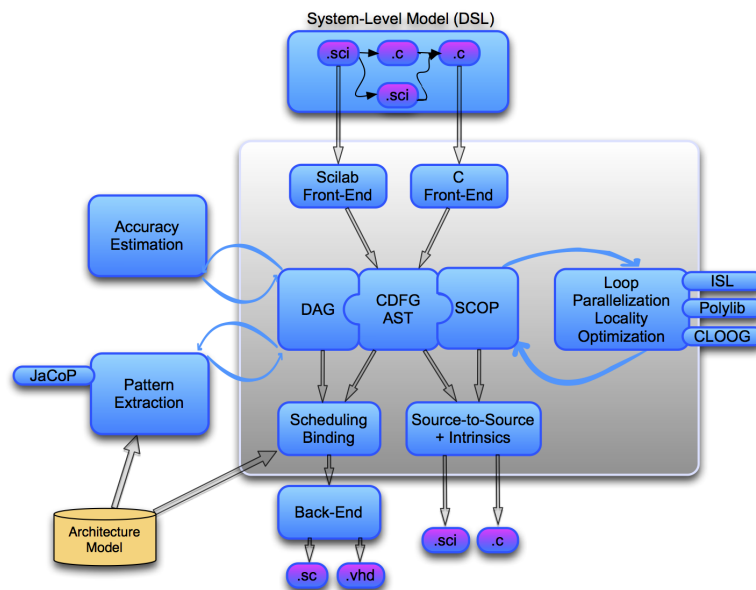


Figure 2. CAIRN's general software development framework.

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, 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. Recent developments in Gecos 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 put to provide a coarse-grain parallelization engine targeting the data-flow actor model in the context of the COMPA ANR project.

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 2014 allowed to have a complete compatibility with GeCos. The structure of each node in the graph has been changed to simplify the graph modifications. The Octave software has been added instead of Matlab for LTI and recursive systems conversion. A development has started to replace Matlab/Octave tool by a C code algorithm to reduce optimization time. In the context of the ANR DEFIS project, the ID.Fix tool has been reorganized to be integrated in the DEFIS toolflow.

In 2014, ID.Fix has been demonstrated during University Booth at IEEE/ACM DATE.

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

Participants: Christophe Wolinski [corresponding author], François Charot.

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* [117]. 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.

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 [79] 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).



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

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 [95], 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>.

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



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 Zyggy 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).

CELTIQUE Project-Team

4. New 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. Its 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.

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 harvested 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: Alan Schmitt [correspondant], Martin Bodin.

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, including the source code, is available at <http://jscert.org/>.

CIDRE Project-Team

5. New Software and Platforms

5.1. Intrusion Detection and Privacy

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; **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 Language (**ADeLe**) proposed by our team; **Netzob** is an open-source tool for reverse engineering, traffic generation and fuzzing of communication protocols; a log visualization tool called **ELVIS** (Extensible Log VISualization) has been implemented in order to test our approaches for log exploration.

In addition, the team participate to the development of **GEPETO** (GEOPrivacy-Enhancing TOolkit), an open source software for managing location data (in cooperation with the CNRS Lab. LAAS, Toulouse). GEPETO can be used to visualize, sanitize, perform inference attacks, and measure the utility of a particular geolocated dataset.

These tools are still under development in the team. Nevertheless, there are not new. For more details, please see previous activity reports.

DIONYSOS Project-Team

4. New Software and Platforms

4.1. T3devKit testing toolkit and IPv6 test suites

Participant: César Viho.

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>.

4.2. Interoperability Assessment

Participant: César Viho.

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>.

4.3. 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`.

Pierre L'Ecuyer is also developing in Montreal a library, *Stochastic Simulation in Java* (SSJ), providing facilities for generating uniform and nonuniform random variates, computing different measures related to probability distributions, performing goodness-of-fit tests, applying quasi-Monte Carlo methods, collecting (elementary) statistics, and programming discrete-event simulations with both events and processes.

DIVERSE Project-Team

5. New Software and Platforms

5.1. Kermeta

Participants: Benoit Combemale [correspondant], Olivier Barais, Arnaud Blouin, Didier Vojtisek, Benoit Baudry, Thomas Degueule, David Mendez, Erwan Bousse, Francois Tanguy, Fabien Coulon.

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 Ecore industrial standard and aligned with the EMOF 2.0 OMG standard, the Kermeta language consists in an 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.

Kermeta 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.

In 2014, we have continued the refactoring of Kermeta to leverage on Xtend. The Kermeta action language is now defined as an extension of Xtend, by 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 non model-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 innovative solutions for MDE.

More precisely, in addition to an Xtend extension dedicated to model manipulation, we started to integrate in Kermeta various facilities to support software language engineering (slicing, pruning, reuse, variability management, etc). In 2014, we improved this software language engineering feature (currently named k3sle/Melange) in order to offer a functional model typing system allowing safe model polymorphism. This system enables reuse of algorithms and transformations across different metamodels, as well as language inheritance, evolution and interoperability.

Moreover, while this version of Kermeta is a DSML development workbench that provides 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.

According to Google Scholar ⁰, the Kermeta platform was used or cited in more than 1300 papers.

5.2. 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.

⁰<http://scholar.google.fr/scholar?q=kermeta+model>

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). The feature models formalism has been studied for more than 20 years [98], and it is widely used in the industry [100]. 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 perform complex variability management tasks: extraction of feature models from software artifacts [87], product line evolution [89], management of multiple models [88] [75], [76], model-based validation of SPLs [22], large scale configuration of feature models [122], etc. The level of maturity of the FAMILIAR platform is TRL 3 (*i.e.* 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 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);
- support of advanced variability constructs (e.g., attributes, multi-features, meta-information);
- Web-based, comprehensive environment (WebFML [42]).

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 and tool demonstrations [42], [25] 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 industrial transfer activity with companies such as Thales. FAMILIAR is involved in several research projects (e.g., 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 familiar-project.github.com.

- Version: 1.3
- Programming language: Java, Scala

5.3. Kevoree

Participants: Olivier Barais [correspondant], Johan Bourcier, Noel Plouzeau, Benoit Baudry, Maxime Tricoire, Jacky Bourgeois, Inti G. Herrera, Ivan Paez, Francisco Acosta, Mohamed Boussaa.

Kevoree is an open-source models@runtime platform ⁰ to properly support the dynamic adaptation of distributed systems. Models@runtime basically pushes the idea of reflection [132] 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.

⁰<http://www.kevoree.org>

Kevoree has been influenced by previous work that we carried out in the DiVA project [132] and the Entimid project [135]. With Kevoree we push our vision of models@runtime [131] 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-system⁰.
- SpringSource Dynamic Module⁰
- GCM-Proactive⁰
- OSGi⁰
- Chef⁰
- Vagran⁰

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 Comparch conference in July 2014 and at the Middleware conference in december 2014. See also the web page <http://www.kevoree.org>.

In 2014, we mainly created a new implementation in JavaScript and we created an implementation for system containers for driving resources using Kevoree. We also use Kevoree in the context of Mohammed's PhD to create testing infrastructure on-demand.

- Version: 5.1.4
- Programming language: Java, Scala, Kermeta, Kotlin, Javascript

⁰<http://frascati.ow2.org>

⁰<http://spring.io/>

⁰<http://proactive.inria.fr/>

⁰<http://www.osgi.org>

⁰<http://wiki.opscode.com/display/chef/Deploy+Resource>

⁰<http://vagrantup.com/>

DREAM Project-Team

5. New 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. Platforms

The Dream project-team, in collaboration with their applicative partners, has proposed and maintains several important software platforms for its main research topics.

5.2.1. Platform: *Environmental decision-support systems*

Participants: Marie-Odile Cordier, Christine Largouët, Véronique Masson, Yulong Zhao.

SACADEAU: the SACADEAU system is an environmental decision software (cf. 4.2) 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. A more elaborated version allows to launch simulations, to learn rules on-line and to access to two recommendation action algorithms. This year, we have developed a new visualization tool designed to compare two sets of rules learnt from simulations. The user can choose one (or more) rule(s) to compare from one set of rules, and a distance to apply from several multidimensional distances. The most similar rules in the second set of rules are found and the comparison can be easily visualized. The examples covered by "similar" rules can also be presented to the user by highlighting shared positive and negative covered examples. 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 [57] for a presentation.

ECOMATA: The ECOMATA tool-box provides means for qualitative modeling and exploring ecosystems and for aiding to design environmental guidelines. We have proposed a new qualitative approach for ecosystem modeling (cf. 4.2) 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 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>.

PATURMATA: The Paturmata tool-box provides means for qualitative modeling and exploring agrosystems, specifically management of herd based on pasture [5]. 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.

5.2.2. Platform: Pattern Mining

Participants: Thomas Guyet, René Quiniou.

QTempIntMiner: the QTEMPINTMINER (Quantitative Temporal Interval Miner) data mining software implements several algorithms presented in [46] and [3] (QTIAPRIORI and QTIPREFIXSPAN). The software is mainly implemented in Matlab. It uses the Mixmod toolbox [33] 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 software includes one new algorithm based on the separation of the set of interval to extract more efficiently but less accurately the time interval in temporal patterns. This new algorithm version is still under evaluation on simulated and real datasets (care pathways).

The following website gives many details about the algorithms and provides the latest stable implementation of QTEMPINTMINER: <http://www.irisa.fr/dream/QTempIntMiner/>.

5.2.3. Platform: Diagnostic and Monitoring Systems

Participants: Marie-Odile Cordier, René Quiniou, Sophie Robin.

Odisseptale: the Odisseptale software implements 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.

ManageYourself: the ManageYourself software comes from a collaborative project between Dream and the Telelogos company aiming at monitoring smartphones from a stream of observations made on the smartphone state.

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.

5.3. TraceSquiz: reduction of captured trace volume

Participants: Serge Vladimir Emteu Tchagou, Alexandre Termier.

TraceSquiz is a software developed in collaboration with STMicroelectronics. Its goal is to reduce the volume of execution trace captured during endurance tests of multimedia applications. It uses anomaly detection techniques to "learn" regular parts of the trace and only capture the irregular ones. The software is written in C++.

DYLISS Project-Team

5. New Software and Platforms

5.1. Platforms and toolboxes

Among others, a goal of the team is to facilitate interplays between tools for biological data analysis and integration. Our tools are based on formal systems. They aim at guiding the user to progressively reduce the space of models (families of sequences of genes or proteins, families of key 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 Mobylye or Galaxy interfaces. Tools are developed in collaboration with the GenOuest resource and data center hosted in the IRISA laboratory, including their computer facilities [\[more info\]](#).

We present here three toolboxes which each contain complementary tools with respect to their targeted sub-domain of bioinformatics.

5.1.1. Integrative Biology: (constraint-based) toolbox for network filtering

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 **A**utomatic **R**econstruction of **M**etabolic networks (AuReMe). In this workflow, we use heterogeneous sources of data with identifiers from different namespaces. MeMap (**M**etabolic network **M**apping) consists in mapping identifiers from different namespaces to a unified namespace. Then, MeMerge (**M**etabolic network **M**erge) 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 [\[57\]](#) [\[50\]](#) [\[python package\]](#)[\[web server\]](#).

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 [\[48\]](#) [\[python package\]](#).

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 [\[26\]](#) [\[web server\]](#).

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 input data. It also provides the visualization of this analysis with a friendly environment to encourage users of different disciplines to analyze their regulatory networks [\[5\]](#) [\[web server\]](#)[\[cytoscape plugin\]](#) .

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 [4] [Python package][web server].

Unifier. [*input*: sbml file with Palsson's metabolites identifiers *output*: sbml file with standard identifiers for metabolites]. This software is a Decision Support Tool to help biologists to normalize a file, containing Palsson's identifiers to refer to reactions and metabolites, using well known identifiers. Submit a list of Palsson identifiers to retrieve the corresponding database entries. Typically it maps with Metacyc identifiers but it would be used with Kegg or other databases later. A Unifier web service will be soon available.

NetWikiMaker. This tool generates (half) automatically a wiki on our reconstruction workflow. It contains information and data about the network reconstruction process such as different versions of draft metabolic networks files, parameters of tools, log files. It also displays the reactions, genes and metabolites that the workflow has found to be involved in the metabolic network, and provides a powerful search tool.

5.1.2. Dynamics and invariant-based prediction

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 search 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 ⁰. [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 [14][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 [1][matlab package].

5.1.3. Sequence annotation

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 ⁰ [9]) [software]

⁰<http://www.cellnopt.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 [49][3] [web server]

5.1.4. Integration of toolboxes and platforms in webservices

Most of our software were designed as "bricks" that can be combined through workflow application such as Mobylye. It is worth considering them into larger dedicated environments to benefit from the expertise of other research groups.

Web servers In collaboration with the GenOuest resource center, most of our tools are made available through several web portals.

- The **mobylye@GenOuest portal** is the generic web server of our resource center. It hosts the *ingranalysis*, *meneco*, *caspo*, *lombarde* and *shogun* tools [website].
- The **Mobylye@Biotempo server** is a mobylye portal for system biology with formal approaches. It hosts the *memap*, *memerge*, *meneco*, *ingranalysis*, *cadbiom* and *pogg* tools [website].
- The **GenOuest galaxy portal** now provides access to most tools for integrative biology and sequence annotation (access on demand).

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 [repository].

5.2. New tools for integrative biology

Participants: Anne Siegel [contact], Jeanne Cambefort [contact], Guillaume Collet, Damien Eveillard, Sylvain Prigent, Marie Chevallier.

The tools MeMap and MeMerge were complemented with new tools in order to analyze reference networks from literature database and to visualize the product of reconstructed metabolic networks.

Unifier. [*input*: SBML file with Palsson's metabolites identifiers *output*: sbml file with standard identifiers for metabolites]. This software is a Decision Support Tool to help biologists to normalize a file, containing Palsson's identifiers to refer to reactions and metabolites, using well known identifiers. Submit a list of Palsson identifiers to retrieve the corresponding database entries. Typically it maps with Metacyc identifiers but it would be used with Kegg or other databases later. A Unifier web service will be soon available.

NetWikiMaker. This tool generates (half) automatically a wiki on our reconstruction workflow. It contains information and data about the network reconstruction process such as different versions of draft metabolic networks files, parameters of tools, log files. It also displays the reactions, genes and metabolites that the workflow has found to be involved in the metabolic network, and provides a powerful search tool.

⁰<http://crispi.genouest.org/>

5.3. New tools for dynamics

Participants: Jérémie Bourdon [contact], Jeanne Cambefort [contact], Damien Eveillard, Anne Siegel, Nathalie Théret, Santiago Videla [contact].

In 2014, the tool caspo was extended to new functionalities.

caspo: Cell ASP Optimizer In the new version of caspo, *automated inference* of logical networks from experimental data allows for identifying admissible large-scale logic models saving a lot of efforts and without any a priori bias. Next, once a family a logical networks has been identified, one can suggest or *design new experiments* in order to reduce the uncertainty provided by this family. Finally, one can look for *intervention strategies* (i.e. inclusion minimal sets of knock-ins and knock-outs) that force a set of target species or compounds into a desired steady state. Altogether, this constitutes a pipeline for automated reasoning on logical signaling networks. Hence, the aim of caspo is to implement such a pipeline providing a powerful and easy-to-use software tool for systems biologists. [\[doc and download as a python package\]](#)[\[web server\]](#).

ESTASYS Exploratory Action

5. New Software and Platforms

5.1. The Plasma Statistical Model Checker

Participants: Axel Legay [Coordinator], Sean Sedwards, Benoît Boyer, Louis-Marie Traonouez, Kevin Corre.

5.1.1. PLASMA

Statistical model checking (SMC) is a fast emerging technology for industrial scale verification and optimisation problems. In recognition of this, our group is developing a Platform for Learning and Advanced Statistical Model checking Algorithms: PLASMA.

PLASMA (see <https://project.inria.fr/plasma-lab/>) was conceived to have high performance and be extensible, using a proprietary virtual machine [48]. Since SMC requires only an executable semantics and is not constrained by decidability, we can easily implement different modelling languages and logics. Our involvement in the DANSE⁰ and DALi⁰ European projects has also made us aware of the need to provide efficient verification for externally implemented simulators. We thus devised PLASMA-lab, a modular SMC library that allows external users to tightly integrate their own code with our efficient SMC algorithms and integrated development environment [47]. PLASMA-lab has now been successfully integrated with DESYRE⁰, Scilab⁰ and MATLAB⁰.

The PLASMA-lab architecture is now the basis of our free-standing tool,⁰ which includes all the modelling languages, logics and algorithms developed by our group. In particular, we have recently developed cutting edge algorithms for rare events [50], [49], [26], nondeterminism [28], [34], [37] and learning [14], [41].

5.2. Quail

Participants: Axel Legay [Coordinator], Fabrizio Biondi [Coordinator], Jean Quilbeuf.

Privacy is a central for Systems of Systems and interconnected objects. We propose QUAIL, a tool that can be used to quantify privacy of components. QUAIL is the only tool able to perform an arbitrary-precision quantitative analysis of the security of a system depending on private information. Thanks to its Markovian semantics model, QUAIL computes the correlation between the system's observable output and the private information, obtaining the amount of bits of the secret that the attacker will infer by observing the output. QUAIL is open source and can be downloaded at <https://project.inria.fr/quail/>.

QUAIL is able to evaluate the safety of randomized protocols depending on secret data, allowing to verify a security protocol's effectiveness. QUAIL can also be used to find previously unknown security vulnerabilities in software systems and security protocols. The tool can verify whether a protocol is protecting its secret in a perfect way, and quantify how much the secret is exposed to being revealed otherwise.

QUAIL has been used to quantify whether voting protocols respect the anonymity of the voters, proving that preference ranking voting schemes are more secure than single preference ones. It has also been applied to the security of smart grids and a number of classic examples like dining cryptographers, authentication protocols and grades protocol.

⁰<http://www.danse-ip.eu>

⁰<http://www.ict-dali.eu>

⁰<http://www.ales.eu.com>

⁰<http://www.scilab.org>

⁰<http://www.mathworks.com>

⁰<https://project.inria.fr/plasma-lab>

Since its initial release in 2013, QUAIL's algorithm has been improved employing abstract trace exploration and statistical estimation techniques, making it thousands of times faster than the initial version and outperforming other comparable analysis tools on most use cases.

5.3. PyEcdar

Participants: Axel Legay [Coordinator], Louis-Marie Traonouez [Coordinator].

One of the main difficulties with Systems of Systems is to describe the connection and interactions between the components. We propose PYECDAR as a solution to this problem. PYECDAR (<https://project.inria.fr/pyecdar/>) is a free software that analyses timed games and timed specifications. The goal of the tool is to allow a fast prototyping of new analysis techniques. It currently allows to solve timed games based on timed automata models. These can be extended with adaptive features to represent dynamicity and to model software product lines.

The tool has been originally developed to analyze the robustness of timed specifications, in extension of the tool ECDAR (<http://people.cs.aau.dk/~adavid/ecdar/>). As ECDAR, it allows to compose components specifications based on Timed I/O Automata (TIOA), and it implements timed game algorithms for checking consistency and compatibility. Additionally, it features original methods for checking the robustness of these specifications.

The tool has been later extended to analyse adaptive systems. It therefore implements original algorithms for checking featured timed games against requirements expressed in the timed AdaCTL logic.

The tool is written in Python with around 3'000 lines of code. It uses a Python console as user interface, from which it can load TIOA components from XML files written in the UPPAAL format (<http://www.uppaal.org/>), and design complex system by combining the components using a simple algebra. Then, it can analyze these systems, transform them and save them in a new XML file.

FLUMINANCE Project-Team

5. New Software and Platforms

5.1. DenseMotion software - Estimation of 2D dense motion fields

Participant: 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 and 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

Participant: 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

Participant: 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. New Software and Platforms

5.1. Next Generation Sequencing

Participants: Alexan Andrieux, Gaëtan Benoit, Charles Deltel, Erwan Drezen, Dominique Lavenier, Claire Lemaitre, Antoine Limasset, Pierre Peterlongo, Chloé Riou, Guillaume Rizk.

GATB: Genome Analysis Tool Box

The GATB software toolbox aims to lighten the design of NGS algorithms. It offers a panel of high-level optimized building blocks to speed-up the development of NGS tools related to genome assembly and/or genome analysis. The underlying data structure is the de Bruijn graph, and the general parallelism model is multithreading. The GATB library targets standard computing resources such as current multicore processor (laptop computer, small server) with a few GB of memory. From high-level API, NGS programming designers can rapidly elaborate their own software based on domain state-of-the-art algorithms and data structures. The GATB library is written in C++ and is available under the GNU Affero GPL License. [contact: D. Lavenier] <https://gatb.inria.fr>

Mapsembler: targetted assembly

The Mapsembler tool enables the micro assembly of one or several area(s) of interest. It takes as input one or more read set(s) and a one or more sequences fragments used as "starters" of each micro-assembly. This task provides a way to check the existence/absence of an area for which the user has an *a priori* interest. Moreover, for each extended "starter", the output is either a flat fasta sequence or a portion of the assembly graph. In this latter case, Mapsembler offers a visualization interface on which each graph (including the read coverage per read set) can be visualized, annotated, and manipulated. [contact: P. Peterlongo] <http://colibread.inria.fr/mapsembler2/>

Leon: NGS data compressor

Leon is a lossless compression software that achieves compression of DNA sequences of high throughput sequencing data, without the need of a reference genome. Techniques are derived from assembly principles that better exploit NGS data redundancy. A reference is built de novo from the set of reads as a probabilistic de-Bruijn graph stored in a Bloom filter. Each read is encoded as a path in this graph, storing only an anchoring kmer and a list of bifurcations indicating which path to follow in the graph. This new method will allow to have compressed read files containing its underlying de-Bruijn Graph, thus directly re-usable by many tools relying on this structure. Leon achieved encoding of a *C. elegans* reads set with 0.7 bits/base, outperforming state of the art reference-free methods. Leon is available under the GNU Affero GPL License. [contact: C. Lemaitre] <https://gatb.inria.fr/software/leon/>

Bloocoo: read corrector

Bloocoo is a k-mer spectrum-based read error corrector, designed to correct large datasets with a very low memory footprint. It uses the disk streaming k-mer counting algorithm contained in the GATB library, and inserts solid k-mers in a bloom-filter. The correction procedure is similar to state-of-the-art approaches. Bloocoo yields similar results while requiring far less memory: as an example, it can correct whole human genome re-sequencing reads at 70 x coverage with less than 4GB of memory [32]. [contact: C. Lemaitre] <https://gatb.inria.fr/bloocoo-read-corrector/>

MindTheGap: insertion variant detection

MindTheGap is a software that performs detection and assembly of DNA insertion variants in NGS read datasets with respect to a reference genome. It takes as input a set of reads and a reference genome. It outputs two sets of FASTA sequences: one is the set of breakpoints of detected insertion sites, the other is the set of assembled insertions for each breakpoint. For each breakpoint, MindTheGap either returns a single insertion sequence (when there is no assembly ambiguity), or a set of candidate insertion sequences (due to ambiguities) or nothing at all (when the insertion is too complex to be assembled). MindTheGap performs de novo assembly using the de Bruijn Graph implementation of GATB. Hence, the computational resources required to run MindTheGap are significantly lower than that of other assemblers. [contact: C. Lemaitre] <http://mindthegap.genouest.org/>

TakeABreak: de novo inversion variant discovery

TakeABreak is a tool that can detect inversion breakpoints directly from raw NGS reads, without the need of any reference genome and without de novo assembling the genomes. Its implementation is based on the Genome Assembly Tool Box (GATB) library, and has a very limited memory impact allowing its usage on common desktop computers and acceptable runtime (Illumina reads simulated at 80x coverage from human chromosome 22 can be treated in less than two hours, with less than 1GB of memory). TakeABreak is available under the GNU Affero GPL License. [contact: C. Lemaitre] <http://colibread.inria.fr/software/takeabreak/>

discoSnp: de novo SNP discovery

The discoSnp tool detects isolated SNPs given one, two or more raw read set(s) without using any reference genome. discoSnp ranks predictions and outputs quality and coverage per allele. Compared to finding isolated SNPs using a state-of-the-art assembly and mapping approach, discoSnp requires significantly less computational resources, shows similar precision and recall values, and highly ranked predictions are less likely to be false positives. [contact: P. Peterlongo] <http://colibread.inria.fr/discosnp/>

5.2. High throughput sequence comparisons

Participants: Sébastien Brillet, Erwan Drezen, Dominique Lavenier, Pierre Peterlongo, Ivaylo Petrov.

KLAST: bank-to-bank alignment search tool

KLAST is a fast, accurate and NGS scalable bank-to-bank sequence similarity search tool providing significant accelerations of seeds-based heuristic comparison methods, such as the Blast suite. KLAST is a new optimized implementation of the PLAST algorithm to which several improvements have been made in 2014. KLAST is fully designed to compare query and subject comprised of large sets of DNA, RNA and protein sequences. It is significantly faster than original PLAST, while providing comparable sensitivity to BLAST and SSearch algorithms. KLAST contains a fully integrated data-filtering engine capable of selecting relevant hits with user-defined criteria (E-Value, identity, coverage, alignment length, etc.). Klast is developed with the Korilog Company and an academic version is now freely available for the scientific community [contact: D. Lavenier]. [34] <https://koriscale.inria.fr/klast/>

COMMET: de novo comparison of metagenomic datasets

Commet is an extension of the Comparead tool that proposes to compute similarity between set of raw non assembled (and usually non-assemblable with current state of the art assemblers) reads. Commet enables to factorize computations when n read sets have to be compared all together. Moreover, Commet proposes a new representation of sub-read sets that has the main advantages to save huge disk space and to enable efficient logical operations between sub-read sets. [contact: P. Peterlongo] <https://colibread.inria.fr/software/commet/>

5.3. 3D Protein structures

Participants: Douglas Goncalves, Antonio Mucherino.

MD-jeep version 0.2

MD-jeep is the result of a strong collaboration among Antonio Mucherino, Leo Liberti, Carlile Lavor and Nelson Maculan. Over the years, PhD and postdoc students under our supervision have also been contributing to this research topic. The new method for the computation of atomic coordinates in MD-jeep v.0.2 was developed in collaboration with Douglas Soares Gonçalves [13], who was a postdoc student in Rennes for one year [contact: A. Mucherino]. <http://www.antoniomucherino.it/en/mdjeep.php>

HYBRID Project-Team

5. New Software and Platforms

5.1. OpenViBE

Participants: Anatole Lécuyer [contact], Marsel Mano, 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 30000 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 8.2.7). More information, downloads, tutorials, videos, documentations are available on the [OpenViBE website](#).

5.2. Collaviz

Participants: Thierry Duval, Thi Thuong Huyen Nguyen [contact].

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 programming 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. A collaboration with [DIVERSE team](#) intended 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.

HYCOMES Team

5. New 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 [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).

5.2. Flipflop and TnF-C++: Test and Flip Net Synthesis Tools for the Automated Synthesis of Surgical Procedure Models

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.

TnF-C++ is a robust and portable re-implementation of Flipflop, developed in 2014 and integrated in the S3PM toolchain. Both software have been designed in the context of the S3PM project on surgical procedure modeling and simulation (see section 7.1).

I4S Project-Team

5. New Software and Platforms

5.1. ISTL

Participant: Qinghua Zhang.

ISTL is a software realizing numerical computations of the inverse scattering transform for electrical transmission lines. It provides an efficient solution to experimentally determining the distributed characteristic impedance of an electrical transmission line from the reflection coefficient measured at one end of the line. Its current applications are in the fields of electrical cable fault diagnosis and of civil engineering structure monitoring. In addition to inverse scattering transform algorithms, ISTL includes a numerical simulator generating reflection coefficients of user-specified transmission lines and a graphical user interface. It is registered at Agence pour la Protection des Programmes (APP) under the number IDDN.FR.001.120003.000.S.P.2010.000.30705. See <http://people.rennes.inria.fr/Qinghua.Zhang/istl.html>.

5.2. PEGASE

Participants: Vincent Le Cam, Mathieu Le Pen, Laurent Mevel, Michael Doehler.

I4S is actually finalizing the setup of a new platform named PEGASE 2.0 as the technological successor of the previous PEGASE platform developed by IFSTTAR.

The new version of PEGASE keeps the best of its previous version in its main vocation, to be a generic high level Wireless Sensor Platform.

- What does not change between PEGASE 1 and 2.0: Based on various feedback from application fields, results from real structures monitored by PEGASE, and due to the rapid obsolescence of electronic devices, the design of the new PEGASE platform has been launched in 2013. Some of the main functions of PEGASE does not change but are reinforced.
 - Software genericity: use of a Linux embedded OS to make any application developed independently from the hardware, to make the user able to manage the system without any physical and heavy operations.
 - Hardware genericity: with a principle of daughter and mother boards, each redundant need is embedded (processing, memory, timing, GPS, energy, etc) which each pluggable daughter board implements a specific function (sensing, 3G, Ethernet, communication, signal processing and relay control).
 - Accurate time synchronization: based on an original GPS and PPS algorithm, PEGASE platform is one of the only board able to time-stamp data from sensors or any event with an accuracy of some micro-seconds Universal Time.
- What's new on PEGASE 2 platform ?

Previous principles are maintained or extended. Full electronic design from scratch occurred in 2014 to maximise its capacities in terms efficiency, cost, energy consumption, etc. Its main characteristics are

- Important software evolutions: the platform embedded a real Linux kernel (not μ Linux as previously done for memory size questions). This new kernel allows the perception of PEGASE 2 as real PC (without screen and mouse) providing important functions as MMU, software upgrade.
- Important hardware evolutions and integration

- * A very advanced GPS module (Ublox Neo 6T) to allow more accurate time synchronization (up to 100 nanoseconds)
- * An embedded energy harvesting module (and not from a daughter board as previously done) to recover energy from DC sources if available or a solar cell, while managing the low of dis/charging Lithium-Ion battery and the MMPT algorithms
- A Single Development Kit (SDK) fully (re) coded in C++ (and not C-object as proposed before) that permit a real capitalization of software developments and knowledges implemented (such as algorithms for SHM). Based on an UML model
- A major evolution in PEGASE concept consists in providing a generic web-tool to monitor PEGASE platform (whatever is the version) and many others wireless and commercial devices. A reproach addressed to previous concept resides in the fact that PEGASE was too focused on providing a generic wireless platform. But a quite big work was still necessary to monitor the devices. This new concept has been already sold to companies (such as SNCF or Cofiroute) and allows:
 - To create one independent instance of the Supervisor by client
 - Each instance of the Supervisor works 100% in the cloud with a secured access (https + login/password). Thus final users can operate it from anywhere in the world : at instrumented site level as at desk or during travel, etc.
 - For each client, the possibility to create an infinity of instrumentation projects
 - For each instrumentation project, to associate as many sensors as required by the application. Sensors can be PEGASE 1 or PEGASE 2 boards as many others : Labjack devices, some National Instruments acquisition boards, Meteo-France sources...

The list of the devices known by the Supervisor is open and is supposed, year after year, to be completed. Thus, next PEGASE projects aims at providing not only some wireless sensors platform but also a modern, full-clouded, monitoring application. Moreover the 2015 *R&D* program plans to add a very interesting function from the scientific point of view: a Matlab plug-in. The idea consists in linking the data flow managed by supervisor directly and automatically to some Matlab sources codes uploaded on the web platform. The Supervisor will compile the original Matlab files. They are dynamically compiled on an embedded Matlab runtime library on the cloud server. Thus, once the the specifications about data format are written and took into account by developers, scientist can dynamically test and operate its Matlab models uploaded on the Supervisor.

IPSO Project-Team (section vide)

KerData Project-Team

5. New Software and Platforms

5.1. Major Software

5.1.1. BlobSeer

Participants: Loïc Cloatre, 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*. This year we have mainly focused on the deployment in production of the BlobSeer software on IBM's cluster at Montpellier, in the context of the ANR MapReduce project. To this end, several bugs were solved, and several optimizations were brought to the communication layer of BlobSeer. To showcase the benefits of BlobSeer on this platform we focused on the Terasort benchmark. Currently, preliminary tests on Grid5000 with this benchmark show that BlobSeer performs better than HDFS for block sizes lower than 2 MB. We have also improved the continuous integration process of BlobSeer by deploying daily builds and automatic tests on Grid5000.

Users: Work is currently in progress in several formalized projects (see previous section) to integrate and leverage BlobSeer as a data storage back-end in the reference cloud environments: a) Microsoft Azure; b) the Nimbus cloud toolkit developed at Argonne National Lab (USA); and c) the OpenNebula IaaS cloud toolkit developed at UCM (Madrid).

URL: <http://blobseer.gforge.inria.fr/>

License: GNU Lesser General Public License (LGPL) version 3.

Status: This software is available on Inria's forge. Version 1.0 (released late 2010) registered with APP: IDDN.FR.001.310009.000.S.P.000.10700.

A *Technology Research Action* (ADT, *Action de recherche technologique*) started in November 2012 for two years, aiming at robustifying the BlobSeer software and making it a safely distributable product. This project is funded by Inria *Technological Development Office* (D2T, *Direction du Développement Technologique*). Loïc Cloatre has been hired as a senior engineer for the second year of this project, as a successor of Zhe Li, starting in February 2014.

5.1.2. Damaris

Participants: Matthieu Dorier, Orçun Yildiz, Lokman Rahmani, Shadi Ibrahim, Gabriel Antoniu.

Contact: Gabriel Antoniu.

Presentation: Damaris is a middleware for multicore SMP nodes enabling them to handle data transfers for storage and visualization efficiently. 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). Damaris 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. Version 1.0 was released in November 2014 and enables other approaches such as the use of dedicated nodes instead of dedicated cores.

Users: Damaris has been preliminarily evaluated at NCSA/UIUC (Urbana-Champaign, IL, USA) with the CM1 tornado simulation code. CM1 is one of the target applications of the Blue Waters supercomputer in production at, 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 four large-scale simulations (CM1, OLAM, Nek5000, GTC).

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.

5.2. New Software

5.2.1. *Omnisc'IO*

Participants: Matthieu Dorier, Shadi Ibrahim, Gabriel Antoniu.

Contact: Matthieu Dorier

Presentation: Omnisc'IO is a middleware integrated in the POSIX and MPI-I/O stacks to observe, model and predict the I/O behavior of any HPC application transparently. It is based on formal grammars, implementing a modified version of the Sequitur algorithm. Omnisc'IO has been used on Grid'5000 with the CM1 atmospheric simulation, the LAMMPS molecular dynamics simulation, the GTC fusion simulation and the Nek5000 CFD simulation. Omnisc'IO was subject to a publication at SC14.

Users: Omnisc'IO is currently used only within the KerData team.

URL: <http://omniscio.gforge.inria.fr/>

License: GNU Lesser General Public License (LGPL) version 3.

Status: This software is available on Inria's forge. Registration with APP is in progress.

5.2.2. *Darshan-Web*

Participants: Matthieu Dorier, Thomas Bouguet.

Contact: Matthieu Dorier

Presentation: Darshan-Web is a web interface for Darshan-Ruby, based on Ruby on Rails and AJAX technologies. It allows to navigate through many Darshan log files and display graphs on demand, directly on a web browser. A demo of Darshan-Web is available at <http://darshan-web.irisa.fr/>, which includes 2 months of logs from ANL's Intrepid supercomputer. The code of this demo is available and can be installed and used by the community.

Users: The KerData team is currently seeking potential users, in particular from Argonne National Laboratory, and will push the development further according to potential users' feedback.

URL: <http://darshan-ruby.gforge.inria.fr/>

License: GNU Lesser General Public License (LGPL) version 3.

Status: Prototype and demo available on demand.

5.2.3. *JetStream*

Participants: Radu Tudoran, Alexandru Costan, Gabriel Antoniu.

Contact: Alexandru Costan

Presentation: JetStream is a middleware solution for batch-based, high-performance streaming across cloud data centers. JetStream implements a set of context-aware strategies for optimizing batch-based streaming, being able to self-adapt to changing conditions. Additionally, the system provides multi-route streaming across cloud data centers for aggregating bandwidth by leveraging the network parallelism. It enables easy deployment across .Net frameworks and seamless binding with event processing engines such as StreamInsight.

Users: JetStream is currently used at Microsoft Research ATLE Munich for the management of the Azure cloud infrastructure.

License: Microsoft Public License.

Status: Prototype and demo available.

5.2.4. *OverFlow*

Participants: Radu Tudoran, Alexandru Costan, Gabriel Antoniu.

Contact: Alexandru Costan.

Presentation: OverFlow is a uniform data management system for scientific workflows running across geographically distributed sites, aiming to reap economic benefits from this geo-diversity. The software is environment-aware, as it monitors and models the global cloud infrastructure, offering high and predictable data handling performance for transfer cost and time, within and across sites. OverFlow proposes a set of pluggable services, grouped in a data-scientist cloud kit. They provide the applications with the possibility to monitor the underlying infrastructure, to exploit smart data compression, deduplication and geo-replication, to evaluate data management costs, to set a tradeoff between money and time, and optimize the transfer strategy accordingly.

Users: Currently, OverFlow is used for data transfers by the Microsoft Research ATLE Munich team as well as for synthetic benchmarks at the Politehnica University of Bucharest.

License: GNU Lesser General Public License (LGPL) version 3.

Status: Registration of the latest version with APP is in progress

5.2.5. *iHadoop*

Participants: Tien Dat Phan, Shadi Ibrahim.

Contact: Shadi Ibrahim

Presentation: *iHadoop* is a Hadoop simulator developed in Java on top of SimGrid to simulate the behavior of Hadoop and therefore accurately predict the performance of Hadoop in normal scenarios and under failures.

Users: *iHadoop* is an internal software prototype, which was initially developed to validate our idea for exploring the behavior of Hadoop under failures. *iHadoop* has preliminarily evaluated within our group and it has shown very high accuracy when predicating the execution time of a Map-Reduce application. We intend to integrate *iHadoop* within the SimGrid distribution and make it available to the SimGrid community.

License: GNU Lesser General Public License (LGPL) version 3.

Status: Available on Inria's forge. Registration with APP is in progress.

LAGADIC Project-Team

5. New 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 license, ViSP, an open source library available from <http://team.inria.fr/lagadic/visp>. 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 allow tracking of visual cues (dots, segments, ellipses...), 3D model-based tracking of known objects or template tracking. Simulation capabilities are also available. ViSP and its full functionalities are presented in Fig. 1 and described in [5].

This year, we continued our efforts to improve the software by ensuring the compatibility with third-party libraries that evolves a lot like CMake 3.0.0 and OpenCV 3.0.0 and by enlarging the compatibility with exotic platforms like RaspberryPi. We also fixed some issues, allowed the model-based tracker to consider circles. We introduced new bar code and face detection but also tracking capabilities. Moreover, we completely re-factored the capabilities concerning keypoint detection and matching. We improved the documentation by providing new tutorials covering the main capabilities of the software. A new release was produced in February. The source code tarball was downloaded 1000 times. With the help of the community, this release was packaged for Debian and Ubuntu 14.04. A new release is in preparation.

Concerning ROS community, all the existing packages in “vision_visp” ROS stack (see http://wiki.ros.org/vision_visp) were updated and ported to indigo 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 indoor and outdoor environments and it allows localizing in real time (45Hz) a robot or a person carrying out a mobile camera.

5.3. HandiViz software

Participants: Marie Babel [correspondant], François Pasteau.

The HandiViz software proposes a semi-autonomous navigation framework of a wheelchair relying on visual servoing. It has been registered to the APP (“Agence de Protection des Programmes”) as an INSA software (IDDN.FR.001.440021.000.S.P.2013.000.10000) and is under GPL license.

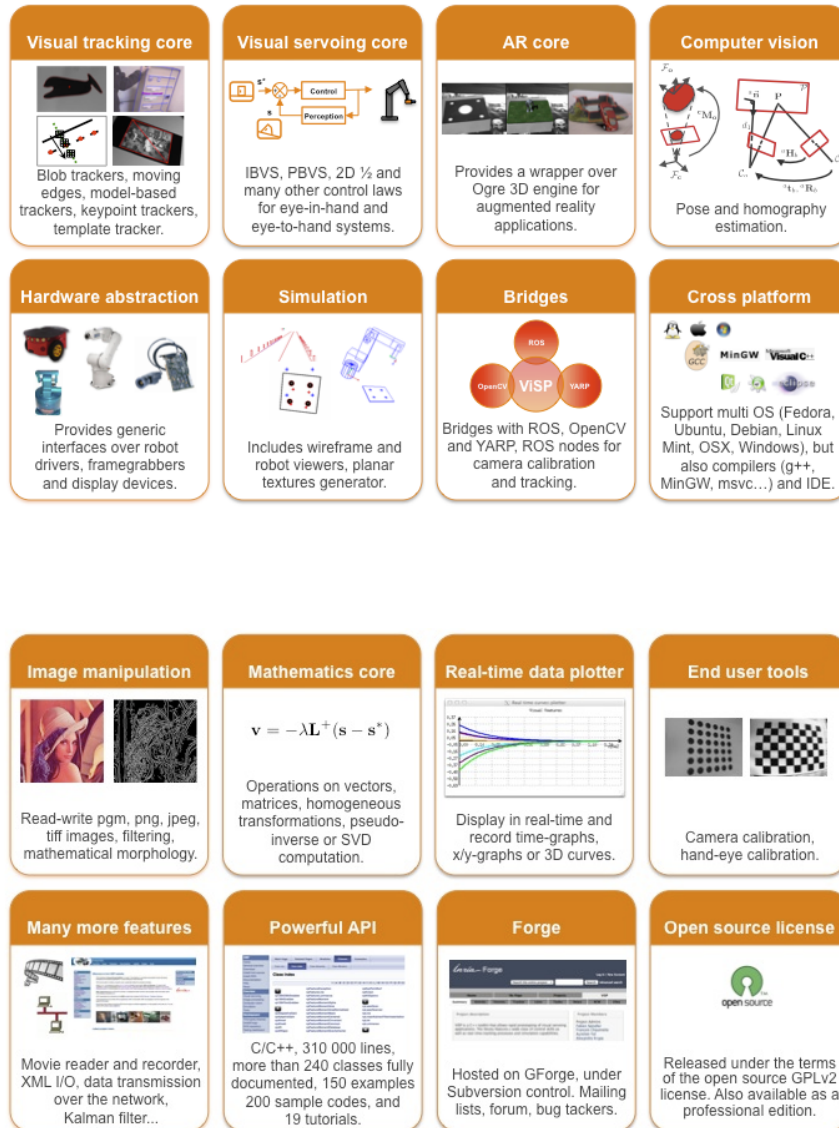


Figure 1. This figure highlights ViSP main capabilities for visual tracking, visual servoing, and augmented reality that may benefit from computer vision algorithms. ViSP allows controlling specific platforms through hardware abstraction or in simulation. ViSP provides also bridges over other frameworks such as OpenCV and ROS. All these capabilities are cross-platform. Moreover, for easing the prototyping of applications, ViSP provides tools for image manipulation, mathematics, data plotting, camera calibration, and many other features. ViSP powerful API is fully documented and available on Inria's forge as an open source software under GPLv2 license.

5.4. Platforms

5.4.1. 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.

Seven papers published by Lagadic in 2014 enclose results validated on this platform [12], [18], [21], [24], [47], [51], [52].

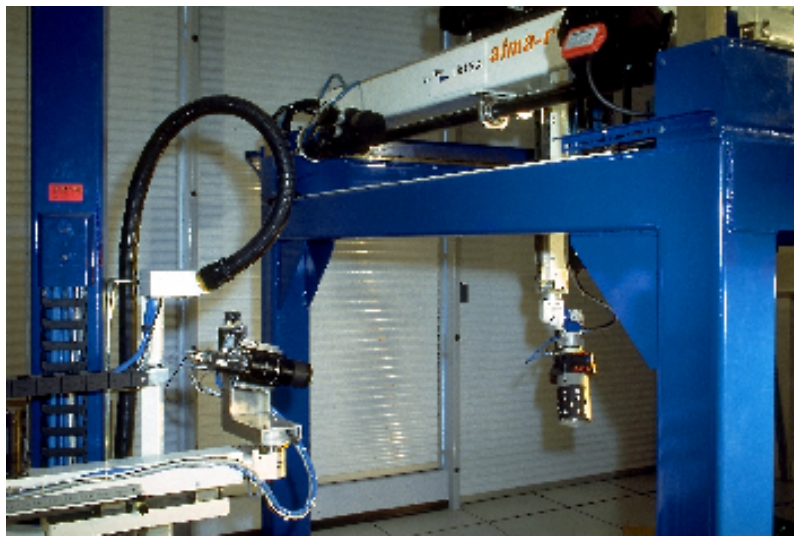


Figure 2. Lagadic robotics platforms for vision-based manipulation

5.4.2. Mobile robotics platforms

Participants: Fabien Spindler [correspondant], Erwan Demairy, Marie Babel, Patrick Rives.

5.4.2.1. Indoor 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. 3 .a), and, in Rennes, a Robotino from Festo (see Fig. 3 .b) and Pioneer 3DX from Adept (see Fig. 3 .c). 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 6.2.1), we have in Rennes a six wheel electric wheelchair from Penny and Giles Drives Technology (see Fig. 3 .d) and a five wheel electric wheelchair from You-Q (see Fig. 3 .e). 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 lets us acquire the user intention through the joystick position and control the wheelchair by applying corrections to its motion. The wheelchairs have been fitted with cameras and eleven ultrasound sensors to perform the required servoing for assisting handicapped people.

Note that eleven papers exploiting the indoors mobile robots were published this year [16], [29], [30], [31], [33], [37], [43], [41], [42], [56], [58].

5.4.2.2. *Outdoor mobile robots*

The team exploits also Cycab urban electrical cars (see Figs. 3 .f and 3 .g). 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.

Two papers published by Lagadic in 2014 enclose experimental results obtained with these outdoor mobile robots [11], [14].

5.4.2.3. *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 indoor 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:

- 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.
- demonstrate the social based navigation methods on the Hannibal platform (see Section 6.2.3)

5.4.3. *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 probe positioning, soft tissue tracking or robotic needle insertion tasks described in Section 6.5.

This platform is composed by two Adept Viper six degrees of freedom arms (see Fig. 4 .a). 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 designed an experimental setup to test an autonomous robotic needle insertion method based on visual servoing 6.5.3. The experimental setup is composed with a gelatin phantom simulating soft tissues, a flexible biopsy needle actuated by an Adept Viper arm and a 3D ultrasound probe held by the second Adept Viper arm (see Fig. 4 .b).

This year, six papers enclose experimental results obtained with this platform [13], [34], [35], [48], [49], [50].

5.4.4. *Humanoid robot*

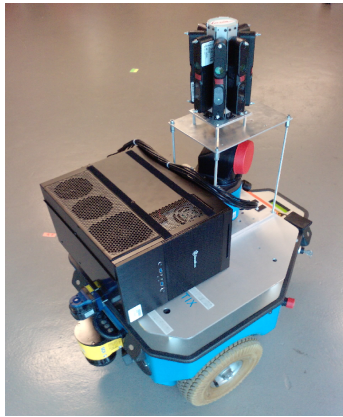
Participants: Giovanni Claudio, Fabien Spindler [correspondant].

Romeo is a humanoid robot from Aldebaran Robotics which is intended to be a genuine personal assistant and companion. In September, we were the first of the four European research laboratories that acquire a Romeo. For the moment only the upper part of the body (arms, head) is working. This research platform is now being used to validate our researches. We developed a first demonstration that make use of visual servoing and visual tracking approaches developed in the team to grasp a box and deliver it to a human (see Fig. 5).

5.4.5. *Unmanned Aerial Vehicles (UAVs)*

Participants: Fabrizio Schiano, Paolo Robuffo Giordano.

In 2014 the team also started some activities involving perception and control for single and multiple quadrotor UAVs, especially thanks to a grant from “Rennes Métropole” (see Section 8.1.4). To this end, we purchased two quadrotors from Mikrokopter GmbH, Germany (Fig. 6 .a), and one quadrotor from 3DRobotics, USA (Fig. 6 .b). These quadrotors will be used as robotic platforms for testing a number of single and multiple flight control schemes with a special attention on the use of onboard vision as main sensory modality.



(a)



(b)



(c)



(d)



(e)

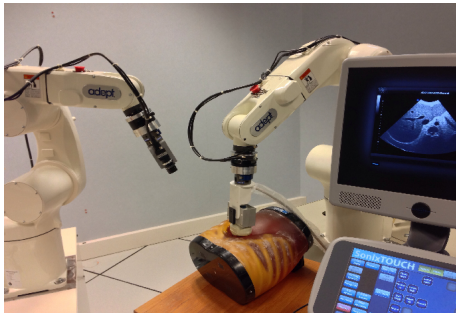


(f)



(g)

Figure 3. a) Hannibal platform, b) Robotino, c) Pioneer P3-DX robot, d) wheelchair from Penny and Giles Drives Technology, e) wheelchair from You-Q, f) Cycab available in Rennes, g) one of the Cycabs available in Sophia Antipolis.



(a)



(b)

Figure 4. a) 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 or biopsy needles. b) Robotic setup for autonomous needle insertion by visual servoing.

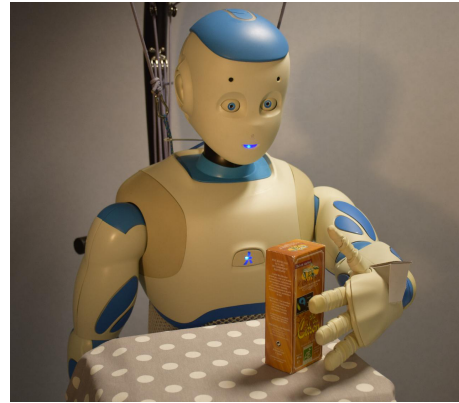


Figure 5. Romeo experimental platform.



(a)



(b)

Figure 6. a) Quadrotor XL1 from Mikrokopter, b) Quadrotor Iris from 3DRobotics

LINKMEDIA Project-Team

5. New Software and Platforms

5.1. Software

5.1.1. News and updates

5.1.1.1. Peyote

Participants: Sébastien Campion, Hervé Jégou [correspondent].

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, it is modular and as such can be used for scientific experimentation on new descriptors or indexation methods. Peyote is used in the AABOT software and was used for the Mediaeval Placing task and the Trecvid Instance Search task.

Peyote is used by Lamark, a start-up that is currently being incubating at Inria Rennes.

Last APP deposit: IDDN.FR.001.420008.001.S.A.2012.000.21000.

5.1.1.2. Yael

Participant: Hervé Jégou [correspondent].

This software is jointly maintained by Matthijs Douze, from Inria Grenoble.

Yael [30] is a C/python/Matlab library providing implementations of computationally demanding functions. In particular, the library provides very optimized functions for k-means clustering and exact nearest neighbor search. It is maintained and continuously improved. This year, we have in particular added a few tutorials implementing two simple image search systems, see http://yael.gforge.inria.fr/tutorial/tuto_imgindexing.html.

The current release (v401) was registered at APP under no IDDN.FR.001.220014.002.S.P.2010.000.10000 on July 2014.

5.2. The AllGO web services

Participants: Sébastien Campion [correspondent], Guillaume Gravier.

Available at <http://allgo.irisa.fr>, the AllGO platform allows for the easy deployment of the technology developed in the team as web services. The engineer hired by SED in October 2013 developed several new features that enable software providers to deploy autonomously their algorithm. Dedicated hardware equipment was also purchased in 2014, composed by a main server with 1.3 TB of storage and 3 nodes for computing task and setup, and will be available in 2015. In addition to a strong involvement in the development of the platform, LINKMEDIA contributed several services.

5.3. Experimental platform

Participant: Sébastien Campion [correspondent].

Our experimental platform, consisting of dedicated equipments to experiment on very large collections of multimedia data, was upgraded in 2014. In order to replace old hardware, we acquired 380 TB of additional disk space. Divided in two categories, the first part (140 TB) must replace in 2015 our current network area storage where datasets are stored. The second part (240 TB) is dedicated to a distributed storage filesystem (CEPH), used to store our experimental results with high IO performances for use with the mesocluster IGRIDA.

MIMETIC Project-Team

5. New Software and Platforms

5.1. Populate

Participants: Carl Jorgensen, Fabrice Lamarche [contact].

Populate is a toolkit dedicated to task scheduling under time and space constraints in the field of behavioral animation. It is currently used to populate virtual cities with pedestrian performing different kind of activities implying travels between different locations. However the generic aspect of the algorithm and underlying representations enable its use in a wide range of applications that need to link activity, time and space. The main scheduling algorithm relies on the following inputs: an informed environment description, an activity an agent needs to perform and individual characteristics of this agent. The algorithm produces a valid task schedule compatible with time and spatial constraints imposed by the activity description and the environment. In this task schedule, time intervals relating to travel and task fulfillment are identified and locations where tasks should be performed are automatically selected.

The software provides the following functionalities:

- A high level XML dialect that is dedicated to the description of agents activities in terms of tasks and sub activities that can be combined with different kind of operators : sequential, without order, interlaced. This dialect also enables the description of time and location constraints associated to tasks.
- An XML dialect that enables the description of agent's personal characteristics.
- An informed graph describes the topology of the environment as well as the locations where tasks can be performed. A bridge between TopoPlan and Populate has also been designed. It provides an automatic analysis of an informed 3D environment that is used to generate an informed graph compatible with Populate.
- The generation of a valid task schedule based on the previously mentioned descriptions.

With a good configuration of agents characteristics (based on statistics), we demonstrated that tasks schedules produced by Populate are representative of human ones. In conjunction with TopoPlan, it has been used to populate a district of Paris as well as imaginary cities with several thousands of pedestrians navigating in real time.

MYRIADS Project-Team

4. New Software and Platforms

4.1. ConPaaS

Contact: Guillaume Pierre, Guillaume.Pierre@irisa.fr

URL: <http://www.conpaas.eu/>

Status: Version 1.4.2

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, Dzenan Softic, Genc Tato, Teodor Crivat.

Impact: ConPaaS is recognized as one of the major open-source PaaS environments. It is being developed by teams in Rennes, Amsterdam, Berlin and Ljubljana. Technology transfer of ConPaaS technology is ongoing in the context of the MC-DATA EIT ICT Labs project.

4.2. HOCL-tools

Contact: Cédric Tedeschi, Cédric.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 higher 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 so as to support 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): Matthieu Simonin, Cédric Tedeschi, Javier Rojas Balderrama.

Impact: The compiler is used as a tool within the team to develop HOCL programs. The decentralized workflow execution support has been extensively used to produce results published and presented at several conferences. It is also used in the framework of the DALHIS⁰ associated team, as a workflow template executor, integrated with the TIGRES workflow manager developed at the Lawrence Berkeley National Lab. It is supported by the GinFlow ADT funded by Inria.

4.3. Merkat

Contact: Nikolaos Parlavantzas, Nikolaos.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 [49], [48] [3]. 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. The development was initiated in the framework of Stefania Costache PhD's thesis.

Active contributors (from the Myriads team): Stefania Costache, Christine Morin, Nikolaos 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.

4.4. Meryn

Contact: Nikolaos Parlavantzas, Nikolaos.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 [51]. The current Meryn prototype was implemented using shell scripts, 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 was developed in the framework of Djawida Dib's PhD thesis [10].

Active contributors (from the Myriads team): Djawida Dib, Christine Morin, Nikolaos Parlavantzas.

Impact: Meryn is not yet distributed as open source.

4.5. Resilin

⁰<http://project.inria.fr/dalhis>

Contact: Christine Morin, Christine.Morin@inria.fr

URL: <http://resilin.inria.fr>

Status: Version 1.0

License: GNU Affero GPL

Presentation: Resilin [6] 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, Céline Merlet, Christine Morin, Nikolaos Parlavantzas, Matthieu Simonin.

Impact: Resilin is being used in the MOAIS project-team at Inria Grenoble - Rhône Alpes.

4.6. Snooze

Contact: Christine Morin, Christine.Morin@inria.fr

URL: <http://snooze.inria.fr>

Status: Version 2.1.5

License: GPLv2

Presentation: Snooze [53], [52], [54] [4] is 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 piece of 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 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 language libraries (*libcloud*, *jeloud* ...). Snooze also provides a web interface to control the system.

Snooze was used as a building box for two internships projects during the summer of 2014. The EC2 interface was used to execute Hadoop jobs configured by Resilin software. As a result we show that (1) the EC2 interface was expressive enough to work with a higher level tool and (2) the control over Snooze allow a better placement of data chunks for Hadoop jobs which leads to a better reliability

of the execution of the different jobs. The second internship topic took part in a collaboration with the Northeastern University of Boston. The goal was to build a *Checkpoint as a Service* system. The service allows users to execute their computations in a cloud environment in a reliable way. Periodic checkpoints are saved making it possible to restore the computation from a previous state in case of failures. This work is described in [31].

Active contributors (from Myriads team): Jiajun Cao, Gene Cooperman, Eugen Feller, Yvon Jégou, David Margery, Christine Morin, 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. Snooze development was supported by the Snooze ADT funded by Inria from October 2012 to September 2014.

4.7. Virtual Execution Platform (VEP)

Contact: Yvon Jégou, Yvon.Jegou@inria.fr

URL: <http://project.inria.fr/vep/>

Status: Version 2.2

License: BSD

Presentation: Virtual Execution Platform

(VEP) [57] is a Contrail (<http://contrail-project.eu>) 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 OpenStack IaaS frameworks were initially supported. During the VEP-S EIT ICT Labs activity in 2014, VEP was extended with a new OCCI IaaS driver which allows to control any IaaS framework providing a standard OCCI API. Support for the new OCCI SLA proposition from OGF has also been added and allows to represent the VEP CEEs in a standard format. Finally, during this activity, the Zabbix open source distributed monitoring system was integrated to VEP.

Active contributors (from Myriads project-team): Roberto-Gioacchino Cascella, Florian Dudouet, Filippo Gaudenzi, Yvon Jégou, Christine Morin, Arnab Sinha.

Impact: VEP is part of Contrail software stack. External users can experiment with it using the open testbed operated by Myriads team. Technology transfer of VEP technology is ongoing in the context of the VEP-S EIT ICT Labs activity.

PANAMA Project-Team

5. New Software and Platforms

5.1. 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 development was jointly achieved by the PAROLE team in Nancy and the TEXMEX team in Rennes through an Inria funded ADT (Action de Développement Technologique). PANAMA contributed to the development by coordinating and performing user tests, and to the dissemination in a Show-and-Tell ICASSP poster [58].

While the first implementation was in Matlab, the new implementation is in C++ (for core functions), with Matlab and Python user scripts. Version 2, including speedup and new features was released in 2014 and can be downloaded from <http://bass-db.gforge.inria.fr/fasst/>.

SAGE Project-Team

5. New Software and Platforms

5.1. Platforms

5.1.1. Platform H2OLab

Participants: Jean-Raynald de Dreuzy, Jocelyne Erhel [correspondant], Grégoire Lecourt, 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 7.2.1). 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>.

5.2. Hydrogeology

5.2.1. GRT3D

Participants: Édouard Canot, Jocelyne Erhel [correspondant].

- Version: version 2.0, April 2014
- 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 and C2S@EXA partners (see sections 7.2.5 , 7.2.3), in particular Andra. 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.

5.2.2. 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 5.2.2 .

5.2.3. GENFIELD

Participants: Jean-Raynald de Dreuzy, Jocelyne Erhel, Grégoire Lecourt, Géraldine Pichot [correspondant].

- Version: version 1.0, December 2014
- Programming language: C++
- Abstract: GENFIELD allows the generation of log-normal correlated fields. It is based on a spectral method and uses the FFTW library. Parallelism is implemented using MPI communications. GENFIELD is used in hydrogeology to model natural fields, like hydraulic conductivity or porosity fields.
- Current work: paper about algorithms 6.4.7 .

5.3. High Performance Scientific Computing

5.3.1. 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.

5.3.2. MUESLI

Participant: Édouard Canot [corresponding author].

Muesli is a library designed to help in coding scientific problems in Fortran using a vector-oriented syntax like Matlab. One of its aims is therefore to speed-up the development process. It contains all the necessary materials to work numerically with a dynamic array (dynamic in size, shape, type, and storage structure), called mfArray. Muesli includes all or some parts of the following numerical libraries: Blas and Lapack, Arpack, Minpack, Slatec, Sparskit, SuiteSparse, Metis, Triangle, RngStreams, and other routines based on ACM algorithms.

The key points of Muesli is to efficiently solve large ODE/DAE systems (which come from, e.g., PDE problems after using the method of lines) or large non-linear minimization problems (where Jacobian matrices can be provided in a sparse format). The user can easily monitor the whole integration process and have access to tools to fix the singularity of the system of equations.

The latest version of Muesli is 2.9.5 (2014-10-03). More information can be found at: <http://people.irisa.fr/Edouard.Canot/muesli>.

5.3.3. Zohour

Participant: Édouard Canot [correspondant].

Zohour is a node-based adaptive 2D mesh algorithm, written in Fortran 2003. A basic rectangular, regular set of nodes is recursively refined. Then the cells come from the Voronoi tessellation. While the domain is currently limited to a rectangular shape, its strength is three-fold:

- first, computing the flux via a Finite Element or Finite Volume method is both simple and accurate because each cell-side of cells is the bisection of two nodes;
- second, the transition between zones of different levels of refinement is more progressive than other methods, leading to a smaller number of nodes for the whole mesh;
- third, during successive refinements when dealing with a transient problem, interpolation is needed only by the new nodes, limiting the numerical errors.

It is planned for use in the HeMaTiS code (5.4.1) in order to get a refined mesh zone around the phase change surface.

See also the web page <http://people.irisa.fr/Edouard.Canot/zohour>.

5.4. Heat diffusion in soils

5.4.1. HeMaTiS

Participants: Édouard Canot [correspondant], Salwa Mansour.

HeMaTiS (**H**eat and **M**ass **T**ransfer in **S**oils) is a set of Finite Volume programs (variants concern different geometrical configurations: 1D, 1D-radial, 2D, 3D-axisymmetric) for computing the transient heat diffusion in soils when there is a phase change of water. Currently, the soil is modelled by a heterogeneous porous medium having constant thermo-physical properties, and the porous medium is saturated with water. The phase change is treated by means of the Apparent Heat Capacity method. In the near future, we plan to use an unsaturated model (but limited to small water content), and an effective thermal conductivity which depends on the local humidity (this latter law may reveal hysteresis behaviour). The software is written in Fortran 95 and is based on the Muesli library (5.3.2). A Computer Algebra System (Maple or Maxima) is used to compute the Jacobian matrix.

5.4.2. TPIP

Participants: Édouard Canot [correspondant], Salwa Mansour.

TPIP (**T**hermal **P**roperties by **I**nverse **P**roblem) is a program which aims at estimating the thermo-physical of a saturated porous medium after a strong heating which leads to the phase change of the water contained in the pores, knowing the experimental heating curves history at few selected points. The least-square criterion is used, in which sensitivity coefficients are the solution of a huge, complex PDE system in order to take into account the phase change of water. These equations for the sensitivity coefficients are therefore obtained via a Computer Algebra System (Maple or Maxima). In many aspects, the forward problem is similar to the HeMaTiS code (5.4.1), and like it, is based on Muesli (5.3.2). Two different minimization algorithms may be used, Damped Gauss-Newton or Levenberg-Marquardt. A special procedure has been applied in order to obtain a robust convergence, by changing some parameters of the forward problem during the iterations.

5.4.3. GLiMuH

Participants: Édouard Canot [correspondant], Salwa Mansour.

The GLiMuH code (**G**rains with **L**iquid **M**eniscus **u**nder **H**eating) is devoted to the understanding of how heat diffuses in an assembly of solid grains separated by air and water. In the pendular regime, the quantity of water is very small, leading to liquid bridges between the grains. In the current approximation, the grains are spherical in shape, and the numerical simulation is done in a 3D axisymmetric coordinate system. The shape of the liquid/gas interface is computed by integrating a differential algebraic system of equations, with a given quantity of water per unit volume of the porous medium, and under the constraint of a given contact angle between the liquid/gas interface and the solid boundaries. The numerical results allow us to estimate the effective thermal conductivity of a real wet granular medium, which is required to establish more realistic models for the HeMaTiS code (5.4.1).

SERPICO Project-Team

5. New Software and Platforms

5.1. Software for live cell imaging

Participants: Charles Kervrann [(contact)], Patrick Bouthemy, 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) [48]. 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: Mobylye@SERPICO <http://mobylye-serpico.rennes.inria.fr/cgi-bin/portal.py#forms::Motion2D>.

Collaborator: 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) [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: Mobylye@SERPICO <http://mobylye-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 ...

Commercial licence agreements: Innopsys, Roper Scientific, Photometrics, Nikon (2015).

Collaborators: Jérôme Boulanger and Jean Salamero (UMR 144 CNRS-Institut Curie, STED team), Peter Elbau (RICAM Linz, Austria) and Jean-Baptiste Sibarita (UMR 5091, University of Bordeaux 2).

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) 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 [35].

On-line demo: Mobylye@SERPICO <http://mobylye-serpico.rennes.inria.fr/cgi-bin/portal.py#forms::Hullkground>.

Collaborators: Anatole Chessel and Jean Salamero (UMR 144 CNRS-Institut Curie, STED team).

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://equip.es.igdr.univ-rennes1.fr/en/tips/Software/TubuleJ/>.

Collaborators: Sophie Blestel and Denis 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 [19]. On real data, this segmentation method extracts the most contrasted regions of microtubules, and 3D visualization is improved.

Collaborators: Sophie Blestel and Denis Chrétien (UMR 6290, CNRS-University of Rennes 1).

5.3. Image Processing software distribution and Mobylye platform

Participants: Tinaherinantenaina Rakotoarivelo, Thierry Pécot [(contact)], Charles Kervrann.

Mobylye@SERPICO (guest)
set email | sign-in | activate | sign-out
refresh workspace

SERPICO team (INRIA Rennes - Bretagne Atlantique) is partner of France-BioImaging

Welcome | Forms | Data Bookmarks | Jobs | Tutorials

Welcome to Mobylye, a portal for bioinformatics analyses

Space time RePresentation, Imaging and cellular dynamics of molecular Complexes

Programs available

- o **Backwarping:** Warp sequence with parametric motion model
- o **CRFMovingSpotDetection:** Detecting moving spots/vesicles using Conditional Random Fields
- o **HotSpotDetection:** Robust detection of fluorescence accumulation over time in video-microscopy
- o **Hullkground:** Separation of moving and non moving part in a sequence
- o **KLTracker:** Track vesicle and POI in image sequences
- o **Motion2D:** Estimate 2D parametric motion model
- o **MS-Detect:** Detecting moving objects in image sequences by background subtraction
- o **ND-SAFIR:** Denoise N-Dimensional images
- o **Optical-flow:** Compute Optical Flow between 2 images
- o **OpticalFlowStack:** Compute Optical Flow between each pair of images in a TIFF stack

Credits
Mobylye is a platform developed jointly by the Institut Pasteur Biology IT Center and the Ressource Parisienne en Bioinformatique Structurale. More information about this project can be found here.

SERPICO FRANCE-BIOIMAGING Inria

Figure 2. Mobylye@SERPICO web portal.

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>).
- **Mobylye@SERPICO web portal:** An on-line version of the image processing algorithms has been developed using the Mobylye framework (Institut Pasteur, see <http://mobylye.pasteur.fr/>). The main role of this web portal (see Fig. 2) is to demonstrate the performance of the programs developed by the team: C-CRAFT[13], ATLAS[23], HOTSPOTDETECTION[51], HULLKGROUND[35], KLTRACKER[50], MOTION2D[49], MS-DETECT[37], ND-SAFIR[4] and OPTICALFLOW. The web interface makes our image processing methods available for biologist users at Mobylye@SERPICO

(<http://mobyale-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 [35], MOTION2D [49], HOTSPOTDETECTION [51]. The C-CRAFT algorithm [13] has been developed for the image processing ICY platform (<http://icy.bioimageanalysis.org/>).
- *Institut Curie CID iManage database*: The microscopy facility of Institut Curie has co-developped a commercial database system (CID iManage/Strand Avadis company). The database can be searched via meta-data and includes menu selections that enable to run remote processing from a cluster. We have integrated ND-SAFIR and HULLKGROUND in the interface environment to allow the database users to process their images easily, and store associated results and parameters used.

Collaborators: Charles Deltel (Inria Rennes SED) and Perrine Paul-Gilloteaux (UMR 144 CNRS-Institut Curie, STED team and PICT-IBiSA).

SIROCCO Project-Team

5. New 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 a 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 exemplar-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. The developments have been pursued in 2014, in particular by introducing a Poisson blending step in order to improve the visual quality of the inpainted video. 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).

SUMO Project-Team

5. New 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 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 (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). 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. 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](#). This year, the source code of DAXML has been submitted at the APP, and a distribution with free ad-hoc licence will follow in 2015.

TACOMA Team

5. New Software and Platforms

5.1. THEGAME

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 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 TACOMA since June 2013.

TASC Project-Team

5. New Software and Platforms

5.1. Platforms

5.1.1. CHOCO

Participants: Nicolas Beldiceanu, Jean-Guillaume Fages, Xavier Lorca [correspondant], Thierry Petit, Charles Prud'Homme [main developer], Rémi Douence.

CHOCO is a Java discrete constraints library integrating within a same system *explanations*, *soft constraints* and *global constraints* (90000 lines of source code). In 2014 developments were focusing on the following aspects:

- For second consecutive year, **CHOCO** has participated at the **MiniZinc Challenge**, an annual competition of constraint programming solvers. In competition with 16 other solvers, **CHOCO** has won three bronze medals in three out of four categories (Free search, Parallel search and Open class).
- Five versions have been released all year long, the last one (v3.3.0, Dec. 17th) has the particularity to be promoted on **Maven Central Repository**. The major modifications were related to a simplification of the API but also improvement of the overall solver.
- A User Guide is now available: 164 pages describing how to use **CHOCO**, together with a new **website**.
- Finally, **Charles Prud'homme** and Jean-Guillaume Fages, the main contributors of **CHOCO**, have defended their Phd, publishing at the same time their work in the source code. In particular, an extension of **CHOCO** now provides support for constraints involving graph variables.

5.1.2. IBEX

Participants: Ignacio Araya, Clément Carbonnel, Gilles Chabert [correspondant], Benoit Desrochers, Luc Jaulin, Bertrand Neveu, Jordan Ninin, Ignacio Salas Donoso, Gilles Trombettoni.

IBEX (Interval-Based EXplorer) is a C++ library for solving nonlinear constraints over real numbers. The main feature of Ibex is its ability to build solver/paver strategies declaratively through the contractor programming paradigm. It also comes with a black-box solver and a global optimizer.

In 2014 the work on IBEX has focused on the following points.

- Global optimizer:
 - Rigorous mode in the global optimizer (certification of the feasibility of strict equality constraints for the minimum found). This includes Newton-based inflation iteration, Hansen test for underconstrained systems (see Global Optimization using Interval Analysis, E. Hansen, 1992).
 - Unconstrained local search algorithm (quasi-Newton method with trust regions).
 - Rejection test based on first-order conditions (see First Order Rejection Tests For Multiple-Objective Optimization, A. Goldsztejn et al. [42]).
 - Multiple selection technique in exploration (see A new multisection technique in interval methods for global optimization, L.G. Casado, Computing, 2000)
- Contractors:
 - Existentially-quantified constraints, (see Contractor Programming, [8]).
 - Mohc contractor, (see Exploiting Monotonicity in Interval Constraint Propagation, I. Araya et al., [41]).

- Q-intersection, (see Q-intersection Algorithms for Constraint-Based Robust Parameter Estimation, C. Carbonnel et al., AAAI 2014, [27]).
- Contractor based on pixel maps (started in Oct 2014, still in progress, see Using set membership methods for robust underwater robot localization, PhD, J. Sliwka).
- Miscellaneous
 - Everyday code improvement (around 400 commits in 2014).
 - Symbolic processing features (symbol occurrence splitting, function construction from strings, progress in differentiation with vector/matrix operations).
 - numerous bug fixes (especially in the inner arithmetic routines).

5.1.3. Global Constraint Catalog

Participants: Nicolas Beldiceanu [correspondant], Mats Carlsson, Sophie Demassey, 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://sofdem.github.io/gccat/> (at item *working version*) containing 431 constraints, 4070 pages and 1000 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 focusing on:

1. maintaining the content of the catalogue,
2. making more easy the navigation within the pdf version,
3. continuing the redesign of the figures using **TikZ**: 200 figures were converted and 100 figures remain to be converted, and adding new illustrations (150 figures).
4. updating the web version of the catalogue (see <http://sofdem.github.io/gccat/>).

5.1.4. AIUR

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. The current system allows both on-line and off-line learning.

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 github.com/AIUR-group/AIUR. AIUR finished 4th to *StarCraft*tm AI competitions organized at the conferences AIIDE 2014 and CIG 2014.

5.1.5. GHOST

Participant: Florian Richoux [correspondant].

GHOST (General meta-Heuristic Optimization Solving Tool) is a template C++11 library designed for *StarCraft: BroodWartm*, under the terms of the GNU GPL v3 licence and is about 7500 lines long. GHOST implements a meta-heuristic solver aiming to solve any kind of combinatorial and optimization RTS-related problems represented by a CSP/COP [36]. The solver handles dedicated geometric and assignment constraints in a way that is compatible with very strong real time requirements. The source code as well as documentation pages are available at github.com/richoux/GHOST.

This framework is a deep extension of an ad-hoc solver. Although GHOST has been developed recently (during Summer 2014), it got itself quickly noticed by a French video-game developing company. We are starting discussion about a technology transfer of GHOST.

TEA Project-Team

5. New Software and Platforms

5.1. The Eclipse project POP

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

The distribution of project POP⁰ is a major achievement of the ESPRESSO project. The Eclipse project POP is a model-driven engineering front-end to our open-source toolset Polychrony. It was finalised in the frame of project OPEES, as a case study: by passing the POLARSYS qualification kit as a computer aided simulation and verification tool. This qualification was implemented by CS Toulouse in conformance with relevant generic (platform independent) qualification documents. Polychrony is now distributed by the Eclipse project POP on the platform of the POLARSYS industrial working group. Team TEA aims at continuing its dissemination to academic partners, as to its principles and features, and industrial partners, as to the services it can offer.

Technically, project POP is composed of the Polychrony toolset, under GPL license, and its Eclipse framework, under EPL license.

The Polychrony toolset. 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.

⁰Polychrony on POLARSYS (POP), an Eclipse project in the POLARSYS Industry Working Group, 2013. <https://www.POLARSYS.org/projects/POLARSYS.pop>

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.

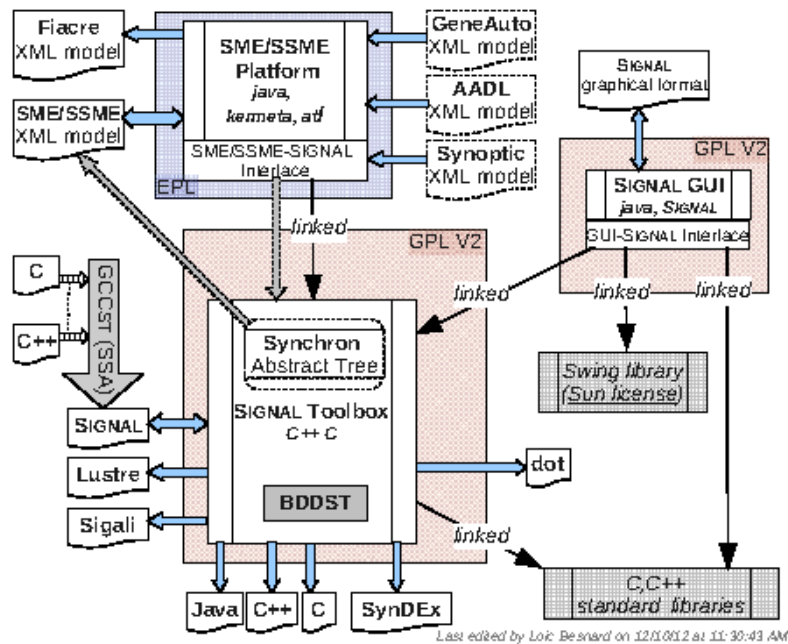


Figure 1. The Polychrony toolset high-level architecture

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.

The Eclipse POP Framework. 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 ⁰: all Signal operators (e.g. arithmetic, clock synchronization), model (e.g. process frame, module), and construction (e.g. iteration, type declaration).

⁰SIGNAL V4-Inria version: Reference Manual. Besnard, L., Gautier, T. and Le Guernic, P. <http://www.irisa.fr/espresso/Polychrony>, 2009

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

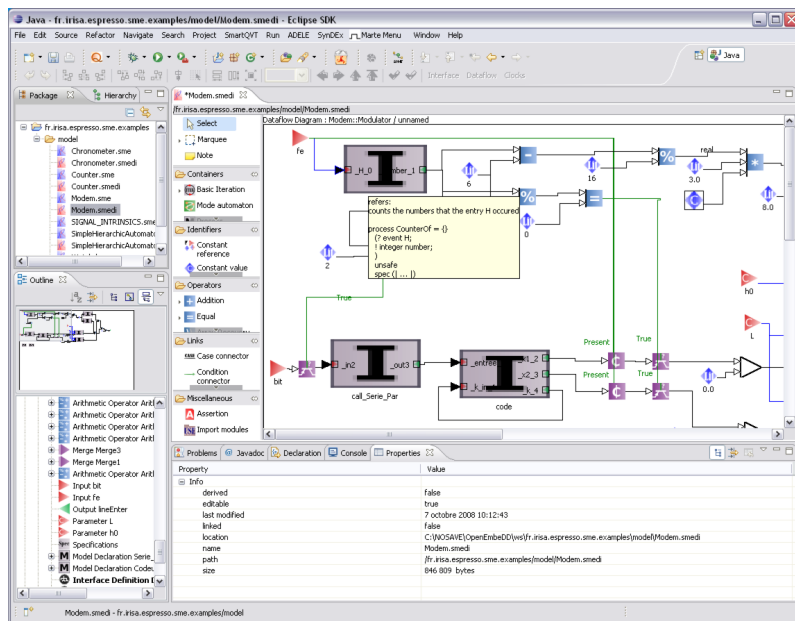


Figure 2. The Eclipse POP Environment

It also provides a graphical modelling 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 modelling 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 modelling in Signal, we split the modelling 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 site⁰. 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 Foundation 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.

5.2. Integrated Modular Avionics design using Polychrony

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

⁰Polychrony Update Site for Eclipse plug-ins. <http://www.irisa.fr/espresso/Polychrony/update>, 2009.

The Apex interface, defined in the ARINC standard ⁰, 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). 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 [4] 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.

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

Participants: Adnan Bouakaz, Thierry Gautier, Jean-Pierre Talpin.

We have proposed a dataflow design model [2] of SCJ/L1 applications ⁰ 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 ⁰ 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.

⁰ARINC Report 651-1: Design Guidance for Integrated Modular Avionics. Airlines Electronic Engineering Committee, 1997

⁰Safety critical Java technology specification. JSR-302, Year = 2010

⁰Static checking of safety critical Java annotations. Tang, D. Plsek, A. and Vitek, J. International Workshop on Java Technologies for Real-Time and Embedded Systems, 2010

VISAGES Project-Team

5. New Software and Platforms

5.1. Shanoir

Participants: Justine Guillaumont, Michael Kain, Yao Yao, Christian Barillot.

Shanoir (Sharing NeuroImaging Resources) is an open source neuroinformatics platform designed to archive, structure, manage, visualize and share neuroimaging data with an emphasis on multi-centric collaborative research projects (Figure 2). It provides a user-friendly interface, a secure web access and offers an intuitive workflow to facilitate the collecting and retrieving of neuroimaging data from multiple sources and a wizard to make the completion of metadata easy. Shanoir comes along many features of neuroimaging data management systems along with research-oriented data imaging organization and enhanced data accessibility, support multi-centers clinical studies on subjects or group of subjects and other functionalities such as anonymization of data. 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.

Shanoir APP registration number is: IDDN.FR.001.520021.003.S.A.2008.000.31230

See also the web page <http://www.shanoir.org>

- Keywords: neuroimaging, ontology, sharing neuroimages
- Version: 0.5
- 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/Middleware: Windows, Mac et Linux.
- Required library or software: Java 1.6, JBoss server, JBoss Seam, JSF, JPA Hibernate, EJB, Rich-faces, Faceless, Ajax4JSF, Dcm4chee.
- Programming language: Java / J2EE
- Documentation: see the website

5.2. ShanoirUploader

Participants: Justine Guillaumont, Michael Kain, Christian Barillot.

The ShanoirUploader (Fig. 3) is a desktop application on base of JavaWebStart (JWS). The application can be downloaded and installed using an internet browser. It 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 application 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://shanoir.gforge.inria.fr> as the ShanoirUploader documentation is integrated on this page.

- Keywords: neuroimaging, ontology, sharing neuroimages
- Version: 0.1
- 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 license for the moment
- Type of human computer interaction: desktop application on base of JavaWebStart (JWS), web service (SOAP messages based)
- OS/Middleware: Linux, Windows and Mac
- Required library or software : Java SDK, installed on client machine
- Programming language: Java
- Documentation : see the website

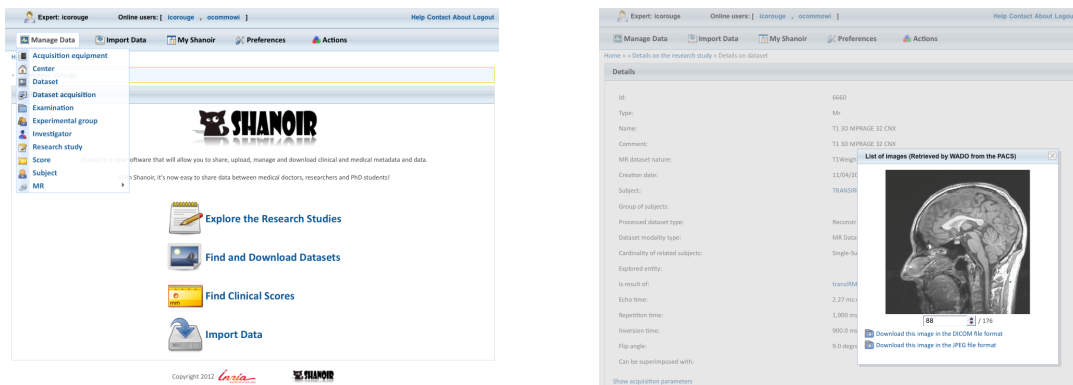


Figure 2. The SHANOIR software is a web application to share, archive, search and visualize neuroimaging data.

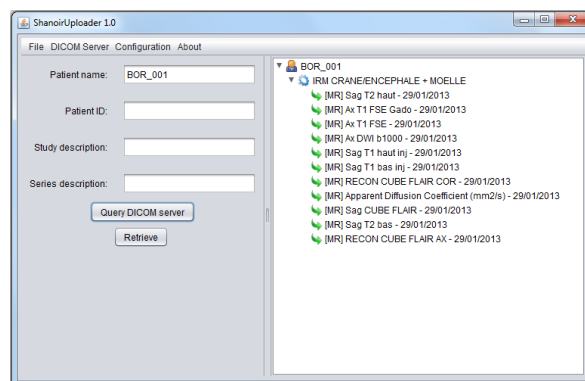


Figure 3. The ShanoirUploader software is a desktop application designed to interact with a PACS to query and retrieve the data stored on any PACS.

5.3. iShanoir

Participants: Michael Kain, Christian Barillot.

iShanoir (Fig. 4) is an iOS application, designed for iPhone and iPad. On base of this application a Shanoir server can be accessed. For this the Shanoir SOAP web-services are called. iShanoir can be used to access and navigate in the data tree structure, stored on a Shanoir server. iShanoir displays as well additional meta data corresponding to the data entities in the tree structure. On base of these informations image files (NIFTI and DICOM) can be selected and downloaded on a local iPhone/iPad in a temporary cache. From this cache the files can be opened and displayed with a corresponding viewer, the user already has to have installed on his device. This project is the result of the internship of H el ene G erome in the team. An APP registration is in progress.

See also the web page <http://shanoir.gforge.inria.fr> as the iShanoir documentation is integrated on this page.

- Keywords: neuroimaging, ontology, sharing neuroimages
- Version: 0.1
- Software benefit: offers access to data stored on a Shanoir server from native iOS devices, like iPhones and iPads
- License: no defined license for the moment
- Type of human computer interaction: mobile iOS Cocoa Touch application with web service connection
- OS/Middleware: iOS
- Required library or software: none
- Programming language: Objective-C
- Documentation : see the website

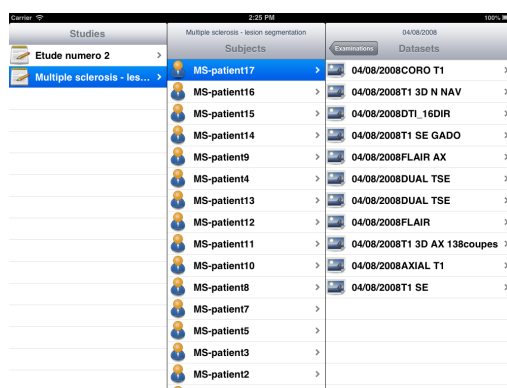


Figure 4. The iShanoir software is a desktop application designed to...

5.4. AutoMRI

Participants: Fang Cao, Isabelle Corouge, Pierre Maurel, Elise Bannier.

AutoMRI Based on MATLAB and the SPM8 toolbox, autoMRI provides complete pipelines to pre-process and analyze various types of images (anatomical, functional, 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 normalization 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 (both pulsed or pseudo-continuous sequences) 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 5).

- 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/>

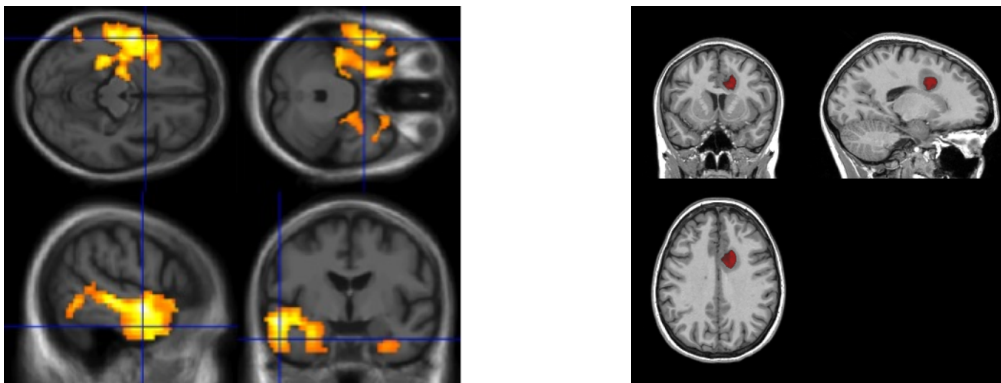


Figure 5. Illustrations of results obtained with autoMRI: Conjunction map showing areas of hypoperfusion and hypometabolism in semantic dementia (right), Detection of relaxometry defect in an MS patient (left).

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 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 2.2.1 has been released in September 2014 for the main distribution platforms. medInria core API source code is also released under a BSD license.

See also Figure 6 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/Middleware: Windows, Mac et Linux.
- Required library or software : Qt, DTK, ITK, VTK.
- Programming language: C++

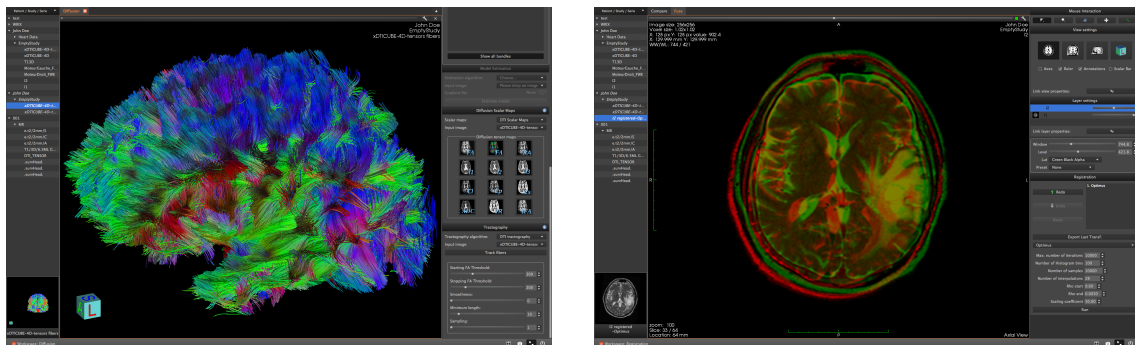


Figure 6. The medInria software platform: Fused view of registered images (right), Tractography overlapped with 3D image (left)

5.6. Anima

Participants: Fang Cao, Laurence Catanese, Olivier Commowick, René-Paul Debroize, Florent Leray, Renaud Hédouin, Guillaume Pasquier.

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 simple 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++

5.7. Integration of EEG and fMRI

Participants: Marsel Mano, Lorraine Perronnet.

Related to the project Hemisfer there have been development of new functions, scripts and demos for the acquisition and processing of the EEG and fMRI data in Real-time. These include:

- Functions for fMRI header info reader, volume reader, motion correction, slice time correction nifty output conversion, real time fMRI initialization, real time fMRI processing, z-score calculation, volume smoother, alignment, etc., functions for real time EEG data acquisition, filtering, power calculation and display.
- Scripts for various protocols used in offline fMRI experiments, real time processing loop for EEG and fMRI.
- Demo for real time acquisition of the EEG and fMRI data, demo for real time processing efficiency of the fMRI data, demo for the real time processing of EEG data, real time z-Score for fMRI data.
- Several small aux functions for I/O interfaces (e.g. com, serial)

In the current stage the prototype also relies on various other free toolboxes (e.g. SPM, pnet)

- Keywords: medical imaging, EEG, fMRI
- Software benefit: integration of EEG and fMRI processing
- Type of human computer interaction: C++ API, shell scripts
- OS/Middleware: Windows, Mac and Linux.
- Required library or software : SPM, pnet.
- Programming language: C++, shell scripts

5.8. Platforms

5.8.1. The Neurinfo Platform

VISAGES is the founding actor of a new experimental research platform which was installed in August 2009 at the University Hospital of Rennes. The University of Rennes 1, Inria, Inserm for the academic side, and the University Hospital of Rennes and the Cancer Institute “Eugene Marquis“ for the clinical side, are partners of this neuroinformatics platform called NeurINFO (<http://www.neurinfo.org>). This platform has been supported under the “Contrat de Projets Etat-Région” (C. Barillot is the PI) and has received a total amount of 5.1 Meuros for the period 2007–2013. European (FEDER), National (through Ministry of research, Inria, Inserm and ANR) and local councils (Brittany Region, Ille et Vilaine, and Rennes Metropolis) have joined their effort to support this operation for a total amount of 5070 keuros (600keuros for the infrastructures, 3670keuros for the equipments and 800keuros for the functioning). This application was set up through the Regional PIMATGI initiative coordinated by INSERM in Brittany (C. Roux). The overall PIMATGI initiative served for the financing of three distinct, but complementary, platforms: NeurINFO, TheraFONC as a technical platform dedicated to therapy guided by functional imaging especially in the oncology domain (Inserm U 650 - LaTIM, Dir. Ch. Roux, Brest), and TherA-Image as a platform dedicated to image guided mini-invasive surgery and therapy especially in the domain of cardio-vascular diseases (U 642 -LTSI, Dir. L. Senhadji, Rennes).

Concerning the NeurINFO Platform, the activity domain is a continuum between methodological and technological research built around specific clinical research projects. The ambition is to do innovation in science, technology and medical technology transfer for the implementation on the clinical field. On the medical field, the translational research domain mainly concerns medical imaging and more specifically the clinical neurosciences. Among them are multiple sclerosis, epilepsy, neurodegenerative, neurodevelopmental and psychiatric diseases, surgical procedures of brain lesions, neuro-oncology and radiotherapy planning. Beyond these CNS applications, the platform is also open to alternative applications. Neurinfo ambitions to support the emergence of research projects based on their level of innovation, their pluri-disciplinarity and their ability to foster collaborations between different actors (public and private research entities, different medical specialties, different scientific profiles). In this context, a new research 3T MRI system (Siemens Verio system) was

acquired in summer 2009 in order to develop the clinical research in the domain of morphological, functional, structural and cellular in-vivo imaging. In 2014 a new equipment for simultaneous recording of EEG and MRI images has been acquired from Brain Product. Visages and its partners in the Neurinfo project are committed to use this new research platform for developing new regional, national and international collaborations around fundamental and applied clinical research projects dealing with in-vivo medical imaging. In 2014, the two engineers running the platform (Elise Bannier and Isabelle Corouge), members of the Visages team, moved from temporary employment contracts to open-ended research engineers contracts.