1. ACES Project-Team (section vide) ................................................................. 4
2. ALF Project-Team .................................................................................. 5
3. ASAP Project-Team ................................................................................ 8
4. ASCOLA Project-Team .......................................................................... 10
5. ASPI Project-Team (section vide) ....................................................... 13
6. ATLANMOD Team ............................................................................. 14
7. CAIRN Project-Team .......................................................................... 20
8. CELTIQUE Project-Team .................................................................. 25
9. CIDRE Project-Team .......................................................................... 26
10. DIONYSOS Project-Team ................................................................. 27
11. DISTRIBCOM Project-Team ............................................................ 28
12. DREAM Project-Team ..................................................................... 29
13. ESPRESSO Project-Team .................................................................. 31
14. FLUMINANCE Project-Team ............................................................. 35
15. I4S Team .......................................................................................... 37
16. IPSO Project-Team (section vide) .................................................... 38
17. KERDATA Team ................................................................................ 39
18. LAGADIC Project-Team ................................................................... 41
19. METISS Project-Team ...................................................................... 45
20. MIMETIC Team ................................................................................. 47
21. MYRIADS Team ............................................................................... 49
22. S4 Project-Team ................................................................................ 53
23. SAGE Project-Team .......................................................................... 54
24. SERPICO Team .................................................................................. 58
25. SYMBIOSE Project-Team .................................................................. 61
26. TASC Project-Team ........................................................................... 64
27. TEMICS Project-Team ...................................................................... 66
28. TEXMEX Project-Team .................................................................... 69
29. TRISKELL Project-Team ................................................................... 74
30. VERTECS Project-Team ................................................................... 76
31. VISAGES Project-Team ..................................................................... 78
32. VR4I Team .......................................................................................... 83
ACES Project-Team (section vide)
5. Software

5.1. Panorama

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

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

5.2. ATMI

Participant: Pierre Michaud.

Contact: Pierre Michaud

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

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

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

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

5.3. STiMuL

Participant: Pierre Michaud.

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

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

STiMuL can be used to model steady-state temperature in 3D circuits consisting of several layers of different materials. STiMuL is based on a rigorous solution to the Laplace equation [9]. The number and characteristics of layers can be defined by the user. The boundary conditions can also be defined by the user. In particular, STiMuL can be used along with thermal imaging to obtain the power density inside an integrated circuit. This power density could be used for instance in a dynamic simulation oriented temperature modeling such as ATMI.

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

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

Participant: Pierre Michaud.

Contact: Pierre Michaud

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

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

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

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

5.5. HAVEGE

Participant: André Seznec.

Contact: André Seznec

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

An unpredictable random number generator is a practical approximation of a truly random number generator. Such unpredictable random number generators are needed for cryptography. HAVEGE (HArdware VoIatile Entropy Gathering and Expansion) is a user-level software unpredictable random number generator for general-purpose computers that exploits the continuous modifications of the internal volatile hardware states in the processor as a source of uncertainty [16]. HAVEGE combines on-the-fly hardware volatile entropy gathering with pseudo-random number generation.

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

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

Visit http://www.irisa.fr/alf/HAVEGE or contact André Seznec.

5.6. Tiptop

Participant: Erven Rohou.

Status: Registered with APP (Agence de Protection des Programmes). Available under GNU General Public License v2.
Tiptop is a new simple and flexible user-level tool that collects hardware counter data on Linux platforms (version 2.6.31+). The goal is to make the collection of performance and bottleneck data as simple as possible, including simple installation and usage. In particular, we stress the following points.

- Installation is only a matter of compiling the source code. No patching of the Linux kernel is needed, and no special-purpose module needs to be loaded.
- No privilege is required, any user can run `tiptop` — non-privileged users can only watch processes they own, ability to monitor anybody’s process opens the door to side-channel attacks.
- The usage is similar to `top`. There is no need for the source code of the applications of interest, making it possible to monitor proprietary applications or libraries. And since there is no probe to insert in the application, understanding of the structure and implementation of complex algorithms and code bases is not required.
- Applications do not need to be restarted, and monitoring can start at any time (obviously, only events that occur after the start of `tiptop` are observed).
- Events can be counted per thread, or per process.

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

For more information, please contact Erven Rohou.
### 5. Software

#### 5.1. WhatsUp: A Distributed News Recommender

**Participants:** Antoine Boutet, Davide Frey, Arnaud Jegou, Anne-Marie Kermarrec.

**Contact:** Antoine Boutet

**Licence:** Open Source

**Presentation:** A Distributed News Recommender

**Status:** Beta version

This work has lead to the development of WhatsUp, a distributed recommendation system aimed to distribute instant news in a large scale dynamic system. WhatsUp has two parts, an embedded application server in order to exchange with others peers in the system and a fully dynamic web interface for displaying news and collecting opinions about what the user reads. Underlying this web-based application lies Beep, a biased epidemic dissemination protocol that delivers news to interested users in a fast manner while limiting spam. Beep is parametrized on the fly to manage the orientation and the amplification of news dissemination. Every user forwards the news of interest to a randomly selected set of users with a preference towards those that have similar interests (orientation). The notion of interest does not rely on any explicit social network or subscription scheme, but rather on an implicit and dynamic overlay capturing the commonalities between users with respect to they are interested in. The size of the set of users to which a news is forwarded depends on the interest of the news (amplification). A centralized version of WhatsUp is already up and running and the decentralized one is still in beta version.

#### 5.2. GossipLib: effective development of gossip-based applications

**Participants:** Davide Frey, Heverson Ribeiro, Anne-Marie Kermarrec.

**Contact:** Davide Frey

**Licence:** Open Source

**Presentation:** Library for Gossip protocols

**Status:** released version 0.7alpha

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 as thus also provides 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 simply by changing one line in a configuration file.

#### 5.3. YALPS

**Participants:** Davide Frey, Heverson Ribeiro, Anne-Marie Kermarrec.

**Contact:** Davide Frey

**Licence:** Open Source

**Presentation:** Library for Gossip protocols

**Status:** released version 0.3alpha
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 feature 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 without a single change in the code. 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 work has been done in collaboration with Maxime Monod (EPFL).

5.4. HEAP: Heterogeneity-aware gossip protocol.

Participants: Davide Frey, Arnaud Jegou, Anne-Marie Kermarrec.
Contact: Davide Frey
Licence: Open Source
Presentation: Java Application
Status: release & ongoing development

This work has been done in collaboration with Vivien Quéma (CNRS Grenoble), Maxime Monod and Rachid Guerraoui (EPFL), and has lead to the development of a video streaming platform based on HEAP, Heterogeneity-Aware gossip Protocol. The platform is particularly suited for environment characterized by heterogeneous bandwidth capabilities such as those comprising ADSL edge nodes. HEAP is, in fact, able to dynamically leverage the most capable nodes and increase their contribution to the protocol, while decreasing by the same proportion that of less capable nodes. During the last few months, we have integrated HEAP with the ability to dynamically measure the available bandwidth of nodes, thereby making it independent of the input of the user.
5. Software

5.1. AWED

Participants: Mario Südholt [correspondent], Ismael Mejia.

The model of Aspects With Explicit Distribution (AWED) supports the modularization of crosscutting functionalities of distributed applications. It addresses the problem that common aspect systems do not provide features for distributed programming. It notably features three main aspect abstractions: remote pointcuts, remotely-executed advice, and distributed aspects.

This year a gray-box model for distributed composition has been built and implemented based on the AWED model using the notion of invasive distributed patterns (see Sec. 6.1 ). Furthermore, the resulting model has been applied to the evolution of grid applications and OpenMRS, an open-source health information system. The AWED system has also been employed in the CESSA project proposal (see Sec. 8.1 ) as a basis for our work on the secure evolution of service-oriented architectures. Finally, the development of a new, more modular, implementation of the AWED system has started in 2011.

AWED is available at http://awed.gforge.inria.fr .

5.2. ECaesarJ, EJava and EScala

Participants: Jacques Noyé [correspondent], Angel Núñez, Jurgen Van Ham.

ECaesarJ is a language developed in the context of the European project AMPLE, as joint work with the Technische Universität Darmstadt (TUD). The basic objective was to provide support for directly mapping the high-level features defined by a software product line onto implementation-level features, beyond standard feature-oriented programming. But the language has much wider applications. ECaesarJ can actually be seen as a language which smoothly integrates Object-Oriented Programming, Feature-Oriented Programming, Aspect-Oriented Programming, and Event-based Programming.

It is an extension of Java with virtual classes and propagating mixin composition (as its ancestor CaesarJ, developed at TUD), but also declarative events and state machines. Unlike AspectJ, ECaesarJ does not include a class-like concept of aspect. Instead, it deals with pointcuts and pieces of advice as (implicit) events and event handlers, which are standard class members. This makes it possible to use standard inheritance to reuse and refine them. Explicit events can also be used when events must be explicitly triggered as in traditional event-based programming. Finally, in the same way as pointcuts can be composed using logical operators, declarative events can be defined as a composition of other events.

This provides a symmetric version of AOP where virtual classes can be used to deal with structural aspects whereas events can be used to deal with behavioral aspects.

In ECaesarJ, a class can also include, as class members, state transitions. Combining this with virtual classes makes it possible to define, at the programming language level, refinable hierarchical state machines. The combination of state machines and events provides, in particular, effective language support for the State design pattern as well as a form of Event-based AOP.

EJava and EScala are more recent developments of the same ideas applied to Java and Scala, respectively. EJava benefits from Java tooling with an eclipse plugin developed with the Spoofax Language Workbench. Unlike EJava and ECaesarJ, EScala makes it possible to dynamically register and unregister event handlers. It also benefits from a more efficient, compiler-based, implementation. As ECaesarJ, EScala is joint work with TUD.

Prototype implementations of these languages are available through http://ecaesarj.gforge.inria.fr/.
5.3. Entropy

Participants: Jean-Marc Menaud [correspondent], Fabien Hermenier, Adrien Lèbre, Hien Nguyen Van, Rémy Pottier, Thomas Chavrier, Guillaume Le Louët.

Entropy is a virtual machine (VM) manager for clusters. The current prototype acts as an infinite control loop, which performs a globally optimized placement according to cluster resource usage, scheduler objectives and administrative rules.

Relying on an encapsulation of jobs into VMs, Entropy enables the implementation of finer scheduling policies through cluster-wide context switches, i.e., permutations of VMs present in the cluster. It thus supports a more flexible use of cluster resources and frees end-users from the burden of dealing with time estimates.

The major advantage of the Entropy system concerns the cluster-wide context switch process itself. Entropy computes a new viable configuration and an optimized reconfiguration plan. This plan describes the sequences of transitions to perform (i.e. the run, migrate, suspend/resume, stop VM operations) in order to transit from the current situation to the new one. As the cost of each action and the dependencies between them is considered, Entropy reduces the duration of each cluster-wide context switch by performing a minimum number of actions in the most efficient way.

Around this solution, we developed VMScript, a domain-specific language for administration of virtualized grid infrastructures. This language relies on set manipulation and is used to introspect physical and virtual grid architectures, thanks to query expressions, and notably to modify VM placement on machines. VMScript interacts with Entropy and can be used to define administrative placement rules.

In 2011, Entropy has been integrated into a product of a newly founded start-up EasyVirt (see Sec. 7.1 ). Entropy has also been tested or used by our partners Orange Labs, DGFiP (direction Générale des Finances Publiques), Bull, MACIF, Logica.

Entropy is available under the LGPL license at http://entropy.gforge.inria.fr/.

5.4. FPath and FScript

Participants: Thomas Ledoux [correspondent], Frederico Alvares.

FPath and FScript are two domain-specific languages (DSLs) dealing respectively with the navigation and the dynamic reconfiguration of Fractal architectures. FPath is a DSL for querying Fractal architectures. It is restricted to the introspection of architectures by browsing elements identified by their properties or location in the architecture. This focused domain allows FPath to offer a very concise and readable syntax and ensures correctness properties by construction (e.g. any query terminates in a finite time). FScript is a DSL dedicated to the reconfiguration of Fractal component architectures. It enables reconfiguration scripts to modify a Fractal architecture. Like FPath, FScript guarantees several properties by construction, e.g. termination of scripts by excluding the possibility of infinite loops. Moreover the FScript interpreter supports a transactional model of reconfigurations and the preservation of ACID properties.

An adaptation of FPath/FScript to FraSCAti, a component framework providing runtime support for the Service Component Architecture (SCA), has been developed by the INRIA Adam project-team. In that way, software architects are able to navigate using FPath notation through FraSCAti architectures and to reconfigure them with FScript. We have used this adaptation in our recent work [22] for reconfiguring cloud applications in order to reduce the energy footprint in cloud infrastructures.

FScript and its extensions are available under the LGPL license at http://fractal.ow2.org/fscript.

5.5. WildCAT

Participants: Thomas Ledoux [correspondent], Frederico Alvares.
WildCAT is a generic Java framework for context-aware applications. It permits the monitoring of large-scale applications by allowing developers to easily organize and access resources through a hierarchical organization backed with a powerful SQL-like language to inspect sensors data and to trigger actions upon particular conditions. WildCAT proposes two modes to inspect the resources: a pull mode relies on synchronous communication and a push one relies on asynchronous communication. In the pull mode, developers programmatically get and set attributes. In the push mode, developers register listeners on queries expressed over the events generated by the backend.

WildCAT has been developed by the team in the last years. We have used WildCAT in our recent work [22] for allowing cloud applications to listen events notification fired by the cloud infrastructure (e.g. whenever the pricing policy of cloud resources changes) or to detect changes on the application activity (e.g. to detect whenever the number of requests/s sharply increases/decreases) in order to launch the reconfiguration of cloud applications.

WildCAT is available under GPL v2 at http://wildcat.ow2.org.
ASPI Project-Team (section vide)
5. Software

5.1. The ATL Model Transformation Language

URL: http://www.eclipse.org/m2m/atl/

With an eye on the normative work of the OMG (MOF, OCL, QVT, etc.), a new conceptual framework has been developed based on a second generation model transformation language called ATL. Although ATL influenced the OMG standard, the approach is more general as discussed in [8]. In 2004 IBM gave an Eclipse innovation award to the ATL project. In 2007 Eclipse recognized ATL as one central solution for model transformation and promoted it to the M2M project (see Eclipse.org/m2m). There are more than 200 industrial and academic sites using ATL today, and several Ph.D. thesis in the world are based on this work.

In 2011 we started a new evolution phase for ATL. Our mid-term plan is making of ATL the leading solution for building autonomous reactive transformation systems, i.e. transformation networks that can autonomously manage a set of dataflows among the application models.

Following this line, we implemented in [43] 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.

In [31] 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 enable also transformations that generate infinite target models, extending the application space of the model-transformation paradigm.

The latest evolution of the ATL engine is a full reactive version, able to activate the minimal computation for responding to updates or request on the involved models. This engine is studied to scalably support large ATL networks. In this line we also introduced an algorithm for simplifying ATL transformation chains.

5.2. MoDisco (Model Discovery)

URL: http://www.eclipse.org/modisco/

MoDisco is an open source Eclipse project that provides a generic and extensible framework dedicated to the elaboration of Model Driven Reverse Engineering (MDRE) solutions. Gathering contributions from both academics and industrials, the goal of the project is to federate common efforts in the model-based transformation of legacy software systems implemented using different technologies (e.g.: Java, COBOL, C). The first principle is to discover models out of legacy artifacts, representing appropriately all the relevant information, to be then used as part of reverse engineering processes for software understanding, evolution or modernization. Targeted scenarios include software (technical or architectural) migration of large legacy systems, but also retro-documentation, refactoring, quality assurance, etc. Within this context, MoDisco has collaborations with the OMG Architecture Driven Modernization (ADM) Task Force, for which the project provides several reference implementations of its standards: Knowledge Discovery Metamodel (KDM), Software Measurement Metamodel (SMM) and Abstract Syntax Tree Metamodel (ASTM).
The MoDisco framework is composed of a set of Eclipse plugins, and relies on the de-facto standard Eclipse Modeling Framework (EMF) for model handling. Thanks to its modular architecture, it allows completely covering the three steps of a standard MDRE approach: 1) Discovery (i.e. extracting a complete model of the source code), 2) Understanding (i.e. browsing and providing views on this model for a given purpose) and 3) Transformation (evolving the model towards a new technology, architecture, etc). More specifically, as part of its Infrastructure layer, MoDisco offers the set of generic (i.e.; legacy technology-independent) reusable components really useful to build the core of MDRE solutions: Discovery Manager and Workflow for MDRE task orchestration, Model Browser for advanced navigation in complex models, model extension and customization capabilities for understanding (e.g.; views definition), etc. As part of its Technologies layer, it provides an advanced support for the Java, JEE and XML technologies, including complete metamodels, corresponding model discoverers, transformations, code generators, customizations, query libraries, etc.

MoDisco (or some of its components) is being used by different partners including other academics, industrials (e.g.; Sodifrance on several of their real modernization projects for their customers) or Eclipse projects (e.g.; Eclipse-MDT Papyrus as developed by CEA). Moreover, the Eclipse-EMFT EMF Facet project has been initiated as a MoDisco spin-off, in order to externalize some features which are not actually specific to reverse engineering problems and thus may be reused in many different contexts (cf. corresponding EMF Facet section).

5.3. EMFtoCSP

URL: http://code.google.com/a/eclipselabs.org/p/emftocsp/

EMFtoCSP is a tool for the verification of precisely defined conceptual models and metamodels. For these models, the definition of the general model structure (using UML or EMF) is supplemented by OCL constraints. The Eclipse Modeling Development Tools (MDT ²) 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 it is 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 problem of constraint logic programming and employs the Eclipse CLP solver ³ to solve it. This way, constraint propagation is exploited to tackle the (generally NP-hard) search.

² http://www.eclipse.org/modeling/mdt/?project=ocl
³ http://eclipseclp.org/
The tool is a continuation of the UMLtoCSP approach [52] developed previously by Jordi Cabot, Robert Clariós and Daniel Riera. It provides a generic plugin framework for Eclipse to solve OCL-annotated models using constraint logic programming. Apart from already supported Ecore and UML metamodels, further metamodels can be added easily in the future. Similarly, other constraint solving back-ends can be integrated. It is provided under the Eclipse Public License.

5.4. AMW (AtlanMod Model Weaver)

URL: http://wiki.eclipse.org/AMW

AMW is an open source Eclipse project, under the Eclipse Public License (EPL), that provides a generic and extensible tooling dedicated to model weaving. It can be used to establish, represent and manage any kind of links (correspondences) between elements coming from different models (or metamodels). These links are stored in separated weaving models, allowing them to be exchanged and reused in various contexts using different modeling techniques. Thus, weaving models are actually relevant in many concrete application scenarios where several models are involved: (model driven) tool interoperability, mapping definition, transformation specification, traceability, model annotation, model merging, model comparison, etc.

The AMW tooling is composed of several Eclipse plugins, and relies on the de-facto standard Eclipse Modeling Framework (EMF) for model handling. The provided model weaving workbench comes notably with a base weaving editor, reusable matching algorithms and weaving model serialization capabilities. These components are fully generic, so that they can be extended and adapted to any specific weaving metamodel (defining any kind of weaving links). AMW also offers a connector allowing to both use and produce weaving models as part of model-to-model transformations written in ATL.

AMW is being used by more than 40 user sites, including research labs and major companies (NASA, BAE, Versata, Mia-Software, Obeo, etc.). Currently part of the Eclipse-GMT project (to be terminated in the coming months), AMW is now in the process of being migrated into the Eclipse-M2M ATL project.

5.5. AM3 (AtlanMod MegaModel Management)

URL: http://wiki.eclipse.org/index.php/AM3

The AtlanMod Megamodel Management tool offers several functionalities for modeling in the large [3], i.e. for handling several related models (either terminal models, metamodels or transformation models) used as part of a complex modeling project.

The main component in AM3 is a generic megamodel manager that allows the user to browse and manipulate a set of related models. This manager knows the semantic relations between all these models. These relations are often associated to a given weaving model allowing not only navigating the traces between models, but also the traces between model elements. Since the links are stored externally as weaving models, the participating models do not get polluted and may be used as they are. Furthermore it is possible to handle multiple traceability chains going through similar models.

AM3 provides also a textual domain-specific language for model management called MoScript (URL: http://wiki.eclipse.org/MoScript). With MoScript, users can automate model management tasks by means of textual scripts written in an extension of the OCL language. For instance, user may write queries (based on model content, structure, relationships, and behaviour derived through on-the-fly simulation) to retrieve models from model repositories, manipulate them (e.g., by running transformations on sets of models), and store them back in the repository. MoScript also allows to populate and update the megamodel automatically by doing reverse engineer of simple modeling artifact repositories.

The generic tool for megamodel management has been used by different partners for several use cases like operationalization of chains of transformations.
5.6. Virtual EMF (Model Virtualization)

URL:  http://code.google.com/a/eclipselabs.org/p/virtual-emf/

Virtual EMF is an Eclipse plugin built on top of EMF that enables the creation and manipulation of virtual models, i.e., models whose elements do not contain concrete data, but are rather proxies to elements contained in other models. The idea is related to that of model composition, as it aims capturing the (often overlapping) concepts a set of models as one single global model. This is a frequently faced problem as, in complex scenarios, modelers often have to deal with a large number of heterogeneous and interrelated models, and most times the view a specific kind of user requires does not correspond to any of these models, but is a combination of cross-domain information scattered among several ones.

Current composition techniques rely on the materialization of the composed model, an approach that poses some important limitations in terms of (i) efficiency, as they do not scale (the data duplication mechanism they use implies in extra memory usage and time-consuming generation of the composed model), (ii) synchronicity, as updates in the composed model are not propagated to the original ones (or vice-versa), thus losing consistency, or even (iii) interoperability, as in some cases the composed model requires a specific API/tool to be handled.

Virtual EMF allows overcoming the limitatins above. A virtual model provides to tools/user the illusion of working with a regular model whereas, in fact, all model access and manipulation requests are transparently redirected to its set of virtualized models. It serves as a centralized and transparent access point to a set of interconnected models, allowing users to easily compose, weave and link them, thus providing the following beneficial properties:

- **Interoperability**: it behaves as a normal model. Therefore, compatibility with existing EMF-based solutions/tools (e.g. models transformations, model editors, ...) is guaranteed;
- **Synchronisation**: changes are automatically and transparently propagated between virtual and original models;
- **Scalability**: support for very big models;
  - low memory usage: no data duplication, direct access to original model elements;
  - faster generation time: no need for (time-consuming) information cloning operations (e.g. executing a model transformation);
- **Genericity**: support for several types of inter-model relationships (e.g. merge, association, filter) and extension capabilities for their semantics.

Virtual EMF is available as an open-source project on Eclipse Labs. It has been presented in a talk in EclipseCon Europe 2011 and contributed by the AtlanMod team to the CESAR project.

5.7. EMF Facet

URL:  http://www.eclipse.org/modeling/emft/facet/

EMF Facet is an open source Eclipse project, under the Eclipse Public License (EPL), that provides a generic and extensible framework dedicated to the dynamic and non-intrusive extension of models. It can be used to extend already existing metamodels with additional concepts and properties, the corresponding models being then transparently augmented, reduced or modified accordingly at runtime. Such a metamodel extension is called a facet, and can be specified on top of any metamodel in EMF Ecore. The underlying mechanism is based on the runtime execution of queries on the models corresponding to the faceted metamodels. Facets are notably particularly relevant for obtaining different views on existing models without having to actually alter them with any extra data.
The EMF Facet framework is composed of several Eclipse plugins, and relies on the de-facto standard Eclipse Modeling Framework (EMF) for model handling. The facet definitions are stored as facet models, allowing them to be exchanged and reused in various contexts. The queries can be implemented using any suitable query language (e.g.; ATL, OCL, Java, XPath), as far as the corresponding adaptors exist and are correctly registered within the framework. The proposed tooling includes dedicated editors for creating, editing and saving both facet and query definitions, the implemented support for Java, OCL and ATL queries, a Table Editor for visualizing query results. An advanced support for the model display customization (e.g.; icons, colors, fonts) is also provided as part of the framework.

EMF Facet is currently intensively used in MoDisco for extracting and displaying different specific views from large models of legacy systems. Its extension and customization capabilities are actually integrated into several MoDisco components, such as notably the MoDisco Model Browser. However, different other integration possibilities will be also explored in the future.

5.8. Portolan (Model-Driven Cartography)
URL: http://code.google.com/a/eclipselabs.org/p/portolan/

Processing large amounts of data to extract useful information is an essential task within companies. To help in this task, visualization techniques have been commonly used due to their capacity to present data in synthesized views, easier to understand and manage. However, achieving the right visualization display for a data set is a complex cartography process that involves several transformation steps to adapt the (domain) data to the (visualization) data format expected by visualization tools. With its simple core principles and set of base generic techniques (metamodeling, model transformation, model weaving, etc), Model-Driven Engineering (MDE) provides the relevant support for bridging the gap between data sets and visualization tools and thus for designing and implementing Cartography solutions. The proposed Portolan prototype is a concrete illustration of both a model-based and model-driven Cartography platform. Thus, the objective of Portolan is to facilitate the identification of interoperability solutions between tools by:

1. discovering (at least semi-automatically) maps of given situations in terms of deployed tools and relationships between them;
2. easily navigating and editing these maps;
3. augmenting or specializing them with both manually-entered and computed information;
4. visualizing them, using different customizable ways, in order to facilitate their understanding.

To this aim, the Portolan platform integrates:
- a set of default DSLs like GraphML, KML, Excel;
- visual displays based on Prefuse, Google Maps;
- modeling tools such as ATL language, Ecore modeler, etc.

It includes also an extension mechanism allowing the tool customization for advanced users. This recently developed generic tooling for cartography has already been used during the first action of our collaboration with BNP Paribas, as well as in the context of the IDM++ project.

5.9. The AmmA ToolBox

ATL, AMW, TCS, MoDisco, and AM3 are among the most important Eclipse.org components produced by the AtlanMod team. However there are also other components and a lot of functionalities, examples, and use cases made available and necessary to express solutions to many problems. The whole set of contributions composes the AmmA platform.
5.10. Industrialization strategy for research prototypes

Research labs, as priority innovation providers, are also indirectly key actors of the Software Engineering market. However, even if they already initiate the promotion of many new innovations to the industry, an important collaborative effort is still needed in order to actually transfer the corresponding techniques or technologies from the research lab to the company. Based on the AtlanMod concrete experience with the previously mentioned open source tools/projects, we have extracted a pragmatic approach [37] for transforming the results of scientific experimentation into practical industrial solutions.

While dealing with innovation, this approach is also innovation-driven itself, as the action is actually conducted by the research lab via a technology transfer. Three different partners are directly involved in this process, using open source as the medium for maintaining a constant interaction between all of them:

- **Use Case Provider.** Usually a company big enough to have to face real complex industrial scenarios which need to be solved (at least partially) by applying new innovative principles and techniques;
- **Research Lab.** Usually a group from a research institute (public or private) or university evaluating the scientific relevance of the problems, identifying the research challenges and prototyping possible solutions;
- **Technology Provider.** Usually a small or medium company, with a particular technical expertise on the given domain or Software Engineering field, building and delivering the industrial version of the designed solutions;

From our past and current experience, three main characteristics of this industrialization business model can be highlighted:

- **Win-win situation.** Each partner can actually focus on its core activity while also directly benefiting from the results obtained by the others (notably the research lab can continue to do research);
- **Application-driven context.** The end-user need is at the origin of the process, which finally makes the developed solution actually relevant;
- **Iterative process.** The fact of having three distinct partners requires different regular and consecutive exchanges between all of them.
CAIRN Project-Team

5. Software

5.1. Panorama

Besides the development of new reconfigurable architectures, the need for efficient compilation flow is stronger than ever. Challenges come from the high parallelism of these architectures and also from new constraints such as resource heterogeneity, memory hierarchy and power constraints and management. We aim at defining a highly effective software framework for the compilation of high-level specifications into optimized code executed on a reconfigurable hardware platform. Figure 2 shows the global framework that we are currently developing.

Our approach assumes that the application is specified as a hierarchical block diagram of communicating tasks expressing data-flow or control, where each task is expressed using languages like C, Signal, Scilab or Matlab, and is then transformed into an internal representation by the compiler front-end. Then, our framework is based on applying some high-level transformations onto the internal representation.

Different internal representations are used depending on the targeted transformations or the targeted architectures.

- The classical Control and Data Flow Graph (CDFG) is the main internal formalism of our framework. It is the basis for transformations like code optimizations, fixed-point transformations, instruction-set extraction or scheduling. Gateways will be provided from CDFG to other supported formalisms.
- The Hierarchical Conditional Dependency Graph (HCDG) format is also used as the internal representation for pattern-based transformations.

Figure 2. CAIRN’s general software development framework
• Other internal representations like Signal Flow Graphs (SFG) and Polyhedral Reduced Dependence Graph (PRDG) will be used respectively for application accuracy estimation and loop parallelization techniques.

Finally, back-end tools enable the generation of code like VHDL for the hardwired or reconfigurable blocks, C for embedded processor software, and SystemC for simulation purposes (e.g. fixed-point simulations). The compiler front-end, the back-end generators, the transformation toolbox as well as the different internal representations and their respective gateways are based on a single framework: the Gecos framework.

Besides CAIRN’s general design workflow, and in order to promote research undertaken by CAIRN, several hardware and software prototypes are developed. Among those, some distributed software are presented in this report: Gecos a flexible compilation platform, ID.Fix an infrastructure for the automatic transformation of software code aiming at the conversion of floating-point data types into a fixed-point representation, UPaK and Durase for the compilation and the synthesis targeting reconfigurable platforms, and Interconnect Explorer a high-level power and delay estimation tool for on-chip interconnects.

5.2. Gecos

Participants: Steven Derrien [correspondant], Daniel Menard, Kevin Martin, Maxime Naulet, Antoine Floch, Antoine Morvan, Clément Guy, Amit Kumar.

The Gecos (Generic Compiler Suite) project is an open source Eclipse-based C compiler infrastructure developed in the CAIRN group since 2004 that allows for fast prototyping of complex compiler passes. Gecos was designed so as to address part of the shortcomings of existing C/C++ infrastructures such as SUIF and LLVM.

Gecos is a 100% Java based implementation and is based on modern software engineering practices. It uses Eclipse plugin as an underlying infrastructure and thus takes benefits of its plugin mechanism to be easily extensible. Gecos follows Model Driven Software Engineering techniques and rely on Eclipse Modeling Framework. The framework 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 now serves as a backbone infrastructure to many group members (Upak, Durase, ID.Fix). In 2011, the work has focused on extending the loop analysis transformation framework, which now includes an OpenMP static analysis tool (developed jointly with Colorado State University) that was presented in June at the 7th International Workshop on OpenMP [39]. The software engineering challenges posed by optimizing compiler also happen to be a novel and promising application field for the MDE community, which led to joint publication [45] with members from CSU and the Triskell EPI team at the IEEE/ACM Models conference in October 2011. This cross fertilization between MDE and Compilers is the core topic of Clément Guy’s PhD thesis supervised by members of CAIRN (S. Derrien) and Triskell (J.M. Jezequel and B. Combemale).

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

Participants: Daniel Menard [correspondant], Olivier Sentieys, Romuald Rocher, Nicolas Simon, Quentin Meunier.

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 three main modules corresponding to the fixed-point conversion (Fix.Conv), the accuracy evaluation (Acc.Eval) and the dynamic range evaluation (Dyn.Eval).
The different developments carried-out in 2011 allow obtaining a fixed-point conversion tool handling functions, conditional structures and repetitive structures having a fixed number of iterations during time. For the accuracy evaluation (Acc.Eval), conditional structures and correlation between noise sources have been considered. For the dynamic range evaluation (Dyn.Eval), the method based on the Karhunen-Loève Expansion (KLE) have been implemented. It allows determining the dynamic range for a given overflow probability.

The development of this tool has been achieved thanks to an INRIA post-doc in the context of S2S4HLS project until August 2011, and a University of Rennes graduate engineer from November 2011 in the context of DEFIS ANR project and different students during their training period.

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

Participants: Christophe Wolinski [correspondant], François Charot, Antoine Floch.

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 [119]. 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 [correspondant], François Charot, Antoine Floch.

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 Sentiays [correspondant], Olivier Berder, Romain Fontaine, Arnaud Carer, Samuel Mouget, Steven Derrien.
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 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.

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, 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. SoCLib: Open Platform for Virtual Prototyping of Multi-Processors System on Chip
Participants: François Charot [correspondant], Laurent Perraudeau, Charles Wagner.

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

5.8. OCHRE: On-Chip Randomness Extraction

Participants: Olivier Sentieys [correspondant], Arnaud Carer, Arnaud Tisserand.

Ochre is a set of synthesizable VHDL models for true and pseudo random number generation and hardware accelerated statistical tests. It includes IP cores of different oscillator-based TRNGs, different PRNGs (linear feedback shift registers, cellular automata, AES) and several statistical tests (FIPS 140-2, AIS31, Diehard). This set of IPs has been used to design Ochre V1 and V2 chips and were delivered under license to a company.
4. Software

4.1. Javalib

Participants: Frédéric Besson [correspondant], David Pichardie, Vincent Monfort.

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: 2.2
- Programming language: Ocaml

4.2. SAWJA

Participants: Frédéric Besson [correspondant], David Pichardie, Vincent Monfort.

Sawja is a library written in OCaml, relying on Javalib to provide a high level representation of Java bytecode programs. It name comes from Static Analysis Workshop for JAva. Whereas Javalib is dedicated to isolated classes, Sawja handles bytecode programs with their class hierarchy and with control flow algorithms. Moreover, Sawja provides some stackless intermediate representations of code, called JBir and A3Bir. The transformation algorithm, common to these representations, has been formalized and proved to be semantics-preserving. See also the web page http://sawja.inria.fr/.

- Version: 1.2
- Programming language: Ocaml

4.3. 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.

- Version: 3.1
- Programming language: Ocaml
5. Software

5.1. Intrusion Detection

Members of Supélec have developed several intrusion detectors.

Blare implements our approach of illegal information flow detection at the OS level. This implementation is a modification of a standard Linux kernel and it monitors information flows between typical OS containers as files, sockets or IPC. System active entities are processes viewed as black-boxes as we only observe their inputs and outputs. Detection at the OS level is in some cases too coarse-grained to avoid the generation of false positives and to detect attacks targeting the application logic. Even if it remains convenient to define the security policy at the OS-level, sound illegal information flow detection implies an additional detection at the language level. This has led us to implement a detector for Java applications, JBlare, to complement the detection at the OS level. JBlare extends the OS-level one by refining the observation of information flows at the language level.

GNG is an intrusion detection system that correlates different sources (such as different logs) in order to identify attacks against the system. The attack scenarios are defined using the Attack Description Langage (ADeLe) proposed by our team, and are internally translated to attack recognition automatons. GNG intends to define time efficient algorithms based on these automatons to recognize complex attack scenarios.

SIDAN (Software Instrumentation for Detecting Attacks on Non-control-data) is a tool that aims to instrument automatically C-language software with assertions whose role is to detect attacks against the software. This tool is implemented as a plugin of the FRAMA-C framework that provides an implementation of static analysis techniques.

5.2. Privacy

GEPETO (GEoPrivacy-Enhancing TOolkit) is an open source software for managing geolocated data (currently in development in cooperation with LAAS). GEPETO can be used to visualize, sanitize, perform inference attacks and measure the utility of a particular geolocated dataset. For each of these actions, a set of different techniques and algorithms can be applied. The global objective of GEPETO is to enable a user to design, tune, experiment, and evaluate various sanitization algorithms and inference attacks as well as visualizing the following results and assessing their utility.

5.3. Reliable Programming

The Prometeus project, part of the Inria Gforge, is a software environment for reliable programming. The basic elements of Prometeus are Eva, a component-based framework and Adam, a set of group communication services. Eva is an implementation of a component model that aims at supporting the development of distributed abstractions and high-level communication protocols. Adam is a library of agreement components, based on the component model implemented by Eva. The central element of the Adam library is GAC (Generic Agreement Component). It implements a generic and adaptive fault-tolerant consensus algorithm that can be customized to cope with the characteristics of the environment. Moreover, thanks to a set of versatile methods, its behavior can be tuned to fit the exact needs of a specific agreement problem. A range of fundamental Adam components are implemented as specializations of this GAC component. The Adam library currently includes the most important components for reliable distributed programming (Group Membership, Atomic Broadcast).
5. Software

5.1. T3devKit testing toolkit and IPv6 test suites

**Participants:** Anthony Baire, 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 required to execute a test suite. 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](http://www.irisa.fr/tipi).

5.2. Interoperability Assessment

**Participants:** Anthony Baire, Nanxing Chen, César Viho.

In previous works we have developed a software toolkit named T3DevKit for easing the development of TTCN-3 tests. The original tool could only run on POSIX systems with the gcc tool-chain. This year we re-factored the build system using the `waf` build automation tool, which allowed us to integrate with other operating systems and tool-chains (especially MSVC on Windows). This work allowed us to study interoperability issues between tools in the TTCN-3 standard and a poster was presented at the TTCN-3 User Conference 2011. We also presented an introduction tutorial for T3DevKit at this conference [76]. A method to generate and to execute passive interoperability test suites on recorded traces has been proposed in [40].

5.3. Performance and dependability evaluation

**Participants:** Gerardo Rubino, Bruno Sericola, Bruno Tuffin.

We develop software tools for the evaluation of two classes of models: Markov models and reliability networks. The main objective is to quantify dependability aspects of the behaviors of the modeled systems, but other aspects of the systems can be handled (performance, performability, vulnerability). The tools are specialized libraries implementing numerical, Monte Carlo and Quasi-Monte Carlo algorithms.

One of these libraries has been developed for the Celar (DGA), and its goal is the evaluation of dependability and vulnerability metrics of wide area communication networks (WANs). The algorithms in this library can also evaluate the sensitivities of the implemented dependability measures with respect to the parameters characterizing the behavior of the components of the networks (nodes, lines).

We are also developing tools with the objective of building Markovian models and to compute bounds of asymptotic metrics such as the asymptotic availability of standard metrics of models in equilibrium, loss probabilities, blocking probabilities, mean backlogs,....). A set of functions designed for dependability analysis is being built under the name `DependLib`. 
5. Software

5.1. SOFAT

Participants: Loïc Hélouët [correspondant], Rouwaida Abdallah.

SOFAT is the acronym for Scenario Oracle and Formal Analysis Toolbox. As this name suggests it is a formal analysis toolbox for scenarios. Scenarios are informal descriptions of behaviors of distributed systems. SOFAT allows the edition and analysis of distributed systems specifications described using Message Sequence Charts, a scenario language standardized by the ITU [Z.120]. The main functionalities proposed by SOFAT are the textual edition of Message Sequence Charts, their graphical visualization, the analysis of their formal properties, and their simulation. The analysis of the formal properties of a Message Sequence Chart specification determines if a description is regular, local choice, or globally cooperative. Satisfaction of these properties allow respectively for model-checking of logical formulae in temporal logic, implementation, or comparison of specifications. All these applications are either undecidable problems or unfeasible if the Message Sequence Chart description does not satisfy the corresponding property. The SOFAT toolbox implements most of the theoretical results obtained on Message Sequence Charts this last decade. It is regularly updated and re-distributed. The purpose of this software is twofold:

- Provide a scenario based specification tool for developers of distributed applications
- Serve as a platform for theoretical results on scenarios and partial orders

SOFAT provides several functionalities, that are: syntactical analysis of scenario descriptions, Formal analysis of scenario properties, Interactive Simulation of scenarios when possible, and diagnosis. This year, SOFAT was extended with code synthesis functionalities, allowing to generate communicating automata, promela code, or rest based web services from HMSCs. A new release of the software is expected before the end of the year.

See also the web page [http://www.irisa.fr/distribcom/Prototypes/SOFAT/index.html](http://www.irisa.fr/distribcom/Prototypes/SOFAT/index.html).

- AMS: Order; lattices; ordered algebraic structures
- APP: IDDN.FR.001.080027.000.S.P.2003.00.10600
- Programming language: Java
5. Software

5.1. Introduction

The pieces of software described in this section are prototypes implemented by members of the project. They are not available through the APP. Any interested person should contact relevant members of the project.

5.2. QTempIntMiner: quantitative temporal sequence mining

QT\text{EMP}I\text{NT}M\text{INER} (Quantitative Temporal Interval Miner) is a software that implements several algorithms presented in [42] and [8].

The software is mainly implemented in Matlab. A standalone application is now available. It uses the Mixmod toolbox [28] to compute multi-dimensional Gaussian distributions. The main features of QT\text{EMP}I\text{NT}M\text{INER} are:

- a tool for generating synthetic noisy sequences of temporal events,
- an implementation of the QT\text{EMP}I\text{NT}M\text{INER}, QTIAP\text{RIORI} and QTIPREFIX\text{SPAN} 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.

This year the software has been updated to include two new algorithms: QTIAP\text{RIORI} and QTIPREFIX\text{SPAN}. The software has been used to compare the efficiency of three algorithms. The software is currently applied to the characterization of cardiac arrhythmias.

The following website gives many details about the algorithms and provides the latest stable implementation of QT\text{EMP}I\text{NT}M\text{INER}: http://www.irisa.fr/dream/QTempIntMiner/.

5.3. Sacadeau: qualitative modeling and decision-aid to preserve the water quality from pollutants as herbicides

SACADEAU is a software that implements the SACADEAU transfer model presented in section 8.2.1. 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 and to learn rules on-line. This year, we have developed this new version by enabling access to two recommendation action algorithms (see section 6.3.5). The user can choose different parameters (set of classification rules from which actions will be built, parameters concerning action feasibility, etc) before asking for action recommending process, and then easily visualize the characteristics of situations to improve (polluted ones) compared with the different recommended actions. The software is mainly in Java.
The following website is devoted to the presentation of the SACADEAU: http://www.irisa.fr/dream/SACADEAU/.

5.4. Ecomata

We have proposed a new qualitative approach for ecosystem modeling based on timed automata (TA) formalism combined to a high-level query language for exploring scenarios. EcoMata is a tool-box for modeling and exploring qualitatively trophic-food web using this approach. To date, it is dedicated to ecosystems that can be modeled as a collection of species (prey-predator systems) under various human pressures and to environmental disturbances. This tool is made of two main parts: the Network Editor and the Query Launcher. The Network Editor let a stakeholder describe the trophic food web in a graphical way (the species icons and interactions between them). Only few ecological parameters are required and the user can save species in a library. The number of qualitative biomass levels is set as desired. An efficient algorithm generates automatically the network of timed automata. EcoMata provides also a dedicated window to help the user to define different fishing pressures, a nice way being by using chronograms. In the Query Launcher, the user selects the kind of query and the needed parameters (for example the species biomass levels to define a situation). Results are provided in a control panel or in files that can be exploited later. Several additional features are proposed in EcoMata: building a species library, import/export of ecosystem model, batch processing for long queries, etc. EcoMata is developed in Java (Swing for the GUI) and the model-checker called for the timed properties verification is UPPAAL.

The following website is devoted to the presentation of the ECOMATA: http://oban.agrocampus-ouest.fr:8080/ecomata.
4. Software

4.1. The Polychrony toolset

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

The Polychrony toolset is an Open Source development environment for critical/embedded systems. It is based on Signal, a real-time polychronous data-flow 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 the 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 platform, a front-end to the Signal toolbox in the Eclipse environment. The SME platform is distributed under EPL license.
- GCCst, a back-end to GCC that generates Signal programs (not yet available for download).

The Polychrony toolset also provides:

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

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

- The Polychrony toolset public web site: [http://www.irisa.fr/espresso/Polychrony](http://www.irisa.fr/espresso/Polychrony). This site, intended for users and for developers, contains downloadable executable and source versions of the software for different platforms, user documentation, examples, libraries, scientific publications and implementation documentation. In particular, this is the site for the new open-source distribution of Polychrony.
- The INRIAForge: [https://gforge.inria.fr](https://gforge.inria.fr). This site, intended for internal developers, contains the whole sources of the environment and their documentation.
- The TOPCASED distribution site: [http://www.topcased.org](http://www.topcased.org). This site provides the current reference version of the SME platform, including the executable of the Signal toolbox.
The Polychrony toolset currently runs on Linux, MacOS and Windows systems. The Geensoft company, now part of Dassault Systèmes, supplies a commercial implementation of Polychrony, called RT-Builder, used for industrial scale projects (see www.geensoft.com).

4.2. The Eclipse interface

Participants: Loïc Besnard, Yann Glouche, Huafeng Yu, François Fabre, Yue Ma.

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 [35]: all Signal operators (e.g. arithmetic, clock synchronization), model (e.g. process frame, module), and construction (e.g. iteration, type declaration).

The meta-model primarily aims at making the language and services of the Polychrony environment available to inter-operation and composition with other components (e.g. AADL, Simulink, GeneAuto) within an Eclipse-based development tool-chain. 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 tool-chain.

![Figure 7. Eclipse SME Environment.](image)

It also provides a graphical modeling framework allowing to design applications using a component-based approach. Application architectures can be easily described by just selecting components via drag and drop, creating some connections between them and specifying their parameters as component attributes. Using the modeling facilities provided with the Topcased framework, we have created a graphical environment for Polychrony (see figure 7) called SME (Signal-Meta under Eclipse). To highlight the different parts of the modeling in Signal, we split the modeling of a Signal process in three diagrams: one to model the interface of the process, one to model the computation (or dataflow) part, and one to model all explicit clock relations and dependences. The SME environment is available through the Espresso update site [27], in the current OpenEmbeDD distribution [26], or in the TopCased distribution [28].

4.3. Integrated Modular Avionics design using Polychrony

Participants: Thierry Gautier, Paul Le Guernic, Jean-Pierre Talpin.
The Apex interface, defined in the ARINC standard [29], 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 [30]). 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 [7] 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.

4.4. Multi-clocked mode automata

Participants: Jean-Pierre Talpin, Thierry Gautier, Christian Brunette.

Gathering advantages of declarative and imperative approaches, mode automata were originally proposed by Maraninchi et al. to extend the functionality-oriented data-flow paradigm with the capability to model transition systems easily and provide an additional imperative flavor. Similar variants and extensions of the same approach to mix multiple programming paradigms or heterogeneous models of computation [36] have been proposed until recently, the latest advance being the combination of stream functions with automata in [38]. Nowadays, commercial toolsets such as the Esterel Studio’s Scade or Matlab/Simulink’s Stateflow are largely inspired from similar concepts.

While the introduction of preemption mechanism in the multi-clocked data-flow formalism Signal was previously studied by Rutten et al. in [51], no attempt has been made to extend mode automata with the capability to model multi-clocked systems and multi-rate systems. In [53], we extend Signal-Meta with an inherited meta-model of multi-clocked mode automata. A salient feature is the simplicity incurred by the separation of concerns between data-flow (that expresses structure) and control-flow (that expresses a timing model) that is characteristic to the design methodology of Signal.

While the specification of mode automata in related works requires a primary address on the semantics and on compilation of control, the use of Signal as a foundation allows to waive this specific issue to its analysis and code generation engine Polychrony and clearly exposes the semantics and transformation of mode automata in a much simpler way by making use of clearly separated concerns expressed by guarded commands (data-flow relations) and by clock equations (control-flow relations).

4.5. Hyper-text source documentation of Polychrony

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

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

A high-level architectural view of the Polychrony toolset is given in Figure 8.
Figure 8. The Polychrony toolset high-level architecture
5. Software

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

Participants: Thomas Corpetti, Patrick Héas, Etienne Mémin.

This code allows the computation from two consecutive images of a dense motion field. The estimator is expressed as a global energy function minimization. The code enables the choice of different data model and different regularization functional depending on the targeted application. Generic motion estimator for video sequences or dedicated motion estimator for fluid flows can be specified. This estimator 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](http://fluid.irisa.fr).

5.2. 2DLayeredMotion software - Estimation of 2D independent mesoscale layered atmospheric motion fields

Participants: Patrick Héas, Etienne Mémin.

This software enables to estimate a stack of 2D horizontal wind fields corresponding to a mesoscale dynamics of atmospheric pressure layers. This estimator is formulated as the minimization of a global energy function. It relies on a vertical decomposition of the atmosphere into pressure layers. This estimator uses pressure data and classification clouds maps and top of clouds pressure maps (or infra-red images). All these images are routinely supplied by the EUMETSAT consortium which handles the Meteosat and MSG satellite data distribution. The energy function relies on a data model built from the integration of the mass conservation on each layer. The estimator also includes a simplified and filtered shallow water dynamical model as temporal smoother and second-order div-curl spatial regularizer. The estimator may also incorporate correlation-based vector fields as additional observations. These correlation vectors are also routinely provided by the Eumetsat consortium. This code corresponds to research studies published in IEEE transaction on Geo-Science and Remote Sensing. It can be freely downloaded on the FLUID web site [http://fluid.irisa.fr](http://fluid.irisa.fr).

5.3. 3DLayeredMotion software - Estimation of 3D interconnected layered atmospheric motion fields

Participants: Patrick Héas, Etienne Mémin.

This software extends the previous 2D version. It allows (for the first time to our knowledge) the recovery of 3D wind fields from satellite image sequences. As with the previous techniques, the atmosphere is decomposed into a stack of pressure layers. The estimation relies also on pressure data and classification clouds maps and top of clouds pressure maps. In order to recover the 3D missing velocity information, physical knowledge on 3D mass exchanges between layers has been introduced in the data model. The corresponding data model appears to be a generalization of the previous data model constructed from a vertical integration of the continuity equation. This research study has been recently accepted for publication in IEEE trans. on Geo-Science and Remote Sensing. A detailed description of the technique can be found in an Inria research report. The binary of this code can be freely downloaded on the FLUID web site [http://fluid.irisa.fr](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 obtained 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.
I4S Team

5. Software

5.1. COSMAD

With the help of former engineers, I4S team has developed and maintained a Scilab toolbox devoted to modal analysis and vibration monitoring of structures or machines subjected to known or ambient (unknown) excitation. This software (COSMAD 3.64) has been registered at the APP under the number IDDN.FR.001.210011.002.S.A.2003.000.20700

A list of test-cases (simulators, laboratory test-beds, real structures) for which COSMAD has been used is available on I4S website. The problem is to identify the eigenstructure (eigenvalues and observed components of the associated eigenvectors) of the state transition matrix of a linear dynamical system, using only the observation of some measured outputs summarized into a sequence of covariance matrices corresponding to successive time shifts. Other services are

- Output-only and Input/Output subspace-based identification,
- Automated on-line identification package,
- Subspace-based identification through moving sensors data fusion,
- Damage detection and monitoring,
- Damage localization,

The modules have been tested by different partners, especially the French industrial partners, EADS, Dassault and Sopemea, within the FLITE2 project, by partners from the past CONSTRUCTIF project, and within the framework of bilateral contracts with SNECMA and SVS.

Based on intensive internal evaluation of the toolbox, on both simulated and real data sets, EADS Space Transportation and CNES have been investigating how to use the toolbox for the exploitation of the Ariane 5 flight data sets.

This Scilab toolbox continues to play the role of a programming and development environment for all our newly designed algorithms.

5.2. Prototypes

Three software have been deposed to the Agency of Program Protection, i.e.

- 1/ Fast multi-order Stochastic Subspace Identification (FMO-SSI) IDDN.FR.001.100017.000.S.P.2011.000.20700
- 2/ Multi-setup Stochastic Subspace Identification (MS-SI) IDDN.FR.001.100016.000.S.P.2011.000.20700
- 3/ Multi-order confidence interval computation for single-setup and multi-setup Stochastic Subspace Identification (MOCI-SSI) IDDN.FR.001.100018.000.S.P.2011.000.20700

They will be transferred to partners and industrial contracts, starting in 2011.
IPSO Project-Team (section vide)
KERDATA Team

5. Software

5.1. BlobSeer

Contact: Gabriel Antoniu, gabriel.antoniu@inria.fr.
Participants from the KerData team: Alexandra Carpen-Amarie, Diana Moise, Viet-Trung Tran, Alexandru Costan, Gabriel Antoniu, Luc Bougé.

Presentation: BlobSeer is the core software platform for most current projects of the KerData team. It is a data storage service specifically designed to deal with the requirements of large-scale data-intensive distributed applications that abstract data as huge sequences of bytes, called BLOBs (Binary Large OBjects). It provides a versatile versioning interface for manipulating BLOBs that enables reading, writing and appending to them.

BlobSeer offers both scalability and performance with respect to a series of issues typically associated with the data-intensive context: scalable aggregation of storage space from the participating nodes with minimal overhead, ability to store huge data objects, efficient fine-grain access to data subsets, high throughput in spite of heavy access concurrency, as well as fault-tolerance.

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

5.2. Damaris

Contact: Gabriel Antoniu, gabriel.antoniu@inria.fr.
Participants from the KerData team: Matthieu Dorier, Gabriel Antoniu.

Presentation: Damaris is a middleware for multicore SMP nodes enabling them to efficiently handle data transfers for storage and visualization. The key idea is to dedicate one or a few cores of each SMP node to the application I/O. It is developed within the framework of a collaboration between KerData and the Joint Laboratory for Petascale Computing (JLPC). The current version enables efficient asynchronous I/O, hiding all I/O related overheads such as data compression and post-processing. On-going work is targeting fast direct access to the data from running simulations, and efficient I/O scheduling.

Users: Damaris has been preliminarily evaluated at NCSA (Urbana-Champaign) with the CM1 tornado simulation code. CM1 is one of the target applications of the Blue Waters supercomputer developed by at NCSA/UIUC (USA), in the framework of the INRIA/UIUC Joint Lab (JLPC). Work is currently in progress to use Damaris as an intermediate data layer optimizing simulation/visualization coupling for several HPC scientific applications intended to run on the Blue Waters.

URL: http://damaris.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.
5.3. Derived software

Derived from BlobSeer, two additional platforms are currently being developed within KerData: 1) Pyramid, a software service for array-oriented active storage developed within the framework of the PhD thesis of Viet-Trung Tran (see Section 6.7); and 2) TomusBlobs, a PaaS-level storage service for Azure clouds developed within the framework of the thesis of Radu Tudoran in relation to the A-Brain project. These platforms have not been publicly released yet.
5. Software

5.1. ViSP: a visual servoing platform

Participants: Fabien Spindler [correspondant], Filip Novotny, Eric Marchand, François Chaumette.

Since 2005, we develop and release under the terms of the GPLv2 licence, ViSP, an open source library that 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 basic visual features (points, straight lines, circles, spheres, cylinders, frames, image moments...) that can be combined together, and image processing algorithms that allows tracking of visual cues (dots, segments, ellipses,...) or tracking of 3D model-based objects. Simulation capabilities are also available. ViSP and its full functionalities are presented in Fig. 1 and described in [ 7 ].

![ViSP software architecture](image-url)

This year, we continued to improve the software and documentation quality. A new version available at [http://www.irisa.fr/lagadic/visp/visp.html](http://www.irisa.fr/lagadic/visp/visp.html) was released in October 2011. To ease ViSP installation, we provide also precompiled ViSP SDK including pre-built ViSP library and headers.

This last release code has been downloaded 400 times during the first month of availability. It 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 a graduate course delivered at MIT, at IFMA Clermont-Ferrand and ESIR Rennes engineer schools. ViSP is now also a ROS stack and ViSP 3D model-based tracker has been proposed by the community as a ROS package (see [http://www.ros.org/wiki/vision_visp](http://www.ros.org/wiki/vision_visp)).
5.2. Development work: Robot vision platforms

Participants: Fabien Spindler [correspondant], Romain Tallonneau.

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.

This platform is by far the most-used one by Lagadic members (9 papers published by Lagadic in 2011 enclose results validated on it or data acquired on it). Note that this platform is also open to researcher from other labs. For example, the work done in [24] was validated on the Gantry robot.

These equipments require specific hardware, but also software maintenance actions and new developments in order to make them evolve. Training and assistance of the users, presentation of demonstrations also form part of the daily activities.

To improve the panel of demonstrations and to highlight our research activities, we have developed a new demonstration that combines 3D model-based visual tracking and visual servoing techniques provided in ViSP (see Section 5.1) to pick up cubes in order to build a tower. One of the challenges was to automate the initial object localization requested to initialize the tracker. At this end we have developed a generic template pose estimation algorithm based on Surf points of interest matched with the corresponding points provided in a database computed offline during a learning step.

![Figure 2. Lagadic robotics platforms for vision-based manipulation](image)

5.3. Development work: Medical robotics platforms

Participants: Fabien Spindler [correspondant], Alexandre Krupa.

This platform is composed by two robots, a six degrees of freedom Hippocrates medical arm designed by the Sinters company (see Fig. 3.a) and an Adept Viper S850 arm (see Fig. 3.b). 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 the robot end-effector.
The research and experiments concerning ultrasound visual servoing applied to positioning or tracking tasks conducted with this medical robotics platforms are described in Section 6.3. Note that 4 papers published in 2011 by Lagadic enclose results validated on it.

![Lagadic medical robotics platforms: a) Hippocrate medical robot, b) Viper robot arm equipped with a SonixTouch 3D ultrasound probe.](image)

**5.4. Development work: Cycab**

**Participants:** Fabien Spindler [correspondant], Andrea Cherubini.

The Cycab is a four wheel drive autonomous electric car dedicated to vision-based autonomous navigation (see Fig. 4). A pan-tilt head (Biclops PTM) equipped with a firewire Marlin camera with about 70 degrees field of view is mounted on the front bumper, as well as a Sick LDMRS laser rangefinder. Concerning the computer units, the Cycab is equipped with two microprocessors dedicated to the low level control of the vehicle actuators and a laptop dedicated to high level visual navigation. They are connected through an internal CAN bus. The camera, the pan-tilt head and the laser rangefinder are connected to the laptop. The research and experiments conducted with the Cycab are described in Section 6.2.3. Note that 4 papers published by Lagadic in 2011 enclose experimental results obtained with the Cycab.
Figure 4. Lagadic Cycab vehicle
5. Software

5.1. Audio signal processing, segmentation and classification toolkits

Participant: Guillaume Gravier.

Guillaume Gravier is now with the TEXMEX group but this software is being used by several members of the METISS group.

The SPro toolkit provides standard front-end analysis algorithms for speech signal processing. It is systematically used in the METISS group for activities in speech and speaker recognition as well as in audio indexing. The toolkit is developed for Unix environments and is distributed as a free software with a GPL license. It is used by several other French laboratories working in the field of speech processing.

In the framework of our activities on audio indexing and speaker recognition, AudioSeg, a toolkit for the segmentation of audio streams has been developed and is distributed for Unix platforms under the GPL agreement. This toolkit provides generic tools for the segmentation and indexing of audio streams, such as audio activity detection, abrupt change detection, segment clustering, Gaussian mixture modeling and joint segmentation and detection using hidden Markov models. The toolkit relies on the SPro software for feature extraction.

Contact: guillaume.gravier@irisa.fr

5.2. Irene: a speech recognition and transcription platform

Participant: Guillaume Gravier.

Guillaume Gravier is now with the TEXMEX group but this software is being used by several members of the METISS group.

In collaboration with the computer science dept. at ENST, METISS has actively participated in the past years in the development of the freely available Sirocco large vocabulary speech recognition software [113]. The Sirocco project started as an INRIA Concerted Research Action now works on the basis of voluntary contributions.

The Sirocco speech recognition software was then used as the heart of the transcription modules within a spoken document analysis platform called IRENE. In particular, it has been extensively used for research on ASR and NLP as well as for work on phonetic landmarks in statistical speech recognition.

In 2009, the integration of IRENE in the multimedia indexing platform of IRISA was completed, incorporating improvements benchmarked during the ESTER 2 evaluation campaign in December 2008. Additional improvements were also carried out such as bandwidth segmentation and improved segment clustering for unsupervised acoustic model adaptation. The integration of IRENE in the multimedia indexing platform was mainly validated on large datasets extracted from TV streams.

Contact: guillaume.gravier@irisa.fr
http://gforge.inria.fr/projects/sirocco

5.3. MPTK: the Matching Pursuit Toolkit

Participants: Rémi Gribonval, Ronan Le Boulch.
The Matching Pursuit ToolKit (MPTK) is a fast and flexible implementation of the Matching Pursuit algorithm for sparse decomposition of monophonic as well as multichannel (audio) signals. MPTK is written in C++ and runs on Windows, MacOS and Unix platforms. It is distributed under a free software license model (GNU General Public License) and comprises a library, some standalone command line utilities and scripts to plot the results under Matlab.

MPTK has been entirely developed within the METISS group mainly to overcome limitations of existing Matching Pursuit implementations in terms of ease of maintainability, memory footprint or computation speed. One of the aims is to be able to process in reasonable time large audio files to explore the new possibilities which Matching Pursuit can offer in speech signal processing. With the new implementation, it is now possible indeed to process a one hour audio signal in as little as twenty minutes.

Thanks to an INRIA software development operation (Opération de Développement Logiciel, ODL) started in September 2006, METISS efforts have been targeted at easing the distribution of MPTK by improving its portability to different platforms and simplifying its developers’ API. Besides pure software engineering improvements, this implied setting up a new website with an FAQ, developing new interfaces between MPTK and Matlab and Python, writing a portable Graphical User Interface to complement command line utilities, strengthening the robustness of the input/output using XML where possible, and most importantly setting up a whole new plugin API to decouple the core of the library from possible third party contributions.

Collaboration : Laboratoire d’Acoustique Musicale (University of Paris VII, Jussieu).

Contact : remi.gribonval@irisa.fr


5.4. FASST

Participants: Emmanuel Vincent [correspondant], Alexey Ozerov, Frédéric Bimbot.

FASST is a Flexible Audio Source Separation Toolbox in Matlab, designed to speed up the conception and automate the implementation of new model-based audio source separation algorithms.
MIMETIC Team

5. Software

5.1. HPTS++: Hierarchical Parallel Transition System ++

Participants: Stéphane Donikian [contact], Fabrice Lamarche [contact].

HPTS++ is a platform independent toolkit to describe and handle the execution of multi-agent systems. It provides a specific object oriented language encapsulating C++ code for interfacing facilities and a runtime kernel providing automatic synchronization and adaptation facilities.

The language provides functionalities to describe state machines (states and transitions) and to inform them with user specific C++ code to call at a given point during execution. This language is object oriented and supports concepts such as polymorphism and inheritance (state machines and user defined C++ classes). The compilation phase translates a state machine in a C++ class that can be compiled separately and linked through static or dynamic libraries. The runtime kernel includes a scheduler that handles parallel state machines execution and that provides synchronization facilities such as mutual exclusion on resources, dead lock avoidance, notions of priorities and execution adaptation in accordance with resources availability.

HPTS++ also provides a task model. Thanks to this model, the user can describe primitive behaviors through atomic tasks and combine them with operators (sequence, parallelism, loops, alternatives...). These operators are fully dynamic. Hence they can be used at runtime to rapidly create complex behaviors.

5.2. MKM: Manageable Kinematic Motions

Participants: Richard Kulpa [contact], Franck Multon.

We have developed a framework for animating human-like figures in real-time, based on captured motions. This work was carried-out in collaboration with the M2S Laboratory (Mouvement, Sport, Santé) of the University Rennes 2.

In this software, we propose a morphology-independent representation of the motion that is based on a simplified skeleton which normalizes the global postural informations. This formalism is not linked to morphology and allows very fast motion retargetting and adaptation to geometric constraints that can change in real-time. This approach dramatically reduces the post production time and allows the animators to handle a general motion library instead of one library per avatar.

The framework provides an animation library which uses the motions either obtained from our off-line tool (that transforms standard formats into our morphology-independent representation) or parameterized models in order to create complete animation in real-time. Several models are proposed such as grasping, orientation of the head toward a target. We have also included a new locomotion model that allows to control the character directly using a motion database.

In order to create realistic and smooth animations, MKM uses motion synchronization, blending and adaptation to skeletons and to external constraints. All those processes are performed in real-time in an environment that can change at any time, unpredictably.

All these features have been used to anticipate and control the placement of footprints depending on high level parameters. This link between control and behavior levels will be used for reactive navigation in order to have realistic motion adaptations as well as to deal with constrained environments.

5.3. TopoPlan: Topological Planner and Behaviour Library

Participant: Fabrice Lamarche [contact].
TopoPlan (Topological Planner) is a toolkit dedicated to the analysis of a 3D environment geometry in order to generate suitable data structures for path finding and navigation. This toolkit provides a two step process: an off-line computation of spatial representation and a library providing on-line processes dedicated to path planning, environmental requests...

TopoPlan is based on an exact 3D spatial subdivision that accurately identifies floor and ceiling constraints for each point of the environment. Thanks to this spatial subdivision and some humanoid characteristics, an environment topology is computed. This topology accurately identifies navigable zones by connecting 3D cells of the spatial subdivision. Based on this topology several maps representing the environment are extracted. Those maps identify obstacle and step borders as well as bottlenecks. TopoPlan also provides a runtime library enabling the on-line exploitation of the spatial representation. This library provides several algorithms including roadmap-based path-planning, trajectory optimization, footprint generation, reactive navigation and spatial requests through customizable spatial selectors.

TopoPlan behavior is a library built on top of TopoPlan and MKM providing several behaviors described thanks to the HPTS++ task model. Its goal is to provide a high level interface handling navigation and posture adaptation within TopoPlan environments. Provided behaviors include:

- A behavior handling fully planned navigation toward an arbitrary destination. This behavior precisely handles footprint generation within constrained environments such as stairs for instance.
- A behavior controlling an MKM humanoid to follow a trajectory specified by the user.
- A behavior controlling MKM to follow a list of footprints given by the user.
- A behavior adapting the humanoid posture to avoid collision with ceiling. This behavior runs in parallel of all other behaviors and adapts humanoid motion when needed without any user intervention.
- A behavior handling reactive navigation of virtual humans. This behavior plan a path to a given target and follows the path while avoiding collisions with other navigating entities.

Those behaviors have been built using the HPTS++ task model. Thus, they can be easily combined together or with other described behaviors through task operators.
MYRIADS Team

5. Software

5.1. SAFDIS

Contact: Jean-Louis Pazat, Jean-Louis.Pazat@irisa.fr
URL: http://www.irisa.fr/myriads/software/folder.2011-12-13.8949308917/
Status: Version 1.0
License: TBD

Presentation: SAFDIS (Self Adaptation for Distributed Services) is a generic framework allowing the self-adaptation of distributed service based applications within a highly volatile context. Compared to other adaptation frameworks, the main advantages of SAFDIS are its genericity, its distributed nature and the focus on SOAs. SAFDIS is in its final implementation and testing phase within the Myriads team and is being used with a real life use case for emergency services.

The current implementation of SAFDIS is based on a Java OSGi implementation. SAFDIS is written in Java and organized into OSGi bundles. SAFDIS is not tight to any specific operating system and work within any JAVA 1.6 platform. An OSGi implementation is needed (such as the Apache Felix http://felix.apache.org or Equinox eclipse.org/equinox implementations). In order to benefit from the reactive adaptation tools, the Jess engine is also needed as an OSGi bundle (http://www.jessrules.com).

5.2. HOCL-tools

Contact: Cédric Tedeschi, Cedric.Tedeschi@irisa.fr
Status: Version 1.0 to be released
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 team. Within HOCL, following the chemical metaphor, computations can be regarded as chemical reactions, and data can be seen as molecules which participate in these reactions. If a certain condition is held, the reaction will be triggered, thus continuing until it gets inert: no more data can satisfy any computing conditions. To realize this program paradigm, a multiset is implemented to act as a chemical tank, containing necessary data and rules. An HOCL program is then composed of two parts: chemical rule definitions (reaction rules) and multiset definition (data). More specifically, HOCL provides the high order: reaction rules are molecules that can be manipulated like any other molecules. In other words, HOCL programs can manipulate other HOCL programs.

An HOCL compiler was developed using java to execute some chemical programs expressed with HOCL. This compiler is based on the translation of HOCL programs to java code. As a support for service coordination and service adaptation (refer to Section 6.4), we recently extended the HOCL compiler with the support of decentralized workflow execution. Works around the implementation of a distributed multiset gave birth to an underlying layer for this compiler, making it able to deploy HOCL programs transparently over large scale platforms. This last part is currently considered to be interfaced with the current HOCL compiler. All these features are planned to be released under the common name of HOCL-tools.

• Version: 1
5.3. XtreemOS

Contact: Yvon Jégou, Yvon.Jegou@inria.fr
Status: Version 3.0
License: GPL-2/BSD depending on software packages composing the system

Presentation: XtreemOS is a Grid Operating system based on Linux with native support for virtual organizations. Three flavours of XtreemOS are developed for individual PCs, clusters and mobile devices (PDA, notebooks and smartphones). XtreemOS has been developed by the XtreemOS consortium. The third public version of XtreemOS has been released in December 2010. XtreemOS has been presented at the Contraill summer school, Presqu’île de Giens, France in July 2011 (Y. Jégou), SC’11, Seattle, USA (Y. Jégou, Ch. Morin) in November 2011. XtreemOS software is a set of services developed in Java, C++ and C. XtreemOS cluster version leverages KERRIGHED single system image operating system. A permanent testbed composed of computers provided by several XtreemOS partners has been public since fall 2010. In 2011, XtreemOS has been packaged for the OpenSuse Linux distribution. Ready-to-use XtreemOS virtual machine images have been made available for the community.

Active contributors (from Myriads team): Amine Belhaj, Jérôme Gallard, Rémy Garrigue, Yvon Jégou, Christine Morin, Yann Radenac, Pierre Riteau.

5.4. Contrail Virtual Execution Platform (VEP)

Contact: Yvon Jégou, Yvon.Jegou@inria.fr
URL: http://www.contrail-project.eu
Status: Version 1.0
License: BSD

Presentation: Virtual Execution Platform (VEP) is a Contrail service that sits just above IaaS layer at the service provider end of the Contrail cloud federation. The VEP 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 integrates a monitoring system where the major events about the application, mainly resource usage, are made available to the user.

The VEP service provides a RESTful interface and can be exploited directly by users on top of the provider IaaS. OpenNebula and OCCI-based IaaS interfaces are currently supported.

Active contributors (from Myriads team): Roberto-Gioacchino Cascella, Florian Dudouet, Piyush Harsh, Yvon Jégou, Christine Morin.

5.5. Snooze

Contact: Christine Morin, Christine.Morin@inria.fr
URL: http://www.irisa.fr/myriads/software/snooze/
Status: Version 1.0
License: TBD
Presentation: Snooze [56], is a novel VM management framework for private clouds which is designed to scale across thousands of nodes. Unlike the existing cloud management frameworks, Snooze utilizes a self-organizing hierarchical architecture and performs distributed VM management. Particularly, VM management tasks are performed by multiple managers, with each manager having only a partial view of the system. Moreover, fault-tolerance is provided at all levels of the hierarchy by replication and integrated leader election algorithm. Consequently, the system is able to self-heal and continue its operation despite system component failures. In addition, VM monitoring is integrated into the framework and a generic scheduling engine exists to support advanced (e.g., consolidation) scheduling policies. Last but not least, a RESTful command line interface (CLI) exists to support virtual cluster (VC) definitions and management (i.e., start, shutdown, destroy, suspend, etc.) as well hierarchy visualization and exporting in GraphML format.

Snooze is fully implemented from scratch in Java and currently comprises approximately 15,000 lines of maintainable abstractions-based code. The leader election algorithm is built on top of the Apache ZooKeeper [59] highly available and reliable coordination system. In order to provide a uniform interface to the underlying hypervisors and support transparent VM monitoring and management, Snooze integrates the libvirt virtualization library.

The first Snooze prototype [41] has been developed and evaluated on Grid5000 experimental testbed.

Active contributors (from Myriads team): Eugen Feller, Christine Morin.

5.6. Resilin

Contact: Christine Morin, Christine.Morin@inria.fr
URL: http://www.irisa.fr/myriads/software/resilin/
Status: Version 0.1
License: TBD

Presentation: Resilin [45] is a system to easily create execution platforms over distributed cloud resources for executing MapReduce computations. Resilin implements the Amazon Elastic MapReduce web service API with resources from other clouds than Amazon EC2, such as private and community clouds. Resilin allows users to perform MapReduce computations on other infrastructures than Amazon EC2, and offers more flexibility: users are free to select different types of virtual machines, different operating systems or newer Hadoop versions. Users only have to submit computations to the service through a web service API, and Resilin takes care of provisioning, configuring and managing cloud-based Hadoop execution platforms, potentially using multiple clouds.

Resilin is implemented in the Python language. It uses the boto library in order to interact with EC2-compatible clouds. Resilin has been evaluated on the Grid’5000 experimental testbed. Our comparison with the Amazon Elastic MapReduce service shows similar performance.

Active contributors (from Myriads team): Pierre Riteau, Christine Morin.

5.7. Saline

Contact: Christine Morin, Christine.Morin@inria.fr
URL: https://www.grid5000.fr/mediawiki/index.php/VMdeploy
Status: Version V1.0 (experimental)
License: BSD
Presentation: **Saline** (former called VMdeploy) is a generic framework to deploy and manage encapsulated user jobs in virtual machines (VMs) at grid level by moving them from one site to another transparently for the encapsulated jobs [58]. Moreover, **Saline** is non-intrusive and can be used with any non-modified Grid resource management systems (RMSs).

**Saline** deploys and configures a set of VMs according to the user needs. Then, periodically, **Saline** takes snapshots of the running VMs and saves them on a dedicated node in an efficient way [57]. In addition, **Saline** checks the status of the running VMs. If something wrong happens i.e. one or more VMs have failed due for instance to a node failure or to the arrival of a higher priority job, **Saline** redeploys the set of VMs from the latest snapshot taken on new available resources provided by the Grid RMS. Thanks to **Saline**, the redeployment of the snapshot is done in a transparent way from the encapsulated job point of view.

In its current implementation, **Saline** is programmed in bash and C. It uses Libvirt in order to create, to snapshot, and to restart the VMs. It means that **Saline** can deploy and manage KVM and XEN VMs or any other VMs usable with Libvirt. In addition, the architecture of **Saline** is very modular in order to have a clear and easily extensible code.

Active contributors (from Myriads team): Jérôme Gallard, Christine Morin
5. Software

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 [18]. The purpose of Modal Interfaces is to provide a formal support to contract based design methods in the field of system engineering. Modal Interfaces enable compositional reasoning methods on I/O reactive systems.

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

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

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

5.2. Synet: A General Petri-Net Synthesis Toolbox

Participant: Benoît Caillaud.

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

Synet is a software tool for the synthesis of bounded and unbounded Petri-nets, based on the theory of regions [31]. It can synthesize Petri-nets from automata or regular expression and can be configured by command-line options to synthesize nets modulo graph isomorphism or language equality. Petri nets computed by Synet can be displayed using the GraphViz 2D graph layout software, or saved to a file for further transformation and analysis.

The tool actually implements two linear-algebraic synthesis methods: A first method uses the simplex algorithm and the second one is based on the computation of extremal rays of polyhedral cones, using Chernikova’s algorithm [34]. Both methods imply that the input graphs are given by extension. Nevertheless, Synet yields good performances on many practical use-cases and is the only tool supporting unbounded net synthesis.

The main application of Synet is the synthesis of communicating distributed protocols and controllers [30]. Synthesis is constrained to produce so-called distributables nets [33], a class of nets that can be turned into networks of communicating automata by automated methods. This allows to divide the synthesis problem in two steps: Given the specification of a protocol as a finite automaton, (i) synthesize (if exists) a distributable net, and then (ii) derive a network of communicating automata from the distributable net. While the second step is automatic and straightforward, the first step is in essence a computer assisted design task, where the distributed Petri-net synthesis algorithm helps the designer to refine the protocol specification into a graph isomorphic to the marking graph of a distributable net.
5. Software

5.1. H2OLab

Participants: Jocelyne Erhel [correspondant], Aurélien Le Gentil, Géraldine Pichot, Baptiste Poirriez, Nadir Soualem.

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 software and a database which are interfaced through the web portal H2OWEB. The platform H2OLab is an essential tool for the dissemination of scientific results. Currently, software and database are shared by the partners of the Micas project (see 8.1.2 ). 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. GW-UTIL

Participants: Jocelyne Erhel, Aurélien Le Gentil, Géraldine Pichot [correspondant], Baptiste Poirriez, Nadir Soualem.

The software GW-UTIL allows to discretize PDE for flow and transport in aquifers and to deal with stochastic models. It contains a set of utilitary modules for geometry, input, output, random numbers, visualization, parallel computing, numerical algorithms, etc. A package is devoted to launch applications.

See also the web page http://h2olab.inria.fr .

- Version: version 1.0, May 2008
- APP: registered
- Programming language: C++

5.3. GW-NUM

Participants: Jocelyne Erhel, Aurélien Le Gentil, Géraldine Pichot [correspondant], Baptiste Poirriez, Nadir Soualem.

The software GW-NUM is a set of generic modules to discretize PDE of flow and transport in 2D computational domains in order to deal with stochastic models. Methods for flow simulations are either Finite Volume on structured meshes or Mixed Finite Element with unstructured meshes. Method for transport simulations is a particle tracker for advection and a random walker for diffusion. Uncertainty Quantification method is Monte-Carlo. For flow computations, the involved linear system is solved by external software devoted to sparse matrices.

See also the web page http://h2olab.inria.fr .

- Version: version 1.0, May 2008
- APP: registered
- Programming language: C++

5.4. MP-FRAC

Participants: Jocelyne Erhel, Aurélien Le Gentil, Géraldine Pichot [correspondant], Baptiste Poirriez, Nadir Soualem.
The software MP-FRAC aims at modelling and simulating numerically flow in a fractured aquifer. The physical domain is a network of fractures, either deterministic or stochastic, with a permeability field either deterministic or stochastic. The software computes the velocity field in the aquifer, by assuming that the medium is saturated and that flow is steady-state. Physical equations are stochastic PDEs, handled by a Monte-Carlo method. This non intrusive approach generates a set of random samples, which are used for simulations. Then, the software analyzes statistically the flow in the stochastic case. The objective is to characterize hydraulic properties in Discrete Fracture Networks. The software MP-FRAC handles a simulation corresponding to one sample, whereas Monte-Carlo method is implemented in a generic way by the software GW-NUM. The software is specific of the physical model (Discrete Fracture Network) and of the application (steady-state flow). Generic numerical methods to discretize PDE are implemented in the software GW-NUM. See also the web page [http://h2olab.inria.fr](http://h2olab.inria.fr).

- Version: version 1.0, May 2008
- APP: registered
- Programming language: C++

### 5.5. PARADIS

**Participants:** Jocelyne Erhel, Aurélien Le Gentil, Géraldine Pichot [correspondant], Baptiste Poirriez, Nadir Soualem.

The software PARADIS aims at modelling and simulating numerically flow in a porous aquifer and transport by convection-diffusion of an inert solute. The porous medium is heterogeneous, with a stochastic or deterministic permeability field. A first step computes the velocity filed in the aquifer, by assuming that the medium is saturated and that flow is steady-state. A second step computes the distribution of solute concentration, by assuming a transport by convection and by molecular diffusion. Physical equations are stochastic PDEs, handled by a Monte-Carlo method and discretized by numerical methods. This non intrusive approach generates a set of random samples, which are used for simulations. Then, the software analyzes statistically the flow in the stochastic case. The objectives are to determine asymptotic laws of transport, to characterize pre-asymptotic behavior and to define global laws.

The software PARADIS handles a simulation corresponding to one sample, whereas Monte-Carlo method is implemented in a generic way by the software GW-NUM. The software is specific of the physical model (heterogeneous porous medium) and of the application (steady-state flow then transport with macro-dispersion). Generic numerical methods to discretize PDE are implemented in the software GW-NUM. See also the web page [http://h2olab.inria.fr/](http://h2olab.inria.fr/).

- Version: version 1.0, May 2008
- APP: registered
- Programming language: C++

### 5.6. GRT3D

**Participants:** Édouard Canot, Jocelyne Erhel [correspondant], Souhila Sabit, Nadir Soualem.

Reactive transport modeling has become an essential tool for understanding complex environmental problems. It is an important issue for MoMaS partners (see section 8.1.1), in particular Andra (see section 7.1). We have developed a method coupling transport and chemistry, based on a method of lines such that spatial discretization leads to a semi-discrete system of algebraic differential equations (DAE system). The main advantage is to use a complex DAE solver, which controls simultaneously the timestep and the convergence of Newton algorithm. Another approach, called SIA, is to use a fixed-point method to solve the nonlinear system at each timestep.
The software suite GRT3D has three executable modules:

- **SIA1D**: Sequential Iterative Approach for 1D domains;
- **GDAE1D**: Global DAE approach for 1D domains;
- **GDAE3D**: Global DAE approach for 1D, 2D or 3D domains.

- Version: version 1.0, April 2011
- APP: registered
- Programming language: C

### 5.7. GPREMS

**Participants:** Édouard Canot, Jocelyne Erhel [correspondant], Désiré Nuentsa Wakam, Nadir Soualem.

GPREMS implements a robust hybrid solver for large sparse linear systems that combines a Krylov subspace method as accelerator with a Schwarz-based preconditioner. This preconditioner uses an explicit formulation associated to one iteration of the multiplicative Schwarz method. The Newton-basis GMRES, which aims at expressing a good data parallelism between subdomains is used as accelerator.

See also the web page [http://www.irisa.fr/sage/](http://www.irisa.fr/sage/).

- Version: version 1.0, May 2008
- APP: registered
- Programming language: C++

### 5.8. DGMRES

**Participants:** Jocelyne Erhel [correspondant], Désiré Nuentsa Wakam.

DGMRES implements a preconditioner based on adaptive deflation, which can be used with any preconditioner for the GMRES algorithm. It is distributed with the free software PETSC.

See also the web page [http://www.irisa.fr/sage/](http://www.irisa.fr/sage/).

- Version: version 1.0, June 2011
- APP: distributed with PETSC
- Programming language: C

### 5.9. AGMRES

**Participants:** Jocelyne Erhel [correspondant], Désiré Nuentsa Wakam.

AGMRES implements an augmented subspace approach, based on adaptive deflation, which can be used with any preconditioner for the GMRES algorithm. It also implements a Newton basis for enhancing parallelism. It will be distributed with the free software PETSC.

See also the web page [http://www.irisa.fr/sage/](http://www.irisa.fr/sage/).

- Version: version 1.0, November 2011
- APP: soon distributed with PETSC
- Programming language: C

### 5.10. PPAT: pseudo-spectrum

**Participants:** Édouard Canot [corresponding author], Bernard Philippe.
PPAT (Parallel PATH following software) is a parallel code, developed by D. Mezher, W. Najem (University of Saint-Joseph, Beirut, Lebanon) and B. Philippe. This tool can follow the contours of a functional from \( \mathbb{C} \) to \( \mathbb{R}^+ \). The present version is adapted for determining the level curves of the function \( f(z) = \sigma_{\min}(A - zI) \) which gives the pseudospectrum of matrix \( A \).

The algorithm is reliable: it does not assume that the curve has a derivative everywhere. The process is proved to terminate even when taking into account roundoff errors. The structure of the code spawns many independent tasks which provide a good efficiency in the parallel runs.

The software can be downloaded under the GPL licence from: http://sourceforge.net/projects/ppat.

5.11. MUESLI: Scientific computing

Participant: Édouard Canot [corresponding author].

Doing linear algebra with sparse and dense matrices is somehow difficult in scientific computing. Specific libraries do exist to deal with this area (e.g. BLAS and LAPACK for dense matrices, SPARSKIT for sparse ones) but their use is often awful and tedious, mainly because of the large number of arguments which must be used. Moreover, classical libraries do not provide dynamic allocation. Lastly, the two types of storage (sparse and dense) are so different that the user must know in advance the storage used in order to declare correctly the corresponding numerical arrays.

MUESLI is designed to help in dealing with such structures and it provides the convenience of coding in Fortran with a matrix-oriented syntax; its aim is therefore to speed-up development process and to enhance portability. It is a Fortran 95 library split in two modules: (i) FML (Fortran Muesli Library) contains all necessary material to numerically work with a dynamic array (dynamic in size, type and structure), called \( \text{mfArray} \); (ii) FGL (Fortran Graphics Library) contains graphical routines (some are interactive) which use the \( \text{mfArray} \) objects.

MUESLI includes some parts of the following numerical libraries: Arpack, Slatec, SuiteSparse, Triangle, BLAS and LAPACK.

Linux is the platform which has been used for developing and testing MUESLI. Whereas the FML part (numerical computations) should work on any platform (e.g. Win32, Mac OS X, Unix), the FGL part is intended to be used only with X11 (i.e. under all UNIXes).

Last version of MUESLI is 2.5.2 (29 nov 2011). More information can be found at: http://www.irisa.fr/sage/edouard/canot/muesli/

5.12. CANARD: BEM for surface flows

Participant: Édouard Canot [corresponding author].

When dealing with non-linear free-surface flows, mixed Eulerian-Lagrangian methods have numerous advantages, because we can follow marker particles distributed on the free-surface and then compute with accuracy the surface position without the need of interpolation over a grid. Besides, if the liquid velocity is large enough, Navier-Stokes equations can be reduced to a Laplace equation, which is numerically solved by a Boundary Element Method (BEM); this latter method is very fast and efficient because computing occur only on the fluid boundary. This method is applied to the spreading of a liquid drop impacting on a solid wall and to the droplet formation at a nozzle; applications take place, among others, in ink-jet printing processes.

The code used (CANARD) has been developed with Jean-Luc Achard (LEGI, Grenoble) for fifteen years and is used today mainly through collaborations with Carmen Georgescu at UPB (University Polytechnica of Bucarest, Romania), and with Alain Glière (CEA-LETI, Grenoble).
5. Software

5.1. nD-SAFIR: Image denoising software

Participants: Charles Kervrann, Patrick Bouthemy.

The nD-SAFIR software (APP deposit number: IDDN.FR.001.190033.002.S.A.2007.000.21000 / new release 3.0 in 2012) written in C++, JAVA and MATLAB, removes additive Gaussian and non-Gaussian noise in still 2D or 3D images or in a 2D or 3D image sequences (with no motion computation) (see Figure 1) [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 associate with each pixel (or voxel) 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, ...).

Academic licence agreements: Institut Curie, CNRS, ENS Ulm, Oxford university, Weizmann Institute, UCSF San-Francisco, Harvard university, Berkeley university, Stanford university, Princeton university, Georgia-Tech, Kyoto university, IMCB Singapore ...

Partners: J. Boulanger, J. Salamero (UMR 144 CNRS Institut Curie), P. Elbau (RICAM Linz, Austria), J.B. Sibarita (UMR 5091 University of Bordeaux 2)

5.2. Fast2D-SAFIR: Fast denoising of large 2D images

Participant: Charles Kervrann.

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 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, ...).

5.3. PBED: Patch-based event detection

Participant: Charles Kervrann.

The PBED software written in C++ automatically quantifies in space and time the number of sudden and transient events observed in fluorescence (WF, TIRF) microscopy. The algorithm parameters are calibrated from the comparison of image patches expected to distinguish sudden appearing/vanishing fluorescent spots/particles from other motion behaviors such as lateral movements [1] and [23]. Two statistical procedures are proposed respectively to control the number of false alarms (Benjamini-Hochsberg, Bonferonni). The algorithm is mainly used to statistically explore the effect of several biological perturbations on the rate of transient events detected on the pilot biological model (e.g. Langerin-YFP endocytic-recycling trans-membrane protein).

Partners: J. Boulanger, A. Gidon, A. Chessel, B. Cinquin, J. Salamero (UMR 144 CNRS Institut Curie)
5.4. HullkGround: Background subtraction by convex hull estimation

**Participant:** Charles Kervrann.

The HULLKGROUND software (APP deposit number: IDDN.FR.001.400005.000.S.P.2009.000.21000) written in JAVA (plug-in IMAGEJ (http://rsbweb.nih.gov/ij/)) 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 $\alpha$ shapes-based scale-space method [24].

**Partners:** A. Chessel and J. Salamero (UMR 144 CNRS Institut Curie)

5.5. TubuleJ: Straightening of microtubule cryo-EM projection views

**Participant:** Charles Kervrann.

The TUBULEJ software (APP deposit number: IDDN.FR.001.240023.000.S.P.2011.000.21000) written in JAVA (plug-in IMAGEJ (http://rsbweb.nih.gov/ij/)) 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.

**Partners:** S. Blestel and D. Chrétien (UMR 6026 CNRS University of Rennes 1)
5.6. Cryo-Seg: Segmentation of tomograms in cryo-electron microscopy

**Participant:** Charles Kervrann.

The Cryo-Seg software written in C++ has been developed to detect microtubule structures and helical structures in 2D cryo-electron microscope images (see Figure 2). Cryo-electron tomography allows 3D observation of biological specimens in their hydrated state. Segmentation is formulated as Maximum A Posteriori estimation problem and exploits image patches to take into account spatial contexts (Markov Random Fields). Because of the contrast anisotropy in the specimen thickness direction, the whole tomogram is segmented section by section, with an automatic update of reference patches. This algorithm has been evaluated on synthetic data and on cryo-electron tomograms of in vitro microtubules. On real data, this segmentation method extracts the most contrasted regions of microtubules, and 3D visualization is improved.

**Partners:** S. Blestel and D. Chrétien (UMR 6026 CNRS University of Rennes 1)

*Figure 2. Cryo-Seg software: Segmentation of 3D microtubules in a cryo-EM tomogram (left) and 2D view (right) (UMR 6026 CNRS University of Rennes 1)*
SYMBIOSE Project-Team

5. Software

5.1. Main softwares

Participants: Olivier Collin [correspondant], Dominique Lavenier, François Coste, Olivier Sallou, Romaric Sabas, Guillaume Rizk, Andres Burgos.

We highlight here 3 softwares of the team which received considerable care this year, in particular to improve their ergonomy and diffusion. In the following sections, all softwares of the team will be described, classified according to their applicative domain.

5.1.1. Biomaj : Data synchronization and processing workflow

BioMAJ (BIOlogie Mise A Jour) is a workflow engine dedicated to data synchronization and processing. The Software automates the update cycle and the supervision of the locally mirrored databank repository. Thanks to the funding of INRIA’s ADT, the BioMAJ software has been ergonomically improved and is diffusion enhanced. It is now part of a Linux distribution (Debian-med). The tool is now used on many bioinformatics core facilities in France and Europe. It is used as an infrastructure tool but also as a key component of new resources. For example the AnnotQTL tool relies heavily on BioMAJ. Another example is popgenie, an integrative explorer of the Populus genome in Sweden has been built on top of BioMAJ.

[Web site: http://biomaj.genouest.org]

5.1.2. GASSST: Short reader mapper for large genomic dataset

GASSST is a short read mapper allowing very large genomic dataset to be processed. It takes as input raw data (reads) coming from next generation sequencing machines and map them over full genomes. In 2011, the GASSST software has been tuned to meet industrial requirements and transfered to the GenomeQuest Company. A specific license agreement has been set up between INRIA and GenomeQuest for integrating GASSST into the GenomeQuest NGS tool suite.

web site: http://www.irisa.fr/symbiose/projects/gassst/

5.1.3. Protomata learner: fine characterization of protein families

Protomata-Learner V2.0 is a tool to infer weighted automata for the characterization of (structural or functional) families of proteins from a sample of (unaligned) sequences belonging to the family. Protomata-Learner has been completely rewritten thanks to the ADT "Suite logicielle pour la modélisation de familles protéiques par automates": based on a better formalisation and thanks to the implementation of efficient weighting techniques, this new version is significantly faster and gives better results. Special care has been given to the integration of the different programs to propose an easy-to-use suite.

Protomata-Learner has been tested and improved on real use-case thanks to collaborations established in Lepidolf and Pelican ANR projects. New scanning algorithms (Forward scores) and procedures for choosing automatically the best set of parameters have been developed. New signatures for the studied families of proteins have been established and are used for the predictions of candidates by our partners.

[Web site: http://tools.genouest.org/tools/protomata/]
5.2. Bioinformatics community tools

Participants: Olivier Collin [contact], Olivier Sallou, Charles Deltel, François Moreews, Anthony Bretaudeau, Delphine Naquin, Aurélien Roult, Romaric Sabas, Claudia Hériveau.

- **BioMAJ** See first section above.
- **GRISBI** The GRISBI project is aiming to set up a grid infrastructure devoted to the Bioinformatics community. This infrastructure is built upon the resources available on different bioinformatics facilities through gLite middleware. [Web site: http://www.grisbio.fr]
- **Mobylenet** In partnership with other bioinformatics platforms, GenOuest is setting up a distributed network of bioinformatics resources built upon web portals based on the Mobyle platform. [Web site: http://mobylenet.rpbs.univ-paris-diderot.fr:8080/]
- **MetaData platform** Seqcrawler is an indexing platform for biological meta data and sequences, providing a google like web interface. It can scale from single computers to the cloud. [Web site: http://seqcrawler.sourceforge.net/]
- **DrMotifs** DrMotifs is a new software resource aiming at the integration of different software commonly used in pattern search and discovery. This resource will also integrate new software elaborated by the Symbiose team. [Web site: http://www.drmotifs.org] [Blog site: http://drmotifs.genouest.org]

5.3. Parallel softwares

Participants: Dominique Lavenier [contact], Charles Deltel, Erwan Drezen, Guillaume Chapuis, Guillaume Rizk.

- **PLAST:** intensive bank sequence comparison. PLAST is a parallel version of BLAST-like software targetting multiple parallel hardware such as FPGA accelerator or GPU boards. [web site: http://www.irisa.fr/symbiose/projects/plast/]
- **SLICEE** (Service Layer for Intensive Computation Execution Environment) is part of the BioWIC project. This software proposes (1) to abstract the calls to the cluster scheduler by handling command submission; (2) to take care of exploring the data parallelism with data specific methods; (3) to manage data using a cache references mechanism and route data between tasks. [Web site: http://vapor.gforge.inria.fr/]
- **QTL-map** is a GPU parallel version of the QTLMap Software developed in cooperation with INRA [web site: http://www.inra.fr/qtlmap]

5.4. Softwares for Next Generation Sequencing data

Participants: Dominique Lavenier [contact], Pierre Peterlongo, Guillaume Rizk, Rayan Chikhi.

- **GASSST:** short reads mapper. See first section above.
- **kisSnp and kisSplice : variant identification without the use of a reference genome.** kisSnp is a tool to find single nucleotide polymorphisms (SNP) by comparing two sets of raw NGS reads. [web site: http://alcovna.genouest.org/kissnp/] KisSplice finds alternative splicings but also short insertions, deletions and duplications, SNPs and sequencing errors in one or two RNA-seq sets, without assembly nor mapping on a reference genome. [web site: http://alcovna.genouest.org/kissplice/]
- **Blastree:** is a tool for computing intensive approximate pattern matching in a string graph. [web site: http://alcovna.genouest.org/blastree/]
- **Mapsembler:** targeted assembly software. Mapsembler takes as input a set of NGS raw reads and a set of input sequences (starters). It first determines if each starter is read-coherent, e.g. whether the reads confirm the presence of each starter in the original sequence. Then for each read-coherent starter, Mapsembler outputs its sequence neighborhood as a linear sequence or as a graph, depending on the user choice. [web site: http://alcovna.genouest.org/mapsembler/]
5.5. Genome structure

Participants: Jacques Nicolas [contact], Catherine Belleannée, Pierre Peterlongo, Raoul Vorc’h, Anthony Bretaudeau, Olivier Sallou.

- **CRISPI: CRISPR identification.** CRISPI is a user-friendly web interface with many graphical tools and facilities allowing extracting CRISPR (Clustered Regularly Interspaced Short Palindromic Repeats), finding out CRISPR in personal sequences or calculating sequence similarity with spacers. web site: [http://crispi.genouest.org](http://crispi.genouest.org)

- **Logol** is a language and a tool to define biological patterns to look for in one or more sequences (dna/rna/proteins). Patterns can be complex: the tool allows the use of variables to look for repetitions for example, the use of gaps and morphisms (reverse word complement for example), etc. web site: [http://www.genouest.org/spip.php?article758](http://www.genouest.org/spip.php?article758)

5.6. Protein sequence and structure

Participants: Rumen Andonov [contact], François Coste, Andres Burgos, Pavel Senin.

- **A_purva: Scoring similarities between proteins.** A_purva is a Contact Map Overlap maximization (CMO) solver. Given two protein structures represented by two contact maps, A_purva computes the amino-acid alignment which maximizes the number of common contacts. web site: [http://apurva.genouest.org](http://apurva.genouest.org)

- **Protomata learner: fine characterization of protein families** See first section above.

5.7. Systems biology

Participants: Anne Siegel [contact], Michel Le Borgne, François Moreews, Anthony Bretaudeau.

- **Bioquali: confront knowledge-based regulatory models with data.** Bioquali tests the consistency between an interaction graph and transcriptomic data. It outputs nodes in the network whose variation cannot be globally explained by the other available observations. web site: [http://bioquali.genouest.org](http://bioquali.genouest.org)  Cytoscape java web start
5. Software

5.1. CHOCO

Participants: Nicolas Beldiceanu, Alexis De Clerq, Sophie Demasse, Jean-Guillaume Fages, Narendra Jussien [correspondant], Arnaud Letort, Xavier Lorca [correspondant], Arnaud Malapert, Julien Menana, Thierry Petit, Charles Prud’homme [correspondant].

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

1. Providing a complete solver independent specification of explanation algorithms, data structure for encoding «nogoods» and treatment algorithms. A reference implementation is being made within the new version of our solver CHOCO.
2. Design and development of a dedicated language to specify the propagation and the search heuristics of constraint solvers.
3. Providing efficient implementation of filtering algorithms for constraints such as tree, increasing_sum, cumulative with resource overload.
4. Providing an implementation of a probabilistic model for alldifferent.


5.2. IBEX

Participants: Gilles Chabert [correspondant], Rémi Douence.

IBEX (Interval-Based EXplorer) is a C++ library for solving nonlinear constraints over real numbers (25000 lines of source code). The main feature of Ibex is its ability to build solver/paver strategies declaratively through the contractor programming paradigm. Ibex includes a parser of the QUIMPER language (QUick Interval Modeling and Programming in a bounded-ERror context) and is currently used in several academic research labs.

G. Chabert and R. Douence (ASCOLA) have contributed in 2011 to the ongoing redesign of the architecture IBEX, the goal being to make it more flexible to cope with specific problems, and more easy to use. The link to the system and documentation is http://www.emn.fr/z-info/ibex/.

5.3. Global Constraint Catalog

Participants: Nicolas Beldiceanu [correspondant], Sophie Demasse, Mats Carlsson, Helmut Simonis.

The global constraint catalog presents and classifies global constraints and describes different aspects with meta data. It consist of

1. a pdf version that can be downloaded from http://www.emn.fr/z-info/sdemasse/gccat/ (at item last working version) containing 360 constraints, 3000 pages and 700 figures,
2. an on line version accessible from the previous address,
3. meta data describing the constraints (button PL for each constraint, e.g., alldifferent.pl),
4. an online service (i.e., a constraint seeker) which provides a web interface to search for global constraints, given positive and negative ground examples.
This year developments were focussing on:

1. maintaining the catalogue,
2. deploying an on-line constraint seeker [16] (see http://seeker.mines-nantes.fr/ and http://4c.ucc.ie/~hsimonis/seekerhelp.html for explanation how to use),
3. providing the negation for constraints defined by automata (with and without counter),
4. defining properties of constraints arguments, and
5. providing modelling examples as well as points of interests and common misunderstanding for core constraints.

N. Beldiceanu, S. Demassey, M. Carlsson (SICS, Sweden) and H. Simonis (4C, Ireland) have contributed in 2011. The link to the global constraint catalog is http://www.emn.fr/z-info/sdemasse/gccat/.
TEMICS Project-Team

5. Software

5.1. Oriented wavelet based image codec

**Participant:** Christine Guillemot [contact person].

This still image codec is based on oriented wavelet transforms developed in the team. The transform is based on wavelet lifting locally oriented according to multiresolution image geometry information. The lifting steps of a 1D wavelet are applied along a discrete set of local orientations defined on a quincunx sampling grid. To maximize energy compaction, the orientation minimizing the prediction error is chosen adaptively. This image codec outperforms JPEG-2000 for lossy compression. This software has been registered at the APP (Agence de Protection des Programmes) under the number IDDN.FR.001.260024.000.S.P.2008.000.21000.

5.2. M3DPlayer: 3D video player

**Participant:** Vincent Jantet [contact person].

A 3D player - named M3DPlayer - supporting rendering of a 3D scene and navigation within the scene has been developed. It integrates as a plug-in the 3D model-based video codec of the team. From a video sequence of a static scene viewed by a monocular moving camera, the 3D model-based video codec allows the automatic construction of a representation of a video sequence as a stream of textured 3D models. 3D models are extracted using stereovision and dense matching maps estimation techniques. A virtual sequence is reconstructed by projecting the textured 3D models on image planes. This representation enables 3D functionalities such as synthetic objects insertion, lightning modification, stereoscopic visualization or interactive navigation. The codec allows compression at very low bit-rates (16 to 256 kb/s in 25Hz CIF format) with a satisfactory visual quality. It also supports scalable coding of both geometry and texture information. The first version of the software was registered at the Agency for the Protection of Programmes (APP) under the number IDDN.FR.001.130017.000S.P.2003.000.41200.

A second version of the player has been registered at the APP (Agence de Protection des Programmes) under the number IDDN.FR.001.090023.000.S.P.2008.000.21000. In 2009-2010, we focused on improving the rendering engine, based on recent OpenGL extensions, to be able to render the viewed scenes on an auto-stereoscopic display with low-end graphic cards. In our case, auto-stereoscopic display requires the rendering of eight 1920x1200 frames instead of just one for a standard display. This player is also used to render LDI (Layered Depth Images) and LDV (Layered Depth Videos) and to visualize 3D scenes on autostereoscopic displays taking multiple input views rendered from the LDI representation.

5.3. Depth maps extractor in mono-view (M3dAnalyzer2)

**Participant:** Josselin Gauthier [contact person].

This software estimates depth maps from a video captured by a unique camera moving in a static 3D environment with Lambertian surfaces. These sequences are of interest to specialized applications such as augmented reality, remote-controlled robots operating in hazardous environments or remote exploration by drones. This software has been filed at the APP (Agence de Protection des Programmes) under the number IDDN.FR.001.110031.000.S.P.2010.000.31235.

5.4. Depth maps extractor in multi-view (MV2MVD)

**Participant:** Josselin Gauthier [contact person].
Figure 1. Depth maps extracted for the kendo sequence (right and left views) with DERS (3rd column) and TEMICS software (middle column).

This software estimates depth maps from multi-view videos, to provide Multi-View plus Depth (MVD) videos. MVD videos can be used to synthesize virtual views of the scene, or to render a different number of views than captured in the original video, for instance on an auto-stereoscopic display. This software produces depth maps of higher quality than those generated by the Depth Estimation Reference Software from the MPEG-3DV group, in terms of virtual views synthesis quality. This software has been filed at the APP (Agence de Protection des Programmes) under the number IDDN.FR.001.110034.000.S.P.2010.000.31235.

5.5. LDI builder

Participants: Vincent Jantet [contact person].

This software constructs a Layered Depth Image (LDI) representation of un-rectified Multi-View + Depth (MVD) sequences. The Incremental construction scheme reduces inter-layer correlation. The generated I-LDI is compatible with the M3DPlayer, permitting 3D visualisation and free viewpoint rendering of the 3D scene. The software also implements a virtual-view rendering technique which significantly reduces ghosting artefacts by eliminating untrusted texture boundaries detected in depth maps, as well as cracking artefacts thanks to an epipolar geometry aided inpainting method.

5.6. ADT PICOVIN

Participants: Ronan Boitard, Laurent Guillo [contact person], Thomas Guionnet, Tangi Poirier.

The ADT Picovin is a technological development action, which works closely with the project-team TEMICS. This is a development structure which gives its support to the project-team to integrate new and relevant algorithms into the state-of-the-art codec and to take part in standardization.

The ITU-T Study Group 16 (VCEG) and ISO/IEC JTC 1/SC 29/WG 11 (MPEG) have created in 2010 the Joint Collaborative Team on Video Coding (JCT-VC) in order to develop a new generation video coding standard that will further reduce by 50
In 2011, the ADT mainly focused on the improvement and integration of algorithms dedicated to intra prediction. A part of our work was integrated in the HM 1.0 and then submitted and presented as a proposal in Daegu during the 4th JCT-VC meeting in January 2011. Bit rate gains that we obtained were significant but to the detriment of encoding and decoding times. That is why, all along this year, the ADT tried to reach the best tradeoff between performances and encoding/decoding times. Our solution based on linear combination of template matching predictors has been improved by taking into account several shapes of template. The very last integration in the HM (HM4.0) shows that it performs well especially for the class “screen content”. A new very promising intra prediction method, Short Distance Intra Prediction (SDIP) is being tested as part of a Core Experiment in the HM 4.0. Once it is associated to our approach, bit save gains are additive. These results will be presented during the 7th JCT-VC meeting in Geneva in November 2011.

During 2011, the ADT also took part in cross checks which aims at evaluating and testing tools studied in core experiments. As part of cross checks the ADT has run 9 tests jointly with companies such as Technicolor, Mitsubushi, Huawei, Qualcomm and Canon.

This ADT started in October 2008. It will go on for one more year through the ADT PICOVIN-P. During this year, one permanent engineer from the SED Rennes (development and experimentation department of INRIA Rennes) and one senior engineer specialized in video compression are involved in the ADT. It is supported by the technological development department of INRIA.
TEXMEX Project-Team

5. Software

5.1. Software

5.1.1. New Software

5.1.1.1. Babaz

**Participants:** Jonathan Delhumeau, Guillaume Gravier, Hervé Jégou [correspondent].

*The deposit of this software at APP is currently being processed (submitted). The software is available from its homepage, namely [http://babaz.gforge.inria.fr/](http://babaz.gforge.inria.fr/).*

Babaz is a audio database management system with an audio-based search function, which is intended for audio-based search in video archives.

It is licensed under the terms of the GNU General Public License v3.0.

5.1.1.2. Bag-of-colors

**Participants:** Sébastien Campion [correspondent]. Hervé Jégou.

*Joint work with Christian Wengert (Kooba Inc.) and Matthijs Douze (INRIA LEAR and SED project-teams)*

This package implements the color descriptor proposed in our ACM Multimedia paper [48], which improves the previous color histogram representation.

The bag-of-colors software corresponds to two packages:

- The (reference) Matlab package, which was co-developed with Christian Wengert and Matthijs Douze;
- The python package (translated) was translated from Matlab by Sébastien Campion.

The Matlab version of this package is available on Github at [https://github.com/kooaba/bag-of-color/](https://github.com/kooaba/bag-of-color/).

The python version is available on the gforge INRIA server, and might be available on request.

5.1.1.3. BonzaiBoost

**Participant:** Christian Raymond [correspondent].

*The software homepage is available at [http://bonzaiboost.gforge.inria.fr/](http://bonzaiboost.gforge.inria.fr/).*

Bonzaiboost stands for boosting over small decisions trees. bonzaiboost is a general purpose machine-learning program based on decision tree and boosting for building a classifier from text and/or attribute-value data. Currently one configuration of bonzaiboost is ranked first on [http://mlcomp.org](http://mlcomp.org) a website which propose to compare several classification algorithms on many different datasets.

5.1.1.4. Don Quixotte

**Participant:** Teddy Furon [correspondent].

*This software was developed in collaboration with project-team TEMICS (P. Meerwald)*

Don Quixotte a software suite in C for Tardos Fingerprinting code (Code generation, collusion, and accusation with single and/or joint decoding).

5.1.1.5. Rare Event

**Participant:** Teddy Furon [correspondent].

*This software was developed in collaboration with project-team ASPI (F. Cérou, A. Guyader)*

Rare Event is a Matlab package for rare event probabilities and extreme quantiles estimations.
5.1.2. Most active software started before 2011

5.1.2.1. Bigimbaz

**Participant:** Hervé Jégou [correspondent].

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

Bigimbaz is a platform originally developed in the LEAR project-team, and now co-maintained by TEXMEX. It integrates several contributions on image description and large-scale indexing: detectors, descriptors, retrieval using bag-of-words and inverted files, and geometric verification.

5.1.2.2. kertrack

**Participant:** Sébastien Campion [correspondent].

Visual graphical interface for tracking visual targets based on particle filter tracking or based on mean-shift. The deposit of this software at APP is currently being processed.

5.1.2.3. mozaic2d

**Participant:** Sébastien Campion [correspondent].

Creation of spatio-temporal mosaic based on dominant motion compensation. It depends on the Motion2D library, which computes the dominant motion, and then adjust the images by back-warping. The deposit of this software at APP is currently being processed.

5.1.2.4. PimPy

**Participant:** Sébastien Campion [correspondent].

*The software homepage is available here:* [http://pim.gforge.inria.fr/pimpy/](http://pim.gforge.inria.fr/pimpy/).

*First APP deposit: IDDN.FR.001.260038.000.S.P.2011.000.40000*

PimPy stands for Indexing Multimedia with Python (or Platform for Indexing Multimedia with Python). The aim of this module is to provide a convenient and high level API to manage common multimedia indexing tasks. It includes severals features. It is used, in particular
- to retrieve video features, such as histogram, binarized DCT descriptor, SIFT, SURF, etc;
- to detect video cuts and dissolve (GoodShotDetector);
- for fast video frame access (pyffas);
- for raw frame extraction, or video segment extraction and re-encoding;
- to search a video segment in another video (content based retrieval);
- to perform scene clustering.

5.1.2.5. Pqcodes

**Participant:** Hervé Jégou [correspondent].

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

*First APP deposit: IDDN.FR.001.220012.000.S.P.2010.000.10000*

A new version of the software at APP is currently being processed.

Pqcodes is a library which implements the approximate k nearest neighbor search method of [18]. This software has been transferred to Technicolor in August 2011.

5.1.2.6. python-geohash

**Participant:** Sébastien Campion [correspondent].

*The deposit of this software at APP is currently being processed.*

Implementation of the Geometric Hashing algorithm of [85] to check if geometrical consistency between pairs of images.
5.1.2.7. Samusa

**Participant:** Sébastien Campion [correspondent].

*This software is jointly maintained with Guillaume Gravier.*

Samusa enable to detect speech and/or musical segment in multimedia content.

5.1.2.8. Yael

**Participant:** Hervé Jégou [correspondent].

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

**APP deposit:** IDDN.FR.001.220014.000.S.P.2010.000.10000

A new version of the software at APP is currently being processed.

Yael is a C/python/Matlab library providing (multi-threaded, Blas/Lapack, low level optimization) implementations of computationally demanding functions. In particular, it provides very optimized functions for k-means clustering and exact nearest neighbor search.

5.1.2.9. TVSearch

**Participant:** Sébastien Campion [correspondent].

TVSearch is a content based retrieval search engine used to search and propagate manual annotation such as advertisement in a TV corpora. Based on a binary DCT descriptor, it used GPU card to compute exhaustive Hamming distance between the query and database. For example, a query of 11 seconds in 21 days on television (504 hours) is done in 9 seconds. (i.e., bit-rate of 2,3 days/second) TVSearch offer a web services API using the HTTP/REST protocol.

The deposit of this software at APP is currently being processed.

5.1.2.10. AVSST

**Participant:** Sébastien Campion [correspondent].

AVSST is an Automatic Video Stream Structuring Tool. First, it allows the detection of repetitions in a TV stream. Second, a machine learning method allows the classification of programs and inter-programs such as advertisements, trailers, etc. Finally, the electronic program guide is synchronized with the right timestamps based on dynamic time warping. A graphical user interface is provided to manage the complete workflow.

5.1.3. Other softwares

Several software programs have been developed in the team over the years:

**I-D**

(I-D DESCRIPTION (APP deposit number: IDDN.FR.001.270047.000.S.P.2003.000.21000)),

**ASARES,** is a symbolic machine learning system that automatically infers, from descriptions of pairs of linguistic elements found in a corpus in which the components are linked by a given semantic relation, corpus-specific morpho-syntactic and semantic patterns that convey the target relation. (IDDN.FR.001.0032.000.S.C.2005.000.20900),

**ANA M ORPHO,** detects morphological relations between words in many languages (IDDN.FR.001.050022.000.S.P.2008.000.20900),

**DIVATEX** is a audio/video frame server. (IDDN.FR.001.320006.000.S.P.2006.000.40000),

**NAVITEX** is a video annotation tool. (IDDN.FR.001.190034.000.S.P.2007.000.40000),

**TELEMEX** is a web service that enables TV and radio stream recording.

**VIDSIG** computes a small and robust video signature (64 bits per image).

**VIDSEG** computes segmentation features such as cuts, dissolves, silences in audio track, changes of ratio aspect, monochrome images. (IDDN.FR.001.250009.000.S.P.2009.000.40000),

**ISEC,** web application used as graphical interface for image searching engines based on retrieval by content.

**GPU-KMEANS,** implementation of k-means algorithm on graphical process unit (graphic cards)

**CORRESPONDENCE ANALYSIS** computes a factorial correspondence analysis (FCA) for image retrieval.
GPU CorrESpondence Analysis, is an implementation of the previous software Correspondence Analysis on graphical processing unit (graphical card).

CAVIZ is an interactive graphical tool that allows to display and to extract knowledge from the results of a Correspondence Analysis on images.

KIWI (standing for Keywords Extractor) is mostly dedicated to indexing and keyword extraction purposes.

Topic Segmenter, is a software dedicated to topic segmentation of texts and (automatic) transcripts.

S2E (Structuring Events Extractor) is a module which allows the automatic discovery of audiovisual structuring events in videos.

2PAC, build classes of words of similar meanings ("semantic classes") specific to the use that is made of them in that given topic. (IDDN.FR.001.470028.000.S.P.2006.000.40000)

FAESTOS, (Fully Automatic Extraction of Sets of keywords for TOpic characterization and Spotting) is a tool composed of a sequence of statistical treatments that extracts from a morpho-syntactically tagged corpus sets of keywords that characterize the main topics that corpus deals with. (IDDN.FR.001.470029.000.S.P.2006.000.40000)

FISHNET, Fishnet is an automatic web pages grabber associated with a specific theme.

MATCH MAKER, semantic relation extraction by statistical methods.

IRISA News Topic Segmenter (IRINTS), automatically segments speech transcripts into topic-consistent parts.

IRISAPHON, produce phonetic words.

5.2. Demonstrations

Participants: Morgan Bréhinier, Sébastien Campion [correspondent], Guillaume Gravier.

The gradual migration of television from broadcast diffusion to Internet diffusion offers tremendous possibilities for the generation of rich navigable contents. However, it also raises numerous scientific issues regarding de-linearization of TV streams and content enrichment. In this demonstration, we illustrate how speech in TV news shows can be exploited for de-linearization of the TV stream. In this context, de-linearization consists in automatically converting a collection of video files extracted from the TV stream into a navigable portal on the Internet where users can directly access specific stories or follow their evolution in an intuitive manner.

Structuring a collection of news shows requires some level of semantic understanding of the content in order to segment shows into their successive stories and to create links between stories in the collection, or between stories and related resources on the Web. Spoken material embedded in videos, accessible by means of automatic speech recognition, is a key feature to semantic description of video contents. At IRISA/INRIA Rennes, we have developed multimedia content analysis technology combining automatic speech recognition, natural language processing and information retrieval to automatically create a fully navigable news portal from a collection of video files.

The demonstration was presented in several workshops (Quaero CTC workshop, Journée INRIA Industrie La Télévision du Futur) and a video has been made available online on the portal of the EIT ICT Labs OpenSEM project.

See the demo at http://texmix.irisa.fr.

5.3. Experimental platform

Participants: Laurent Amsaleg, Sébastien Campion [correspondent], Patrick Gros, Pascale Sébillot.
Until 2005, we used various computers to store our data and to carry out our experiments. In 2005, we began some work to specify and set-up dedicated equipment to experiment on very large collections of data. During 2006 and 2007, we specified, bought and installed our first complete platform. It is organized around a very large storage capacity (155TB), and contains 4 acquisition devices (for Digital Terrestrial TV), 3 video servers, and 15 computing servers partially included in the local cluster architecture (IGRIDA).

In 2010, we have acquired a new large memory server with 144GB of RAM which is used for memory demanding tasks, in particular to improve the speed of building index or language model. The previous server dedicated to this kind of jobs (acquired in 2008) has been upgraded to 96GB of RAM.

A dedicated website has been developed in 2009 to provide a user support. It contains useful information such as references of available and ready to use software on the cluster, list of corpus stored on the platform, pages for monitoring disk space consumption and cluster loading, tutorials for best practices and cookbooks for treatments of large datasets.

In 2008, we build up a corpus of multimedia data. It consists in a continuous recording (6 months) of two TV channels and three radios. It also includes web pages related to these contents captured on broadcaster’s website. This corpus is to be used for different studies like the treatment of news along the time and to provide sub-corpus like TV news within the Quaero project (see below). The manual annotation of all the TV programs is under progress.

This platform is funded by a joint effort of INRIA, INSA Rennes and University of Rennes 1.
TRISKELL Project-Team

5. Software

5.1. Kermeta

Participants: Didier Vojtisek [correspondant], Olivier Barais, Cédric Bouhours, Xavier Dolques, Jacques Falcou, François Fouquet, Marie Gouyette, Jean-Marc Jézéquel, Hajanirina Johary Rambelontsalama.

Nowadays, object-oriented meta-languages such as MOF (meta-object Facility) are increasingly used to specify domain-specific languages in the model-driven engineering community. However, these meta-languages focus on structural specifications and have no built-in support for specifications of operational semantics. Integrated with the industrial standard Ecore and aligned with the OMG standard EMOF 2.0, the Kermeta language consists in a extension to these meta languages to support behavior definition. The language adds precise action specifications with static type checking and genericity at the meta level. Based on object-orientation and aspect orientation concepts, the Kermeta language adds model specific concepts. It is used in several use cases:

- to give a precise semantic of the behavior of a metamodel which then can be simulated.
- to act as a model transformation language.
- to act as a constraint language.

The development environment built for the Kermeta language provides an integrated workbench based on Eclipse. It offers services such as : model execution, text editor (with syntax highlighting, code autocompletion), additional views and various import/export transformations.

Thanks to Kermeta it is possible to build various frameworks dedicated to domain specific metamodels. Those frameworks are organised into MDKs (Model Development Kits). For example, Triskell proposes MDKs to work with the metamodels such as Java5, UML2, RDL (requirements), Ecore, Traceability,...

In 2011, Kermeta tooling has been refactored into a version 2.0.x in order to ease the integration of various MOF related languages in the tool chain. This new version also focuses on a fully compiled mode that allows to deploy kermeta programs in production environments.

See also the web page: [http://www.kermeta.org](http://www.kermeta.org).

- APP: IDDN.FR.001.420009.000.S.P.2005.000.10400
- Version: 2.0.1
- Programming language: Java, Scala, Kermeta

5.2. Kevoree

Participants: Olivier Barais [correspondant], François Fouquet, Erwan Daubert, Johann Bourcier, Gregory Nain, Noël Plouzeau.

The Kevoree project[^1] defines a framework dedicated to distributed systems design, using the *models at runtime* paradigm and a component-based software architecture approach. This framework offers a high-level abstraction for managing components and their interactions. It also provides concepts to describe the underlying infrastructure: resources, logical nodes and their topology.

Kevoree also provides a set of tools to manipulate model abstraction easily, relying in part on a Domain Specific Language (DSL) called KevScript. This DSL makes the architecture model modifications easier. Our DSL can also be used in a reasoning engine to dynamically adapt the running system by applying some changes at different level (SaaS, PaaS and IaaS). Kevoree has several runtime platform implementations, allowing execution of applications on various devices (*e.g.* JavaSeE, Android, µController such as Arduino, and cloud virtual nodes).

[^1]: [http://kevoree.org](http://kevoree.org)
5.3. Pramana

Participants: Benoit Baudry, Juan-Jose Cadavid Gomez, Benoit Combemale, Xavier Dolques, Hajanirina Johary Rambelontsalama, Didier Vojtisek [correspondant].

Pramana is an open-source tool, which automatically generates valid instances of a metamodel. These instances can then be used for analysis, verification, simulation or validation of the metamodel. The core mechanism for model generation relies on the bounded constraint-solver of Alloy, a lightweight model checker developed at the MIT. Alloy is integrated in Kermeta to allow the generation of instances of Ecore or Kermeta metamodels. Pramana implements this integration through a series of transformations and analysis, all implemented in Kermeta.

Metamodel instances can be used as input data for model transformation testing, and in particular for the testing of Kermeta code. For this purpose Pramana includes the K-Yeti module implementing a binding between the Kermeta language and the generic testing framework Yeti. This module allows a Kermeta user to automatically run test cases.

See also the web page https://www.irisa.fr/triskell/Softwares/pramana.

- Version: 1.0
- Programming language: Java, Alloy, Kermeta
5. Software

5.1. TGV

**Participant:** Thierry Jéron.

TGV (Test Generation with Verification technology) is a tool for test generation of conformance test suites from specifications of reactive systems \[^4\]. It is based on the IOLTS model, a well defined theory of testing, and on-the-fly test generation algorithms coming from verification technology. Originally, TGV allows test generation focused on well defined behaviors formalized by test purposes. The main operations of TGV are (1) a synchronous product which identifies sequences of the specification accepted by a test purpose, (2) abstraction and determinisation for the computation of next visible actions, (3) selection of test cases by the computation of reachable states from the initial states and co-reachable states from accepting states. TGV has been developed in collaboration with Vérimag Grenoble and uses libraries of the CADP toolbox (VERIMAG and VASY). TGV can be seen as a library that can be linked to different simulation tools through well defined APIs. An academic version of TGV is distributed in the CADP toolbox and allows test generation from Lotos specifications by a connection to its simulator API. TGV has been registered at APP (Agence de Protection des Programmes) under deposit number IDDN.FR.001.310012.00.R.P.1997.000.2090.

5.2. STG

**Participant:** Thierry Jéron.

STG (Symbolic Test Generation) is a prototype tool for the generation and execution of test cases using symbolic techniques. It takes as input a specification and a test purpose described as IOSTS, and generates a test case program also in the form of IOSTS. Test generation in STG is based on a syntactic product of the specification and test purpose IOSTS, an extraction of the subgraph corresponding to the test purpose, elimination of internal actions, determinisation, and simplification. The simplification phase now relies on NBAC, which approximates reachable and coreachable states using abstract interpretation. It is used to eliminate unreachable states, and to strengthen the guards of system inputs in order to eliminate some *Inconclusive* verdicts. After a translation into C++ or Java, test cases can be executed on an implementation in the corresponding language. Constraints on system input parameters are solved on-the-fly (i.e. during execution) using a constraint solver. The first version of STG was developed in C++, using Omega as constraint solver during execution. This version has been deposited at APP under number IDDN.FR.001.510006.000.S.P.2004.000.10600.

A new version in OCaml has been developed in the last years. This version is more generic and will serve as a library for symbolic operations on IOSTS. Most functionalities of the C++ version have been re-implemented. Also a new translation of abstract test cases into Java executable tests has been developed, in which the constraint solver is LUCKYDRAW (VERIMAG). This version has also been deposit at APP and is available for download on the web as well as its documentation and some examples.

Finally, in collaboration with ULB, we implemented a prototype SMACS, derived from STG, that is devoted to the control of infinite system modeled by STS.

5.3. SIGALI

**Participant:** Hervé Marchand.
SIGALI is a model-checking tool that operates on ILTS (Implicit Labeled Transition Systems, an equational representation of an automaton), an intermediate model for discrete event systems. It offers functionalities for verification of reactive systems and discrete controller synthesis. It is developed jointly by the ESPRESSO and VERTECS 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 \cite{6} \cite{28}. SIGALI is connected with the Polychrony environment (ESPRESSO 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.
5. Software

5.1. Vistal

Participant: Alexandre Abadie.

VistaL is a software platform of 3D and 3D+t image analysis allowing the development of generic algorithms used in different contexts (rigid and non-rigid registration, segmentation, statistical modelling, calibration of free-hand 3D ultrasound system and so on, diffusion tensor image processing, tractography). This software platform is composed of generic C++ template classes (Image3D, Image4D, Lattice and so on) and a set of 3D/3D+t image processing libraries. VistaL is a multi-operating system environment (Windows, Linux/Unix...). A web site presenting the project has been developed, precompiled packages and the SDK are now available. VistaL APP registration number is: IDDN.FR.001.200014.S.P.2000.000.21000. See also the web page http://vistal.gforge.inria.fr.

![Figure 1](image)

(a) (b) (c)

Figure 1. Some ViSTAL results screenshots: a) The ViSTAL Logo, b) ViSTAL Brain surface and sulci modelisation, c) The ROI3D Extraction view

- Keywords: medical image processing, image analysis, registration, segmentation, denoising
- Software benefit: New methodological image processing, some GPU based algorithms, easy to use C++ library
- APP: IDDN.FR.001.200014.S.P.2000.000.21000
- License: Licence Propriétaire
- Type of human computer interaction: C++ API and less complete Python API
- OS/Middleware: Windows, Mac et Linux
- Required library or software: CMak (GPL) - ITK (BSD) - VTK (BSD) - Boost (BSD) - Libxml++ (LGPL) - CppUnit (LGPL)
- Programming language: C/C++, Python
- Documentation: Documentation Doxygen, documentation utilisateur.
5.2. Vistal-Tools

**Participant:** Alexandre Abadie.

The Vistal-Tools are a set of command line binaries based on the VisTaL library. These programs allow users to perform batch mode processing as well as scripting complex processing workflows. The most popular Vistal-Tools are NLMEANS (perform a NLMEANS filtering of 3D or 4D volumes), Registration (encapsulate the most common rigid registration algorithms), Tractography (track fibers from a DTI volume), etc.

5.3. Online applications

**Participant:** Alexandre Abadie.

Online applications offers a web service for testing the tools developed by the members of the VISAGES team: denoising based on Non Local Mean algorithm (3D and 2D) (NLMEAN), 3D rigid registration, brain symmetry plan estimation. This application support the main formats used in medical imaging data: Nifti-1, Analyze7.5, Mha, GIS. The applications are available at this url [http://www.irisa.fr/visages/benchmarks](http://www.irisa.fr/visages/benchmarks). More than 2000 processes have been benchmarked to date using this service.

5.4. CLARCS: C++ Library for Automated Registration and Comparison of Surfaces

**Participants:** Alexandre Abadie, Sylvain Prima.

In collaboration with Benoit Combes, within the 3D-MORPHINE ARC project ([http://3dmorphine.inria.fr](http://3dmorphine.inria.fr)), we conceived and implemented a C++ library (named CLARCS) for the automated analysis and comparison of surfaces. One of the primary goal of this library is to allow the assessment and quantification of morphological differences of free-form surfaces from medical or paleoanthropological data.

- **APP:** IDDN.FR.001.130002.000.S.P.2011.000.21000
- **Programming language:** CC++

CLARCS was presented at the MeshMed MICCAI workshop ([http://www2.imm.dtu.dk/projects/MeshMed/2011/index.html](http://www2.imm.dtu.dk/projects/MeshMed/2011/index.html)) [27] and is to be distributed through a dedicated website ([http://clarcs.inria.fr](http://clarcs.inria.fr)). We also developed a surface viewer (named 'Surface').

- **APP:** IDDN.FR.001.110019.000.S.P.2011.000.21000
- **Programming language:** C++, Python

5.5. SUBANA: SUrface-BAsed Neuronavigation on Atlas for TMS

**Participant:** Sylvain Prima.

In collaboration with Charles Garraud ([http://www.syneika.com](http://www.syneika.com)), Benoit Combes and Pierre Hellier ([http://serpico.rennes.inria.fr](http://serpico.rennes.inria.fr)), we developed a software for i) the automated surface reconstruction of the face and skull cap from sparsely acquired points and ii) the automated nonlinear registration of free-form surfaces. The latter step is implemented using the CLARCS library ([http://clarcs.inria.fr](http://clarcs.inria.fr)). The primary goal of this software is the surface-based neuronavigation for transcranial magnetic stimulation. The method was presented at the MeshMed MICCAI workshop ([http://www2.imm.dtu.dk/projects/MeshMed/2011/index.html](http://www2.imm.dtu.dk/projects/MeshMed/2011/index.html)) [30].

- **APP:** IDDN.FR.001.440010.000.S.P.2010.000.31230
- **Patent:** was granted, but the reference number is unknown
- **Programming language:** C++

5.6. Shanoir

**Participants:** Guillaume Renard, Alexandre Abadie, Bernard Gibaud, Christian Barillot.
Shanoir (Sharing NeurOImaging Resources) is an open source neuroinformatics platform designed to share, archive, search and visualize neuroimaging data. It provides a user-friendly secure web access and offers an intuitive workflow to facilitate the collecting and retrieving of neuroimaging data from multiple sources and a wizard to make the completion of metadata easy. Shanoir comes along many features such as anonymization of data, support for multi-centres clinical studies on subjects or group of subjects.

Shanoir APP registration number is: IDDN.FR.001.520021.000.S.P.2008.000.31230
See also the web page http://www.shanoir.org

- Keywords: neuroimaging, ontology, sharing neuroimage
- Software benefit: full featured neuroimaging management system with additional web services
- APP: IDDN.FR.001.200014.S.P.2000.000.21000
- 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, Richfaces, Faceless, Ajax4JSF, Dcmntk, Dcm4chee.
- Programming language: Java
- Documentation: see the website

5.7. QtShanoir

Participants: Alexandre Abadie, Olivier Commowick, Guillaume Renard.

QtShanoir is a C++ Qt based library for querying data from a Shanoir server. For those who don’t know what is shanoir, see the shanoir website at http://shanoir.org. QtShanoir uses the soap based webservice provided by a shanoir server to get and display studies, patients, data with their associated metadata. In QtShanoir, you will find a set of Qt widgets (inherited from a QWidget object) that you can embed in your Qt application.

An APP registration is in progress and the library has been released in October under the LGPL license. See http://qtshanoir.gforge.inria.fr.

- Keywords: medical imaging, dicom
- Software benefit: offers a great solution to query a Shanoir server. Can be easily re-used in larger Qt applications
- License: no defined licence for the moment
- Type of human computer interaction: C++ library
- OS/Middleware: Linux, Windows and Mac
- Required library or software: Qt
- Programming language: C++
- Documentation: http://qtshanoir.gforge.inria.fr/html

5.8. QtDcm

Participant: Alexandre Abadie.

QtDcm is a C++ library implementing a widget that can be re-used with the Qt development framework. With this new widget, it is now easy to view the content of a Dicom CD-Rom, to manage Dicom Query/Retrieve from a PACS and to convert downloaded data in the nifti format (easy to use medical image format). QtDcm APP registration number (2010) is: IDDN.FR.001.490036.000.S.P.2010.000.31230 A new APP registration is in progress and the library has been released in October under the LGPL license. See http://qtdcm.gforge.inria.fr.
Computational Medicine and Neurosciences - Software - Project-Team VISAGES

See also the web page https://www.irisa.fr/visages/members/aabadie/demos

- Keywords: medical imaging, dicom
- Software benefit: offers a great solution to query medical images storage server (Dicom PACS). Can be easily re used in larger Qt applications
- APP: IDDN.FR.001.490036.000.S.P.2010.000.31230
- License: no defined licence for the moment
- Type of human computer interaction: C++ library
- OS/Middleware: Linux, Windows and Mac
- Required library or software: Qt, Dcmtk, dcm2nii (optional)
- Programming language: C++
- Documentation: http://qtdcm.gforge.inria.fr/html

5.9. AutoMRI

Participant: Camille Maumet.

autoMRI is an SPM-based set of tools to study structural and functional MRI data. This software is currently made up of three modules: autofMRI, autoVBM and autoROI. autofMRI produces statistical maps of activations and deactivations at the group or the subject level based on functional MRI data. It can deal with block or event-related designs and is highly configurable in order to fit to a wide range of needs. autoVBM performs between-group voxel-based morphometric analysis in order to outline regions of grey (or white) matter volume reduction and increase. To further study a morphometric or a functional analysis, regions of interest analysis can be performed with autoROI. This module also provides the user with laterality indexes.

- Keywords: fMRI, MRI, SPM, automation
- Software benefit: Automatic MRI data analysis based on SPM. Once the parameters are set, the analysis can be run without human interaction.
- APP: Coming soon
- License: Ceccil
- Type of human computer interaction: Matlab function (script, no GUI)
- OS/Middleware: Linux/Windows
- Required library or software: Matlab, SPM, SPM toolboxes: Marsbar, LI-toolbox, NS
- Programming language: Matlab
- Documentation: Available

5.10. Medinria

Participants: Alexandre Abadie, Clément Philipot, 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) for two years, starting from late 2010. Th Visages team 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 is currently being packaged for the main distribution platforms and will be released in the first two weeks of January 2012.
See also the web page  http://med.inria.fr

- Keywords: medical imaging, diffusion imaging, registration, filtering, user-friendly interface
- Software benefit: user-friendly interface to cutting-edge research tools for research clinicians. Straightforward to add functionalities through plugins.
- License: core: BSD, plugins: choice of each team.
- Type of human computer interaction: Qt-based GUI
- OS/Middelware: Windows, Mac et Linux.
- Required library or software : Qt, DTK, ITK, VTK.
- Programming language: C++

5.11. EMPROS

Participant: Elise Bannier.

EMPROS stands for "Event Related Emotional Prosody Recognition fMRI Task". This software implements a paradigm, i.e., a sequence of stimuli to be proposed to a subject, in order to study the perception of emotions with functional MRI. The subject hears meaningless but emotionally charged pseudo-words or onomatopoeas and selects the evoked emotion among 5 emotions (joy, fear, sadness, anger, neutral) by pushing a button. The response of the subject is registered while a BOLD fMRI acquisition images his/her brain. This paradigm aims at detecting the cortical areas involved in emotional perception.

This software will be distributed as open source code.

- APP: IDDN.FR.: APP registration in progress.
- Patent: under application
- Programming language: E-Basic
- Programming software: E-Prime v2.0

5.12. IOGAT

Participant: Elise Bannier.

IOGAT stands for "Iowa Gambling Task for Event Related fMRI". This software implements a paradigm, i.e., a sequence of stimuli to be proposed to a subject, in order to study the decision making process with functional MRI. The subject is presented with 4 decks of cards. Each deck is associated with a gain or a loss of money in a non random way: 2 of them are advantageous to the subject whereas the other 2 are disadvantageous. The subject is asked to pick up cards, choosing freely the deck he/she picks up the card from, so as to maximize his/her gains. While the subject performs this task, his/her brain is imaged with a BOLD fMRI acquisition. This paradigm is designed to localize the cortical areas involved in the decision making process.

This software will be distributed as open source code.

- APP: IDDN.FR.: APP registration in progress.
- License: the software is being licensed to CHU Besancon
- Patent: under application
- Programming language: E-Basic
- Programming software: E-Prime v2.0
5. Software

5.1. OpenMASK: Open-Source platform for Virtual Reality

Participants: Alain Chauffaut [contact], Ronan Gaugne [contact], Georges Dumont, Thierry Duval, Laurent Aguereotype, Florian Nouviale.

OpenMASK (Open Modular Animation and Simulation Kit) is a federative platform for research developments in the VR4i team. Technology transfer is a significant goal of our team so this platform is available as OpenSource software (http://www.openmask.org).

OpenMASK is a C++ software platform for the development and execution of modular applications in the fields of animation, simulation and virtual reality. The main unit of modularity is the simulated object (OSO) which can be viewed as frequent or reactive motors. It can be used to describe the behavior or motion control of a virtual object as well as input devices control like haptic interfaces. Two OSO communicate with synchronous data flows or with asynchronous events.

We provide Model Driven Tools to help building OpenMASK applications without tedious and repeated coding and to improve reusability. Within Eclipse environment we offer an editor and a C++ code generator to design and build objects classes. The current OpenMASK 4.2 release is now based on MPI for distribution service, Ogre3D for visualization service. One can benefit of new interaction tools for local or remote collaborative applications.

5.2. GVT : Generic Virtual Training

Participants: Bruno Arnaldi, Valérie Gouranton [contact], Florian Nouviale, Andrés Saraos-Luna.

The aim of GVT software is to offer personalized VR training sessions for industrial equipments. The most important features are the human and equipment security in the VR training (in opposition to the real training), the optimization of the learning process, the creation of dedicated scenarios, multiple hardware configurations: laptop computer, immersion room, distribution on network, etc.

The actual kernel of GVT platform is divided into two main elements that rely on innovative models we have proposed: LORA and STORM models.

- A Behavior Engine. The virtual world is composed of behavioral objects modeled with STORM (Simulation and Training Object-Relation Model).
- A Scenario Engine. This engine is used to determine the next steps of the procedure for a trainee, and its state evolves as the trainee achieves actions. The scenario is written in the LORA language (Language for Object-Relation Application).

A commercialized version of GVT, which includes a pedagogical engine developed in CERV laboratory, proposes training on individual procedures. A prototype is also available that enables users to train on collaborative procedures with one another or with virtual humans.

In the ANR Corvette 7.1.4 and in the FUI SIFORAS 7.1.2, new features of GVT Software are proposed.

5.3. OpenViBE Software

Participants: Anatole Lécuyer [contact], Laurent Bonnet, Jozef Legény, Yann Renard.

OpenViBE is a free and open-source software devoted to the design, test and use of Brain-Computer Interfaces.
The OpenViBE platform consists of a set of software modules that can be integrated easily and efficiently to design BCI applications. Key features of the platform are its modularity, its high-performance, its portability, its multiple-users facilities and its connection with high-end/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 LGPL-V2 licence, and it was officially released in June 2009. Since then, the OpenViBE software has already been downloaded more than 300 time, and it is used by numerous entities worldwide.

Our first international tutorial about OpenViBE was held at the International BCI Meeting in June 2010 (Monterey, US), with around 30 participants.

More information, downloads, tutorials, documentation, videos are available on OpenViBE website: http://openvibe.inria.fr