Activity Report 2011

Section Software

Edition: 2012-03-22
ADAM Project-Team

5. Software

5.1. Introduction

We intend to develop a number of software to evaluate and validate our solutions. We will complete our development by experimentation, benchmarks and deployment in multi-paradigm platforms. We list our actual software that we intend to continue and to extend in the ADAM project-team.

5.2. CALICO

Participants: Laurence Duchien, Antonio de Almeida Souza Neto, Anne-Françoise Le Meur.

Modern software is characterized by a need for constant and rapid evolution, such as in the mobile domain. To facilitate the development and the rapid evolution of complex systems, software engineering approaches have been proposed, such as software architecture and agile software development. However, current solutions offer poor support to enable the development of a reliable system.

In this context we propose CALICO, an agile development framework for the design and evolution of safe component-based and service-oriented software. The agile software development relies on an iterative and incremental development cycle that allows the architect to iterate between the design of the architecture and the debug of the software in its execution context. At each iteration, the architect can evolve its software and check the consistency of its evolution through the execution of static and dynamic analysis tools. Thus, during the design and the evolution of the system, the architect can use a set of metamodels to specify the structure of the architecture and its various quality of services requirement. During the deployment, CALICO instantiates the system on the target runtime platform from the models specified and keeps them synchronized with the software during its execution. Through this means, the architect has a conceptual view which allows him to reason on the critical software properties during its evolution. Moreover, in order to check these evolutions, CALICO provides a unifying framework which allows reuse of many static analysis tools of software architectures and dynamic debugging tools, that were scattered in different existing platforms. Thus, each change can be statically analyzed on the conceptual view before being propagated to the software system. Dynamic analysis are based on data values available during the execution only. The capture of these values is done through automatic instrumentation of the software system.

Globally, CALICO enables reliable evolution even if the underlying platforms does not natively provide this support. The current version handles four component-based and service-oriented platforms. Moreover, the benchmarks that we have performed show that CALICO is usable for the design and development of safe applications up to 10,000 components and services, which corresponds to the maximal load of most runtime platforms. CALICO has been developed in the context of Guillaume Waignier’s PhD thesis [79].

CALICO is an open source software available at http://calico.gforge.inria.fr.

5.3. Fractal

Participants: Christophe Demarey, Philippe Merle [correspondant], Romain Rouvoy, Lionel Seinturier.

FRACTAL is a modular, extensible and programming language agnostic component model that can be used to design, implement, deploy and reconfigure systems and applications, from operating systems, middleware platforms to graphical user interfaces [53], [55], [67]. FRACTAL has been designed by both Inria and France Telecom R&D.

FRACTAL is also a LGPL open source software project hosted by the OW2 international consortium and is available at http://fractal.ow2.org [59].
Philippe Merle is the leader of the OW2 FRACTAL open source project. The ADAM project-team actively contributes to this project, and more specifically on the following modules:

- **AOKell** is an aspect-oriented implementation of the FRACTAL component model [76].
- **Fraclet** is an attribute-oriented programming model enabling the rapid development of FRACTAL components [73].
- **Fractal ADL** is the extensible architecture definition language for FRACTAL associated to an open FRACTAL component-based toolchain.
- **Fractal Distribution** is the module to produce packaged releases of the FRACTAL project.
- **Fractal Documentation** is the module to produce the whole documentation of the FRACTAL project.
- **Fractal Eclipse Plugin** is a plugin to create FRACTAL projects within the Eclipse IDE [57], [58]. This work was supported by an Inria ODL and is contributed to the FUI MIND project.
- **Fractal Explorer** is a framework to build graphical consoles to introspect and manage FRACTAL components dynamically at runtime.
- **FScript** is a scripting language for both introspection and reconfiguration of FRACTAL software systems.
- **Juliac** is an extensible framework for generating and compiling the code of FRACTAL component-based systems. Juliac is registered with the APP (Agence pour la Protection des Programmes) under reference FR.001.230007.000.S.P.2009.000.10600.
- **Koch** is an implementation of the FRACTAL component model where components have a component-based control membrane.

### 5.4. FraSCAti

**Participants:** Christophe Demarey, Damien Fournier, Rémi Méliisson, Philippe Merle [correspondant], Christophe Munilla, Romain Rouvoy, Lionel Seinturier.

FraSCAti is a runtime platform for the Service Component Architecture (SCA) component framework. SCA is an initiative for unifying Service Oriented Architectures (SOA) and Component-Based Software Engineering (CBSE). SCA is supported by the Open SOA consortium, which includes partners, such as IBM, Oracle, Sun and Iona, and is standardized by the OASIS consortium (see at [http://www.oasis-opencsa.org/scac](http://www.oasis-opencsa.org/scac)).

FraSCAti includes Tinfi, which provides a SCA personality for the FRACTAL component model. Thanks to the openness of this latter model, the necessary code elements (so called controllers and membranes) have been designed and developed to customize FRACTAL and to end up with components owning both a FRACTAL personality and a SCA personality. As far as we know, this result, which has been presented in [75], is original and is the first one to concretely demonstrates that FRACTAL is open and flexible enough to implement different component personalities. Moreover, Tinfi reuses the aspect-oriented concepts defined in FAC [70] for component-based programming and allows integrating smoothly non functional concerns (so called intents and policy sets in SCA terms). FraSCAti and Tinfi have been implemented by reusing modules developed in the context of the FRACTAL project, and among others, the Juliac FRACTAL compiler.

The development of the FraSCAti platform is conducted in the context of some current and past funded projects (ICT FP7 SOA4All Integrated Project, ANR ARPEGE ITEmIS project, FUI EasySOA project, Inria ADT Adapt).

FraSCAti is a LGPL open source software, hosted by the OW2 consortium since November 2008 at [http://frascati.ow2.org](http://frascati.ow2.org). FraSCAti is registered with the APP (Agence pour la Protection des Programmes) under reference FR.001.050017.000.S.P.2010.000.10000.

### 5.5. SPACES

**Participants:** Russel Nzekwa, Daniel Romero [correspondant], Romain Rouvoy, Lionel Seinturier.
SPACES is a context mediation middleware that follows the REpresentational State Transfer (REST) principles [61]. The current implementation of SPACES is based on the COSMOS context framework [54], [72] and the COMANCHE web server [53]. Both COSMOS and COMANCHE are based on the FRACTAL component model and use the JULIA implementation of the FRACTAL runtime environment [53].

The main features of the current SPACES implementation are presented below:

1. **Ubiquitous connectors**: SPACES defines connectors that encapsulate the distribution concern. These connectors expose the COSMOS context nodes as REST resources with logical associated URLs, and enable interactions between consumers and producers via different communication protocols and the discovery of the available context sources. The current SPACES implementation supports interaction using the HTTP and twitter [65] protocols. For discovery, the implementation uses the Service Location Protocol (SLP) [63].

2. **Context Representation**: Following the REST principles, SPACES supports multiple representations of the context information: JSON [56], XML and Java serialization.

3. **Quality of context (QoC) information**: The QoC properties are incorporated as service attributes in the SLP advertisements of the context information.

4. **Context selection**: The restrictions in terms of QoC of the required context information are expressed as LDAP filters [77]. SPACES benefits from the LDAP based queries of SLP to select the context providers.

We use XStream 1.3.13 [51] and JSON-lib 2.2.34 [50] to serialize context information as XML and JSON documents. For SLP and twitter we employ jSLP 1.0.0 [71] and twitter4j 2.0.6 [80].

SPACES is registered with the APP (Agence pour la Protection des Programmes) under reference IDDN 10-500002-000.

5.6. **ApplIDE**

**Participants**: Laurence Duchien, Christophe Demarey, Clément Quinton [correspondant].

ApplIDE is directly connected to the work of Carlos Parra’s PhD and Ubino ADT’s work which covers the definition and implementation of a Context-Aware Dynamic Software Product Line (DSPL) named CAPucine. It provides a set of tools for selection of features, metamodel transformation and code generation for mobile applications [40]. The current implementation of ApplIDE addresses transformation from CAPucine metamodel towards SCA metamodel, and Spoon EMF metamodel. The transformations were formerly written with Acceleo tool, which is a dedicated language for transformation, enhancing the readability. ApplIDE meta models are based on the Eclipse Modeling Framework. Code generators are all written in Acceleo.

ApplIDE is registered with the APP (Agence pour la Protection des Programmes) under reference IDDN.FR.001.500004.000.S.A.2010.000.10600.
5. Software

5.1. AmbiDexter

**Participants:** Bas Basten [correspondent], Jurgen Vinju.

**Characterization:** A-3-up4, SO-4, SM-2-up3, EM-2-up3, SDL-4-up5, OC-DA-3-CD-3-MS-3-TPM-3.
**WWW:** [http://homepages.cwi.nl/~basten/ambiguity/](http://homepages.cwi.nl/~basten/ambiguity/)

**Objective:** Statically detect ambiguity of context-free grammars for programming languages, as fast and precise as possible.

**Users:** Authors of context-free grammars of programming languages in SDF2, Rascal, ANTLR, etc

**Impact:** This is the first usable ambiguity detection tool, aiming to solve the Achilles’ heal of context-free general parsing.

**Competition:** AmbiDexter is the fastest and most accurate tool currently available.

**Engineering:** AmbiDexter was developed by one person and will be maintained by another. It is 25 LOC in Java and distributed as a component of the Rascal IDE.

**Publications:** [14], [9], [2], [1]

5.2. Derric

**Participants:** Tijs van der Storm, Jeroen van den Bos [correspondent].

**Characterization:** A-2-up3, SO-4, SM-2-up3, EM-3, SDL-3-up4, OC-DA-3-CD-3-MS-3-TPM-3.
**WWW:** [http://svn.rascal-mpl.org/derric/](http://svn.rascal-mpl.org/derric/)

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

**Users:** Digital forensic investigation specialists

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

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

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

**Publications:** [27], [13]

5.3. Pacioli

**Participants:** Tijs van der Storm, Paul Griffioen [correspondent].

**Characterization:** A-2-up3, SO-4, SM-2, EM-3, SDL-3-up4, OC-DA-3-CD-3-MS-3-TPM-3.
**WWW:** [http://svn.rascal-mpl.org/pacioli/](http://svn.rascal-mpl.org/pacioli/)

**Objective:** Encapsulate all the variability in the construction of modeling and analysis tools in computational auditing

**Users:** Financial auditing experts

**Impact:** Pacioli is an experiment with a big potential in the field of computational auditing. It operates as a vehicle now for experimenting with new ideas in this field. The goal is to tackle the enormous complexity in the (trading) of companies using high level modeling and analysis techniques.

**Competition:** Pacioli competes with less specialized and less formal business analysis tooling, mostly based on spreadsheets.

**Engineering:** Pacioli is a part Java, part Rascal project written by one person.
5.4. Rascal

Participants: Paul Klint, Jurgen Vinju [correspondent], Tijs van der Storm, Bas Basten, Jeroen van den Bos, Mark Hills, Bert Lisser, Arnold Lankamp, Atze van der Ploeg, Vadim Zaytsev, Anastasia Izmaylova, Anya Helene Bagge.

Characterization: A5, SO-4, SM-4, EM-4, SDL-4-up5, OC-DA-3-CD-3-MS-3-TPM-3.
WWW: http://www.rascal-mpl.org

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

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

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

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

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

Publications: [21], [28], [29], [22][6], [7]

5.4.1. Novelties

- Re-design of embedded grammar formalism including semantic disambiguation facilities.
- Extremely fast top-down context-free general parsing algorithm in cubic time and space.
- Parse error reporting via partial parse trees (useful in incremental syntax highlighting and incremental type analysis).
- Auto-indent feature for code generation templates.
- Significant extensions and improvements of software visualization library, such as hierarchical graphs and smaller set of more powerful primitives for charts and interactive features.
- Significant improvements to online documentation and interactive tutor environment.
- “ToLaTex” mode to include Rascal code in papers.
- ShellExec library for inter-acting via pipes with external programs.
- Bridge to Maude and K.
- Generalized function dispatch to arbitrary pattern dispatch.
- New module composition mechanism “extend” next to “import”.
- Ambiguity diagnostics library and parse tree visualizations as a first step towards more grammarware in the IDE.
- A command-line interface to run a single Rascal program.
• Fixed a number of memory leaks in the IDE.
• IDE features for mixed Java/Rascal projects.
• Rational numbers.
• Formal concept analysis library.
• Enhanced SDF2 to Rascal translation.
• Redesigned and simplified abstract grammar format.
• Added “break”, “continue” and “fail” statements for back-tracking and continuation control.
• Radically changed internal design from Visitor to Interpreter design pattern (using an automated refactoring).

5.5. IDE Meta-tooling Platform

Participants: Jurgen Vinju [correspondent], Arnold Lankamp, Anya Helene Bagge.

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

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

WWW: http://www.eclipse.org/imp

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

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

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

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

Engineering: IMP is a long-lived project of many contributors, which is managed as an Eclipse incubation project at eclipse.org.

Publications: [3]

Jurgen Vinju and Arnold Lankamp contribute significantly to the development of IMP. Their effort is focused on the maintenance and optimization of a general purpose symbolic representation library for source code artifacts, called “PDB”. PDB stands for Program DataBase. For more information, please visit http://www.eclipse.org/imp.

The Rascal language itself was accepted by Eclipse as a contribution to the IMP project. This will further strengthen the collaboration between the IMP and the Rascal team as well as generate a wider audience for Rascal.
5.6. Ensō

**Participant:** Tijs van der Storm [correspondent].

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

**WWW:** [http://www.ensō-lang.org](http://www.ensō-lang.org)

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

**Users:** All programmers.

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

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

**Engineering:** Ensō is less than 7000 lines of (bootstrapped) Ruby code.

5.7. Software Language Processing Suite

**Participant:** Vadim Zaytsev [correspondent].

**Characterization:** A3-up4, SO-4, SM-3, EM-2up3, SDL-2, OC-DA-4-CD-4-MS-4-TPM-4

**WWW:** [http://slps.sourceforge.net](http://slps.sourceforge.net)

**Objective:** The project facilitates exposition and comparison of approaches and techniques on language processing.

**Users:** Computer science students, teachers, engineering and practitioners

**Impact:** SLPS contains the largest collection of grammars for programming languages directly recovered from documentation, as well as the largest collection of source-to-source grammar formalisms translators and other related grammarware.

**Engineering:** SLPS is a large collection of scripts and programs written by Ralf Lämmel and Vadim Zaytsev.

5.7.1. Novelties

- New grammars: Ada, Dart, Eiffel, Fortran, Modula, Mediawiki, ...(now a total of 41)
- Grammar Tank: a new collection of 54 small grammars for research purposes
- TestMatch: a tool for grammar-based differential testing of ANTLR grammars and for nonterminal matching based of parsing generated test data (in collaboration with Ralf Lämmel).
- Grammar Hunter: a tool for automated notation-parametric grammar recovery (will also be a Rascal library).

5.8. Demo Light for Composing Models

**Participants:** Jan van Eijck [correspondent], Floor Sietsma.

**Characterization:** A2,SO-3,SM-1,EM-2,SDL-2,OC-4

**WWW:** [http://homepages.cwi.nl/~jve/software/demolight0/](http://homepages.cwi.nl/~jve/software/demolight0/)

**Objective:** Demonstrate epistemic modeling and reasoning

**Users:** Students and researchers in application of epistemic logic

**Impact:** Demo light makes the theory of epistemic reasoning insightful by offering a Haskell library for experimenting with it.

**Engineering:** Demo Light is a Haskell library.
5. Software

5.1. YASS – local homology search

*Actively maintained.*

Software self-assessment following the mechanisms provided by INRIA Evaluation Committee for software evaluation: A-4, SO-3, SM-2, EM-3, SDL-4, DA-4, CD-4, MS-4, TPM-4

Software web site: [http://bioinfo.lifl.fr/yass/](http://bioinfo.lifl.fr/yass/)

Objective: YASS is an open source software devoted to the classical problem of genomic pairwise alignment, and use most of our knowledge to design and implement efficient seeding techniques these last years.

YASS is frequently used; it always receives more than 300 web queries per month (excluding INRIA and Univ-Lille1 local queries), and is also frequently downloaded and cited.

5.2. Carnac – RNA structure prediction

*Actively maintained.*


Software web site: [http://bioinfo.lifl.fr/carnac/](http://bioinfo.lifl.fr/carnac/)

The CARNAC program is for RNA structure prediction by comparative analysis. The web interface also offers 2D visualisation tools and alignment functionalities with gardenia. It has proven to be very fast and very specific compared to its competitors [21].

5.3. TFM-Explorer – Identification and analysis of transcription factor binding sites

*Actively maintained.*


Software web site: [http://bioinfo.lifl.fr/TFM/](http://bioinfo.lifl.fr/TFM/)

The TFM suite is a set of tools for analysis of transcription factor binding sites. locating and analyzing transcription factor binding sites using Position Weight Matrices. In this suite, the TFM-EXPLORER tool is designed to analyze regulatory regions of eukaryotic genomes using comparative genomics and local over-representation.

5.4. Regliss – RNA locally optimal structures

*Actively developed in 2011.*

Software self-assessment: A-2, SO-4, SM-2, EM-2, SDL-4, DA-4, CD-4, MS-4, TPM-4


REGLISS is a tool that studies the energy landscape of a given RNA sequence by considering locally optimal structures. Locally optimal structures are thermodynamically stable structures that are maximal for inclusion: they cannot be extended without producing a conflict between base pairs in the secondary structure, or increasing the free energy. The tool generates all locally optimal structures in a given sequence. Moreover, REGLISS can be used to explore the neighborhood of structures through an energy landscape graph.
5.5. RNAspace – a platform for noncoding RNA annotation

*Actively developed in 2011.*
Software self-assessment: A-5, SO-3, SM-3-up4, EM-2-up3, SDL-4, DA-4, CD-4, MS-4, TPM-4

RNAspace is an open source platform born from a national collaborative initiative. Its goal is to develop and integrate functionalities allowing structural and functional noncoding RNA annotation (see Section 6.2): [http://www.rnaspace.org](http://www.rnaspace.org), and it is distributed under the GPL licence. The project has been awarded by the national IBISA label in autumn 20091.

5.6. CGseq – a toolbox for comparative analysis

*Actively maintained in 2011.*
Software web site: [http://bioinfo.lifl.fr/CGseq/](http://bioinfo.lifl.fr/CGseq/)

CG-seq is a toolbox to identify functional regions in a genomic sequence by comparative analysis using multispecies comparison.

5.7. Biomanycores.org – a community for bioinformatics on manycore processors

*Actively developed in 2011.*
Software self-assessment: A-3-up4, SO-2, SM-2, EM-3, SDL-4-up5, DA-4, CD-4, MS-4, TPM-4

Manycore architectures are an emerging field of research full of promises for parallel bioinformatics. However the usage of GPUs is not so widespread in the end-user bioinformatics community. The goal of the biomanycores.org project is to gather open-source CUDA and OpenCL parallel codes and to provide easy installation, benchmarking, and interoperability. The last point includes interfaces to popular frameworks such as Biopython, BioPerl and BioJava.

The development of Biomanycores is supported by a national ADT2 between BONSAIL, SYMBIOSE (CRI Rennes) and DOLPHIN (CRI Lille). This ADT started in October 2010 and led to the hiring of J.-F. Berthelot (IJD).

In the first year of the ADT, J.-F. Berthelot redesigned and rewrote almost all the existing code. The code base is now stable. He worked on the documentation and on various software engineering aspects such as continuous integration. The second year of the ADT will focus on integrating more applications and targeting bioinformaticians users.

5.8. Norine – a resource for nonribsomal peptides

*Actively developed in 2011.*
Software self-assessment: A-5, SO-3, SM-3-up4, EM-2-up3, SDL-4, DA-4, CD-4, MS-4, TPM-4
Software web site: [http://bioinfo.lifl.fr/norine/](http://bioinfo.lifl.fr/norine/)

Objective: Norine is a public computational resource that contains a database of NRPs with a web interface and dedicated tools, such as a 2D graph viewer and editor for peptides or comparison of NRPs.

Project management: Norine was created and is maintained by members of Bonsai team, in tight collaboration with members of the ProBioGEM lab, a microbial laboratory of Lille1 University.

---

1IBISA is a French consortium for evaluating and funding national technological platforms in life sciences.
2ADT (Action for Technological Development) is an INRIA internal call
Users community: Since its creation in 2006, Norine has gained a universal recognition as the unique database dedicated to non-ribosomal peptides because of its high quality and manually curated annotations. It is queried from all around the world by biologists or biochemists. It receives more than 3000 queries per month. Norine main users come for 13% from the United States of America, for 12% from the United Kingdom, for 5% from China or for 4% from Germany where renowned biology laboratories work on nonribosomal peptides (NRPs) or on their synthetases.

Improvements: This year, the source code has been reorganised by Laurie Tonon, a SED engineer, to use model view controller software architecture, implemented with Struts2.

5.9. GkArrays – indexing high throughput sequencer reads

_Actively maintained._


Software web site: [http://crac.gforge.inria.fr/gkarrays/](http://crac.gforge.inria.fr/gkarrays/)

Objective: Gk-Arrays is a C++ library specifically dedicated to indexing reads produced by high-throughput sequencers. This index allows to answer queries centred on reads. It also takes benefits from the input specificity to lower space consumption.

This library is the result of a collaboration with N. Philippe and T. Commes (IGH laboratory, Montpellier), M. Léonard and T. Lecroq (LITIS laboratory, Rouen) and É. Rivals (LIRMM laboratory, Montpellier). We plan to improve our library in the forthcoming months with the help of Master’s students.
4. Software

4.1. Gaspard 2

Participants: Jean-Luc Dekeyser [correspondant], All DaRT team.

Gaspard2 is an Integrated Development Environment (IDE) for SoC visual co-modeling. It allows or will allow modeling, simulation, testing and code generation of SoC applications and hardware architectures. Its purpose is to provide a single environment for all the SoC development processes:

- High level modeling of applications and hardware architectures
- Application and hardware architecture association (mapping and scheduling)
- Application refactoring
- Deployment specification
- Model to model transformation (to automatically produce models for several target platforms)
- Code generation
- Simulation
- Reification of any stages of the development

The Gaspard2 tool is based on the Eclipse IDE. A set of plugins provides the different functionalities. Gaspard2 provides an internal engine to execute transformation chains. This engine is able to run either QVT (OMG standard) or Java transformations. It is also able to run model-to-text transformations based on Acceleo. The Gaspard2 engine is defined to execute models conform to an internal transformation chains metamodel. A GUI has been developed to specify transformation chain models by drawing them. For the final user, application, hardware architecture, association, deployment and technology models are specified and manipulated by the developer through UML diagrams, and saved by the UML tool in an XMI file format. Gaspard2 manipulates these models through repositories (Java interfaces and implementations) automatically generated thanks to the Ecore specification. Several transformation chains are provided with Gaspard2 to target, from UML models, several execution or simulation platforms (OpenMP, OpenCL, Pthread, SystemC, VHDL, ...). This input language is based on the MARTE UML profile. A tool to generate SIMD configurations derived from the mppSoC model was developed. It allows to automatically generate the VHDL code from a high specification modeled at a high abstraction level (UML model using MARTE profile) based on the IP mppSoC library. The developed tool facilitates to the user to choose a SIMD configuration adapted to his application needs. It has been integrated in the Gaspard environment. Gaspard2 as an educational resource.

The Gaspard2 platform was one of the topics taught in the context of the courses on embedded systems in Telecom Lille and in a Master 2 (TNSI) lecture " Design tools for embedded systems" at the University of Valenciennes. These lectures focused on the potentiality to generate several targets from a subset of the Marte profile and the ability to target system on chip architectures at the TLM level respectively. Furthermore, the model driven engineering characteristics of Gaspard2 are largely detailed in the lecture of Software engineering at Polytech Lille and in the Master of research at university of Lille too.

- See also the web page http://www.gaspard2.org/
- Inria software evaluation: A-2, SO-4, SM-2, EM-1, SDL-2, DA-4, CD-4, MS-4, TPM4
- Version: 2.1.0

4.2. Papyrus

Participants: Cédric Dumoulin [correspondant], Amine El Kouhen, Rahma Yangui.
The Papyrus tool is an UML Development Environment fully compliant with the UML standard and providing all UML diagrams. It is now an Eclipse project (in the incubator state). Papyrus Eclipse can easily be installed in Eclipse from the Eclipse update site. The Papyrus Tool is developed under an Open source license in collaboration with CEA, Atos, Airbus, LIFL.

- See also the web page http://www.eclipse.org/papyrus/
- Software data: plugins number > 150, lines number > 1 million
- Inria software evaluation: A-5, SO-4, SM-4, EM-4, SDL-5, DA-4, CD-4, MS-4, TPM3
- Version: 0.9.0

4.3. Model Driven Factory

Participants: Alexis Muller, Anne Etien [correspondant], Thomas Legrand.

MDFactory is a Model Driven Engineering environment to design, develop and run software production chains. This tool supports our approach based on localized transformation and our Extend operator [96]. It provides a graphical editor to build such production chains with drag and drop from a reusable transformation library. MDFactory is based on the Eclipse platform and the Eclipse Modeling Framework (EMF). It is used to build Gaspard2 integrated transformation chains. This software will be transferred to the start up company Axellience.

- Software data: plugins number around 75
- Evaluation of the software: A 4; SO 4; SM 2; EM 3; SDL 3; DA 4; CD 3; MS 2; TPM 2
- Version: 1.0

4.4. OMEGSI

Participant: Amen Souissi [correspondant].

OMEGSI is an integrated development environment (IDE) for collaborative portals. It allows business process-centered modeling, process simulation, process optimization and full code generation for collaborative portals. The OMEGSI tool is based on the Eclipse IDE. A set of plugins provides the different functionalities. OMEGSI provides an internal engine to execute interactive transformation strategies. This engine (TranS) is written in QVT transformation and able to run any transformation type (QVT, JAVA, Acceleo...). Currently one transformation strategy is provided with OMEGSI to target, from an UML model, the Dolmen execution platform. This input language is based on the MACoP (Modeling and Analysis of Collaborative Portals) UML profile. The fully functional OMEGSI Beginning version is still available on Ecreall website.

- See also the web page http://omegsi.ecreall.com/
- Inria software evaluation: A-3, SO-3, SM-1, EM-2, SDL-4, DA-4, CD-4, MS-4, TPM4
- Version:
DOLPHIN Project-Team

5. Software

5.1. ParadisEO

Participants: Karima Boufaras, Laetitia Jourdan, Arnaud Liefooghe, Thé Van Luong, Nouredine Melab, El-Ghazali Talbi [correspondent], Sébastien Verel.

ParadisEO (PARallel and DIStributed Evolving Objects) is a C++ white-box object-oriented framework dedicated to the flexible design of metaheuristics. See web pages http://paradiseo.gforge.inria.fr/. Based on EO, a template-based ANSI-C++ compliant evolutionary computation library, it is composed of four modules:

- **Paradiseo-EO** provides tools for the development of population-based metaheuristics (evolutionary algorithm, genetic programming, particle swarm optimization, etc.)
- **Paradiseo-MO** provides tools for the development of single solution-based metaheuristics (hill-climbing, tabu search, simulated annealing, iterative local search, variable neighborhood search, incremental evaluation, partial neighborhood, etc.)
- **Paradiseo-MOEO** provides tools for the design of multi-objective metaheuristics (MO fitness assignment, MO diversity preservation, elitism, performance indicators, easy-to-use state-of-the-art algorithms, etc)
- **Paradiseo-PEO** provides tools for the design of parallel and distributed metaheuristics (parallel evaluation, parallel evaluation function, island model)

Furthermore, ParadisEO also introduces tools for the design of distributed, hybrid and cooperative models:

- High level hybrid metaheuristics: coevolutionary and relay models.
- Low level hybrid metaheuristics: coevolutionary and relay models.

The ParadisEO framework has been especially designed to best suit the following objectives:

- **Maximum design and code reuse**: ParadisEO is based on a clear conceptual separation of the solution methods from the problems they are intended to solve. This separation confers to the user a maximum code and design reuse.
- **Flexibility and adaptability**: The fine-grained nature of the classes provided by the framework allows a higher flexibility compared to other frameworks.
- **Utility**: ParadisEO allows the user to cover a broad range of metaheuristics, problems, parallel distributed models, hybridization mechanisms, etc.
- **Transparent and easy access to performance and robustness**: As the optimization applications are often time-consuming the performance issue is crucial. Parallelism and distribution are two important ways to achieve high performance execution. ParadisEO is one of the rare frameworks that provide the most common parallel and distributed models. These models can be exploited in a transparent way, one has just to instantiate their associated provided classes.
- **Portability**: The implemented models are portable on distributed-memory machines as well as on shared-memory multiprocessors, as they use standard libraries such as MPI and PThreads.

This year, with the aim of reenforcing ParadisEO, much works has been established:

- A new design and implementation of the Paradiseo-MO module.
- The addition of local search algorithms for multiobjective optimization.
- The addition of a new module dedicated to parallel metaheuristics on graphics cards.
All the new features is managed via the INRIA’s Gforge project http://paradiseo.gforge.inria.fr.

5.1.1. Paradiseo-MO: a new design and fitness landscape

In the previous version of ParadisEO-MO, each local search algorithm was implemented as a whole, with only a small number of components shared with the others. Moreover, there was no component to trace statistics on local search execution, and no way to implement them easily, in opposition to the ParadisEO philosophy. A new design and implementation of the ParadisEO-MO module has been achieved, allowing one to tackle an optimization problem as a whole, from its analysis to its resolution. In comparison to the previous version of the framework, the modularity has been largely improved, together with an easier reuse of basic components. Another brand new feature of the ParadisEO-MO software framework relates to sampling and statistical tools for fitness landscape analysis.

The new design is based on a clear conceptual separation of the solution methods from the problems they are intended to solve, new concepts are proposed:

- Neighbor : Moves and saves neighbor informations (fitness and more)
- Neighborhood : Describes how to compute all the neighbors
- Evaluation : Can be incremental or full evaluation

Different features are included to improve the set of ParadisEO-MO modular classes combined to develop single solution based metaheuristics:

- General scheme of Local Search algorithms (LS)
- List of Local search algorithms :
  - Hill-climbing (4 different methods)
  - Random Walk (3 different methods)
  - Metropolis Hasting
  - Simulated annealing
  - Tabu search
  - Iterated local search
  - Variable neighborhood search

- New tools to perform fitness landscapes analysis:
  - Density of states
  - Fitness distance correlation
  - Autocorrelation length and function
  - Sampling the local optima by adaptive walks
  - Neutral degree distribution
  - Evolvability of neutral networks by neutral walks
  - Fitness cloud

- New stopping criteria and control method have been added
- Predefined neighborhood operators for standard problem representations
5.1.2. *Paradiseo-MOEO and multiobjective local search*

This year, we particularly improved the module dedicated to multiobjective optimization in terms of local search metaheuristics. As a first step, we focused on a subclass of pure neighborhood search methods. These algorithms can be seen as a generalization of the most basic local search procedure for the multiobjective case. Generally speaking, they combine the definition of a neighborhood structure with the management of a population (or archive) of potentially efficient solutions according to a dominance relation. This archive is iteratively improved by exploring the neighborhood of its own content until no further improvement is possible, or until a stopping condition is satisfied. We denoted them as *Dominance-based Multiobjective Local Search* (DMLS). We also started to implement scalar (preference-based) solution-based local search approaches that should be incorporated in the next version of the platform.

Additionally, some hybridization approaches based on the relay mode have been proposed to hybridize easily evolutionary algorithms with local search during mutation or checkpointing in a multiobjective context. At last, archiving good-quality solutions during the execution of the algorithm is often a large part of the execution time. Thus it was important to provide advanced techniques to reduce this cost. Several solutions proposed in the literature have been implemented. All theses new components have been tested and documented.

5.1.3. *ParadisEO-GPU*

We proposed a pioneering framework called ParadisEO-GPU for the reusable design and implementation of parallel local search metaheuristics (S-Metaheuristics) on Graphics Processing Units (GPU). We have first revisited the ParadisEO-MO software framework to allow its utilization on GPU accelerators focusing on the parallel iteration-level model, the major parallel model for S-Metaheuristics. It consists in the parallel exploration of the neighborhood of a problem solution.

The challenge is on the one hand to rethink the design and implementation of this model optimizing the data transfer between the CPU and the GPU. On the other hand, the objective is to make the GPU as transparent as possible for the user minimizing his or her involvement in its management. From a design point of view, we proposed solutions to this challenge as an extension of the ParadisEO framework. Indeed, a conceptual effort has been done to take into account the aspects related to the GPU architecture and to the ParadisEO-MO module. It has allowed to identify the generic components that are transparent to the user: memory allocation/desallocation on GPU, data transfer between the CPU and the GPU, parallel evaluation of the neighborhood on GPU, structures for the neighborhood evaluation on CPU/GPU, etc.

The first release of the new GPU-based ParadisEO framework has been experimented on the permuted perceptron problem and on the quadratic assignment problem. The preliminary results are convincing, both in terms of flexibility and easiness of reuse at implementation, and in terms of efficiency at execution on GPU.

5.1.4. *New technical features*

Regarding the technical aspects, the compatibility with dependencies taken into account is:

- Checked compatibility with different operating systems
- Reviewed and checked compatibility with new versions of the tools used (Cmake, g++, Visual Studio...)
- Unit test of all additional components, and experiments on classical applications
- A new Website design, with a rearrangement of information based on a set of collected statistics

5.1.5. *Contributions and documentations*

Many investigations were made in this context in order to help users to manipulate the framework.

New documentation:

- The API documentation is available on the ParadisEO Website
- New tutorials
– Hill Climbing
– Neighborhoods (classical and indexed)
– Simulated Annealing and Checkpointing
– Tabu Search
– Iterated Local Search
– Fitness Landscapes Analysis
– Hybrid Lesson

Moreover, a set of implementations for classical problems are now provided as contributions, available within the new version of ParadisEO:

- Single-objective problems: oneMax, queen, quadratic assignment problem, royal road, long path, building block royal road, NK landscapes, NKq landscapes, NKp landscapes, MAX-SAT, unconstrained binary quadratic programming problem, and more.
- Multi-objective problems: traveling salesman problem, quadratic assignment problem, multiple and long path problems.
MINT Team

5. Software

5.1. LibGINA

Participant: Laurent Grisoni [correspondant].

This library has been developed within the context of the ADT GINA, for one of the installations that have been made in collaboration with Le Fresnoy national studio (Damassama, Léonore Mercier). This library is currently being posted as APP, and has been used by Idées-3com small company, in the context of our joint I-lab program. This library allows for use of gestures for command, and is able to handle strong variability into recognized patterns.

Current version: version 1.0

Software characterization: A-2 SO-3 SM-2-up EM-3 SDL-3 OC-DA4-CD4-MS2-TPM4

5.2. 3D interaction using mobile phone

Participants: Samuel Degrande [correspondant], Laurent Grisoni.

This work has been achieved in the context of the Idées-3com I-lab. In this context a module, that allows to use any android based smartphone to control an Explorer module for navigation and interaction with VRML-based content. This module was used as a basis by Idées-3com in their commercial product this year.

Current version: version 1.0

Software characterization: A-2 SO-3 SM-2-up EM-2-up SDL-3 OC-DA4-CD4-MS2-TPM4

5.3. tIO (tactile input & output)

Participants: Paolo Olivo, Nicolas Roussel [correspondant].

tIO is a library designed to facilitate the implementation of doubly tactile interaction techniques (tactile input coupled with tactile feedback) based on the STIMTAC technology. Supporting all current STIMTAC prototypes, it makes it easy to move the system pointer of the host computer according to motions detected on them and adapt their vibration amplitude based on the color of the pointed pixel or the nature of the pointed object. The library includes a set of Qt demo applications that illustrate these two different approaches and makes it easy to “augment” existing Qt applications with tactile feedback. It also makes it possible to supplement or substitute tactile feedback with basic auditory feedback synthesized using portaudio (friction level is linearly mapped to the frequency of a sine wave). This not only facilitates the development and documentation of tactile-enhanced applications but also makes it easier to demonstrate them to a large audience.

Current version: 0.1 - June 2011 (IDDN.FR.001.270005.000.S.P.2011.000.10000)

Software characterization: A2, SO3-up, SM-2, EM2, SDL1.

5.4. libpointing

Participants: Géry Casiez [correspondant], Damien Marchal, Nicolas Roussel.
Libpointing is a software toolkit that provides direct access to HID pointing devices and supports the design and evaluation of pointing transfer functions [16]. The toolkit provides resolution and frequency information for the available pointing and display devices and makes it easy to choose between them at run-time through the use of URIs. It allows to bypass the system’s transfer functions to receive raw asynchronous events from one or more pointing devices. It replicates as faithfully as possible the transfer functions used by Microsoft Windows, Apple OS X and Xorg (the X.Org Foundation server). Running on these three platforms, it makes it possible to compare the replicated functions to the genuine ones as well as custom ones. The toolkit is written in C++ with Python and Java bindings available. It is scheduled to be publicly released in 2012, the licence remaining to be decided.

Web site: http://libpointing.org/

Software characterization: A3, SO3, SM-2, EM2, SDL4
MODAL Team

5. Software

5.1. MIXMOD

Participants: Christophe Biernacki, Serge Iovleff, Remi Lebret, Parmeet Bhatia.

MIXMOD (MIXture MODelling) is the core software of the MODAL team for two reasons. First, MIXMOD concerns main topics of MODAL since it is devoted to model-based supervised, unsupervised and semi-supervised classification for various data situations. Second, MIXMOD is now a well-distributed software since over 250 downloads/month are recorded for several years. Consequently, MIXMOD will be the main software for diffusing future methodological advances of the MODAL team.

MIXMOD is written in C++ (more than 10 000 lines), currently interfaced with Scilab and Matlab and distributed under GNU General Public License. An interface between MIXMOD and R is being developed by Rémi Lebret and will be soon available (during 2012).

Several other institutions participate in the MIXMOD development since several years: CNRS, INRIA Saclay-Île de France, Université de Franche-Comté, Université Lille 1. The software already benefits from several APP depositions and leads also to some international publications.

In addition, an INRIA ADT grant (Parmeet Bhatia) will also develop co-clustering models for continuous, binary and discrete data. It is a strategic development for MIXMOD since offering the ability to structure very large data tables both in lines and columns for different data types. In particular, it opens wide potential applications in biology, marketing, etc.

Serge Iovleff is the main supervisor of software engineers who are recruited for all the previously described tasks. More information about MIXMOD can be easily found on its web page http://www.mixmod.org/.

5.2. AAM

Participant: Serge Iovleff.

The AAM program is a R library implementing Auto-Associative models. Thus it could with few work transformed into a R package. As the AAM is a statistical model, the R language was well-suited for a diffusion inside the scientific community. It is a prototype for testing the AAM models against other kind of non-linear PCA models.

The first release was a scilab program written by Serge Iovleff and Stéphane Girard. It was rewritten in January 2009 and the code is now faster and produces enhanced graphics. The 2009 release is the result of a conjoint work of Serge Iovleff and a M1 internship of the ENS.

More information on the web site http://www.iut-info.univ-lille1.fr/~iovleff/softwares/

5.3. Kerfdr

Participant: Alain Céлиц.

Computation of the local FDR: R package for biostatisticians allowing to estimate FDR and local FDR by kernel density estimation. This package allows also to deal with truncated data and to take into account supervision. More information on the website http://cran.r-project.org/web/packages/kerfdr/

1C. Biernacki, G. Celeux, G. Govaert and F. Langrognet, Model-Based Cluster and Discriminant Analysis with the MIXMOD Software, Computational Statistics and Data Analysis, Vol. 52, no 2, 587–600, 2006
5.4. MetaMa

**Participant:** Guillemette Marot.

metaMA is a specialised software for microarrays. It is a R package which combines either p-values or modified effect sizes from different studies to find differentially expressed genes. The main competitor of metaMA is geneMeta. Compared to geneMeta, metaMA offers an improvement for small sample size datasets since the corresponding modelling is based on shrinkage approaches.

Guillemette Marot is the main contributor and the maintainer of this packages and spent around one year full time for this package between the conception, the implementation, and the documentation. Her PhD advisors (Florence Jaffrézic, Claus-Dieter Mayer, Jean-Louis Foulley) helped her with the conception but she implemented alone the code.

First versions have been posted to the CRAN, the official website of the R software, in 2009. New versions for this package were released in August 2011 in order to take into account remarks from the main users (biologists or biostatisticians analysing gene expression data). This software is routinely used by biologists from INRA, Jouy en Josas (it has been included in a local analysis pipeline) but its diffusion on the CRAN makes it available to a wider community, as attested by the publications citing the software.

More information is available on the website [http://cran.r-project.org/web/packages/metaMA/](http://cran.r-project.org/web/packages/metaMA/).

5.5. STK++

**Participant:** Serge Iovleff.

STK++ is a multi-platform toolkit written in C++ for creating fast and easy to use data mining programs. It offers a large set of templated class in C++ which are suitable for projects ranging from small one-off projects to complete statistical application suites. A C equivalent would be gsl. However, STK++ is developed in C++ in order to get speed and reusability.

As the aim of STK++ is to aid developers to new developments, it proposes essentially interfaces classes and various concrete helping classes, like arrays, numerical methods (QR, SVD), input and output (csv files), random number generators... For instance, some part of the project will be integrated to the co-cluster project (in the MIXMOD software, see Section 5.1 ) actually developed by Parmeet Bathia.

The software is regularly developed since 10 years by Serge Iovleff and it is a work in progress. More information is available on the website [http://www.stkpp.org/](http://www.stkpp.org/) and source repository is here: [https://sourcesup.cru.fr/projects/stk/](https://sourcesup.cru.fr/projects/stk/).

5.6. SMVar

**Participant:** Guillemette Marot.

SMVar is a specialised software for microarrays. This R package implements the structural model for variances in order to detect differentially expressed genes from gene expression data. It performs gene expression differential analysis, based on a particular variance modelling. Its main competitor is the Bioconductor R package limma but limma assumes a common variance between the two groups to be compared while SMVar relaxes this assumption.

Guillemette Marot is the main contributor and the maintainer of this packages and spent around one year full time for this package between the conception, the implementation, and the documentation. Her PhD advisors (Florence Jaffrézic, Claus-Dieter Mayer, Jean-Louis Foulley) helped her with the conception but she implemented alone the code. She received some help from Anne de la Foye (INRA, Clermont-Ferrand) to correct the bugs in the first versions.

---

First versions have been posted to the CRAN, the official website of the R software, in 2009. New versions for this package were released in August 2011 in order to take into account remarks from the main users (biologists or biostatisticians analysing gene expression data). This software is routinely used by biologists from INRA, Jouy en Josas (it has been included in a local analysis pipeline) but its diffusion on the CRAN makes it available to a wider community, as attested by the publications citing the software\(^3\).

More information on the website [http://cran.r-project.org/web/packages/SMVar/index.html](http://cran.r-project.org/web/packages/SMVar/index.html)

---

5. Software

5.1. FXP

Participants: Joachim Niehren [correspondant], Denis Debarbieux, Tom Sebastian.

Software Self-Assessment: A-3, SO-4, SM-3, EM-3, SDL-4

The FXP language is a temporal logic for a fragment of Forward XPath that is suitable for querying XML streams. The FXP library of the Mostrare project of INRIA Lille provides a compiler of the FXP library to nested word automata, efficient query answering algorithm for nested word automata on XML streams, and thus for FXP queries.

FXP is developed in the INRIA transfer project QuiXProc in cooperation with Innovimax. Both a professional and a free version are available. The owner is INRIA.

See also the web page http://fxp.lille.inria.fr/.

- Version: 0-9-2011-03-25

5.2. QuixPath

Participants: Joachim Niehren [correspondant], Denis Debarbieux, Tom Sebastian.

Software Self-Assessment: A-3, SO-4, SM-3, EM-3, SDL-4

The QuiXPath language is a large fragment of Forward XPath with full support for the XML data model. The QuiXPath library provides a compiler from QuiXPath to FXP. Thereby, the efficient query answering algorithms for FXP are lifted to a fragment of Forward XPath. QuiXPath is developed in the INRIA transfer project QuiXProc in cooperation with Innovimax. Both, a free open source and a professional version are available. The ownership of QuiXPath is shared between INRIA and Innovimax. The main application of QuiXPath is its usage in QuiXProc, a professional implementation of the W3C pipeline language XProc owned by Innovimax.

See also the web page http://fxp.lille.inria.fr/.

- Version: QuixPath v1.0.0

5.3. VOLATA

Participant: Fabien Torre [correspondant].


VOLATA provides several machine learning algorithms for attribute-value inference, grammatical inference and inductive logic programming.

See also the web page http://www.grappa.univ-lille3.fr/~torre/Recherche/Softwares/volata/.

- ACM: I.2.6
NON-A Team (section vide)
POPS Project-Team

4. Software

4.1. ASPIRE TDT

Participants: Nathalie MITTON, Loic Schmidt [correspondant], David Simplot-Ryl.

Tag Data Translation (TDT) is an EPCGlobal Inc. standard allowing the translation of identifiers EPC in different representation standard. EPCGlobal standards deal only EPC identifiers. We have extended it to other RFID GS1 and smartcard standards (as ISO 14443 or 15693 and EAN/UPC).

See also the web page http://wiki.aspire.ow2.org/xwiki/bin/view/Main.Documentation/TDT.

- Version: version 0.5

4.2. ASPIRE ALECC

Participants: Nathalie MITTON [correspondant], David Simplot-Ryl, Lei Zhang.

According to the feedback of several RFID application SMEs. They are more likely to accept a light and efficient ALE scheme which only includes the most-used basic modules defined by EPC standard. They desire that such light scheme can be encapsulated and be flexibly used to establish their own RFID application.

The AspireALECC scheme is encapsulated in jar and aims to supply an easy and efficient framework for developers to realize the most used basic operations defined by the EPC ALECC standard.

- Version: 1.0

4.3. EPC TAG CONVERTER

Participants: Roudy Dagher [correspondant], Nathalie MITTON, Loic Schmidt, David Simplot-Ryl.

This module is an EPC-compliant module that aims to convert any tag format into an EPC tag understandable by the middleware.

- Version: 1.0

4.4. EPC TAG GENERATOR

Participants: Roudy Dagher [correspondant], Nathalie MITTON, Loic Schmidt, David Simplot-Ryl.

This module aims to generate tag ids in hexadecimal format in order to validate the EPC grouping and filtering engines of the ALE.

See also the web page http://wiki.aspire.ow2.org/xwiki/bin/view/Main.Documentation.EmbeddedTools/TagGenerator.

- Version: 1.0

4.5. EVe - TCF

Participants: Arnaud Fontaine, Isabelle Simplot-Ryl [correspondant].

Verification of transitive control flow policies on JavaCard 2.x bytecode. Control flow policies expressed using a DSL language are embedded in JavaCard packages (CAP files) using EVe-TCF convert tool. Control flow policies are then statically verified on-device at loading-time thanks to an embedded verifier (designed for smart cards in EVe-TCF). EVe-TCF (Embedded Verifier for Transitive Control Flow) also contains an offline (i.e. PC tool) to simulate on-device loading process of JavaCard 2.x platforms with GlobalPlatform 2.x installed.

- Version: v2.0 - 07/09/2011
4.6. GOLIATH 1.0

Participants: Tony Ducrocq, Nathalie MITTON, David Simplot-Ryl [correspondant], Julien Vandaele.

GOLIATH (Generic Optimized LIghtweight communication stack for Ambient TecHnologies) is a full protocol stack for wireless sensor networks.

4.7. JITS

Participants: Geoffroy Cogniaux, Gilles GRIMAUD [correspondant].

Initial goal of Java was to allow high level software development on small devices. Eventually it founds success and promotion with software deployment on the Web, and more recently as a solution for huge enterprise servers and massive parallel computing. Today small targets are still supported, but with dedicated (Java-like) APIs and VMs. These specific technologies dramatically restrain the context in which Java applications can be deployed.

JITS focuses on these technologies and on enhancements to allow the use of a real Java Runtime Environment and a Java Virtual Machine everywhere by targeting tiny devices such as SmartCards. These devices usually don’t use a Virtual Machine layer over an OS, but expect the Virtual Machine to be the OS. This is possible thanks to the JVM features which can be presented as a specific hardware abstraction for most of them.

See also the web page http://jits.gforge.inria.fr/

• Version: 1

4.8. Light ALE

Participants: Roudy Dagher [correspondant], Nathalie Mitton, Loic Schmidt, David Simplot-Ryl.

In order to provide minimal inventory services, at interface level, subsets of the Reading and the Logical Reader APIs are implemented:

1. Immediate mode: sufficient for user-triggered inventory.
2. Fixed readers configuration: only some properties (Power, Session and InitialQ) can be updated.

The ALE engine manages tag grouping and filtering according to EPC standard patterns in input ECSpec objects. A lightweight custom CODEC was developed as well, in order to decode tag IDs using binary format (array of bytes) and in a garbage-free fashion. A filter engine is also made available for software filtering of tags. This leaves the choice for Reader Connectors to choose the best tradeoff between software and hardware filtering. Note that, because of the Java CDC constraint, the ECSpec and ECReports classes and subclasses were written manually despite of automatic generation from XSD files.

See also the web page http://wiki.aspire.ow2.org/xwiki/bin/view/Main.Documentation.Filtering%26Collection/EmbededALE.

• Version: 1.0

4.9. Light RP

Participants: Roudy Dagher [correspondant], Nathalie Mitton, Loic Schmidt, David Simplot-Ryl.

This wrapper defines the Reader Protocol interface classes that are used to dialog with an RP-compliant reader device. Based on each vendor-specific driver, two implementations were developed in order to provide minimal required services (inventory). Note that the communication with the Reader Device is done locally and directly via method calls. This avoids overhead when using MTB layers for message bindings.

See also the web page http://wiki.aspire.ow2.org/xwiki/bin/view/Main.Documentation/LightRP.

• Version: 1.0

4.10. NFC Light ALE

Participants: Nathalie Mitton [correspondant], Loic Schmidt, David Simplot-Ryl, Lei Zhang.
In order to provide minimal inventory services, at interface level, subsets of the Reading and the Logical Reader APIs are implemented: (1) Immediate mode: sufficient for user-triggered inventory. (ii) Fixed readers configuration: only some properties (Power, Session and InitialQ) can be updated.

The ALE engine manages tag grouping and filtering according to EPC standard patterns in input ECSpec objects. A lightweight custom CODEC was developed as well, in order to decode tag IDs using binary format (array of bytes) and in a garbage-free fashion. A filter engine is also made available for software filtering of tags. This leaves the choice for Reader Connectors to choose the best tradeoff between software and hardware filtering. Note that, because of the Java CDC constraint, the ECSpec and ECRreports classes and subclasses were written manually despite of automatic generation from XSD files. This package has been developed for NFC connection on a mobile phone.

- Version: 1.0

### 4.11. RFID Tag Searcher

**Participants:** Roudy Dagher [correspondant], Nathalie Mitton, Loïc Schmidt, David Simplot-Ryl.

The objective is to use the PDA to look for an item in a given neighborhood. The user would be notified of item proximity via the change of the beep frequency.

Tag Searcher is composed of three main modules:

- The Reader interface is an abstraction of the required services for searching for a tag at a given RF power. The wrappers for CAEN and Intermec readers are straightforward.

- The Ticker class represents a periodic thread that beeps periodically using the standard java call `java.awt.Toolkit.getDefaultToolkit().beep()`. The ticker’s period is synchronized inside the Period-Semaphore class.

- The Scanner class is able to scan for a tag ID and update the Ticker period according to the RF power at which the tag was observed: observation at a small RF power leads to a small tick period, and vice versa.

After testing on both PDAs (i.e. Psion, Intermec), a standalone application with an IHM was developed. It is able to retrieve a list (from a local file or an URL) so that the user choose the item to search for. See also the web page [http://wiki.aspire.ow2.org/xwiki/bin/view/Main.Documentation.EmbeddedTools/TagSearcher](http://wiki.aspire.ow2.org/xwiki/bin/view/Main.Documentation.EmbeddedTools/TagSearcher).

- Version: 1.0

### 4.12. SINGLE

**Participants:** Tony Ducrocq, Nathalie Mitton, David Simplot-Ryl [correspondant].

SINGLE pour Simple IN-door Geo-Localization systEm est une application pour réseaux de capteurs permettant la localisation géographique de capteurs sans fils dans un environnement intérieur.

See also the web page [http://www.senslab.info/](http://www.senslab.info/).

- Version: 1.0
4. Software

4.1. Moose

Participants: Stéphane Ducasse [correspondant], Usman Bhatti, Andre Hora, Nicolas Anquetil, Cyrille Delaunay, Jannik Laval, Tudor Girba [University of Bern].

Web: http://www.moosetechnology.org/

The platform. Moose is a language-independent environment for reverse- and re-engineering complex software systems. Moose provides a set of services including a common meta-model, metrics evaluation and visualization, a model repository, and generic GUI support for querying, browsing and grouping. The development of Moose began at the Software Composition Group in 1997, and is currently contributed to and used by researchers in at least seven European universities. Moose offers an extensible meta-described metamodel, a query engine, a metric engine and several visualizations. Moose is currently in its fourth release and comprises 55,000 lines of code in 700 classes.

The RMoD team is currently the main maintainer of the Moose platform. There are 200 publications (journal, international conferences, PhD theses) based on execution or use of the Moose environment.

The first version running on top of Pharo (Moose 4.0) was released in June 2010. In 2011, Moose saw five releases, with Moose 4.6 in beta since October 2011.

Here is the self-assessment of the team effort following the grid given at http://www.inria.fr/institut/organisation/instances/commission-d-evaluation .

- (A5) Audience : 5 – Moose is used by several research groups, a consulting company, and some companies using it in ad-hoc ways.
- (SO4) Software originality : 4 – Moose aggregates the last results of the teams that use it.
- (SM3) Software Maturity : 3 – Moose is developed since 1996 and got two main redesign phases.
- (EM4) Evolution and Maintenance : 4 – Moose will be used as a foundation of our start up so its maintenance is planned.
- (SDL4) Software Distribution and Licensing : 4 – BSD
- (OC) Own Contribution : (Design/Architecture)DA-4, (Coding/Debugging)-4, (Maintenance/Support)-4, (Team/Project Management)-3

4.2. Pharo

Participants: Stéphane Ducasse, Marcus Denker [correspondant], Damien Pollet, Mariano Martinez-Peck, Veronica Uquillas-Gomez, Igor Stasenko.

Web: http://www.pharo-project.org/

The platform. Pharo is a new open-source Smalltalk-inspired language and environment. It provides a platform for innovative development both in industry and research. By providing a stable and small core system, excellent developer tools, and maintained releases, Pharo’s goal is to be a platform to build and deploy mission critical Smalltalk applications.

The first stable version, Pharo 1.0, was released in 2010. The development of Pharo accelerated in 2011: Version 1.2 and 1.3 have been released, the development branch (1.4a) has seen already over 230 incremental releases as of mid November 2011. For 1.2 and 1.3, over 1000 bug tracker issues have been resolved. In 2011, the community organized five Pharo Sprints, RMoD organized the Deep into Smalltalk School in March 2011.
RMoD is the main maintainer and coordinator of Pharo. It is used widely in both research and industry. With Inria, RMoD is in the process of setting up a Pharo Consortium. There are 25 companies interested in supporting the consortium.

Here is the self-assessment of the team effort following the grid given at [http://www.inria.fr/institut/organisation/instances/commission-d-evaluation](http://www.inria.fr/institut/organisation/instances/commission-d-evaluation).

1. **(A5) Audience**: 5 – Used in many universities for teaching, more than 25 companies.
2. **(SO3) Software originality**: 3 – Pharo offers a classical basis for some aspects (UI). It includes new frameworks and concepts compared to other implementations Smalltalk.
3. **(SM4) Software Maturity**: 4 – Bug tracker, continuous integration, large test suite are on place.
4. **(EM4) Evolution and Maintenance**: 4 – Active user group, consortium is being set up.
5. **(SDL4) Software Distribution and Licensing**: 4 – Pharo is licensed under MIT.
6. **(OC5) Own Contribution**: (Design/Architecture) DA-5, (Coding/Debugging) CD-5, (Maintenance/Support) MS-5, (Team/Project Management) TPM-5

### 4.3. Coral

**Participants**: Damien Pollet [correspondant], Camillo Bruni.


Coral extends the standard Pharo image, to integrate it into the host operating system shell environment and define system commands in Pharo. In term it will provide facilities for image preparation, configuration and deployment.

### 4.4. VerveineJ

**Participants**: Nicolas Anquetil [correspondant], Andre Hora.

**Web**: Inria project [https://gforge.inria.fr/projects/verveinej/](https://gforge.inria.fr/projects/verveinej/).

VerveineJ is a tool to export Java projects into the MSE format, which can then be imported inside Moose (see above). Although VerveineJ is not a research project in itself, it is an important building block for our research in that it allows us to run the Moose platform on legacy Java projects. Another similar tool, Infusion, already existed to fulfill the same needs, but it was closed sources and presented some errors that tainted the results we could obtain.

### 4.5. VerveineSharp

**Participant**: Usman Bhatti [correspondant].

**Web**: Inria project [https://gforge.inria.fr/projects/verveinesharp/](https://gforge.inria.fr/projects/verveinesharp/).

Similar to VervineJ (see above), VerveineSharp is a tool to export C# projects into the MSE format, which can then be imported inside Moose. The reasons for creating this project are the same as for VerveineJ: it is an important building block for our research in that it allows us to run the Moose platform on legacy C# projects. Because C# is a proprietary platform, there are no other tools that can give us the same functionality.
5. Software

5.1. SOFA

SOFA, the Simulation Open Framework Architecture, is an international, multi-institution, collaborative initiative, aimed at developing a flexible and open source framework for interactive simulations. This will eventually establish new grounds for a widely usable standard system for long-term research and product prototyping, ultimately shared by many academic and industrial sites. Over the last two years, the SOFA framework has evolved from an informal collaborative work between the Sim Group at CIMIT, the Alcove, Asclepios and Evasion teams at INRIA into a more structured development project. By proposing a unique architecture allowing the integration of the multiple competencies required for the development of a medical training system, we believe it will be possible to accelerate and foster research activities in the field of interactive medical simulation. The main objectives of the SOFA framework are:

- Simplify the development of medical simulation systems by improving interoperability
- Evaluate and validate new algorithms
- Accelerate the prototyping of simulation systems by promoting component reusability
- Promote collaboration between research groups
- Facilitate technology transfer between research and industry

Our activities around the SOFA framework will be twofold. We will remain one of the leading teams contributing to the design of SOFA, the development of its architecture and its distribution to research groups and industrial partners. In addition, we will use SOFA as a core element of most of our simulations, as a mean to facilitate the integration of results from partners of the national initiative, and to simplify the development of prototypes of simulation systems. For the past few years, there have been a few attempts at designing software toolkits for medical simulation. Examples include [41], GiPSi [30], SPORE [40] or SSTML [27]. These different solutions aim at the same goal: providing an answer (usually Open Source) to the various challenges of medical simulation research and development. Although our aim is similar, we propose a different approach, through a very modular and flexible software framework, while minimizing the impact of this flexibility on the computation overhead. To achieve these objectives, we have developed a new architecture that implements a series of innovative concepts. Also, by developing the SOFA framework collaboratively with scientific experts in the different areas of medical simulation, we believe we can provide state-of-the-art solutions that are generically applicable, yet computationally efficient. The following sections describe in more details our approach to the development of this framework, from a technical standpoint and from the perspective of a collaborative work.

5.1.1. SOFA architecture

Medical simulation relies on a variety of interacting physics-based models, such as rigid structures (e.g. bones), deformable structures (e.g. soft-tissues) and fluids. It also involves anatomical representations through geometrical models, used for visual rendering, collision detection or meshes that will support various computational models. Finally, interactions between these different models need to be efficient, accurate and capable of handling a variety of representations. In some instances, a hierarchy also exists between the various anatomical structures, and needs to be taken into account in the description of the simulated environment. The design of the SOFA architecture, by supporting these various requirements, brings the flexibility needed for academic research. Yet, its very efficient implementation makes it also suitable for professional applications and potentially for product development. This architecture relies on several innovative concepts, in particular the notion of multi-model representation. In SOFA, most simulation components (deformable models, collision models, medical devices, etc.) can have several representations, connected through a mechanism called mapping. Each
representation is optimized for a particular task (e.g. collision detection, visualization) while at the same time improving interoperability by creating a clear separation between the functional aspects of the simulation components. As a consequence, it is possible to have models of very different nature interact together, for instance rigid bodies, deformable objects, and fluids. This is an essential aspect of SOFA, as it will help the integration of new research components. This modular design also facilitates the rapid prototyping of simulation systems, allowing various combinations of algorithms to be tested and compared against each other. At a finer level of granularity, we also propose a decomposition of physical models (i.e. any model that behaves according to the laws of physics) into a set of basic components. In the case of (bio)mechanical models, which are computationally expensive, many strategies have been used to improve computation times or to reduce the complexity of the original model: linear elastic models have often been used instead of more complex non-linear representations, mass-spring methods as an alternative to finite element methods, etc. Each of these simplifications induces drawbacks, yet the importance of these drawbacks depends largely on the context in which they are applied. It becomes then very difficult to choose which particular method is most likely to provide the best results for a given simulation. To address this issue in SOFA we have introduced a finer level of granularity which permits to independently test and compare each component, such as time integration schemes, to see the change in performance or robustness of the simulation, or to test different constitutive models. These changes can be made in a matter of seconds, without having to recompile any of the code, by simply editing an XML file.

5.1.2. Current Results

Version 1.0 RC1 of SOFA was released in December 2011. More than 87,000 downloads of SOFA have been counted as of December 2011. More than 70 researchers, students, engineers have contributed at various degrees to SOFA, for a total of about 700,000 lines of code. Currently, thanks to its advanced architecture, SOFA allows to:

- Create complex and evolving simulations by combining new algorithms with existing algorithms
- Modify most parameters of the simulation by simply editing a XML file
- Build complex models from simpler ones using a scene-graph description
- Efficiently simulate the dynamics of interacting objects using abstract equation solvers
- Reuse and easily compare a variety of available methods
- Transparently parallelize complex computations using semantics based on data dependencies
- Use new generations of GPUs through the CUDA API to greatly improve computation times
Various results and information can be obtained on the SOFA website at http://www.sofa-framework.org. Most of the current results are generic and only aim at validating the different aspects of the SOFA framework. Developments of complex medical simulations have recently started, in particular in the areas of ophthalmic surgery and interventional radiology. We have also started a collaboration with a few companies (Digital Trainers, Didhaptics, B.K.) which are in the process of developing medical applications based on SOFA.

Figure 4. Animation of a chain combining a FEM model, a mass-spring model, a FFD grid, and a rigid body. This example is a perfect illustration of the flexibility of SOFA. Not only several algorithms for rigid or deformable bodies can be part of the same simulation, but they can also interact in a physically correct manner. No constraints between links were pre-defined, instead we relied on collision detection and stiff contact forces to handle the contacts. Using implicit integrator handling dynamically-created groups of interacting objects resulted in a stable simulation.
5. Software

5.1. Introduction

In 2011, SEQUEL continued the development of software for computer games (notably Go) and also developed two novel libraries for functional regression and data mining.

5.2. Computer Games

Participant: Rémi Coulom.

We developed three main softwares for computer games:

- **Crazy Stone** is a top-level Go-playing program that has been developed by Rémi Coulom since 2005. Crazy Stone won several major international Go tournaments in the past. In 2011, its strength improved to 5 dan on the KGS Go Server. It is distributed as a commercial product by Unbalance Corporation (Japan). 5-month work in 2011. URL: http://remi.coulom.free.fr/CrazyStone/

- **Crazy Hanafuda** is a new program to play the Japanese game of Hanafuda. 3 weeks of work in 2011. Discussion are in progress for licensing it.

- **CLOP** is a tool for automatic parameter optimization of game-playing programs. Distributed as freeware (GPL). One month of work in 2011. Available at: http://remi.coulom.free.fr/CLOP/

5.3. Functional Regression

Participant: Hachem Kadri.

A software package in C++ of algorithms for nonlinear functional data analysis using our operator-valued kernel framework (see sec. 6.4.1) is under development. A beta-version of the software can be downloaded at: https://gforge.inria.fr/frs/?group_id=982. The aim of this library is to grow and be shared in our scientific community, and also to be a software resource for our group.

5.4. Data mining library

Participant: Sertan Girgin.

A fully stand-alone library for data mining has been developed, including many classical algorithms for supervised and non-supervised learning. This library is available as an internal resource for the group.
4. Software

4.1. DDNS2

Participant: Caterina Calgaro [correspondant].

The DDNS2 code is a parallel solver for unsteady incompressible Navier-Stokes flows in 2D geometries and primitive variables written in Fortran 95 with MPI as a message-passage library. Mixed finite element methods, with hierarchical basis, are used to discretize the equations and a non overlapping domain decomposition approach leads to an interface problem which involves a Lagrange multiplier corresponding to the velocity (the FETI approach). A dynamical multilevel method is developed locally on each subdomain. Several numerical estimates on the evolution of linear and nonlinear terms allow to construct the multilevel strategy which produces auto-adaptive cycles in time during which different mesh sizes, one for each subdomain, can be considered.

Here is the self-assessment of the team effort following the grid provided by Inria (see: http://www.inria.fr/institut/organisation/instances/commission-d-evaluation): A1, SO3, SM1, EM1, SDL1, DA1, CD4, MS4, TMP1.

4.2. NS3ED

Participant: Caterina Calgaro [correspondant].

The NS3ED code is a solver for steady incompressible Navier-Stokes flows in three-dimensional exterior domains, written in C++. The truncated problem is discretized using an exponential mesh and an equal-order velocity-pressure finite element method, with additional stabilization terms. A bloc-triangular preconditioner is performed for the generalized saddle-point problem.

Here is the self-assessment of the team effort following the grid provided by Inria (see: http://www.inria.fr/institut/organisation/instances/commission-d-evaluation): A1, SO3, SM1, EM1, SDL1, DA1, CD4, MS4, TMP1.

4.3. ns2ddv-M

Participants: Caterina Calgaro [correspondant], Emmanuel Creusé [correspondant], Thierry Goudon, Manuel Bernard.

The NS2DDV-M code is based on a hybrid method coupling FV and FE approaches for solving the variable density Navier-Stokes equation in dimension 2. This original approach for variable density flows is described in [49]. The NS2DDV-M code will be available on the SIMPAF team web page before the end of 2011.

Here is the self-assessment of the team effort following the grid provided by Inria (see: http://www.inria.fr/institut/organisation/instances/commission-d-evaluation): A3, SO3-up4, SM2-up3, EM3, SDL4, DA1, CD4, MS4, TPM4.

Software web site: http://math.univ-lille1.fr/~simpaf/SITE-NS2DDV/home.html

4.4. ns2ddv-C++

Participants: Caterina Calgaro [correspondant], Emmanuel Creusé [correspondant], Thierry Goudon.
The NS2DVD-C++ code is based on a hybrid method coupling FV and FE approaches for solving the variable density Navier-Stokes equation in dimension 2. The code is developed around the GetFem++ and the Bamg softwares. It allows in particular mesh refinement strategies so that very relevant simulations can be reached (as the falling droplet with very high density ratios, see for example [47]). The current version of the code consider the additional terms in the Kazhikhov-Smagulov model.

Here is the self-assessment of the team effort following the grid provided by Inria (see: http://www.inria.fr/institut/organisation/instances/commission-d-evaluation): A1, SO3-up4, SM1, EM2, SDL1, DA1, CD4, MS4, TPM1.

4.5. RTcodes

Participants: Pauline Lafitte [correspondant], Jean-François Coulombel, Christophe Besse, Thierry Goudon.

We have developed a set of numerical codes, written in Scilab, to compute the solutions of the system coupling the Euler equations to the radiation through energy exchanges, in the non equilibrium regime. This covers several situations in the hierarchy of asymptotic problems. The code treats the one-dimensional framework. In particular the code can be used to investigate radiative shocks profiles. The main advantage of our numerical codes is that they do not require any refinement near the singularities. The numerical tests show a very good agreement with the theoretical predictions.

Here is the self-assessment of the team effort following the grid provided by Inria (see: http://www.inria.fr/institut/organisation/instances/commission-d-evaluation): A2, SO3, SM2, EM1, SDL1.

4.6. FPcodes

Participants: Pauline Lafitte [correspondant], Thierry Goudon.

We have developed a numerical code, written in Scilab, to compute the solutions of the two-phase flows equations describing particles interacting with a fluid through friction forces. The code treats one-dimensional situation and is well adapted to describe gravity driven flows in either bubbling or flowing regimes. In particular, it can be used to describe the evolution of pollutants in the atmosphere. The numerical strategy, based on a asymptotic-based scheme, is described in details in [50].

Here is the self-assessment of the team effort following the grid provided by Inria (see: http://www.inria.fr/institut/organisation/instances/commission-d-evaluation): A2, SO3, SM2, EM1, SDL1.

4.7. CLAToolBox

Participants: Christophe Besse [correspondant], Pauline Klein.

As a byproduct of the review paper [39], a user-friendly interface is offered\(^5\) to trial and compare various numerical methods to solve the 1D Schrödinger equation with absorbant boundary conditions. We also mention [42] for a numerical investigation of blow-up phenomena in the nonlinear Schrödinger equation.

4.8. SPARCS

Participants: Christophe Besse, Thierry Goudon [correspondant], Ingrid Lacroix-Violet.

SPARCS is the code developed by Thales Alenia Space for the simulation of the charge phenomena the spacecrafts are subject to. The current version of the code, according to the PhD thesis of O. Chanrion and M. Chaney-Yook performed in collaboration with the team Caiman at Sophia Antipolis, is specialized to geostationary atmospheres. The model consists in the stationary Vlasov-Poisson system, but where instationary effects are taken into account with the boundary condition for the electric field. We participate, in particular through the post doc of N. Vaucheleit, to the elaboration of an improved version of the code which includes parallization optimized procedures, the modeling of the natural difference of potential between different dielectric surfaces of the spacecraft, as well as the possible presence of devices emitting charged particles.

\(^5\) http://math.univ-lille1.fr/~besse/site/recherche/logiciels/index.html