Activity Report 2012

Section Software

ATEAMS Project-Team

4. Software

4.1. Derric

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

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

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

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

Users: Digital forensic investigation specialists

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

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

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

Publications: [8][32], [14]

In 2012 Derric was validated on a large body of image files taken from wikipedia, and a new approach to software optimization via model transformation was developed for optimizing Derric code. We released Derric 1.0 in 2012.

4.2. Basic Voting Theory

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

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

WWW: http://homepages.cwi.nl/~jve/software

Objective: Demonstrate the basic concepts of voting theory.

Users: Students and researchers interested in voting theory.

Impact: This is a demonstrator and a tool for teaching.

Competition: None.

Engineering: Haskell program.

4.3. Rascal

Participants: Paul Klint, Jurgen Vinju [correspondent], Tijs van der Storm, Jeroen van den Bos, Mark Hills, Bert Lisser, Atze van der Ploeg, Vadim Zaytsev, Anastasia Izmaylova, Michael Steindorfer, Ali Afroozeh.

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: [23], [22], [11], [21], [22]

4.3.1. Novelties

- Statically typed access to external data-sources [21]. This includes access to CVS files, spreadsheets, databases, etc.

- Significant improvements to online documentation and inter-active tutor environment.

- Full transparent support for Unicode codepoints.

- Added language-supported quickcheck-style random testing facility (by Wietse Venema, intern), including bridge to JUnit testing framework and IDE support.

- Revived access libraries to CVS, SVN and Git VCSs.

- Added support for JSON export and import, towards Rascal webservices.

- Totally re-implemented and extended debugging interface.

- Priority and associativity mechanism for context-free grammars was completed, such that it can not be used to accidentally remove sentences from a language anymore.

- Reimplementation of the except disambiguation filter with much higher efficiency.

- Improved module import times.

- Reimplemented URI encoding/decoding mechanism for correctness and portability.

- Added semi-automated exam generation and grading feature to the Rascal tutor environment.

- Experimented with strategies for error recovery in context-free general top-down parser.

- Added MissGrant and SuperAwesomeFighter language workbench demonstrations.

- Structured re-design of menus and menu options in the IDE

- Added bindings to Apache statistics libraries
- Created Rascalopedia, a glossary of concepts and terms that are relevant for metaprogrammers. The descriptions are aiming at under-graduate students.
- Two previously designed programmable transformation languages for grammars in a broad sense: the unidirectional XBGF and the bidirectional ΞBGF — have been reimplemented as libraries in Rascal.
- Improved general stability and efficiency.

4.4. IDE Meta-tooling Platform

Participants: Jurgen Vinju [correspondent], Michael Steindorfer.

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

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

WWW: 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`. Currently we are moving the project to Github to explore more different ways of collaboration.

Publications: [3]

4.5. Ensō

Participant: Tijs van der Storm [correspondent].

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

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

Objective: Together with Prof. Dr. William R. Cook of the University of Texas at Austin, and Alex Aloh, 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.

4.6. Turing language

Participants: Davy Landman [correspondent], Tijs van der Storm, Jeroen van den Bos, Vadim Zaytsev, Paul Klint.

Characterization: A3, SO-2, SM-1, EM-1, SDL-5, DA-3-CD-3-MS-3-TPM-3

WWW: http://www.legoturingmachine.org

Objective: This software is used to program the Lego Turing Machine which was built as a piece for the Turing Centennial exposition at CWI. The software features a full fledged Eclipse based IDE for a small programming language which is compiled to Turing machine instructions that run on a Lego machine.

Users: People interested in learning about computation and programming languages.

Impact: the Lego Turing Machine and it's software have reached more than 3 million people via the internet (slashdot, vimeo, youtube) and all Dutch national newspapers.

Competition: none.

Engineering: the hardware is Lego and the software is fully generated from Rascal syntax definitions and IDE construction functions.

4.7. Lua AiR

Participant: Riemer van Rozen [correspondent].

Characterization: A2-up, SO-4, SM-2-up, EM-3-up, SDL-4, DA-3-CD-3-MS-3-TPM-3

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

Objective: This system provides IDE integrated support for static analysis of Lua code. Lua is a scripting language used in game development.

Users: Game programmers and game designers

Impact: Lua AiR is currently a research prototype designed to experiment with and validate the static analysis of a highly dynamic scripting language.

Competition: none.

Engineering: LuA AiR is fully implemented in Rascal.

DART Project-Team

5. Software

5.1. 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 [35] 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 [37]. The Gaspard2 engine is defined to execute models conform to an internal transformation chains meta-model. 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 softwre evaluation: A-2, SO-4, SM-2, EM-1, SDL-2, DA-4, CD-4, MS-4, TPM4
- Version: 2.1.0

DOLPHIN Project-Team

5. Software

5.1. ParadisEO

Participants: Clive Canape, Laetitia Jourdan, Arnaud Liefooghe, Nouredine Melab, Alexandre Quemy, El-Ghazali Talbi [correspondent], Sébastien Verel.

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

- Paradiseo-EO provides tools for the development of population-based metaheuristic (evolutionary and genetic algorithm, genetic programming, particle swarm optimization, etc.)

- Paradiseo-MO provides tools for the development of single solution-based metaheuristics (hill-climbing, tabu search, simulated annealing, iterative local search, variable neighborhood search, incremental evaluation, partial neighborhood, etc.)

- Paradiseo-MOEO provides tools for the design of multi-objective metaheuristics (MO fitness assignment, MO diversity preservation, elitism, performance indicators, easy-to-use state-of-the-art algorithms, etc)

- Paradiseo-PEO provides tools for the design of parallel and distributed metaheuristics (parallel evaluation, parallel evaluation function, island model)

- Paradiseo-SMP provides tools for the design of shared memory parallel metaheuristics (parallel evaluation, island model)

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

- High level hybrid metaheuristics: coevolutionary and relay models.

- Low level hybrid metaheuristics: coevolutionary and relay models.

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

- Maximum design and code reuse: ParadisEO is based on a clear conceptual separation of the solution methods from the problems they are intended to solve. This separation confers to the user a maximum code and design reuse.

- Flexibility and adaptability: The fine-grained nature of the classes provided by the framework allows a higher flexibility compared to other frameworks.

- Utility: ParadisEO allows the user to cover a broad range of metaheuristics, problems, parallel distributed models, hybridization mechanisms, etc.

- Transparent and easy access to performance and robustness: As the optimization applications are often time-consuming the performance issue is crucial. Parallelism and distribution are two important ways to achieve high performance execution. ParadisEO is one of the rare frameworks that provide the most common parallel and distributed models. These models can be exploited in a transparent way, one has just to instantiate their associated provided classes.

- Portability: The implemented models are portable on distributed-memory machines as well as on shared-memory multiprocessors, as they use standard libraries such as MPI and std::threads. ParadisEO supports the most recent version of standard of the C++ programming, c++11.

This year a new module, Paradiseo-SMP, has been released. All the new features is managed via the Inria Gforge project http://paradiseo.gforge.inria.fr. The version 2.0 has been released in Septembre 2012.

5.1.1. Paradiseo-SMP: a new module for shared memory parallel

This year, we released a new module dedicated to shared memory parallel. This module improves the technical mechanisms of Paradiseo-PEO thanks a new software architecture and the new c+11 features.
Paradiseo-SMP implements parallel evaluation, dynamic heterogeneous island model, and their hybridization. The main features are:

- Dynamic Island Model: topology can be changed during the execution.
- Heterogeneous Islands: different kinds of population-based metaheuristics can communicate (evolutionary and genetic algorithm, particle swarm optimization, etc.).
- Island Model and master/slave model can be hybridized.

All these new features are developed in c++11.

5.1.2. New technical features

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

- c++11 features supporting.
- Checked compatibility with different operating systems.
- Reviewed and checked compatibility with new versions of the tools used (CMake, g++, clang, MinGW...).
- Unit and integration test of all additional components, and experiments on classical applications.

5.1.3. Contributions and documentations

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

- New quick start guide is available.
- New tutorials:
    - Tutorials SMP.
    - Tutorials GPU.
- Updated implementation for classical problems.

Self-assessment of the team effort (software criteria: http://www.inria.fr/institut/organisation/instances/commission-d-evaluation)

(A-4-up5) Audience: 4 - Used in many universities for teaching and several companies.

(SO-4) Software Originality: 4 - ParadisEO aggregates the last results of the Dolphin team.

(SM-4) Software Maturity: 4 - Extensive documentation, strong software engineering and testing, regression testing, user feedback ...

(EM-2-up3) Evolution and Maintenance: 2 - Basic maintenance with persistent attention to users.

(SDL-4) Software Distribution and Licensing: 4 - CeCILL license, public source, Windows and Mac installer, Linux packages.

(OC) Own Contribution: (Design/Architecture) DA-4, (Coding/Debugging) CD-4, (Maintenance/Support) MS-4, (Team/Project Management) TPM-4

MODAL Project-Team

5. Software

5.1. Two advances for the MIXMOD software

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

MIXMOD (MIXture MODelling) is an important software for the mΘdal team since it concerns its main topics: model-based supervised, unsupervised and semisupervised classification for various data situations. MIXMOD is now a well-distributed software with over 250 downloads/month are recorded for several years. MIXMOD is written in C++ (more than 10 000 lines) and distributed under GNU General Public License. Several other institutions participate in the MIXMOD development since several years: CNRS, Inria Saclay-Île de France, Université de Franche-Comté, Université Lille 1. The software already benefits from several APP deposits.

An interface between MIXMOD and R (Rmixmod) has been developed by Rémi Lebret and Serge Iovleff and is now available on the CRAN (http://cran.r-project.org/web/packages/Rmixmod/index.html). We expect now a wide impact of MIXMOD on the growing community familiar with R. A paper related to Rmixmod is submitted to an international journal [34].

Until December 2012, Parmeet Bhatia, under scientific supervision of Christophe Biernacki, is developping possibility in MIXMOD to cluster simultaneously continuous and categorical data with the restrictive conditional independence assumption. It is an important first step towards the long term purpose of mΘdal to cluster heterogeneous (or mixed) data sets.

5.2. The blockcluster package

Participants: Christophe Biernacki, Serge Iovleff, Parmeet Bhatia.

blockcluster is a R package for model-based simultaneous clustering of rows and columns, thanks to an Inria ADT grant (Parmeet Bhatia). It is also developed in collaboration with University of Technology of Compiègne. It offers the ability to structure very large data tables both in lines and columns for different data types (continuous, binary and contingency data). In particular, it opens wide potential applications in biology, marketing, etc. It is available online on CRAN (http://cran.r-project.org/web/packages/blockcluster/index.html) for all major platforms (Linux, MacOS, Windows). It also comes with utility functions to visualize data. A paper related to blockcluster is submitted to an international journal [40].

5.3. Cuvclust package

Participant: Guillemette Marot.

cuvclust is a R package dedicated to model-based curve clustering. Considered models include Functional Clustering Mixed Models (FCMM, ie functional clustering with the presence of functional random effects), but also traditional functional clustering model (FCM, without functional random effects), and functional mixed models (FMM, functional random effects without clustering). Estimation is done by maximum likelihood using the EM algorithm, and two criteria are proposed to select the number of clusters, based on integrated likelihoods.
Guillemette Marot was the main contributor of the beta version of the package during her post-doc. Due to several changes in conception and due to planning of extensions in the package by the other contributors of the package, she decided to become a regular contributor and left the maintenance to Franck Picard.

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 these 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 were 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 citations of publications related to the methods implemented in the software.

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

5.5. 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.

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

5.6. Tax3 Software

Participants: Serge Iovleff, Remi Lebret.

Tax3 implements a statistical method providing an analytical framework for high dimensional datasets and complex problems combining several variable types: genetics, genomics, biomarkers and phenotypes

5.7. aam Program

Participant: Serge Iovleff.

aam is a console based program dedicated to the estimation of the semi-linear auto-associative models in a gaussian setting. It is written in C++ and used the STK++ library as support.

5.8. 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, etc.

The software is regularly developed for 10 years by Serge Iovleff and it is a work in progress. The version 0.3 has been released. More information is available on the website http://www.stkpp.org/ and source repository is here: https://sourcesup.cru.fr/projects/stk/

5.9. Scan3D

Participants: Alexandru Amarioarei, Cristian Preda.

Scan3D is a C++ sofware for estimating the distribution of the three-dimensional scan statistics for Bernoulli and Poisson models. It implements the most recent approximation methods available, in particular that developed by the authors providing bounds for the approximation errors [39].

NON-A Project-Team (section vide)

SEQUEL Project-Team

5. Software

5.1. Introduction

In 2012, 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 continued the development of 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 2012, a new version was released in Japan. This new version won a game with a 4-stone handicap against a professional player during the European Go Congress in Bonn, Germany. It is distributed as a commercial product by *Unbalance Corporation* (Japan). 6-month work in 2012. URL: http://remi.coulom.free.fr/CrazyStone/

- ***Crazy Hanafuda*** is a program to play the Japanese card game of Hanafuda. One month of work in 2012. A licence agreement was signed with Unbalance Corporation in January. The Windows 8 version of the program was released commercially in November.

- ***CLOP*** [30] is a tool for automatic parameter optimization of game-playing programs. Distributed as freeware (GPL). One month of work in 2012. Available at: http://remi.coulom.free.fr/CLOP/

5.3. Vowpal Wabbit

Participants: Jérémie Mary, Romaric Gaudel, Thomas Chabin.

Vowpal Wabbit is a GPL project leaded by John Langford at Yahoo! Research and now at Microsoft. The goal is to build a very fast, distributed and large scale machine learning software. https://github.com/JohnLangford/vowpal_wabbit/wiki. We worked on the optimization of the parser and on the memory structures of the i/o. The modifications have been accepted for commit in the main branch and allow an average division by two of all execution times.

SIMPAF Project-Team

5. Software

5.1. ns2ddv-M

Participants: Caterina Calgaro [correspondant (Univ. Lille 1)], Emmanuel Creusé [correspondant (Univ. Lille 1)].

Incompressible Navier-Stokes, Variable Density, Rayleigh-Taylor Instability The NS2DDV-M code is based on a hybrid method coupling FV and FE approaches for solving the variable density Navier-Stokes equation in dimension 2. This original approach for variable density flows is described in [56]. The NS2DDV-M code will be avalaible 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

5.2. ns2ddv-C++

Participants: Caterina Calgaro [correspondant (Univ. Lille 1)], Emmanuel Creusé [correspondant (Univ. Lille 1)], Thierry Goudon.

Incompressible Navier-Stokes, Variable Density, Kazhikhov-Smagulov model, Rayleigh-Taylor Instability, avalanches phenomena The NS2DVD-C++ code is based on a hybrid method coupling FV and FE approaches for solving the variable density Navier-Stokes equation in dimension 2. The code is developed around the GetFem++ and the Bamg softwares. It allows in particular mesh refinement strategies so that very relevant simulations can be reached (as the falling droplet with very high density ratios, see for example [54]. The current version of the code consider the additional terms in the Kazhikhov-Smagulov model.

Webpage : http://math.univ-lille1.fr/~simpaf/SITE-NS2DDV

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

5.3. RTcodes

Participants: Pauline Lafitte [correspondant (ECP)], Jean-François Coulombel [(CNRS & Univ. Nantes)], Christophe Besse [(Univ. Lille 1)], Thierry Goudon [(Inria)], Giovanni Samaey [(KU Leuven)].

Radiative Transfer, Radiative shocks, AP schemes

We have developed a set of numerical codes, written in Scilab, to compute the solutions of the system coupling the Euler equations to the radiation through energy exchanges, in the non equilibrium regime. This covers several situations in the hierarchy of asymptotic problems. The code treats the one-dimensional framework. In particular the code can be used to investigate radiative shocks profiles. The main advantage of our numerical codes is that they do not require any refinement near the singularities. The numerical tests show a very good agreement with the theoretical predictions. See reference [27].

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

5.4. FPcodes

Participants: Pauline Lafitte [correspondant (ECP)], Thierry Goudon [(Inria)], Benjamin Boutin [(Univ. Rennes)].

Fluid-Particles flows, Gravity driven flows, AP schemes

We have developed a numerical code, written in Scilab, to compute the solutions of the two-phase flows equations describing particles interacting with a fluid through friction forces. The code treats one-dimensional situation and is well adapted to describe gravity driven flows in either bubbling or flowing regimes. In particular, it can be used to describe the evolution of pollutants in the atmosphere. The numerical strategy, based on a asymptotic-based scheme, is described in details in [57].

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

5.5. CLAToolBox

Participant: Christophe Besse [correspondant].

Absorbant boundary conditions, Schrödinger equation

As a byproduct of the review paper [45], a user-friendly interface is offered [1] to trial and compare various numerical methods to solve the 1D Schrödinger equation with absorbant boundary conditions. We also mention [50] for a numerical investigation of blow-up phenomena in the nonlinear Schrödinger equation.

5.6. SPARCS

Participant: Christophe Besse.

Vlasov-Poisson system, Euler-Poisson system. Back-Trajectory method

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

5.7. Code-Carmel3D

Participant: Emmanuel Creusé [correspondant (Univ. Lille 1)].

This numerical code, developed in collaboration between EDF R&D and Lille 1 University, is devoted to the electromagnetic fields computation by the use of finite element methods. This code allows in particular to perform nondestructive control by the use of Foucault currents in steam generator pipes, and should be soon coupled with the thermal simulation of Code-Aster. Code-Carmel3D uses the Salomé platform (mesh-generator and post-processing) and Open Turns (uncertainties computation). It will consequently allow to solve multi-physics problems, both for the temporal and harmonic formulations.

---

[1] http://math.univ-lille1.fr/~besse/site/recherche/logiciels/index.html

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

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/

Licence: GPL

YASS is a software devoted to the classical problem of genomic pairwise alignment, and use most of our knowledge to design and implement efficient seeding techniques these last years. It is frequently used, it always receives more than 300 web queries per month (excluding local queries), and is also frequently downloaded and cited.

5.2. Carnac – RNA structure prediction

Actively maintained.
Software self-assessment: **A-4**, **SO-3**, **SM-2**, **EM-3**, **SDL-4**, DA-4, CD-4, MS-4, TPM-4

Software web site : http://bioinfo.lifl.fr/carnac/

Licence: Cecill

The CARNAC program is for RNA structure prediction by comprative 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 [19].

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

Actively maintained.
Software self-assessment: **A-4**, **SO-3**, **SM-2**, **EM-3**, **SDL-4**, DA-4, CD-4, MS-4, TPM-4

Software web site : http://bioinfo.lifl.fr/TFM/

Licence: GPL

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

5.4. Regliss – RNA locally optimal structures

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

Software web site : http://bioinfo.lifl.fr/RNA/regliss/

REGLISS is a tool that studies the energy landscape of a given RNA sequence by generating all locally optimal structures, that are maximal thermodynamically stable structures.

5.5. RNAspace – A platform for noncoding RNA annotation

Actively developped.

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

Software web site : http://www.rnaspace.org/

Licence: GPL

RNAspace is a national collaborative initiative conducted with Genopole Midi-Pyrénées and originally supported by IBISA [1]. The goal is to develop an open source platform for structural and functional noncoding RNA annotation in genomes (see Section 6.2): http://www.rnaspace.org. The project will be pursued within France Génomique (see Section 7.2.1).

5.6. CGseq – A toolbox for comparative analysis

Actively maintained.
Software self-assessment: **A-4**, **SO-3**, **SM-2**, **EM-3**, **SDL-4**, DA-4, CD-4, MS-4, TPM-4

Software web site : http://bioinfo.lifl.fr/CGseq/

Licence: GPL

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

5.7. SortMeRNA – Metatranscriptome classification

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

Software web site: http://bioinfo.lifl.fr/RNA/sortmerna

Licence: GPL

SortMeRNA is a software designed to rapidly filter ribosomal RNA fragments from metatransriptomic data produced by next-generation sequencers. It is capable of handling large RNA databases and sorting out all fragments matching to the database with high accuracy and specificity.

5.8. Biomanycores.org – A community for bioinformatics on manycore processors

Actively developped.
Software self-assessment: **A-3**, **SO-2**, **SM-3**, **EM-3down2**, **SDL-4up5**, OC-4 (DA-4, CD-4, MS-4, TPM-4)

Software web site : http://biomanycores.org/

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

The development of Biomanycores was supported by a national ADT [2] between BONSAI, SYMBIOSE (CRI Rennes) and DOLPHIN (CRI Lille), from October 2010 to October 2012. This ADT led to the hiring of J.-F. Berthelot (IJD) who completely redesigned the existing code and added more applications. Biomanycores has now a comprehensive developer and user documentation, large test suites and continuous integration. In June 2012, the project has been presented during a workshop dedicated to manycore programming (see Section 8.1).

---

[1]IBISA is a French consortium for evaluating and funding national technological platforms in life sciences.
[2]ADT (Action for Technological Development) is an Inria internal call

5.9. Norine – A resource for nonribosomal peptides

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

Software web site : http://bioinfo.lifl.fr/norine/ Norine is a public computational resource that contains a database of NRPs with a web interface and dedicated tools, such as a 2D graph viewer and editor for peptides or comparison of NRPs. Norine was created and is maintained by members of BONSAI team, in tight collaboration with members of the ProBioGEM lab, a microbial laboratory of Lille1 University. Since its creation in 2006, Norine has gained an international recognition as the unique database dedicated to non-ribosomal peptides because of its high quality and manually curated annotations, and has been selected by wwPDB as a reference database. It is queried from all around the world by biologists or biochemists. It receives more than 3000 queries per month. Norine main users come for 13% from the United States of America, for 12% from the United Kingdom, for 5% from China or for 4% from Germany where renowned biology laboratories work on nonribosomal peptides (NRPs) or on their synthetases.

5.10. GkArrays – Indexing high throughput sequencer reads

Actively maintained.
Software self-assessment: **A-3**, **SO-3**, **SM-3**, **EM-2**, **SDL-4**, DA-4, CD-4, MS-4, TPM-3

Software web site : http://crac.gforge.inria.fr/gkarrays/

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

This library is the result of a collaboration with N. Philippe and T. Commes (IGH laboratory, Montpellier), M. Léonard and T. Lecroq (LITIS laboratory, Rouen) and É. Rivals (LIRMM laboratory, Montpellier).

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

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 [36], GiPSi [25], SPORE [35] or SSTML [22]. 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
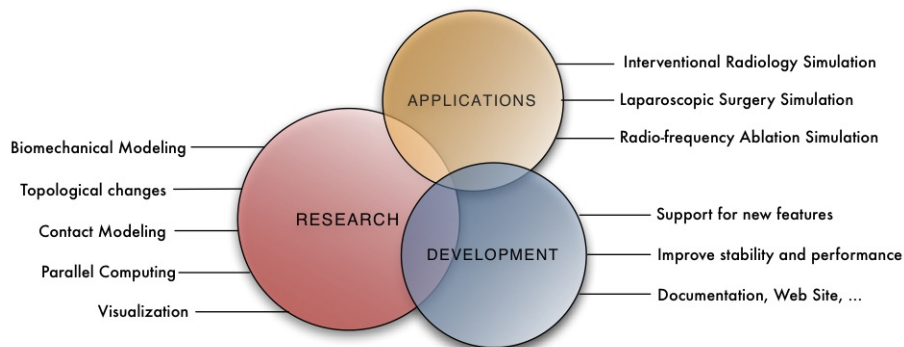
Figure 3. Multidisciplinary research and development of the SOFA framework need to take place simultaneously to quickly advance research in the field of computer-based interactive medical simulation

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 but since October 2012, SOFA is now available through a public and anonymous SVN. More than 137,000 downloads of SOFA have been counted as of November 2012. More than 70 researchers, students, engineers have contributed at various degrees to SOFA, for a total of about 1,200,000 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
- Use embedded Python environment to create interactive and parametric scenes, and interact with 3rd party software

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

ADAM Project-Team

5. Software

5.1. Introduction

We report on the major software systems that are developed by our research group. FRASCATI, PowerAPI, SPACES relate to the first research direction. AppliIDE and CALICO relate to the second one. Finally, FRACTAL is a general purpose component framework that serves as a foundation for most of our work around reconfigurable middleware.

5.2. AppliIDE

Participants: Laurence Duchien, Clément Quinton [correspondant].

AppliIDE is directly connected to the work of Carlos Parra's PhD thesis [116] and Ubinov 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 the selection of features, metamodel transformation and code generation for mobile applications [119]. 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 metamodels are based on the Eclipse Modeling Framework. Code generators are all written in Acceleo.

Inria Evaluation Committee Criteria for Software Self-Assessment: A-3, SO-3, SM-3, EM-2, SDL-2. ApplIDE is registered with the APP (Agence pour la Protection des Programmes) under reference IDDN.FR.001.500004.000.S.A.2010.000.10600.

5.3. CALICO

Participants: Laurence Duchien [correspondant], Antonio de Almeida Souza Neto.

CALICO is an agile development framework for the design and evolution of safe component-based and service-oriented software that has been developed in the context of Guillaume Waignier's PhD thesis [128].

Agile software development relies on an iterative and incremental development cycle that allows the software 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. 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 analyses 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 do not natively provide this support. The current version handles four component-based and service-oriented platforms (FRASCATI, FRACTAL, OPENCCM, OPENCOM). 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.

Inria Evaluation Committee Criteria for Software Self-Assessment: A-3, SO-4, SM-3, EM-3, SDL-4. CALICO is an open source software available at http://calico.gforge.inria.fr.

5.4. Fractal

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

FRACTAL is a modular, extensible, and reflective component framework. The FRACTAL toolchain can be used to design, implement, deploy and reconfigure any kind of software and middleware system. FRACTAL has initially been designed by both Inria and France Telecom R&D.

Inria Evaluation Committee Criteria for Software Self-Assessment: A-4, SO-4, SM-3-up, EM-3-up, SDL-4-up, DA-3, CD-4, MS-4, TPM-4. FRACTAL is a project of the OW2 consortium for open-source middleware. Web site: http://fractal.ow2.org. License LGPL. Some of the research activities around FRACTAL [91], [90], [124] are on top cited publications of the CBSE research community [109]. The ADAM project-team members are among the top committers of the project with 33.8% of all commits and they are the principal contributors for several modules including AOKell [124], Fraclet, the Inria ODL F4E [95], [96], Juliac, Koch. Philippe Merle is the leader of the OW2 FRACTAL project.

5.5. FraSCAti

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

FRASCATI is a service-oriented component-based middleware platform implementing OASIS *Service Component Architecture* (SCA) specifications.The main originality of OW2 FRASCATI is to bring FRACTAL-based reflectivity to SCA, *i.e.*, any FRASCATI software component is equipped with both the SOA capabilities brought by SCA and the reflective capabilities (*i.e.*, introspection and reconfiguration) brought by FRACTAL. Various micro-benchmarks have shown that FRASCATI reflectivity is achieved without hindering its performance relative to the de facto reference SCA implementation, *i.e.*, Apache Tuscany. Non-functional concerns (logging, transaction, security, etc.), so called intents in SCA terms, are also programmed as FRASCATI components and are (un)woven on business components dynamically at runtime, this is based on aspect-oriented concepts defined in FAC [117]. OW2 FRASCATI supports various implementation technologies (SCA Composite, Java, WS-BPEL, Spring Framework, OSGi, Fractal ADL, native C library, Apache Velocity templates, and seven scripting languages as BeanShell, FScript, Groovy, JavaScript, JRuby, Jython, XQuery) for programming services or integrating legacy code, various binding protocols (SOAP, REST, JSON-RPC, UPnP, HTTP servlets, Java RMI, JMS, JGroups) and interface definition languages (WSDL, Java, WADL) for interoperating with existing services. OW2 FRASCATI provides management tools like standalone, Web-based, and JMX-based graphical consoles and a dedicated scripting language for reconfiguring SCA applications. The whole OW2 FRASCATI platform is itself built as a set of reflective SCA components.

Inria Evaluation Committee Criteria for Software Self-Assessment: A-4-up, SO-4, SM-4-up, EM-3-up, SDL-4-up, DA-4, CD-4, MS-4, TPM-4. FRASCATI is a project of the OW2 consortium for open-source middleware. Web site: http://frascati.ow2.org. 208 Kloc (mainly Java). Registered with the APP (Agence pour la Protection des Programmes) under reference FR.001.050017.000.S.P.2010.000.10000. License: LGPL. Embedded into several industrial software systems: EasySOA, Petals Link EasyViper, EasyBPEL, EasyESB, OW2 PEtALS, OW2 Scarbo. Various demonstrators built during funded projects: ANR SCOrWare, FP7 SOA4All, ANR ITEmIS, ANR SALTY, ANR SocEDA, FUI Macchiato, FUI EasySOA, ADT Galaxy and ADT Adapt. Main publications: [19], [123], [111], [112], [98], [97].

5.6. PowerAPI

Participants: Aurélien Bourdon, Adel Noureddine, Romain Rouvoy [correspondant].

PowerAPI is a Scala-based library for monitoring energy at the process-level. It is based on a modular and asynchronous event-driven architecture using the Akka library. PowerAPI differs from existing energy process-level monitoring tool in its pure software, fully customizable and modular aspect which let users precisely define what they want to monitor, without plugging any external device. PowerAPI offers an API which can be used to express requests about energy spent by a process, following its hardware resource utilization (in terms of CPU, memory, disk, network, etc.). Its applications cover energy-driven benchmarking [52], energy hotspots and bugs detection [53], [75] and real-time distributed system monitoring.

PowerAPI is registered with the APP (Agence pour la Protection des Programmes) under reference IDDN.FR.001.400015.000.S.P.2012.000.10000. License: AGPL.

5.7. 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 [100]. The current implementation of SPACES is based on the COSMOS context framework [93], [121] and the COMANCHE web server [91]. Both COSMOS and COMANCHE are based on the FRACTAL component model and the JULIA implementation [91].

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 [108] protocols. For discovery, the implementation uses the Service Location Protocol (SLP) [102].

2. *Context Representation*: Following the REST principles, SPACES supports multiple representations of the context information: JSON [94], 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 [125]. SPACES benefits from the LDAP based queries of SLP to select the context providers.

We use XStream 1.3.13 [89] and JSON-lib 2.2.34 [88] to serialize context information as XML and JSON documents. For SLP and twitter we employ jSLP 1.0.0 [120] and twitter-4j 2.0.6 [129].

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

FUN Team

4. Software

4.1. Distributed ONS

Participants: Nathalie Mitton, Roberto Quilez [correspondant].

This module implements a DHT-based Distributed EPC Global ONS issued from the ANR WINGS project and published in [30]. APP number: IDDN.FR.001.180033.000.S.P.2012.000.10000.

- Version: version 1

4.2. GOLIATH 1.0

Participants: Fadila Khadar [correspondant], Nathalie Mitton.

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

See also the web page https://gforge.inria.fr/projects/goliath/.

4.3. Linear variable energy module for WSNET.

Participants: Tony Ducrocq [correspondant], Nathalie Mitton.

This module is to be integrated in the WSNET event-based simulator for wireless networks. It implements a Linear transmission variable energy module for WSNET.

- Version: 1.0

RMOD Project-Team

4. Software

4.1. Moose

Participants: Stéphane Ducasse [correspondant], Muhammad Bhatti, Andre Hora, Nicolas Anquetil, Tudor Gîrba [University of Bern].

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

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

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

The first version running on top of Pharo (Moose 4.0) was released in June 2010. In February 2012, Moose 4.6 was released.

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

- **(A5)** Audience : 5 – Moose is used by several research groups, a consulting company, and some companies using it in ad-hoc ways.
- **(SO4)** Software originality : 4 – Moose aggregates the last results of several research groups.
- **(SM4)** Software Maturity : 4 – Moose is developed since 1996 and got two main redesign phases.
- **(EM4)** Evolution and Maintenance : 4 – Moose will be used as a foundation of our Synectique start up so its maintenance is planned.
- **(SDL4)** Software Distribution and Licensing : 4 – Moose is licensed under BSD
- **(OC)** Own Contribution : (Design/Architecture)DA-4, (Coding/Debugging)-4, (Maintenance/Support)-4, (Team/Project Management)-4

4.2. Pharo

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

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 and 2012: Versions 1.2 to 1.4 have been released (with more than 2400 closed issues), and the development branch (2.0) has seen already over 398 incremental releases as of mid November 2012. In 2012, RMoD organized the first *Pharo Conference* during two days in May with 60 participants.

Additionally, in November 2012 RMoD launched the Pharo Consortium (http://www.pharo-project.org/community/consortium) and the Pharo Association (http://association.pharo.org/). 25 companies already shown interest in supporting the consortium.

RMoD is the main maintainer and coordinator of Pharo.

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

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

4.3. Fuel

Participants: Martin Dias [Correspondant], Mariano Martinez-Peck.

Web: http://rmod.lille.inria.fr/web/pier/software/fuel

Objects in a running environment are constantly being born, mutating their status and dying in the volatile memory of the system. The goal of serializers is to store and load objects either in the original environment or in another one. Fuel is a general-purpose serializer based on four principles: (1) speed, through a compact binary format and a pickling algorithm which obtains the best performance on materialization; (2) good object-oriented design, without any special help from the virtual machine; (3) specialized for Pharo, so that core objects (such as contexts, block closures and classes) can be serialized too; (4) flexible about how to serialize each object, so that objects are serialized differently depending on the context.

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

- **(A4)** Audience: 4 – Large audience software, usable by people inside and outside the field with a clear and strong dissemination, validation, and support action plan.
- **(SO3)** Software originality : 3.
- **(SM4)** Software Maturity: 4 – Bug tracker, continuous integration, large test suites are on place.
- **(EM4)** Evolution and Maintenance: 4.
- **(SDL4)** Software Distribution and Licensing: 4 – Fuel is licensed under MIT.
- **(OC5)** Own Contribution: (Design/Architecture) DA-5, (Coding/Debugging) CD-5, (Maintenance/Support) MS-5, (Team/Project Management) TPM-5

4.4. Athens

Participant: Igor Stasenko [Correspondant].

Athens is a vector graphics framework for Pharo.

4.5. Citezen

Participants: Damien Pollet [Correspondant], Stéphane Ducasse.

Web: http://people.untyped.org/damien.pollet/software/citezen/

Citezen is a suite of tools for parsing, validating, sorting and displaying BibTeX databases. This tool suite is integrated within the Pier Content Management System (CMS) and both are implemented on top of Pharo. Citezen aims at replacing and extending BibTeX, in Smalltalk; ideally, features would be similar to BibTeX, CrossTeX, and CSL.

4.6. Handles

Participant: Jean-Baptiste Arnaud [Correspondant].

Web: http://jeanbaptiste-arnaud.eu/handles/

An Handle is a first-class reference to a target object. Handles can alter the behavior and isolate the state of the target object. Handles provide infrastructure to automatically create and wrap new handles when required. A real-time control of handles is possible using a special object called metaHandle.

4.7. Hazelnut

Participants: Guillermo Polito [Correspondant], Benjamin van Ryseghem, Nicolas Paez, Igor Stasenko.

Web: http://rmod.lille.inria.fr/web/pier/software/Seed

Traditionally, Smalltalk-based systems are not bootstrapped because of their ability to evolve by self-modification. Nevertheless, the absence of a bootstrap process exposes many problems in these systems, such as the lack of reproducibility and the impossibility to reach certain evolution paths. Hazelnut is a tool that aims to introduce a bootstrap process into these systems, in particular Pharo.

4.8. Jet

Participant: Veronica Uquillas-Gomez [Correspondant].

Jet is a tool to analyze streams of changes. Jet identifies dependencies between changes and sets of changes and supports cherry picking. Moreover, Jet classifies sets of changes based on their dependencies as a way to ease the analysis of changes within the stream and guide system integrators.

4.9. LegacyParsers

Participants: Muhammad Bhatti [Correspondant], Nicolas Anquetil, Guillaume Larcheveque, Esteban Lorenzano, Gogui Ndong.

As part of our research on legacy software and also for the Synectique company), we started to define several parsers for old languages like Cobol for example. This work is important to help us validate our meta-model and tools against a larger range of existing technologies and to discover the limits of our approach. From our initial results, and the in-depth understanding that it gave us, we are formulating new research objectives in meta-model driven reverse engineering. This work is also important for the spin-off company, as being able to work with such technologies is fundamental.

4.10. Mate

Participants: Marcus Denker [Correspondant], Clement Bera, Camillo Bruni.

Mate is the future research-oriented virtual machine for Pharo. Its goal is to serve as a prototype for researchers to experiment with. As a result, the design of Mate is very simple to understand. As of today, Mate consists of an AST interpreter, a new object memory layout, and a simple garbage collector.

4.11. NativeBoost

Participant: Igor Stasenko [Correspondant].

Web: http://code.google.com/p/nativeboost/

NativeBoost is a Smalltalk framework for generating and running machine code from the language side of Pharo. As part of it comes a foreign function interface that enables calling external C functions from Smalltalk code with minimal effort.

4.12. Nabujito

Participants: Camillo Bruni [Correspondant], Marcus Denker.

Nabujito is a new Just In Time compiler implemented as a Smalltalk application, based on NativeBoost, that does not require changes in the virtual machine.

4.13. Nautilus

Participants: Benjamin Van Ryseghem [Correspondant], Stéphane Ducasse, Igor Stasenko, Camillo Bruni, Esteban Lorenzano.

Nautilus is a new source code browser based on the latest infrastructure representations. Its goal is mainly to replace the current system browser that was implemented in the 80s and that doesn't provide optimal tools for the system as it has evolved.

4.14. SourceCity

Participants: Erwan Douaille [Correspondant], Igor Stasenko, Guillaume Larcheveque, Stéphane Ducasse.

Modern systems are too complex. Understanding and analyzing these systems is very hard and tedious (thousand of classes, millions of lines of code). One needs an overview of the system that allows to discover important parts in the system, weak points, suspicious components. SourceCity is a powerful 3D tool that can help to understand quickly how a system works by taking the metaphor of a city buildings. By looking at tall, large, low building, one can identify different properties of the software components being represented.

4.15. Spec

Participants: Benjamin Van Ryseghem [Correspondant], Stéphane Ducasse, Johan Fabry.

Spec is a programming framework for generating graphical user interfaces inspired by VisualWorks' Subcanvas. The goal of Spec is to tackle the lack of reuse experienced in existing tools. Spec serves as a pluggable layer on top of multiple lower-level graphical frameworks. Many improvements have been noticed in Pharo after the introduction of Spec in terms of speed or number of lines of code while we re-implemented existing tools using Spec.

4.16. VerveineJ

Participants: Nicolas Anquetil [Correspondant], Andre Hora, Guillaume Larcheveque.

Web: Inria project 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 fulfil the same needs, but it was closed sources and presented some errors that tainted the results we could obtain.

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

5. Software

5.1. LibGINA

Participant: Laurent Grisoni [correspondant].

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

Current version: version 1.0

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

5.2. 3D interaction using mobile phone

Participants: Samuel Degrande [correspondant], Laurent Grisoni.

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

Current version: version 1.0

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

5.3. tIO (tactile input & output)

Participants: Paolo Olivo, Nicolas Roussel [correspondant], Ibrahim Yapici.

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 [2]. The toolkit provides resolution and frequency information for the available pointing and display devices and makes it easy to choose between them at run-time through the use of URIs. It allows to bypass the system's transfer functions to receive raw asynchronous events from one or more pointing devices. It replicates as faithfully as possible the transfer functions used by Microsoft Windows, Apple OS X and Xorg (the X.Org Foundation server). Running on these three platforms, it makes it possible to compare the replicated functions to the genuine ones as well as custom ones. The toolkit is written in C++ with Python and Java bindings available. It is 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

MOSTRARE Project-Team

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 rewriting tool that removes backward axis, a compiler of the FXP library to nested word automata and an efficient query answering algorithm for nested word automata on XML streams.
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. A new release was published in October 2012.

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

- Version: FXP v1.1.0

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 XPath with full support for the XML data model. The QuiXPath library provides a compiler from QuiXPath to FXP. 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, an professional implementation of the W3C pipeline language XProc own by Innovimax. A new release was published in October 2012.

See also the web page http://code.google.com/p/quixpath/.

- Version: QuixPath v1.1.0

5.3. VOLATA

Participant: Fabien Torre [correspondant].

Software Self-Assessment: A-2, SO-4, SM-2, EM-2, SDL-2

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

5.4. JProGraM

Participant: Antonino Freno [correspondant].

Software Self-Assessment: A-3, SO-3-up, SM-2, EM-3, SDL-4.

JProGraM is a GPL-licensed Java library for machine learning and statistical analysis *over* graphs and *through* graphs. Supported models for vectorial data include e.g. Bayesian networks, Markov random fields, Gaussian mixtures, kernel density estimators, and neural networks, whereas random graph tools include small-world networks, preferential-attachment, exponential random graphs, and spectral models (as well as subgraph sampling algorithms). One strong point of the library is the extensive support for continuous random variables.

See also the webpage http://researchers.lille.inria.fr/~freno/JProGraM.html.